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, 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 [8, 9], and moreover in a series of evolutionary experiments [10], 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 [11, 16, 20, 27], made room for other protagonists, such as, Nice and Forgiving [11], Pavlov [22] and Generous Tit For Tat [23].

In 2004, the Anniversary Iterated Prisoner Dilemma Tournament was run 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 [14, 15, 25].

Yet again, in 2012 another set of strategies was introduced as the dominant set of strategies [24]. 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 [17] 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, 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 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 5 the results are discussed and summarised.

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 [2], 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 [8]. 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 [2], a random list of N strategies is chosen. 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 [3, 12]. 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 which can be found 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, in Sections 3.3 in probabilistic ending and in 3.4 in noisy probabilistic ending.

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 standard tournaments. The strategy's normalised rank distribution in different tournament 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.

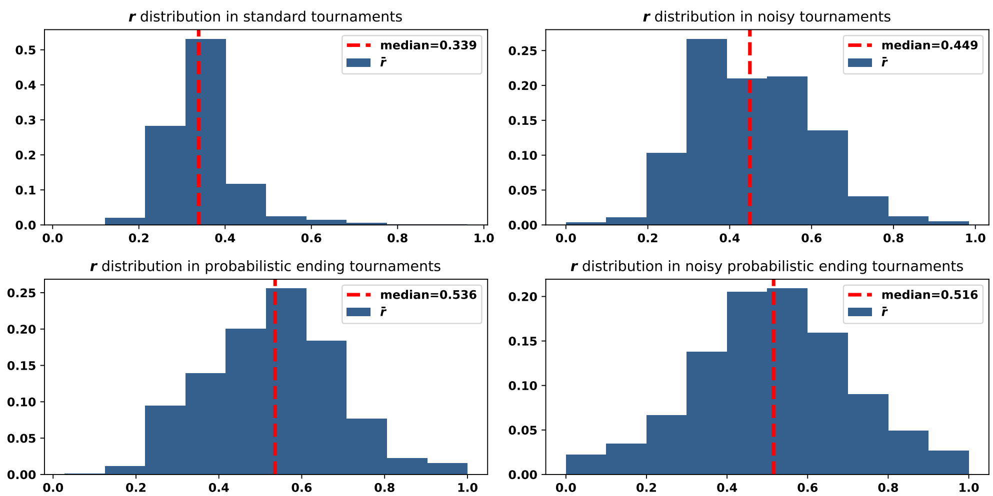


Figure 1: Tit For Tat's r distribution in tournaments.

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 simulated tournaments of this work.

3.1 Top ranked strategies in: Standard tournaments

The results presented in this section are based on 12,285 different standard tournaments. The top 15 performances in standard tournaments are given in Table 3. Ten of these are strategies introduced in [17]. 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 [2] in standard tournaments. Thus their performance is 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 [19] and DBS are both from the the literature. DBS [7] 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

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 3: Standard top performances

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 following sections. The next section present the results of noisy tournaments.

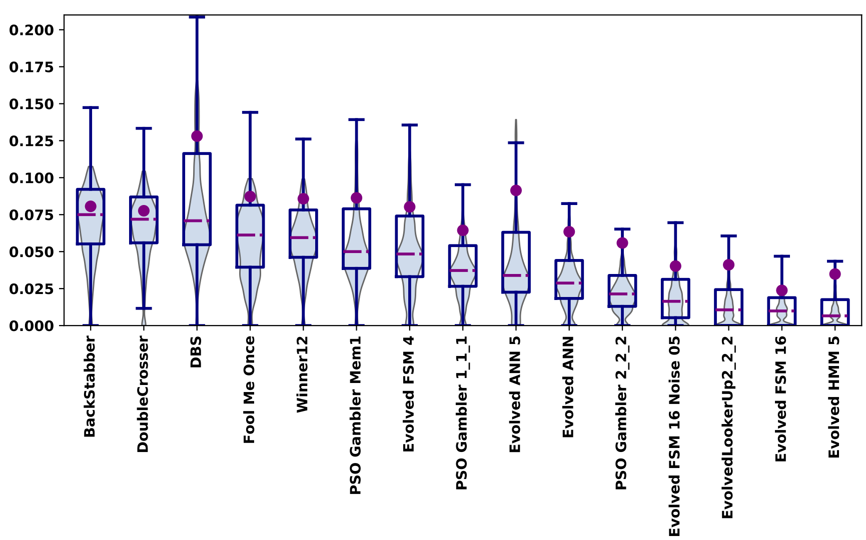


Figure 2: $\bar{r}$ distributions of top 15 strategies.

3.2 Top ranked strategies in: 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 4. The distributions of their corresponding $\bar{r}$ is also given in Figure 3.

The top strategies include the two strategies, Tit For 2 Tats [9] and Hard Tit For 2 Tats [28], 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. ShortMem [13], 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.

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 4: 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 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.

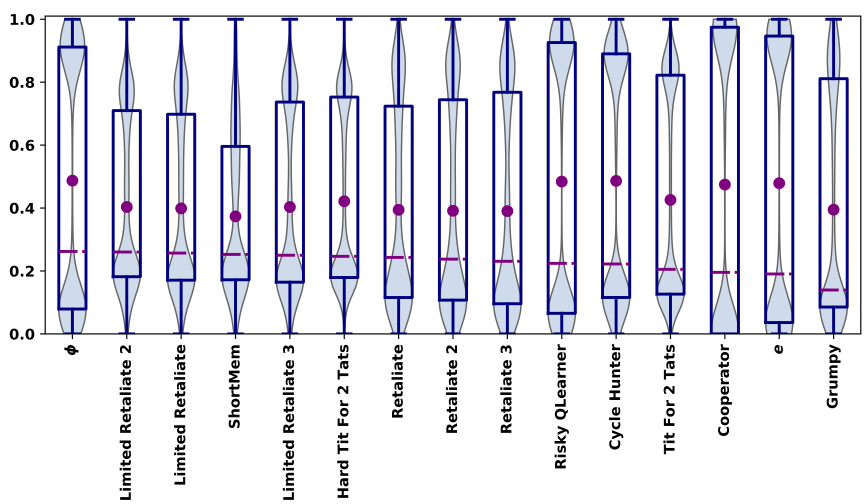


Figure 3: $\bar{r}$ distributions for best performed strategies in noisy tournaments.

3.3 Top ranked strategies: Probabilistic ending tournaments

In this section the performances in probabilistic ending tournaments are discussed. The 15 top ranked strategies are given by Table 5 and Figure 4 gives their respective $\bar{r}$ distributions. Fortress 3, Fortress

4 (both introduced in [5]), Raider [6] and Solution B1 [6] 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 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) [21] 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 tournaments.

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 5: Probabilistic ending top performances

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.

3.4 Noisy & probabilistic ending tournaments

This section presents the results in tournaments that have both noise and an unspecified number of turns. Several of the top ranked have been covered before because they were highly ranked in noisy tournaments as well, Table 6. However, strategies from the top ranks of probabilistic ending tournaments did not rank highly here.

The Retaliates, 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 [29] 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 similar to those of Section 3.2. In comparison, the rest of the distributions are skewed towards 0.3 to 0.4 Figure 5.

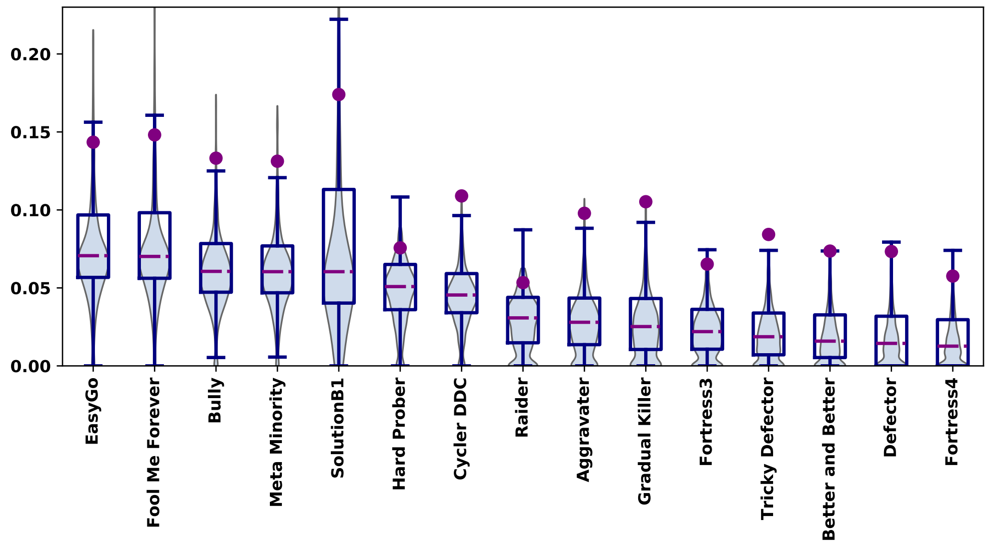


Figure 4: $\bar{r}$ distributions for best performed strategies in probabilistic ending tournaments.

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 6: Noisy and probabilistic ending top performances

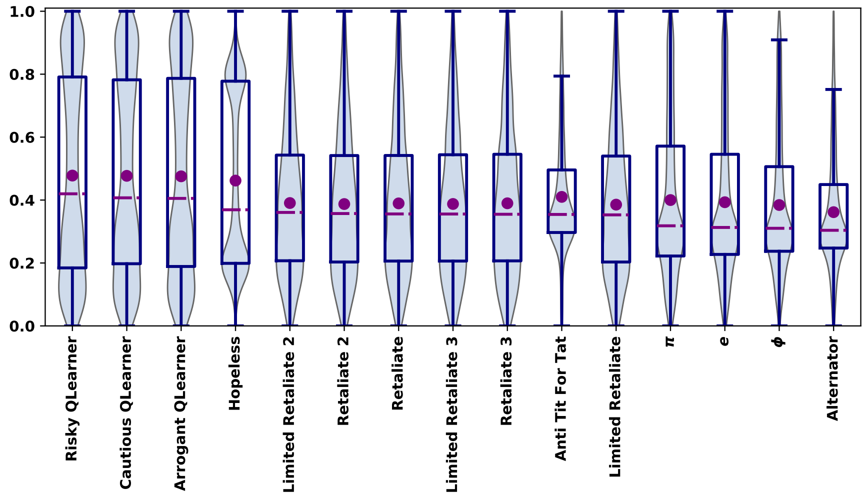


Figure 5: $\bar{r}$ distributions for best performed strategies in noisy probabilistic ending tournaments.

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 complex strategies that have been trained using reinforcement learning techniques in a standard tournament. 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 defections and cooperations 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. However, in these tournaments the means and medians of the distributions were close to 0.4. Indicating that on average these strategies manage to beat only 60% of the opponents.

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

3.5 Top performance in data set

The data set described in Section 2 contains results from 49,140 tournaments and the performances in this section are based on these tournaments. The top ranked 15 strategies are given in Table 7. It includes several strategies that have ranked in the top spots in the separate tournament analysis. The top spaces are overtaken by the Retaliate strategies, followed by BackStabber and DoubleCrosser. DoubleCrosser is just an extension of BackStabber. Nice Meta Winner and NMWE Memory One are strategies based on teams of strategies 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 a similar strategy to Grudger but sometimes forgets and cooperates. Finally, both PSO Gambler and Evolved HMM 5 are strategies introduced in [17].

The top ranked strategies include a good mixture of strategies. PSO Gambler and Evolved HMM 5 are very complex strategies compared to Stein and Rapoport and Grudger that are strategies manually designed. There are strategies based on teams and the Retaliate strategies which have illustrated to do well in noise.

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 7: Top performances in data set

The Retaliates' distributions are very similar to each other, Figure 7, and different compared to the rest of the distributions.

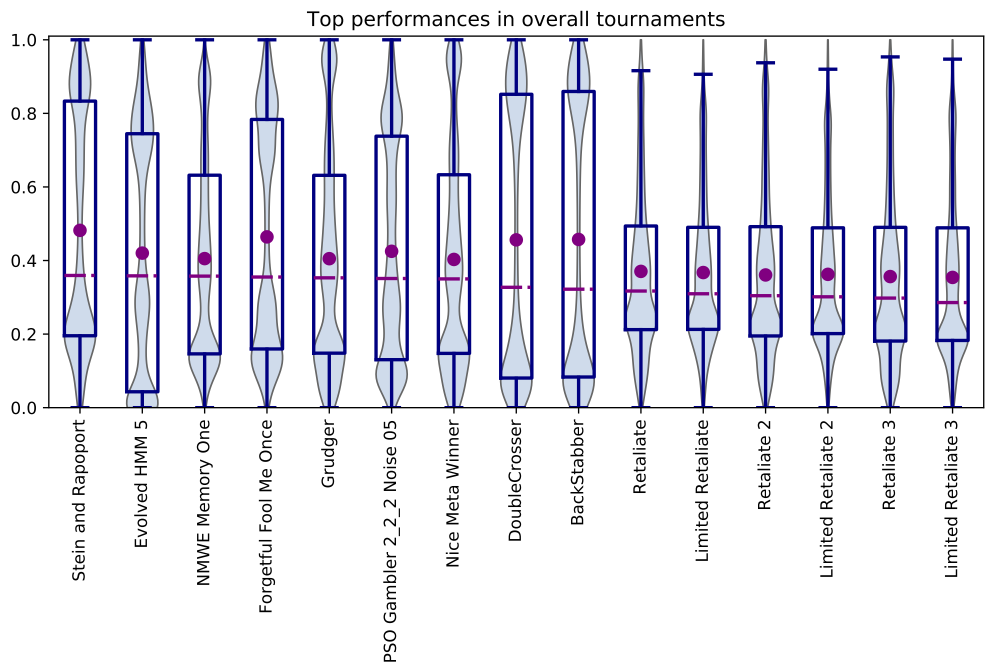


Figure 6: $\bar{r}$ distributions for best performed strategies in data set.

However, are these strategies the answer as to how to play the IPD. The distributions of their ranks are given for all the tournaments individually. Figure illustrates that there are settings that these strategies perform well, that they perform badly and settings that they perform average. Thus the conclusion is that there is not a single dominant strategy in the IPD.

In this section the top ranked performances were presented. This was not done in a single tournament of a given list of opponents, as many in the literature, but over a number of tournaments where the list and

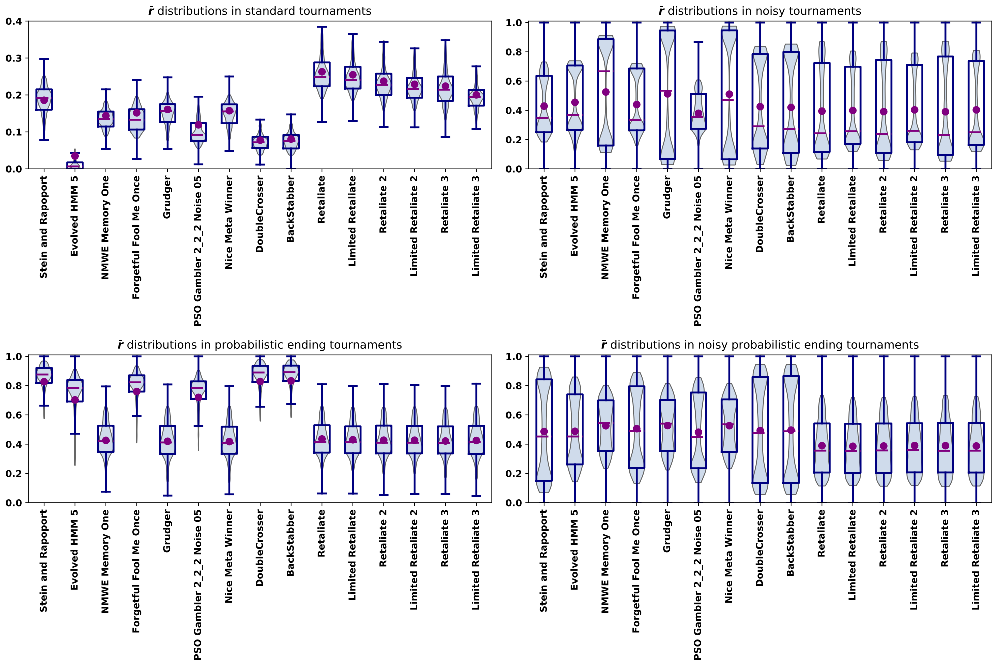


Figure 7: $\bar{r}$ distributions for best performed strategies in data set.

size could vary. Moreover, four different type of tournaments were performed in this work. Even with the large number of tournaments there some think that is evident, there is not a single best strategy. Though some strategies seem to be repeated, there are other tournament types that they are not. The top ranks are occupied but almost different strategies every time. And in the top spots the strategies behaviour differ. There are simple and complex, with small memory and large, deterministic and stochastic. Is there a single dominant strategy in the iterated prisoner's dilemma? We argue not.

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

This section explores the factors that contributed to the successful performance of strategies described in Section 3. This is achieved by separating the strategies performances into successful ones and non. The performances are separated on the normalised rank and the median score using the clustering algorithm $k-$means [4]. The number of clusters is not deterministic but the decision is based on the silhouette coefficients [26].

Consider the performances in the standard tournaments. Figure 8 gives the normalised rank against the median score for three trials of the clustering algorithm with number of clusters being 2, 3 and 4. A number of 2 clusters gives the highest silhouette coefficient (0.66) and so it's selected for the standard tournaments. More specifically, being in cluster 1 corresponds to a low performance whereas being in cluster 2 corresponds to a high performance. These are based on the median normalised rank and median score of each cluster as shown in Table 8.

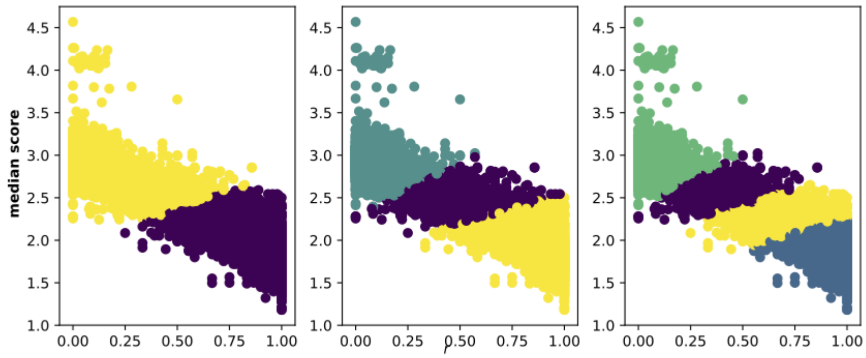


Figure 8: Clustering trials for standard tournaments. A number of 2 clusters has been chosen with a silhouette score of 0.66 against 0.511 and 0.50 respectively.

Cluster	$\bar{r}$	median score
1	0.769	2.065
2	0.263	2.639

Table 8: Median normalised rank and median score of each cluster in standard tournaments.

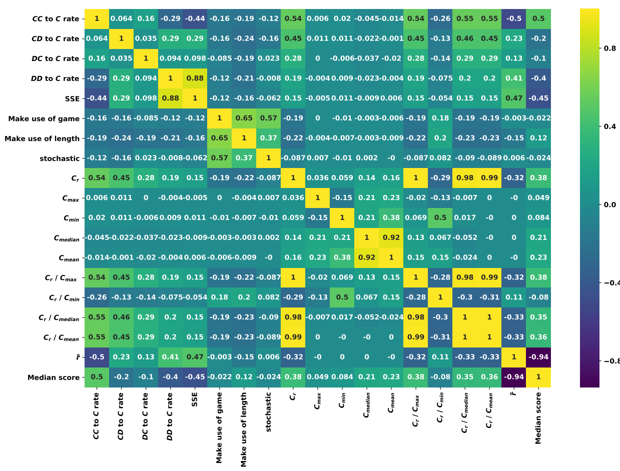
Now that the performances have been clustered the aim is to evaluate which factors affect the performances being in each respective cluster. The factors evaluated in this work are given by Table 9. Several of these factors include measures regarding the strategies and it's behaviour however others are fixed from the tournament or even the type of tournament. The measures from Table 1 are also included.

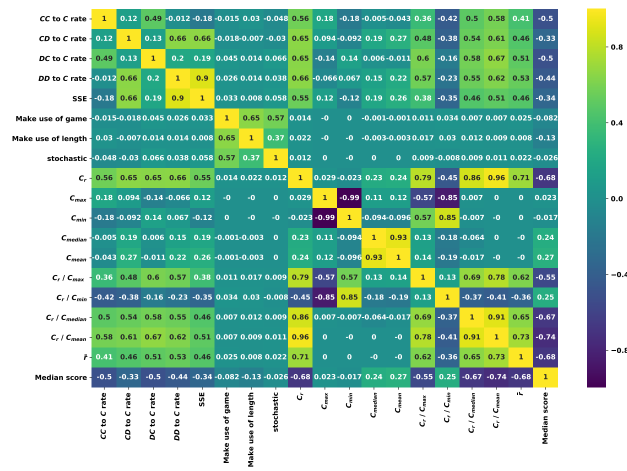
measure	measure explanation	source	value type	min value	max value
stochastic	If a strategy is stochastic	strategy classifier from [2]	boolean	False	True
makes use of game	If a strategy makes used of the game information	strategy classifier from [2]	boolean	False	True
makes use of length	If a strategy 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 [18]	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
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

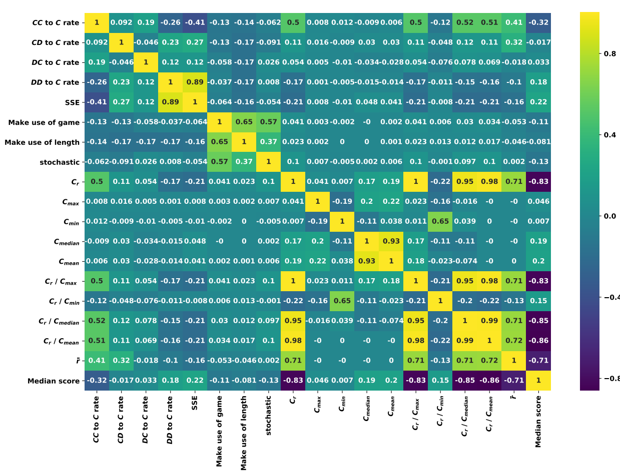
Table 9: Manually calculated/retrieved measures.

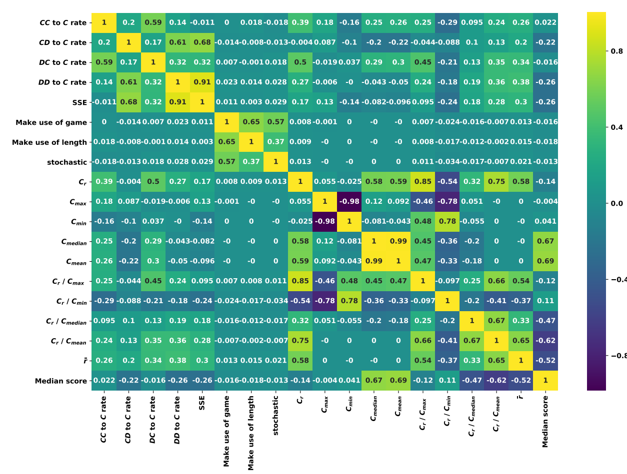
To evaluated how significant each factor contribution was a random forest is applied to the performances of standard tournaments where the target value is their corresponding cluster. Figure 9 gives the importance of each factor. The four factors that have affect more the performances of strategies in standard tournaments are the cooperation ratio compared to the mean, the SSE, the cooperation ratio compared to the median and the cooperating ratio after being betrayed by the opponent.

To delved deeper into the results of the random forest and how each feature has helped to be clustered where you were the tree interpreter method described in [] is used. The strongest contributors to predicting that a performance was clustered in the successful class were the DC to D ratio, the SSE error, the size followed by the cooperating ratio compared to the mean. In comparison the strongest impact on being in the non









successful cluster have the Cd to C rate and the cooperating ration compare to the median.

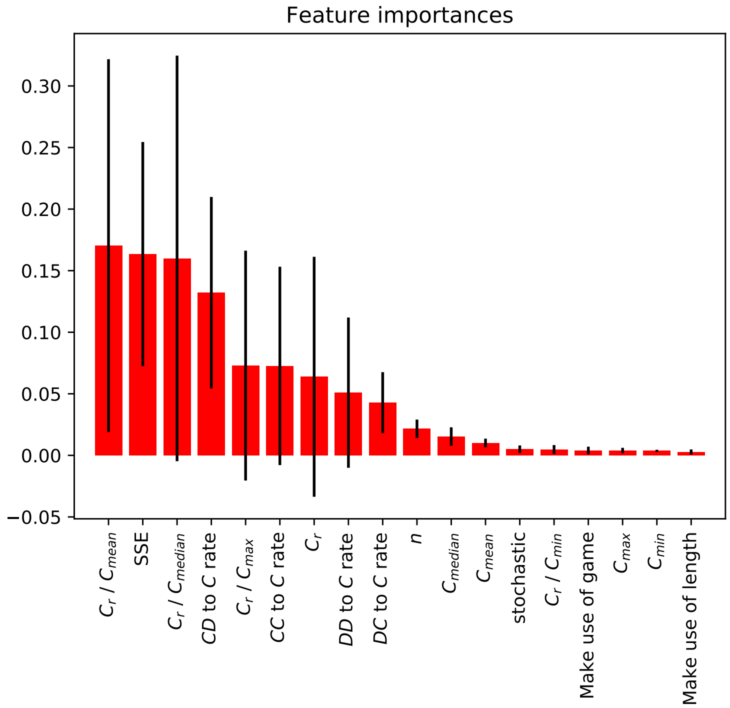


Figure 9: Importance of features in standard tournaments.

	importance	cluster 1	cluster 2
DC to C rate	0.042803	-0.000321	0.000321
SSE	0.163466	-0.000317	0.000317
n	0.021574	-0.000159	0.000159
C_r / C_{mean}	0.170318	-0.000112	0.000112
CC to C rate	0.072634	-0.000091	0.000091
stochastic	0.004977	-0.000064	0.000064
C_{min}	0.003700	-0.000037	0.000037
C_{mean}	0.010065	-0.000031	0.000031
C_r / C_{min}	0.004783	-0.000022	0.000022
Make use of length	0.002762	-0.000016	0.000016
Make use of game	0.003964	-0.000002	0.000002
C_r / C_{max}	0.072897	0.000028	-0.000028
C_{median}	0.015318	0.000028	-0.000028
C_{max}	0.003928	0.000033	-0.000033
DD to C rate	0.050942	0.000062	-0.000062
C_r	0.063826	0.000088	-0.000088
CD to C rate	0.132096	0.000277	-0.000277
C_r / C_{median}	0.159947	0.000346	-0.000346

Figure 10: The average contribution of each factor in standard tournaments based on the tree interpreter method.

The same approach is applied to the tournament types individually and in the merged performances fom the the tournament types. More specifically, in noisy tournament the normalised rank against the median score is given in Figure 11, based on the silhouette coefficient the number of clusters is 3 in noisy tournaments. The performance of each cluster is a follows, being in cluster 1 corresponds to low performance, 2 in medium and 3 high, Table 10.

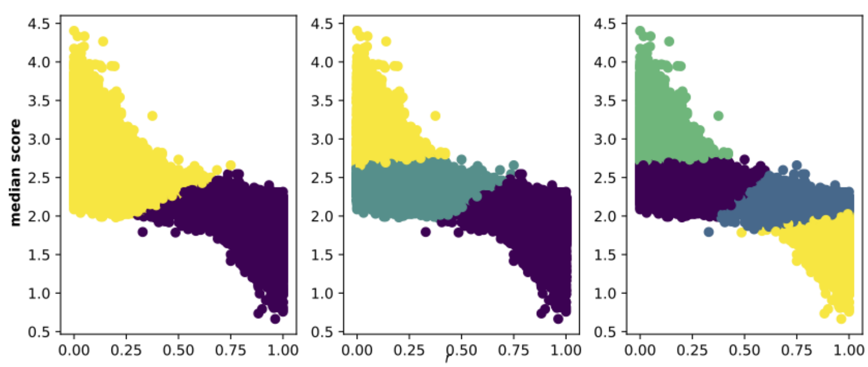


Figure 11: Clustering trials for noisy tournaments. A number of 3 clusters has been chosen with a silhouette score of 0.49 against 2 clusters with a coefficient of 0.47 and 4 with 0.45. respectively.

The two factor that appear to affect the performance in noisy tournaments, based on the average importance, are the cooperating ratio compared to the mean and the mean (Figure 12) with both having a strong effect to the contribution of being in the successful cluster Table 13.

Cluster	$\bar{r}$	median score
1	0.788	2.096
2	0.316	2.290
3	0.097	3.012

Table 10: Median normalised rank and median score of each cluster in noisy tournaments.

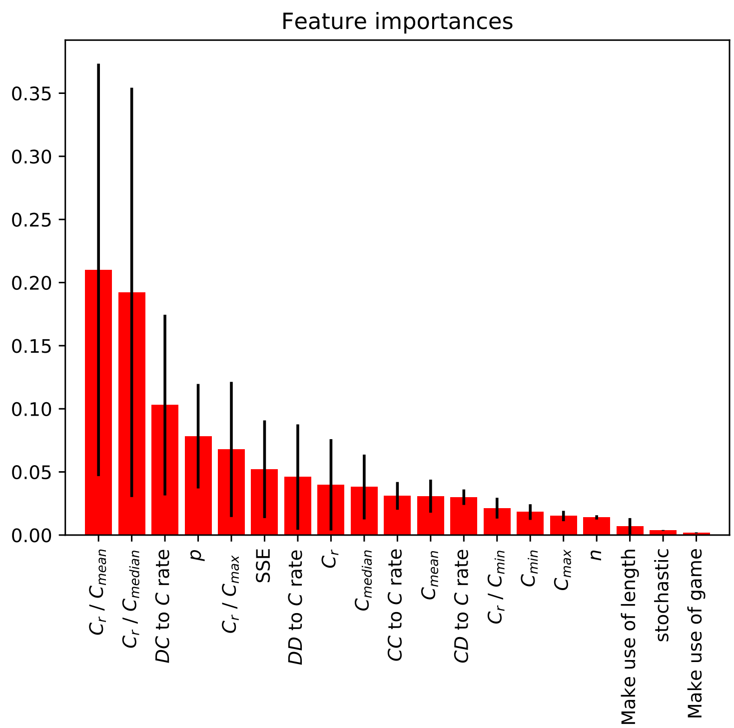


Figure 12: Importance of factors from Table 8 in noisy tournaments.

	importance	cluster 1	cluster 2	cluster 3
C_r / C_{mean}	0.21004	-0.00025	0.00008	0.00017
C_r / C_{median}	0.19222	0.00029	-0.00044	0.00015
C_{median}	0.03806	0.00005	-0.00014	0.00009
p	0.07831	0.00000	-0.00008	0.00008
C_r / C_{max}	0.06782	0.00008	-0.00016	0.00008
SSE	0.05214	-0.00002	-0.00003	0.00005
DD to C rate	0.04593	-0.00010	0.00006	0.00004
n	0.01400	-0.00002	-0.00002	0.00004
DC to C rate	0.10295	-0.00042	0.00041	0.00001
C_{max}	0.01509	0.00006	-0.00007	0.00001
C_r / C_{min}	0.02130	0.00003	-0.00003	0.00000
stochastic	0.00367	0.00001	-0.00001	0.00000
Make use of game	0.00175	0.00005	-0.00003	-0.00001
C_r	0.03979	0.00015	-0.00014	-0.00001
C_{min}	0.01823	0.00009	-0.00005	-0.00004
CC to C rate	0.03104	0.00005	0.00001	-0.00006
Make use of length	0.00685	0.00004	0.00004	-0.00008
C_{mean}	0.03081	0.00007	0.00002	-0.00009
CD to C rate	0.03000	0.00013	0.00001	-0.00014

Figure 13: The average contribution of each factor in standard tournaments based on the tree interpreter method.

Similarly for probabilistic ending tournaments, the cluster are given by igure.. Winner is n = 2 with. 'Clusters: n = 3': 0.51128060607311, 'Clusters: n = 2': 0.6705714418492148, 'Clusters: n = 4': 0.4980365221065629

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] 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.

[5] 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.

[6] 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.

[7] 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.

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

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

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

[11] 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.

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

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

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

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

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

[17] 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.

[18] 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.

[19] 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.

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

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

[22] 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.

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

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.