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 strategies, to the use of sophisticated structures 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; Axelrod-Python. This manuscript make use of Axelrod-Python to conduct a meta analysis of 40000 Iterated Prisoner's Dilemma tournaments. The aim is to evaluate the performance of numerous strategies and finally answer the explore the factors of success in the game.

1 Background

The Iterated Prisoner's Dilemma (IPD) 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 having memory of the prior interactions. 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 [16], 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 IPD. In the 80's following the strong performance of Tit For Tat in both Axelrod's computer tournaments [14, 15], and moreover in a series of evolutionary experiments [16], the strategy was thought as the most robust basic strategy in the IPD. However, the strategy was shown to perform poorly in environments with noise [19, 26, 42, 51]. More robust strategy in such setting were introduced and were the new protagonists, such as, Nice and Forgiving [19], Pavlov [44] and Generous Tit For Tat [45].

The 20$^{\text{th}}$ Anniversary Iterated Prisoner Dilemma Tournament took place in 2004 with 233 entries. The winning strategies were based on a mechanism of teams. A team from Southampton University took advantage of the fact that a participant was allowed to submit multiple strategies. They submitted a total of 60 of strategies that could recognised each other and colluded to increase one members score [24, 25, 49]. Yet again, in 2012 another set of strategies was introduced as the dominant set of strategies [47]. 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 [29] a tournament containing over 200 strategies, zero-determinant, was performed and none of the zero-determinant strategies ranked in top spots. Instead, the top ranked strategies were a set of evolved strategies based lookup tables [13], hidden markov models [29] and finite state automata [40].

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 [3] to simulate a large number of computer tournaments using as many strategies as possible from the literature. The aim is to 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 different tournaments and the data generating process are covered in Section 2. Section 3, covers the best performed strategies for each type of tournament and overall. Section 4, explores the traits which contribute to good performance and finally in Section **??** the results are discussed and concluded in Section 5.

2 Data generating process

For the purposes of this manuscript a data set containing results on IPD tournaments has been generated and is available at. This was done using the open source package Axelrod-Python [3], more specifically, version 3.0.0. Axelrod-Python allows for different types of IPD computer tournaments to be simulated whilst containing a list of over 180 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 [14]. 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 k number of times. The winner of each tournament is based on the average score a strategy achieved and not by number of wins.

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 186 strategies in [3], a random list of N strategies is chosen. The 186 strategies used here are given in the Appendix A. For the given list of strategies a standard, a noisy, a probabilistic ending and a noisy probabilistic ending tournament are performed and repeated k 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 1.

parameter	parameter explanation	min value	max value
N	number of strategies	3	195
k	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 1: Data generation parameters' values

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 [4, 20]. 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;
- $k \leftarrow$ randomly select integer $\in [k_{min}, k_{max}]$;
- $n \leftarrow$ 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, k);
- result noisy $\leftarrow$ Axelrod.tournament(players, n, p, k);
- result probabilistic ending $\leftarrow$ Axelrod.tournament(players, e, k);
- result noisy probabilistic ending $\leftarrow$ Axelrod.tournament(players, p, e, k);

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

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 2.

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, 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 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	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 2: Output result.

The **normalised rank** is a measure which was manually included. The normalised rank, denoted as r, is calculated as a strategy's rank divided by the tournament's size (N). The normalised rank will be used in the next section to evaluate the performance of strategies.

3 Top ranked strategies

This section evaluates the performance of 186 strategies. The performance of each strategy will be evaluated for each type of tournament independently, followed by an evaluation of their performance over all the simulated tournaments of this work. Each strategy could have participated in multiple tournaments of the same type (on average each participated in 5690 different tournaments). For example Tit For Tat has participated in a total of 5569 tournaments of each type. The strategy's normalised rank distribution in these is given in Figure 1. As a result, of the multiple entries of strategies their performance is evaluated based on the **median normalised rank** denoted as $\bar{r}$. A value of $\bar{r} = 0$ corresponds to a strategy winning the tournament where a a value of $\bar{r} = 1$ corresponds to the strategy coming last.

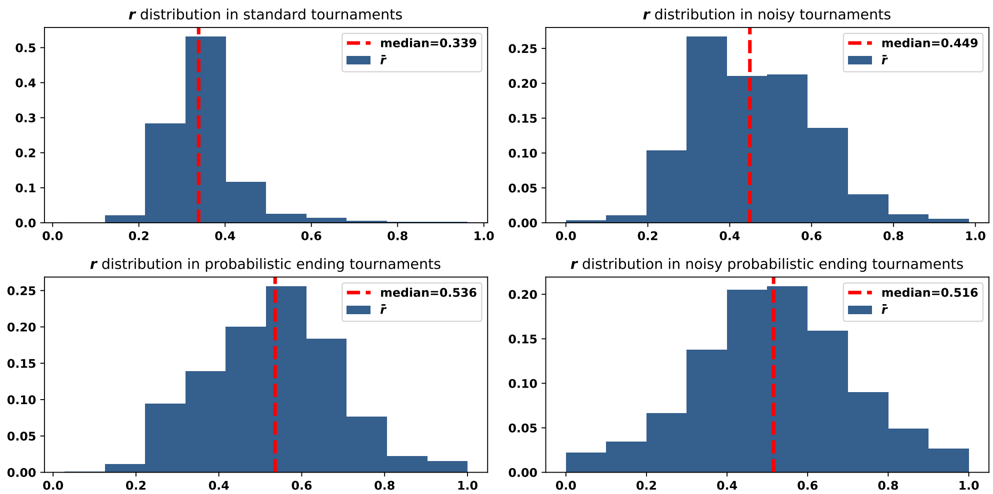

Figure 1: Tit For Tat's r distribution in tournaments. The best performance of the strategy has been in standard tournaments where it achieved a normalised rank of 0.339.

The result of each tournament type are based on 12, 285 trials. The strategies are ranked based on the median normalised rank they achieved and the top 15 strategies of each tournament are given in Table 3.

Standard		Noisy		Probabilistic ending		Noisy probabilistic ending	
Name	$\bar{r}$	Name	$\bar{r}$	Name	$\bar{r}$	Name	$\bar{r}$
Evolved HMM 5	0.00658	Grumpy	0.13953	Fortress4	0.01266	Alternator	0.30390
Evolved FSM 16	0.00990	e	0.19048	Defector	0.01444	ϕ	0.31025
EvolvedLookerUp2 2 2	0.01064	Cooperator	0.19565	Better and Better	0.01587	e	0.31293
Evolved FSM 16 Noise 05	0.01639	Tit For 2 Tats	0.20520	Tricky Defector	0.01869	π	0.31818
PSO Gambler 2 2 2	0.02139	Cycle Hunter	0.22222	Fortress3	0.02198	Limited Retaliate	0.35294
Evolved ANN	0.02874	Risky QLearner	0.22424	Gradual Killer	0.02521	Anti Tit For Tat	0.35429
Evolved ANN 5	0.03390	Retaliate 3	0.23077	Aggravater	0.02797	Retaliate 3	0.35484
PSO Gambler 1 1 1	0.03723	Retaliate 2	0.23762	Raider	0.03077	Limited Retaliate 3	0.35563
Evolved FSM 4	0.04839	Retaliate	0.24309	Cycler DDC	0.04545	Retaliate	0.35588
PSO Gambler Mem1	0.05000	Hard Tit For 2 Tats	0.24658	Hard Prober	0.05085	Retaliate 2	0.35714
Winner12	0.05946	Limited Retaliate 3	0.25000	SolutionB1	0.06040	Limited Retaliate 2	0.36066
Fool Me Once	0.06122	ShortMem	0.25272	Meta Minority	0.06040	Hopeless	0.36913
DBS	0.07087	Limited Retaliate	0.25698	Bully	0.06061	Arrogant QLearner	0.40526
DoubleCrosser	0.07190	Limited Retaliate 2	0.26027	Fool Me Forever	0.07018	Cautious QLearner	0.40711
BackStabber	0.07500	ϕ	0.26201	EasyGo	0.07065	Risky QLearner	0.41989

Table 3: Top performances for each tournament type based on $\bar{r}$.

In standard tournaments 10 of the 15 top strategies are strategies introduced in [29]. These have been trained using reinforcement learning algorithms and are based on finite state automata (FSM), hidden markov models (HMM), artificial neural networks (ANN), lookup tables (LookerUp) and stochastic lookup tables (Gambler). Specifically, the have been trained against the strategy list of [3] in standard tournaments. Thus their performance is to be expected. DoubleCrosser, and Fool Me Once, are not from the literature but from [3]. 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 [39] and DBS are both from the the literature. DBS [12] is strategy specifically designed for noisy environments, however, it ranks highly only in standard ones.

Figure 2 gives the distributions of $\bar{r}$ for the top ranked strategies. The distributions are skewed towards zero and the highest median is at 0.075. This indicates that the top ranked strategies are dominating strategies in standard tournaments. They are very likely to perform well in standard tournament despite the number of opponents, the opponents, the turns etc. This does not hold for all the tournament types as it will be discussed in the later parts.

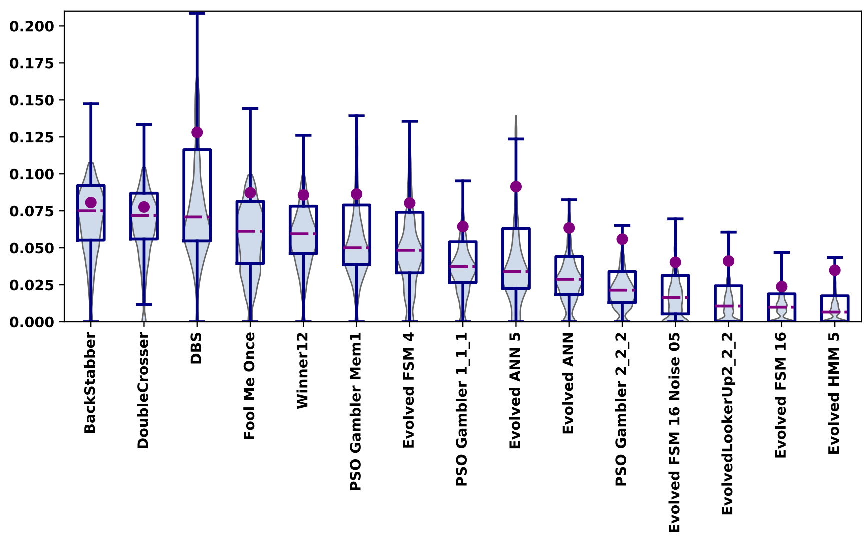

Figure 2: $\bar{r}$ distributions of top 15 strategies in standard tournaments.

The top strategies in noisy tournaments include the two strategies, Tit For 2 Tats [15] and Hard Tit For 2 Tats [52], which are strategies that will defect only after the have received two defections from the opponent. The Retaliate strategies are set of strategies from [3] that start by cooperating but will retaliate once the opponent's wins and defections surpass a curtain threshold. ShortMem [22], Grumpy, e and ϕ are strategies that make decisions based on the cooperations to defections ratio. In 5^{th} and 6^{th} place are the strategies Cycler Hunter and Risky QLearn. Cycler Hunter tries to extort strategies that play cyclically and Risky QLearn uses a Q learning algorithm. Notably, a deterministic and one of the most simple strategies in game is ranked 3^{rd}. That is Cooperator, a strategy that just cooperates.

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 tournaments. The distributions are skewed both towards 0 and 1 which indicates that though the top ranked strategies mainly performed well (medians < 0.3) there are several tournaments that they performed worse than the 60% of the participants.

The 15 top ranked strategies in probabilistic ending tournaments include Fortress 3, Fortress 4 (both introduced in [10]), Raider [11] and Solution B1 [11] which are strategies based on finite state automata introduced

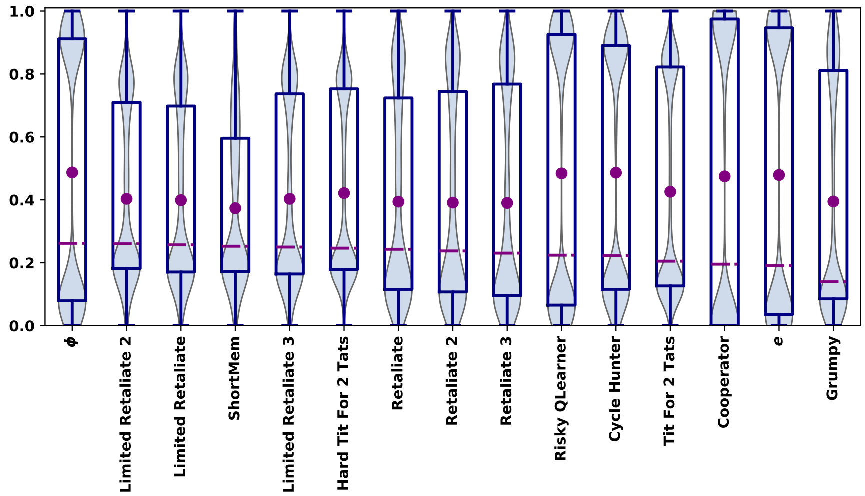

Figure 3: $\bar{r}$ distributions for best performed strategies in noisy tournaments.

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 ones. In probabilistic ending tournaments it appears that the top ranks are mostly occupied by defecting strategies which include Better and Better, Gradual Killer, Hard Prober (all from [1]), Bully (Reverse Tit For Tat) [43] and Defector. Thus, it's surprisingly that EasyGo and Fool Me Forever are ranked 14^{th} and 15^{th}. These strategies are actually the same; they will defect until their opponent defect, then they will cooperate until the end. Both strategies have repeatedly ranked highly as shown in Figure 4 and there are cases for which they were the winners of the tournament.

The distributions of the normalised rank in probabilistic ending tournaments are less variant than those of noisy tournaments. The medians are lower than 0.1 and the distributions are skewed towards 0. Though the large difference between the means and the medians indicates some outliers, the strategies have overall performed well in the tournaments that they participated.

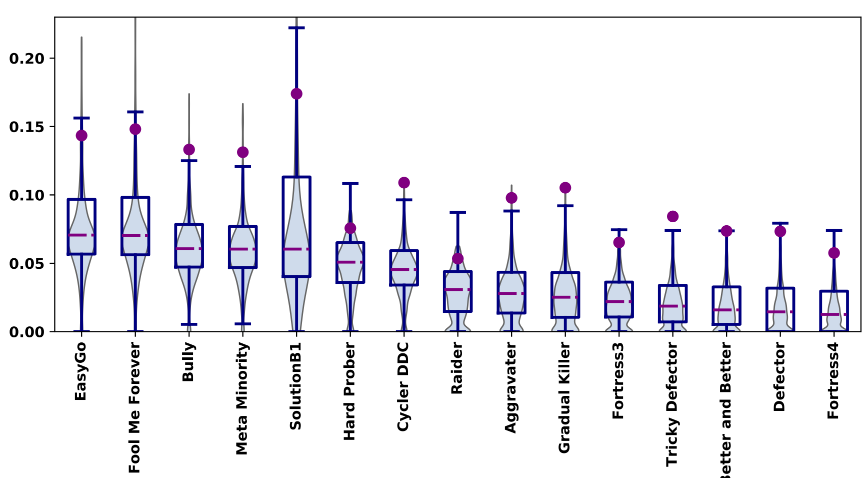

Figure 4: $\bar{r}$ distributions for best performed strategies in probabilistic ending tournaments.

In tournaments that have both noise and an unspecified number of turns several of the top ranked strategies are strategies that were highly ranked in noisy tournaments as well. However, strategies from the top ranks of probabilistic ending tournaments did not rank highly here. The Retaliate set, ϕ, e and Anti Tit For Tat behaviour strategies appear to perform well in noisy environments. So these strategies performed well in this setting as well even though now the turns are not specified. The top ranked strategy is Alternator a strategy that alternate between cooperation and defection. Hopeless [55] is a strategy that will only defect if and only if a mutual cooperation and the last three places are occupied by strategies based on a Q learning algorithm. The three Q learning strategies are the only ones that have bimodal distributions of normalised ranks. In comparison, the rest of the distributions are skewed towards 0.4, Figure 5.

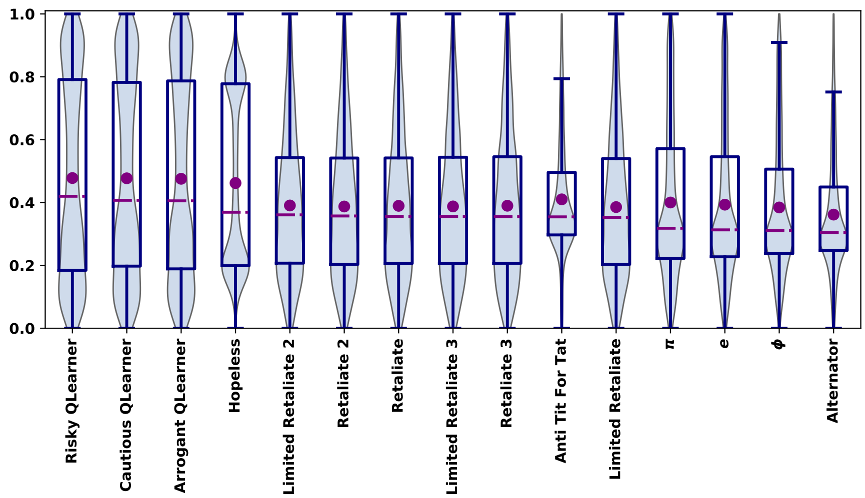

Figure 5: $\bar{r}$ distributions for best performed strategies in noisy probabilistic ending tournaments.

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 4: Top performances in data set

So far the performances have been evaluated separately for each tournament type. The merged data set, which was described in Section 2, contains a total of 49,140 result summaries. The 15 top ranked strategies overall are given in Table 4. The top ranks include strategies that have been mentioned before due to their performance in specific tournaments. The top ranks are overtaken by the set of Retaliate strategies followed by BackStabber and DoubleCrosser. DoubleCrosser performed well in standard tournaments and the strategy is just an extension of BackStabber. The two strategies Nice Meta Winner and NMWE Memory One are strategies based on teams. PSO Gambler and Evolved HMM 5 are trained strategies introduced in [29] and Stein and Rapoport and Grudger are strategies from Axelrod's original tournament where they came 6th and 7th respectively. Forgetful Fool Me Once is based on the same approach as Grudger.

Figure 6 gives the normalised rank distributions of these strategies. It is evident that the Retaliate strategies performances are very similar and their distributions are different in comparison with the rest of the top ranked strategies in the setting.

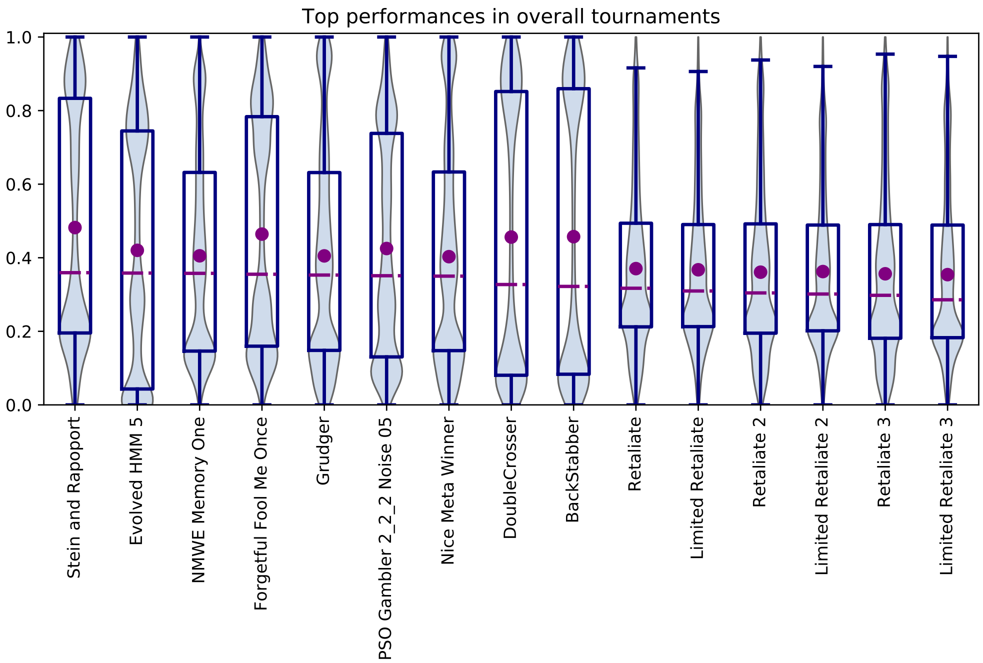

Figure 6: $\bar{r}$ distributions for best performed strategies in data set.

This section presented the top ranked performances.

- In standard tournaments the top ranked spots were dominated by complex strategies that have been trained using reinforcement learning techniques. 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 were basing their decisions on the he defection to cooperation 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 in this setting, as well as finite state automata that have been introduced all by the same authors.
- Moreover, 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 the means and medians of the distributions were close to 0.4. Indicating that on average these strategies managed to consistently perform on the top 40% of the strategies.

- Finally, using all the tournaments results, despite their type, the top ranked strategies were a mixture of behaviours that do well in standard tournaments and tournaments with noise. The top ranks also included two strategies that are based on teams.

In this section the winning strategies in a series of IPD tournaments of different settings were presented. Though there have been instances of strategies that do well in more than one setting, most of the top ranks were occupied by different strategies each time. The top ranks have been occupied by strategies with small memory size, with infinite memory size, sophisticated and deterministic strategies. Even two of the most basic strategic rules, Cooperator and Defector, have been ranked in the top 15 places.

This indicates that in an IPD tournament a winning strategy exist but only for the given environment, however, there is no dominant strategy in the IPD that can do well in any tournament where the settings can vary. The aim of the next section is to understand which are the factors that made these strategies successful. In each setting separately but also overall.

4 Evaluation of performance

The aim of this section is to explore the factors that contribute to a strategy's successful performance. The factors explored are measures regarding a strategy's behaviour but also measures regarding the tournaments the strategies competed in. More specifically the factors that are included in the analysis are given by Table 5 and Table 1.

Axelrod-Python makes use of classifiers to classify strategies according to various dimensions. The classifiers considered in this work are the stochastic classifier, the make use of game and length classifiers. These determine whether a strategy is stochastic or deterministic, whether it makes use of the number of turns or the game's payoffs. The SSE is a measure extortionate behaviour introduced in [34]. A value of 1 indicates no extortionate behaviour at all whereas a value of 0 indicates that a strategy is always trying to extortion the opponent. The rest of the factors considered are the CC to C, CD to C, DC to C, and DD to C rates as well as cooperating ratio of a strategy. The minimum, maximum, medium and median cooperating ratios of each tournament are also included, and finally the number of turns, the size and the probabilities of noise or the game ending.

measure	measure explanation	source	value type	min value	max value
stochastic	If a strategy is stochastic	strategy classifier from [3]	boolean	False	True
makes use of game	If a strategy makes used of the game information	strategy classifier from [3]	boolean	False	True
makes use of length	If a strategy makes used of the number of turns	strategy classifier from [3]	boolean	False	True
SSE	A measure of how far a strategy is from extortionate behaviour	method described in [34]	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
C_r	The cooperating ratio of a strategy	result summary	float	0	1
CC to C rate	The probability a strategy will cooperate after a mutual cooperation	result summary	float	0	1
CD to C rate	The probability a strategy will cooperate after being betrayed by the opponent	result summary	float	0	1
DC to C rate	The probability a strategy will cooperate after betraying the opponent	result summary	float	0	1
DD to C rate	The probability a strategy will cooperate after a mutual defection	result summary	float	0	1

Table 5: Manually calculated/retrieved measures.

The effect of these factors on a strategy's success is evaluated based on the correlation coefficient between the factors, the normalised rank and the median score. The correlation coefficients are given by Table 6.

Note that the correlation for the classifiers is not included because they are binary variables and they will evaluated by a different method.

	Standard		Noisy		Probabilistic ending		Noisy probabilistic ending		Overall	
	r	median score	r	median score	r	median score	r	median score	r	median score
CC to C rate	-0.501	0.501	0.414	-0.504	0.408	-0.323	0.260	0.022	0.108	0.081
CD to C rate	0.226	-0.199	0.456	-0.330	0.320	-0.017	0.205	-0.220	0.281	-0.177
DC to C rate	0.127	-0.100	0.509	-0.504	-0.018	0.033	0.341	-0.016	0.173	-0.088
DD to C rate	0.412	-0.396	0.533	-0.436	-0.103	0.176	0.378	-0.263	0.237	-0.239
SSE	0.473	-0.452	0.463	-0.337	-0.156	0.223	0.305	-0.259	0.233	-0.167
C_r	-0.323	0.384	0.711	-0.678	0.714	-0.832	0.579	-0.135	0.360	-0.124
C_{max}	-0.000	0.049	0.000	0.023	-0.000	0.046	0.000	-0.004	0.000	0.280
C_{min}	0.000	0.084	0.000	-0.017	-0.000	0.007	-0.000	0.041	0.000	-0.250
C_{median}	0.000	0.209	-0.000	0.240	-0.000	0.187	-0.000	0.673	-0.000	0.544
C_{mean}	-0.000	0.229	-0.000	0.271	0.000	0.200	0.000	0.690	-0.000	0.553
C_r / C_{max}	-0.323	0.381	0.616	-0.551	0.714	-0.833	0.536	-0.116	0.395	-0.265
C_r / C_{min}	0.109	-0.080	-0.358	0.250	-0.134	0.150	-0.368	0.113	-0.161	-0.190
C_r / C_{median}	-0.331	0.353	0.652	-0.669	0.712	-0.852	0.330	-0.466	0.294	-0.405
C_r / C_{mean}	-0.331	0.358	0.731	-0.740	0.721	-0.861	0.649	-0.621	0.428	-0.439
p	-	-	-0.000	0.207	-	-	-0.000	-0.650	0.000	-0.256
n	0.000	-0.125	-0.000	-0.024	-	-	-	-	0.000	-0.074
e	-	-	-	-	0.000	0.165	0.000	-0.058	0.000	0.055

Table 6: Correlations table.

In standard tournaments the measures CC to C, C_r, $C_r/C_{\max}$ and the cooperating ratio compared to the median and the mean have a moderate negative effect on the normalised rank and a moderate positive on the median score. The SSE error and the DD to C have the opposite effects. Thus, in standard tournaments behaving cooperative corresponds to a more successful performance. However, even though being cooperative pays of that's not true against defective strategies. Cooperating after a mutual defection lowers a strategy's success. Figure7 confirms these. The winner of standard tournaments almost always cooperated after a mutual cooperation but the 50% only cooperated with a probability of 0.2 after a mutual defection.

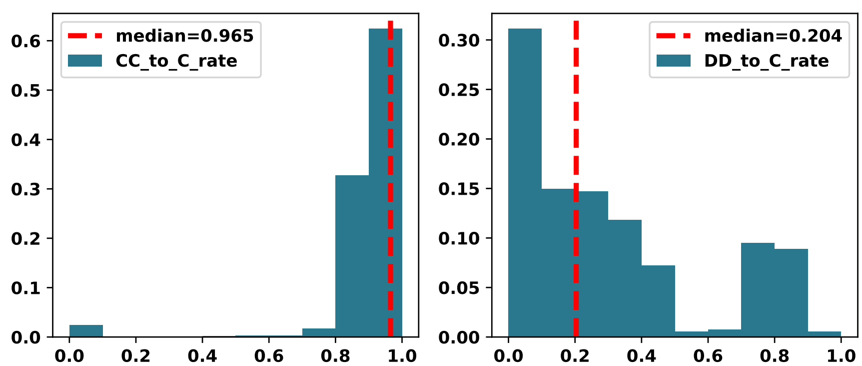

Figure 7: Winners distributions in standard tournaments for CC to C and DD to C.

Compared to standard tournaments in both noisy and in probabilistic ending tournaments the higher the rates of cooperation the lower a strategy's success and median score. Moreover, a strategy would want to cooperate less than both the mean and median cooperator in such settings. In probabilistic ending tournaments the correlations coefficients have a larger values, indicating a stronger effect. Thus a strategy will be punished more by it's cooperative behaviour in such environments. The distributions of the C_r of the winners in both tournaments is given by Figure 8. It confirms that the winners in these two settings were more defective strategies, mainly in probabilistic ending ones.

In noisy probabilistic ending tournaments and in the overall tournaments the only factors that had a moderate

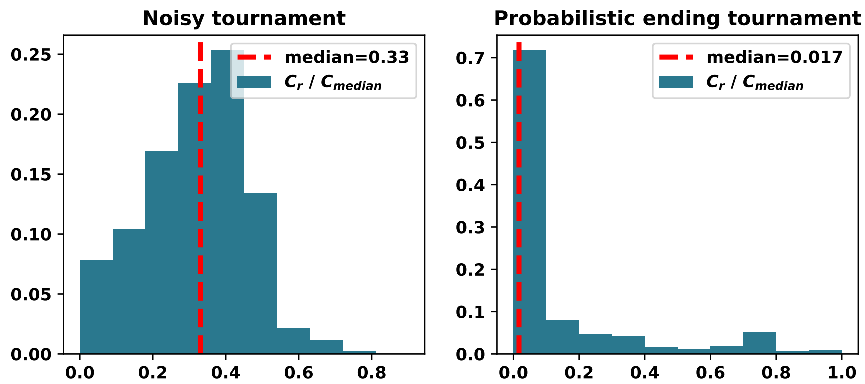

Figure 8: C_r distributions.

affect are $C_r/C_{\text{mean}}, C_r/C_{\text{max}}$ and C_r. In such environments cooperative behaviour appears to be punished by not as much as in noisy and probabilistic ending tournaments.

To further evaluate the effect of factors on performances a random forest classification [21] is applied. Initially, the performances are clustered based on their normalised rank and the median score by a $k-$means algorithm [7]. The number of clusters are not deterministically chosen but are based on the silhouette coefficients [50]. Consider the case of standard tournaments, the chosen number of clusters is 2 and Figure 9 illustrates the trials of clustering the performances in 2, 3 and 4 clusters respectively. The number of chosen clusters for each type and overall are given in Table 7.

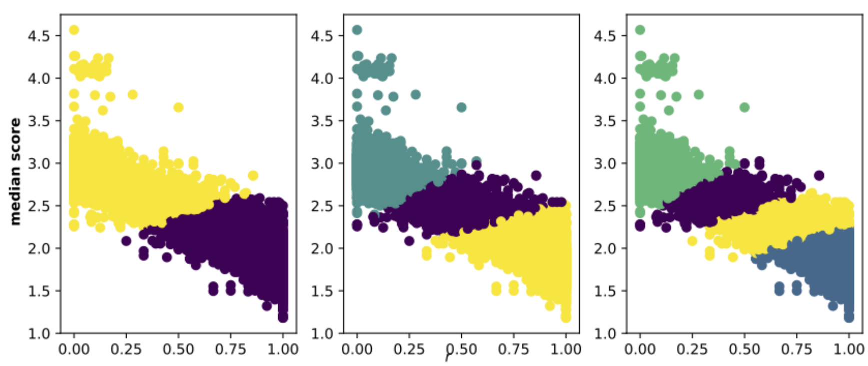

Figure 9: Clustering trials for standard tournaments. A number of 2 clusters has been chosen with a silhouette score of 0.66 against 0.511 and 0.50 respectively.

type	number of clusters	silhouette coefficient
standard	2	0.648
noisy	3	0.493
probabilistic ending	2	0.680
noisy probabilistic ending	4	0.415
overall	3	0.443

Table 7: Number of clusters for each type and overall and the respective silhouette coefficients.

A random forest approach is applied to the data to predict the cluster to which a strategy's performance has been assigned to. The random forest method constructs many individual decision trees and the predictions from all trees are pooled to make the final prediction. The random forest models are trained on a training set of 70% of the tournaments results. The accuracy of each model based on R^2 are given by Table 8. The out of the bag error [30] has also been calculated. The models fit well and the values of the accuracy measures on the test data and the OOB error indicate that they are not over fitting.

type	R^2 training data	R^2 test data	R^2 OBB score
standard	0.998545	0.989890	0.982331
noisy	0.996677	0.950572	0.935383
probabilistic ending	0.999592	0.995128	0.992819
noisy probabilistic ending	0.990490	0.813905	0.791418
overall	0.993396	0.913409	0.898059

Table 8: Accuracy metrics for random forest models.

The importance of the features on the classification task along with their inter trees variability are given by Figure 15. The importance indicates that the two factors that affected performances the most were C_r/C_{median} and C_r/C_{mean}. The classifiers which where not included in the previous analysis appear to have no effect, and several of the factors that are highted by the importance are inline with the correlation results.

The behaviour a strategy adapts in comparison with the mean and median cooperator influence the success of a strategy. To gain a better understanding on the influence of these measures, in each tournament type the distributions of the winner's C_r/C_{median} and C_r/C_{median} are given by Figures 16, 17, 18, 19 and 20. In summary, in a noisy, noisy probabilistic or if a strategy does not know the settings on the environment is about to compete, a strategy should cooperate 60% of times the median/mean cooperator does. In standard tournaments a strategy would want to have the same cooperating ratio as the mean/median and finally in probabilistic ending tournaments a strategy wants to be defective.

The factors which have an effect on the performances were presented here by two approaches which included the correlation coefficients and a random forest analysis. The results of these are discussed in the following section.

5 Conclusion

This manuscript explored a variety of strategic behaviours in the game of the Iterated Prisoner's Dilemma and presented an evaluation of performance in the game. A large number of computer tournaments have been analysis and demonstrated that a single dominant strategy does not exist in the game. Moreover, it was shown that from a series of factors that a strategy has control over the tge factors that can influence it's performance is the cooperating ratio compared to that of it's environment.

A total of 186 strategies were used in this manuscript, which are available via an open source software, the Axelrod-Python. By making use of the software a total of 49,140 computer tournaments results were gathered. In Section 3, the result summaries were used to present the top performances. This was not done over the entire data set. The data set contains results from four different settings, and these were also studied individually. The top performances were presented for all the different combinations of the results summary. Though Retaliate set of families, strategies introduced in [3], did perform well in noisy environments and this feature allowed the strategies to rank well in more than one tournament type, the analysis concluded that there is not a single dominant strategy in the Iterated Prisoner's Dilemma. THough a dominant perform was not highlighted by the analysis of this manuscript the factors that made performances successful were further explored.

In Section 4 an analysis of strategies features was covered. The results of this analysis showed that a strategy's

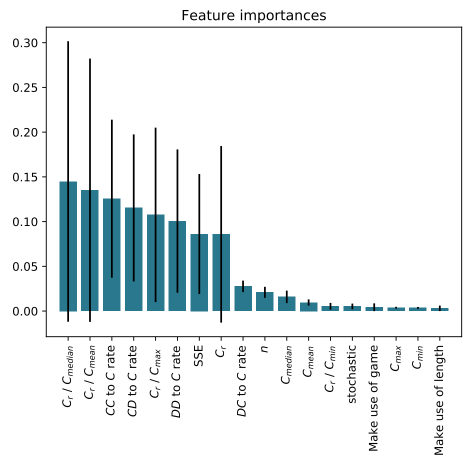

Figure 10: Standard tournaments

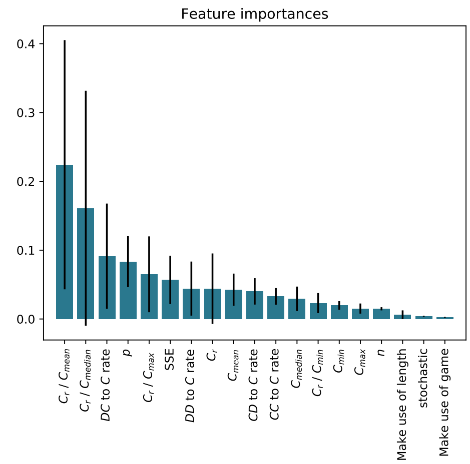

Figure 11: Noisy tournaments

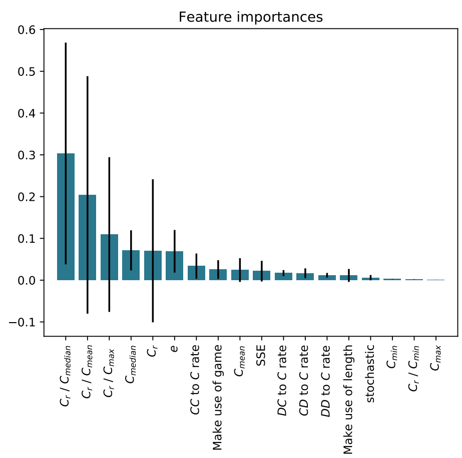

Figure 12: Probabilistic ending tournaments

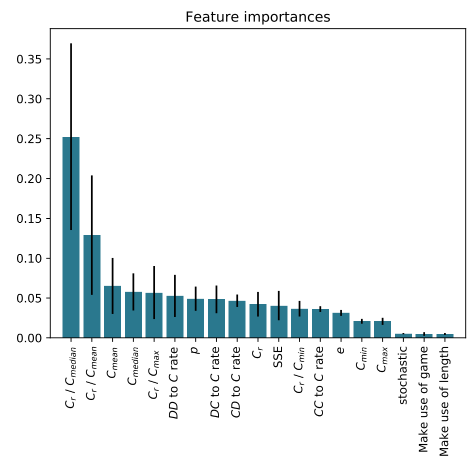

Figure 13: Noisy probabilistic ending tournaments

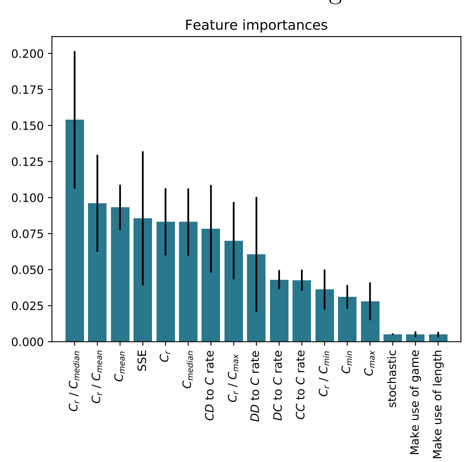

Figure 14: overall ending tournaments

Figure 15: Importance of features in IPD tournaments.

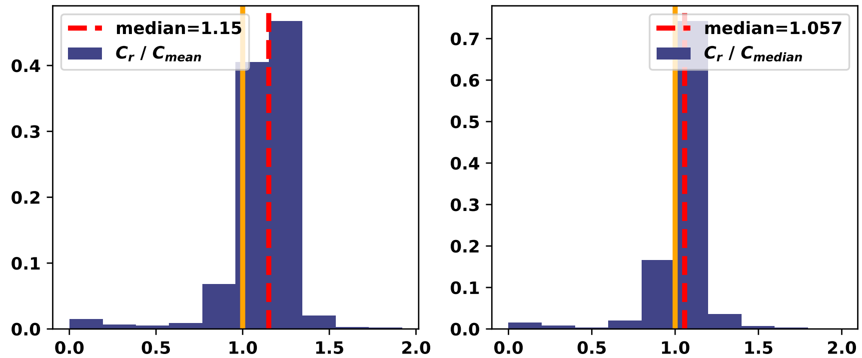

Figure 16: Distributions of C_r/C_{median} and C_r/C_{median} for standard tournaments.

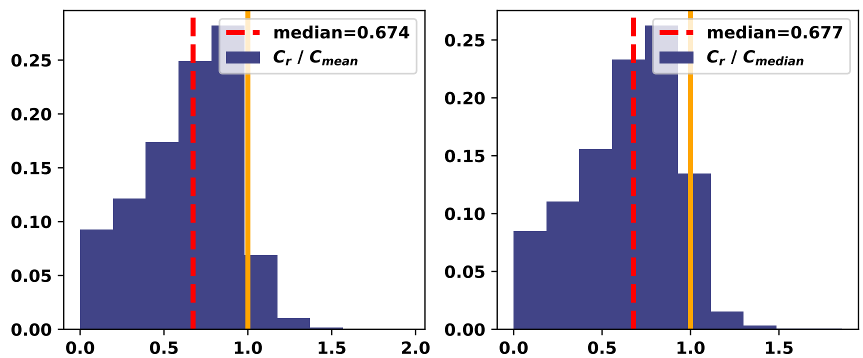

Figure 17: Distributions of C_r/C_{median} and C_r/C_{median} for noisy tournaments.

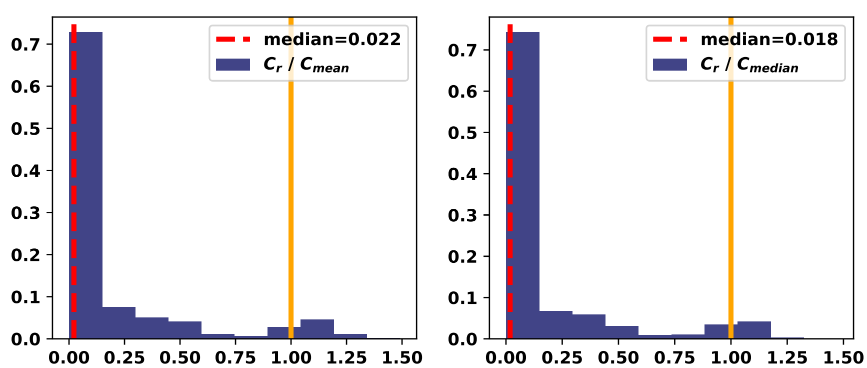

Figure 18: Distributions of C_r/C_{median} and C_r/C_{median} for probabilistic ending tournaments.

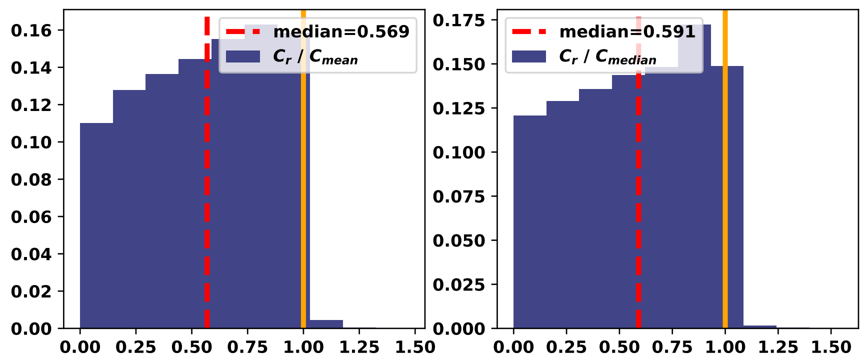

Figure 19: Distributions of C_r/C_{median} and C_r/C_{median} for noisy probabilistic ending tournaments.

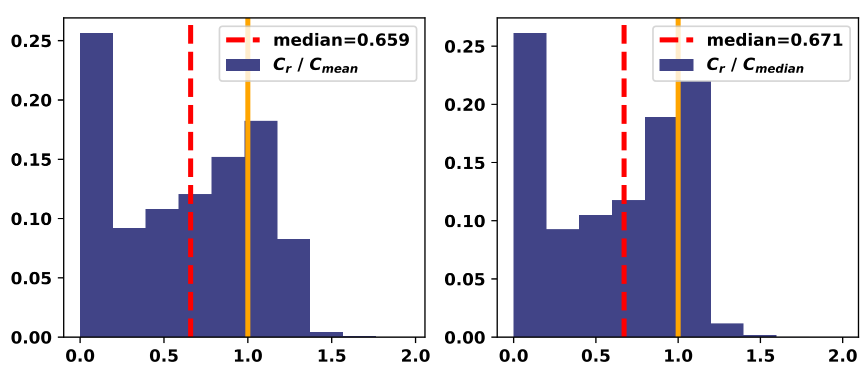

Figure 20: Distributions of C_r/C_{median} and C_r/C_{median} for over all the tournaments.

characteristics such as it's stochasticity and the information it used regarding the game had no effect on the strategy's success. The most important factors have been those that compared the strategy's behaviour to it's environment. The cooperating ratio of the strategy compared to the mean and median cooperator was highlighted as the most important feature in the analysis. More specifically, if a strategy were to enter a tournament with a theory of mind of it's environment it would choose to be the median cooperator in standard tournaments, the defector in probabilistic ending tournaments and to cooperate 60% of the median in noisy and noisy probabilistic tournaments. Lastly, if a strategy was aware of the opponents but not of the setting on the tournament, a strategy would be more likely to be successful if it were to identify the median cooperator and cooperated 60% of the times that they did.

References

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

[2] A strategy with novel evolutionary features for the iterated prisoner's dilemma. *Evolutionary Computation*, 17(2):257–274, 2009.

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

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

[5] Christoph Adami and Arend Hintze. Evolutionary instability of zero-determinant strategies demonstrates that winning is not everything. *Nature communications*, 4(1):2193, 2013.

[6] Eckhart Arnold. Coopsim v0.9.9 beta 6. https://github.com/jecki/CoopSim/, 2015.

[7] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[8] Daniel Ashlock, Joseph Alexander Brown, and Philip Hingston. Multiple opponent optimization of prisoners dilemma playing agents. *IEEE Transactions on Computational Intelligence and AI in Games*, 7(1):53–65, 2015.

[9] Daniel Ashlock and Eun-Youn Kim. Fingerprinting: Visualization and automatic analysis of prisoner's dilemma strategies. *IEEE Transactions on Evolutionary Computation*, 12(5):647–659, 2008.

[10] 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.

[11] 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.

[12] 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.

[13] R. Axelrod. The evolution of strategies in the iterated prisoner's dilemma. *Genetic Algorithms and Simulated Annealing*, pages 32–41, 1987.

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

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

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

[17] Jeffrey S Banks and Rangarajan K Sundaram. Repeated games, finite automata, and complexity. *Games and Economic Behavior*, 2(2):97–117, 1990.

[18] Bruno Beaufils, Jean-Paul Delahaye, and Philippe Mathieu. Our meeting with gradual, a good strategy for the iterated prisoners dilemma. In *Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems*, pages 202–209, 1997.

[19] 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.

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

[21] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

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

[23] Andre LC Carvalho, Honovan P Rocha, Felipe T Amaral, and Frederico G Guimaraes. Iterated prisoners dilemma-an extended analysis. 2013.

[24] J.P. Delahaye. L'altruisme perfectionn. *Pour la Science (French Edition of Scientific American)*, 187:102–107, 1993.

[25] J.P. Delahaye. Logique, informatique et paradoxes, 1995.

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

[27] Marcus R Frean. The prisoner's dilemma without synchrony. *Proceedings of the Royal Society of London B: Biological Sciences*, 257(1348):75–79, 1994.

[28] Marco Gaudesi, Elio Piccolo, Giovanni Squillero, and Alberto Tonda. Exploiting evolutionary modeling to prevail in iterated prisoners dilemma tournaments. *IEEE Transactions on Computational Intelligence and AI in Games*, 8(3):288–300, 2016.

[29] 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.

[30] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[31] Christian Hilbe, Martin A Nowak, and Arne Traulsen. Adaptive dynamics of extortion and compliance. *PloS one*, 8(11):e77886, 2013.

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

[33] Graham Kendall, Xin Yao, and Siang Yew Chong. *The iterated prisoners' dilemma: 20 years on*, volume 4. World Scientific, 2007.

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

[35] David Kraines and Vivian Kraines. Pavlov and the prisoner's dilemma. *Theory and decision*, 26(1):47–79, 1989.

[36] Steven Kuhn. Prisoner's dilemma. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2017 edition, 2017.

[37] Jiawei Li, Philip Hingston, Senior Member, and Graham Kendall. Engineering Design of Strategies for Winning Iterated Prisoner ' s Dilemma Competitions. 3(4):348–360, 2011.

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

[39] 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.

[40] J. H. Miller. The coevolution of automata in the repeated prisoner's dilemma. *Journal of Economic Behavior and Organization*, 29(1):87 – 112, 1996.

[41] Shashi Mittal and Kalyanmoy Deb. Optimal strategies of the iterated prisoner's dilemma problem for multiple conflicting objectives. *IEEE Transactions on Evolutionary Computation*, 13(3):554–565, 2009.

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

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

[44] 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.

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

[46] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[47] 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.

[48] Arthur J Robson. Efficiency in evolutionary games: Darwin, nash and the secret handshake. *Journal of theoretical Biology*, 144(3):379–396, 1990.

[49] A. Rogers, A Rogers, RK Dash, SD Ramchurn, P Vytelingum, and NR Jennings. Coordinating team players within a noisy iterated prisoners dilemma tournament. *Theoretical computer science.*, 377(1-3):243–259, 2007.

[50] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[51] 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.

[52] 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.

[53] E Tzafestas. Toward adaptive cooperative behavior. *From Animals to animals: Proceedings of the 6th International Conference on the Simulation of Adaptive Behavior (SAB-2000)*, 2:334–340, 2000.

[54] Unknown. www.prisoners-dilemma.com. http://www.prisoners-dilemma.com/, 2017.

[55] 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.

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

[57] Jianzhong Wu and Robert Axelrod. How to cope with noise in the iterated prisoner's dilemma. *Journal of Conflict resolution*, 39(1):183–189, 1995.

A List of strategies

The strategies used in this study which are from Axelrod version 3.0.0 [3].

1. ϕ [3]
2. π [3]
3. e [3]
4. ALLCorALLD [3]
5. Adaptive [37]
6. Adaptive Pavlov 2006 [33]
7. Adaptive Pavlov 2011 [37]
8. Adaptive Tit For Tat: 0.5 [53]
9. Aggravater [3]
10. Alternator [16, 41]
11. Alternator Hunter [3]
12. Anti Tit For Tat [31]
13. AntiCycler [3]
14. Appeaser [3]
15. Arrogant QLearner [3]
16. Average Copier [3]
17. Better and Better [1]
18. Bully [43]
19. Calculator [1]
20. Cautious QLearner [3]
21. CollectiveStrategy [2]
22. Contrite Tit For Tat [57]
23. Cooperator [16, 41, 47]
24. Cooperator Hunter [3]
25. Cycle Hunter [3]
26. Cycler CCCCCD [3]
27. Cycler CCCD [3]
28. Cycler CCCDCD [3]
29. Cycler CCD [41]
30. Cycler DC [3]
31. Cycler DDC [41]
32. DBS: 0.75, 3, 4, 3, 5 [12]
33. Davis: 10 [14]
34. Defector [16, 41, 47]
35. Defector Hunter [3]
36. Desperate [55]
37. DoubleResurrection [6]
38. Doubler [1]
39. Dynamic Two Tits For Tat [3]
40. EasyGo [37, 1]
41. Eatherley [15]
42. Eventual Cycle Hunter [3]
43. Evolved ANN [3]
44. Evolved ANN 5 [3]
45. Evolved ANN 5 Noise 05 [3]
46. Evolved FSM 16 [3]
47. Evolved FSM 16 Noise 05 [3]
48. Evolved FSM 4 [3]
49. Evolved HMM 5 [3]
50. EvolvedLookerUp1_1_1 [3]
51. EvolvedLookerUp2_2_2 [3]
52. Feld: 1.0, 0.5, 200 [14]
53. Firm But Fair [27]
54. Fool Me Forever [3]
55. Fool Me Once [3]
56. Forgetful Fool Me Once: 0.05[3]

57. Forgetful Grudger [3]
58. Forgiver [3]
59. Forgiving Tit For Tat (**FTfT**) [3]
60. Fortress3 [10]
61. Fortress4 [10]
62. GTFT: 0.1 -
63. GTFT: 0.3 -
64. GTFT: 0.33 [28, 44]
65. GTFT: 0.7 -
66. GTFT: 0.9 -
67. General Soft Grudger: n=1,d=4,c=2 [3]
68. Gradual [18]
69. Gradual Killer: ('D', 'D', 'D', 'D', 'D', 'C', 'C') [1]
70. Grofman[14]
71. Grudger [14, 17, 18, 55, 37]
72. GrudgerAlternator [1]
73. Grumpy: Nice, 10, -10 [3]
74. Handshake [48]
75. Hard Go By Majority [41]
76. Hard Go By Majority: 10 [3]
77. Hard Go By Majority: 20 [3]
78. Hard Go By Majority: 40 [3]
79. Hard Go By Majority: 5 [3]
80. Hard Prober [1]
81. Hard Tit For 2 Tats (**HTf2T**) [52]
82. Hard Tit For Tat (**HTfT**) [54]
83. Hesitant QLearner[3]
84. Hopeless [55]
85. Inverse[3]
86. Inverse Punisher [3]
87. Joss: 0.9 [14, 52]
88. Level Punisher [6]
89. Limited Retaliate 2: 0.08, 15 [3]
90. Limited Retaliate 3: 0.05, 20 [3]
91. Limited Retaliate: 0.1, 20 [3]
92. MEM2 [38]
93. Math Constant Hunter [3]
94. Meta Hunter Aggressive: 7 players [3]
95. Meta Hunter: 6 players [3]
96. Meta Mixer: 173 players[3]
97. Naive Prober: 0.1 [37]
98. Negation [54]
99. Nice Average Copier[3]
100. Nydegger [14]
101. Omega TFT: 3, 8 [33]
102. Once Bitten [3]
103. Opposite Grudger [3]
104. PSO Gambler 1_1_1[3]
105. PSO Gambler 2_2_2[3]
106. PSO Gambler 2_2_2 Noise 05[3]
107. PSO Gambler Mem1 [3]
108. Predator [10]
109. Prober [37]
110. Prober 2 [1]
111. Prober 3 [1]
112. Prober 4 [1]
113. Pun1 [10]
114. Punisher [3]
115. Raider [11]
116. Random Hunter [3]
117. Random: 0.1
118. Random: 0.3
119. Random: 0.5 [14, 53]
120. Random: 0.7
121. Random: 0.9
122. Remorseful Prober: 0.1 [37]
123. Resurrection [6]
124. Retaliate 2: 0.08 [3]
125. Retaliate 3: 0.05 [3]
126. Retaliate: 0.1 [3]
127. Revised Downing: True [14]
128. Ripoff [9]
129. Risky QLearner[3]
130. SelfSteem [23]
131. ShortMem [23]
132. Shubik [14]
133. Slow Tit For Two Tats [3]
134. Slow Tit For Two Tats 2 [1]
135. Sneaky Tit For Tat [3]
136. Soft Go By Majority [16, 41]
137. Soft Go By Majority: 10 [3]
138. Soft Go By Majority: 20 [3]
139. Soft Go By Majority: 40 [3]
140. Soft Go By Majority: 5 [3]
141. Soft Grudger [37]
142. Soft Joss: 0.9 [1]
143. SolutionB1 [8]
144. SolutionB5 [8]
145. Spiteful Tit For Tat [1]
146. Stochastic Cooperator [5]
147. Stochastic WSLS: 0.05 [3]

148. Suspicious Tit For Tat [18, 31]
149. TF1 [3]
150. TF2 [3]
151. TF3 [3]
152. Tester [15]
153. ThueMorse [3]
154. ThueMorseInverse [3]
155. Thumper [9]
156. Tit For 2 Tats (**Tf2T**) [16]
157. Tit For Tat (**TfT**) [14]
158. Tricky Cooperator [3]
159. Tricky Defector [3]
160. Tullock: 11 [14]
161. Two Tits For Tat (**2TfT**) [16]
162. VeryBad [23]
163. Willing [55]
164. Win-Shift Lose-Stay: D (**WShLSt**) [37]
165. Win-Stay Lose-Shift: C (**WSLS**) [35, 44, 52]
166. Winner12 [39]
167. Winner21 [39]
168. Worse and Worse[1]
169. Worse and Worse 2[1]
170. Worse and Worse 3[1]
171. ZD-Extort-2 v2: 0.125, 0.5, 1 [36]
172. ZD-Extort-2: 0.1111111111111111, 0.5 [52]
173. ZD-Extort-4: 0.23529411764705882, 0.25, 1 [3]
174. ZD-GEN-2: 0.125, 0.5, 3 [36]
175. ZD-GTFT-2: 0.25, 0.5 [52]
176. ZD-SET-2: 0.25, 0.0, 2 [36]

B Acknowledgements

A variety of software have been used in this work:

- The Axelrod library for IPD simulations [3].
- The Matplotlib library for visualisation [32].
- The Numpy library for data manipulation [56].
- The scikit-learn library for data analysis [46].