Understanding responses to environments for the Prisoner's Dilemma: A machine learning approach

Nikoleta E. Glynatsi

Month 2020

Submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy.

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD
School of Mathematics
Ysgol Mathemateg

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Chapter 2

A systematic literature review of the Prisoner's Dilemma.

The Prisoner's Dilemma is a well known game used since the 1950's as a framework for studying the emergence of cooperation; a topic of continuing interest for mathematical, social, biological and ecological sciences. The iterated version of the game, the Iterated Prisoner's Dilemma, attracted attention in the 1980's after the publication of the "The Evolution of Cooperation" and has been a topic of pioneering research ever since. The aim of this paper is to provide a systematic literature review on Prisoner's Dilemma related research. This is achieved by reviewing selected pieces of work and partition the literature into five different sections with each reviewing a different aspect of research. The questions answered in this manuscript are (1) what are the research trends in the field (2) what are the already existing results within the field.

2.1 Introduction

Based on the Darwinian principle of survival of the fittest cooperative behaviour should not be favoured, however, cooperation is plentiful in nature. A paradigm of understanding the emergence of these behaviours is a particular two player non-cooperative game called the Prisoner's Dilemma (PD), originally described in [22].

In the PD each player has two choices, to either be selfless and cooperate or to be selfish and defect. Each decision is made simultaneously and independently. The utility of each player is influenced by its own behaviour, and the behaviour of the opponent. Both players do better if they choose to cooperate than if both choose to defect. However, a player has the temptation to deviate as that player will receive a higher payoff than that of mutual cooperation. Players' payoffs are generally represented by (2.1). Both players receive a reward for mutual cooperation, R, and a payoff P for mutual defection. A player that defects while the other cooperates receives a payoff of T, whereas the cooperator receives S. The dilemma exists due to constraints (2.2) and (2.3).

$$\begin{pmatrix} R & S \\ T & P \end{pmatrix} \tag{2.1}$$

$$T > R > P > S \tag{2.2}$$

$$2R > T + S \tag{2.3}$$

Another common representation of the payoff matrix is given by (2.4), where b is the benefit of the altruistic behaviour and c it's its cost (constraints (2.2) and (2.3) still hold).

$$\begin{pmatrix} b - c & c \\ b & 0 \end{pmatrix} \tag{2.4}$$

Constraints (2.2-2.3) guarantee that it never benefits a player to cooperate, indeed mutual defection is a Nash equilibrium. However, when the game is studied in a manner where prior outcome matters, defecting is no longer necessarily the dominant choice.

The repeated form of the game is called the Iterated Prisoner's Dilemma (IPD) and theoretical works have shown that cooperation can emerge once players interact repeatedly. Arguably, the most important of these works is Robert Axelrod's "The Evolution of Cooperation" [**?**]. In his book Axelrod reports on a series of computer tournaments he organised. In these tournaments academics from several fields were invited to design computer strategies to compete. Axelrod's work showed that greedy strategies did very poorly in the long run whereas altruistic strategies did better. "The Evolution of Cooperation" is considered a milestone in the field but it is not the only one. On the contrary, the PD has attracted attention ever since the game's origins.

This manuscript presents a qualitative description of selected pieces of work. These have been separated into five sections, each reviewing a different aspect of research. The topics reviewed at each section are the following:

- section 2.2, **Origins of the Prisoner's Dilemma**.
- section 2.3, **Axelrod's tournaments and intelligent design of strategies**.
- section 2.4, **Evolutionary dynamics**
- section 2.5, **Structured strategies and training**.
- section 2.6, **Software**.

The aim of this work is to provide a concrete summary of the existing literature on the PD. This is done to provide a review which will allow the research community to understand overall trends in the field, and already existing results.

2.2 Origins of the prisoner's dilemma

The origin of the PD goes back to the 1950s in early experiments conducted at RAND [22] to test the applicability of games described in [?]. The game received its name later the same year. According to [?], Albert W. Tucker (the PhD supervisor of John Nash [?]), in an attempt to deliver the game with a story during a talk described the players as prisoners and the game has been known as the Prisoner's Dilemma ever since.

The early research on the IPD was limited. The only source of experimental results was through human subject research where pairs of participants simulated plays of the game. Human subject research had disadvantages. Humans could behave randomly and in several experiments both the size and the background of the individuals were different, thus comparing results of two or more studies became difficult.

The main aim of these early research experiments was to understand how conditions such as the gender of the participants [?, ?, ?], the physical distance between the participants [?], the effect of their opening moves [?] and even how the experimenter, by varying the tone of their voice and facial expressions [?], could influence the outcomes and subsequently the emergence of cooperation. An early figure that sought out to understand several of these conditions was the mathematical psychologist Anatol Rapoport. The results of his work are summarised in [?].

Rapoport was also interested in conceptualising strategies that could promote international cooperation. Decades later he would submit the winning strategy (Tit for Tat) of the first computer tournament, run by Robert Axelrod. In the next section these tournaments, and several strategies that were designed by researchers, such as Rapoport, are introduced.

2.3 Axelrod's tournaments and intelligently designed strategies

As discussed in Section 2.2, before 1980 a great deal of research was done in the field, however, as described in [?], the political scientist Robert Axelrod believed that there was no clear answer to the question of how to avoid conflict, or even how an individual should play the game. Combining his interest in artificial intelligence and political science Axelrod created a framework for exploring these questions using computer tournaments. Axelrod asked researchers to design a strategy with the purpose of wining an IPD tournament. This section covers Axelrod's original tournaments as well as research that introduced new intelligently designed strategies.

Axelrod's tournaments made the study of cooperation of critical interest. As described in [?], "Axelrod's "new approach" has been extremely successful and immensely influential in casting light on the conflict between an individual and the collective rationality reflected in the choices of a population whose members are unknown and its size unspecified, thereby opening a new avenue of research". In a collaboration with a colleague, Douglas Dion, Axelrod in [?] summarized a number of works that were immediately inspired from the "Evolution of Cooperation", and [?] gives a review of tournaments that have been conducted since the originals.

The first reported computer tournament took place in 1980 [12]. A total of 13 strategies were submitted, written in the programming languages Fortran or Basic. Each competed in a 200 turn match against all 12 opponents, itself and a player that played randomly (called **Random**). This type of tournament is referred to as a round robin. The tournament was repeated 5 times to get a more stable estimate of the scores for each pair of play. Each participant knew the exact number of turns and had access to the full history of each match. Furthermore, Axelrod performed a preliminary tournament and the results were known to the participants. This preliminary tournament is mentioned in [12] but no details were given. The payoff values used for equation (2.1) were $R = 3, P = 1, T = 5$ and $S = 0$. These values are commonly used in the literature and unless specified will be the values used in the rest of the works described here.

The winner of the tournament was determined by the total average score and not by the number of matches won. The strategy that was announced the winner was the strategy submitted by Rapoport, **Tit For Tat**. The success of Tit for Tat came as a surprise. It was not only the simplest submitted strategy, it would always cooperates on the first round and then mimic the opponent's previous move, but it had also won the tournament even though it could never beat any player it was interacting with.

In order to further test the results Axelrod performed a second tournament in 1980 [13]. The second tournament received much more attention and had a total of 62 entries. The participants knew the results of the previous tournament and the rules were similar with only a few alterations. The tournament was repeated 5 times and the length of each match was not known to the participants. Axelrod intended to use a fixed probability (refereed to as 'shadow of the future' [**?**]) of the game ending on the next move. However, 5 different number of turns were selected for each match 63, 77, 151, 308 and 401, such that the average length would be around 200 turns.

Nine of the original participants competed again in the second tournament. Two strategies that remained the same were Tit For Tat and **Grudger**. Grudger is a strategy that will cooperate as long as the opponent does not defect, submitted by James W. Friedman. The name Grudger was give to the strategy in [**?**], though the strategy goes by many names in the literature such as, Spite [17], Grim Trigger [16] and Grim [50]. New entries in the second tournament included **Tit for Two Tats** submitted by John Maynard Smith and **KPavlovC**. KPavlovC, is also known as Simpleton [**?**], introduced by Rapoport or just Pavlov [40]. The strategy is based on the fundamental behavioural mechanism win-stay, lose-shift. Pavlov is heavily studied in the literature and similarly to Tit for Tat it is used in tournaments today and has had many variants trying to build upon it's success, for example **PavlovD** and **Adaptive Pavlov** [**?**].

Despite the larger size of the second tournament none of the new entries managed to outperform the simpler designed strategy. The winner was once again Tit for Tat. Axelrod deduced the following guidelines for a strategy to perform well:

- The strategy would start of by cooperating.
- It would forgive it's opponent after a defection.
- It would always be provoked by a defection no matter the history.
- It was simple.

The success of Tit for Tat, however, was not unquestionable. Several papers showed that stochastic uncertainties severely undercut the effectiveness of reciprocating strategies and such stochastic uncertainties have to be expected in real life situations [?]. For example, in [38] it is proven that in an environment where **noise** (a probability that a player's move will be flipped) is introduced two strategies playing Tit for Tat receive the same average payoff as two Random players. Hammerstein, pointed out that if by mistake, one of two Tit for Tat players makes a wrong move, this locks the two opponents into a hopeless sequence of alternating defections and cooperations [45].

The poor performance of the strategy in noisy environments was also demonstrated in tournaments. In [18, 21] round robin tournaments with noise were performed, and Tit For Tat did not win. The authors concluded that to overcome the noise more generous strategies than Tit For Tat were needed. They introduced the strategies **Nice and Forgiving** and **OmegaTFT** respectively.

A second type of stochastic uncertainty is misperception, where a player's action is made correctly but it is recorded incorrectly by the opponent. In [?], a strategy called **Contrite Tit for Tat** was introduced that was more successful than Tit for Tat in such environments. The difference between the strategies was that Contrite Tit for Tat was not so fast to retaliate against a defection.

Several works extended the reciprocity based approach which has led to new strategies. For example Gradual [17] which was constructed to have the same qualities as those of Tit for Tat except one, **Gradual** had a memory of the game since the beginning of it. Gradual recorded the number of defections by the opponent and punished them with a growing number of defections. It would then enter a calming state in which it would cooperates for two rounds. In a tournament of 12 strategies, including both Tit for Tat and Pavlov, Gradual managed to outperformed them all. A strategy with the same intuition as Gradual is **Adaptive Tit for Tat** [48]. Adaptive Tit for Tat does not keep a permanent count of past defections, it maintains a continually updated estimate of the opponent's behaviour, and uses this estimate to condition its future actions. In the exact same tournament as in [17] with now 13 strategies Adaptive Tit for Tat ranked first.

Another extension of strategies was that of teams of strategies [?, ?, ?] that collude to increase one member's score. In 2004 Graham Kendall led the Anniversary Iterated Prisoner's Dilemma Tournament with a total of 223 entries. In this tournament participants were allowed to submit multiple strategies. A team from the University of Southampton submitted a total of 60 strategies [?]. All these were strategies that had been programmed with a recognition mechanism by default. Once the strategies recognised one another, one would act as leader and the other as a follower. The follower plays as a **Cooperator**, cooperates unconditionally and the leader would play as a **Defector** gaining the highest achievable score. The followers would defect unconditionally against other strategies to lower their score and help the leader. The result was that Southampton had the top three performers. Nick Jennings, who was part of the team, said that "We developed ways of looking at the Prisoner's Dilemma in a more realistic environment and we devised a way for computer agents to recognise and collude with one another despite the noise. Our solution beats the standard Tit For Tat strategy" [?].

2.3.1 Memory one Strategies

A set of strategies that have received a lot of attention in the literature are **memory one** strategies. In [?], Nowak and Sigmund proposed a structure for studying simple strategies that remembered only the previous turn, and moreover, only recorded the move of the opponent. These are called **reactive** strategies and they can be represented by using three parameters (y, p_1, p_2), where y is the probability to cooperate in the first move, and p_1 and p_2 the conditional probabilities to cooperate, given that the opponent's last move was a cooperation or a defection. For example Tit For Tat is a reactive strategy and it can be written as $(1, 1, 0)$. Another reactive strategy well known in the literature is **Generous Tit for Tat** [41].

In [?], Nowak and Sigmund extended their work to include strategies which consider the entire history of the previous turn to make a decision. These are called **memory one** strategies. If only a single turn of the game is taken into account and depending on the simultaneous moves of the two players there are only four possible states that the players could be in. These are:

- Both players cooperated, denoted as CC.
- First player cooperated while the second one defected, denoted as CD.
- First player defected while the second one cooperated, denoted as DC.
- Both players defected, denoted as DD.

Thus a memory one strategy can be denoted by the probabilities of cooperating after each state and the probability of cooperating in the first round, (y, p_1, p_2, p_3, p_4). For example Pavlov's memory one representation is $(1, 1, 0, 0, 1)$.

Memory one strategies made an impact when a specific set of memory one strategies were introduced called **Zero-determinant** (ZD) [43]. The American Mathematical Society's news section [?] stated that "the world of game theory is currently on fire" and in [47] it was stated that "Press and Dyson have fundamentally changed the viewpoint on the Prisoner's Dilemma". ZD are a set of extortionate strategies that can force a linear relationship between the long-run scores of both themselves and the opponent, therefore ensuring that the opponent will never do better than them.

Press and Dyson's suggested ZD strategies were the dominant family of strategies in the IPD. Moreover, they argued that memory is not beneficial. In [6, ?, 28, ?, ?, ?, 30, ?, 47] the effectiveness of ZD strategies is questioned. In [6], it was shown that ZD strategies are not evolutionary stable, and in [47] a more generous set of ZDs, the **Generous ZD**, were shown to outperform the more extortionate ZDs. Finally, in [?, ?, 30, ?], the 'memory does not benefit a strategy' statement was questioned. A set of more complex strategies, strategies that take in account the entire history set of the game, were trained and proven to be more stable than ZD strategies.

This section covered the original computer tournaments of Axelrod and the early success of Tit For Tat in these tournaments. Though Tit For Tat was considered to be the most robust basic strategy, reciprocity was found to not be enough in environments with uncertainties. There are at least two properties, that have been discussed in this section, for coping with such uncertainties; generosity and contrition. Generosity is letting a percentage of defections go unpunished, and contrition is lowering a strategy's readiness to defect following an opponent's defection.

In the later part of this section a series of new strategies which were built on the basic reciprocal approaches were presented, followed by the infamous memory one strategies, the zero-determinant strategies. Though the ZDs can be proven to be robust in pairwise interactions they were found to be lacking in evolutionary settings and in computer tournaments. Evolutionary settings and the emergence of cooperation under natural selection are covered in the next section.

2.4 Evolutionary dynamics

As yet, the emergence of cooperation has been discussed in the contexts of the one shot PD game and the IPD round robin tournaments. In the PD it is proven that cooperation will not emerge, furthermore, in a series of influential works Axelrod demonstrated that reciprocal behaviour favours cooperation when individuals interact repeatedly. But does natural selection favours cooperation? Understanding the conditions under which natural selection can favour cooperative behaviour is important in understanding social behaviour amongst intelligent agents [**?**].

Imagine a mixed population of cooperators and defectors where every time two individuals meet they play a game of PD. In such population the average payoff for defectors is always higher than cooperators. Under natural selection the frequency of defectors will steadily increase until cooperators become extinct. Thus natural selection favours defection in the PD (Figure 2.1). However, there are several mechanisms that allow the emergence of cooperation in an evolutionary context which will be covered in this section.

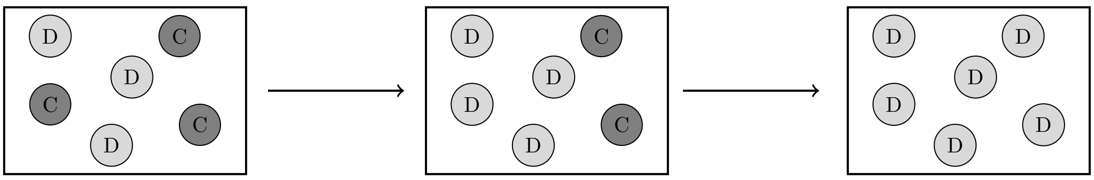


Figure 2.1: Natural selection favours defection in a mixed population of Cooperators and Defectors.

In the later sections of [13], Axelrod discusses an ecological tournament that he performed using the 62 strategies of the second tournament to understand the reproductive success of Tit for Tat. In his ecological tournament the prevalence of each type of strategy in each round was determined by that strategy's success in the previous round. The competition in each round would become stronger as weaker performers were reduced and eliminated. The ecological simulation concluded with a handful of nice strategies dominating the population whilst exploitative strategies had died off as weaker strategies were becoming extinct. This new result led Axelrod to study the IPD in an evolutionary context based on several of the approaches established by the biologist John M. Smith [**?**, **?**, **?**]. John M. Smith was a fundamental figure in evolutionary game theory and a participant of Axelrod's second tournament. Axelrod and the biologist William Donald Hamilton wrote about the biological applications of the evolutionary dynamics of the IPD [**?**] and won the Newcomb-Cleveland prize of the American Association for the Advancement of Science.

In Axelrod's model [14] pairs of individuals from a population played the IPD. The number of interactions between the pairs were not fixed, but there was a probability defined as the importance of the future of the game w, where $0 < w < 1$, that the pair would interact again. In [14] it was shown that for a sufficient high w Tit For Tat strategies would become common and remain common because they were "collectively stable". Axelrod argued that collective stability implied evolutionary stability (ESS) and that when a collectively stable strategy is common in a population and individuals are paired randomly, no other rare strategy can invade. However, Boyd and Lorderbaum in [**?**] proved that if w, the importance of the future of the game, is large enough then no pure strategy is ESS because it can always be invaded by any pair of other strategies. This was also independently proven in [**?**].

All these conclusions were made in populations where the individuals could all interact with each other. In 1992, Nowak and May, considered a structured population where an individual's interactions were limited to its neighbours. More specifically, in [**?**] they explored how local interaction alone can facilitate population wide cooperation in a one shot PD game. The two deterministic strategies Defector and Cooperator, were placed onto a two dimensional square array where the individuals could interact only with the immediate neighbours. The number of immediate neighbours could be either, fourth, six or eight, as shown in Figure 2.2, where each node represents a player and the edges denote whether two players will interact. This topology is refereed to as spatial topology. Each cell of the lattice is occupied by a Cooperator or a Defector and at each generation step each cell owner interacts with its immediate neighbours. The score of each player is calculated as the sum of all the scores the player achieved at each generation. At the start of the next generation, each lattice cell is occupied by the player with the highest score among the previous owner and their immediate neighbours.

Local interactions proved that as long as small clusters of cooperators form, where they can benefit from interactions with other cooperators while avoiding interactions with defectors, global cooperation will continue. Thus, local interactions proved that even for the PD cooperation can emerge. Moreover in [**?**], whilst using the payoff matrix (2.4), it was shown that cooperation will evolve in a structured population as long as the benefit to cost ratio b/c is higher than the number of neighbours. In [**?**], graphs were a probability of rewiring ones connections was considered were studied. The rewire could be with any given node in the graphs and not just with immediate neighbours. Perc et al. concluded that "making new friends" may be an important activity for the successful evolution of cooperation, but also they must be selected carefully and one should keep their number limited.

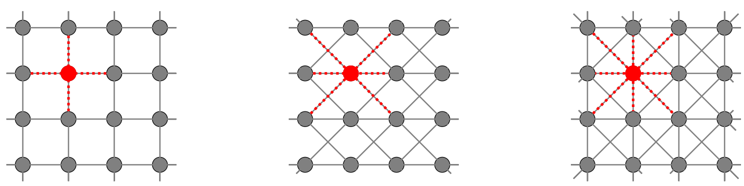


Figure 2.2: Spatial neighbourhoods

Another approach for increasing the likelihood of cooperation by increasing of assortative interactions among cooperative agents, include partner identification methods such as reputation [?, ?, ?], communication tokens [?] and tags [?, ?, ?, ?].

In this section evolutionary dynamics and the emergence of cooperation were reviewed. The following section focuses on strategy archetypes, training methods and strategies obtained from training.

2.5 Structured strategies and training

This section covers strategies that are different to that of intelligent design discussed in Section 2.3. These are strategies that have been **trained** using generic strategy archetypes. For example, in [15] Axelrod decided to explore deterministic strategies that took into account the last 3 turns of the game. As discussed in Section 2.3.1, for each turn there are 4 possible outcomes, CC, CD, DC, DD, thus for 3 turns there are a total of $4 \times 4 \times 4 = 64$ possible combinations. Therefore, the strategy can be defined by a series of 64 C's/D's, corresponding to each combination; this type of strategy is called a lookup table. This lookup table was then trained using a genetic algorithm [?]. During the training process random changes are made to a given lookup table. If the utility of the strategy has increased this change is kept, otherwise not.

In 1996 John Miller considered finite state automata as an archetype [36], more specifically, Moore machines [?]. He used a genetic algorithm to train finite state machines in environments with noise. Miller's results showed that even a small difference in noise (from 1% to 3%) significantly changed the characteristics of the evolving strategies. The strategies he introduced were **Punish Twice**, **Punish Once for Two Tats** and **Punish Twice and Wait**. In [?] finite state automata and genetic algorithms were also used to introduce new strategies. In a series of experiments where the size of the population varied, there were two strategies frequently developed by the training process and more over they were developed only after the evolution had gone on for many generations. These were **Fortess3** and **Fortess4**. Following Miller's work in 1996, the first structured strategies based on neural networks that had be trained using a genetic algorithm was introduced in [?] by Harrald and Fogel. Harrald and Fogel considered a single layered neural network which had 6 inputs. These were the last 3 moves of the player and the opponent, similar to [15]. Neural networks have broadly been used to train IPD strategies since then with genetic algorithms [?, ?, ?] and particle swarm optimization [?].

In [?, ?] both genetic algorithm and particle swarm optimization were used to introduce a series of structured strategies based on lookup tables, finite state machines, neural networks, hidden Markov models [?] and Gambler. Hidden Markov models, are a stochastic variant of a finite state machine and Gamblers are stochastic variants of lookup tables. The structured strategies that arised from the training were put up against a large number of strategies in (1) a Moran process, which is an evolutionary model of invasion and resistance across time during which high performing individuals are more likely to be replicated and (2) a round robin tournament. In a round robin tournament which was simulated using the software [3] and the 200 strategies implemented within the software, the top spots were dominated by the trained strategies of all the archetypes. The top three strategies were **Evolved LookUp 2 2 2**, **Evolved HMM 5** and **Evolved FSM 16**.

In [?] it was demonstrated that these trained strategies would overtake the population in a

Moran process. The strategies evolved an ability to recognise themselves by using a handshake. This recognition mechanism allowed the strategies to resist invasion by increasing the interactions between themselves, an approach similar to the one described in Section 2.4.

Throughout the different methods of training that have been discussed in this section, a spectrum of structured strategies can be found. Differentiating between strategies is not always straightforward. It is not obvious looking at a finite state diagram how a machine will behave, and many different machines, or neural networks can represent the same strategy. For example Figure 2.3 shows two finite automata and both are a representation of Tit for Tat.

(a) Tit for Tat as a finite state machine with 1 state.

(b) Tit for Tat as a finite state machine with 2 states.

Figure 2.3: Finite state machine representations of Tit for Tat. A machine consists of transition arrows associated with the states. Each arrow is labelled with A/R where A is the opponent's last action and R is the player's response. Finite state machines consist of a set of internal states. In (a) Tit for Tat finite state machine consists of 1 state and in (b) of 2.

To allow for identification of similar strategies a method called fingerprinting was introduced in [?]. The method of fingerprinting is a technique for generating a functional signature for a strategy [8]. This is achieved by computing the score of a strategy against a spectrum of opponents. The basic method is to play the strategy against a probe strategy with varying noise parameters. In [?] Tit for Tat is used as the probe strategy. In Figure 2.4 an example of Pavlov's fingerprint is given. Fingerprinting has been studied in depth in [8, ?, ?, ?]. Another type of fingerprinting is the transitive fingerprint [3]. The method represents the cooperation rate of a strategy against a set of opponents over a number of turns. An example of a transitive fingerprint is given in Figure 2.5.

This section covered structured strategies and training methods. In the following section software that has been developed with main aim simulating the IPD is presented.

2.6 Software

The research of the IPD heavily relies on software. This is to be expected as computer tournaments have become the main means of simulating the interactions in an IPD game. Many academic fields suffer from lack of source code availability and the IPD is not an exception. Several of the tournaments that have been discussed so far were generated using computer code, though not all of the source code is available. The code for Axelrod's original tournament is known to be lost and moreover for the second tournament the only source code available is the code for the 62 strategies (found on Axelrod's personal website [?]).

Several projects, however, are open, available and have been used as research tools or educational platforms over the years. Two research tools [1, 3] and two educational tools [?, ?] are briefly mentioned here. Both [1, 3] are open source projects. The "Game of Trust" [?] is an

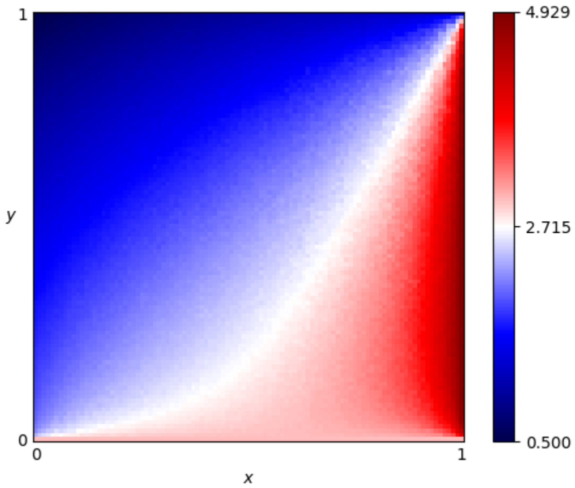


Figure 2.4: Pavlov fingerprinting with Tit for Tat used as the probe strategy. Figure was generated using [3].

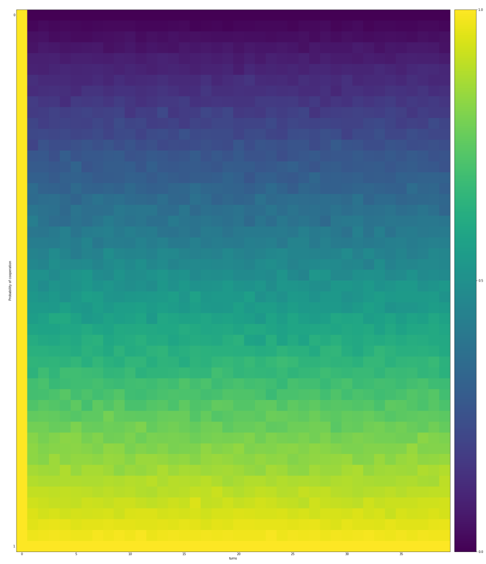


Figure 2.5: Transitive fingerprint of Tit for Tat against a set of 50 random opponents.

on-line, graphical user interface educational platform for learning the basics of game theory, the IPD and the notion of strategies. It attracted a lot of attention due to being "well-presented with scribble-y hand drawn characters" [?] and "a whole heap of fun" [?]. Finally [?] is a personal project written in PHP. It is a graphical user interface that offers a big collection of strategies and allows the user to try several matches and tournament configurations.

PRISON [1] is written in the programming language Java and a preliminary version was launched on 1998. It was used by its authors in several publications, such as [17], which introduced Gradual, and [?]. The project includes a good number of strategies from the literature but unfortunately the last update of the project dates back to 2004. Axelrod-Python [3] is a software used by [?, ?, ?, ?]. It is written in the programming language Python following best practice approaches [5, 19] and contains the largest collection of strategies, known to the author. The strategy list of the project has been cited by publications [?, ?, ?].

2.7 Conclusion

This manuscript presented a literature review on the Iterated Prisoner's Dilemma. The opening sections focused on research trends and published works of the field, followed by a presentation of research and educational software. More specifically, Section 2.2 covered the early years of research. This was when simulating turns of the game was only possible with human subject research. Following the early years, the pioneering tournaments of Axelrod were introduced in Section 2.3. Axelrod's work offered the field an agent based game theoretic framework to study the IPD. In his original papers he asked researchers to design strategies to test their performance with the new framework. The winning strategy of both his tournaments was Tit for Tat. The strategy however came with limitations which were explored by other researchers, and new intelligently designed strategies were introduced in order to surpass Tit for Tat with some contributions such as Pavlov and Gradual.

Soon researchers came to realise that strategies should not just do well in a tournament setting but should also be evolutionary robust. Evolutionary dynamic methods were applied to many works in the field, and factors under which cooperation emerges were explored, as described in Section 2.4. This was not done only for unstructured populations, where all strategies in the population can interact with each other, but also in population where interactions were limited to only strategies that were close to each other. In such topologies it was proven that even in the one shot game, cooperation can indeed emerge.

Evolutionary approaches can offer many insights in the study of the PD. In evolutionary settings strategies can learn to adapt and take over population by adjusting their actions; such algorithms can be applied so that evolutionarily robust strategies can emerge. Algorithms and structures used to train strategies in the literature were covered in Section 2.5. From these training methods several strategies are found, and to be able to differentiate between them fingerprinting was introduced. The research of best play and cooperation has been going on since the 1950s, and for simulating the game software has been developed along the way. This software has been briefly discussed in Section 2.6.

The study of the PD is still an ongoing field research where new variants and new structures of strategies are continuously being explored [?]. The game now serves as a model in a wide range of applications, for example in medicine and the study of cancer cells [?, ?], as well as in social situations and how they can be driven by rewards [?]. New research is still ongoing for example in evolutionarily dynamics on graphs [?, ?, ?].

Chapter 3

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

The research reported in this Chapter has lead in a manuscript, entitled:
"Properties of winning Iterated Prisoner's Dilemma strategies"
Available at: `https://arxiv.org/abs/2001.05911`
Associated data set: [?]
Axerod-Python library version: 3.0.0

The manuscript's abstract is the following:

Researchers have explored the performance of Iterated Prisoner's Dilemma strategies for decades: from the celebrated performance of Tit for Tat, to the introduction of the zero-determinant strategies, to the use of sophisticated learning structures such as neural networks, many new strategies have been introduced and tested in a variety of tournaments and population dynamics. Typical results in the literature, however, rely on performance against a small number of somewhat arbitrarily selected strategies in a very small number of tournaments, casting doubt on the generalisability of conclusions. We analyze a large collection of 195 typically known strategies in 45686 tournaments, present the top performing strategies across multiple tournament types, and distill their salient features. The results show that there is not yet a single strategy that performs well in diverse Iterated Prisoner's Dilemma scenarios. Nevertheless there are several properties that heavily influence the best performing strategies, refining the properties described by R. Axelrod in light of recent and more diverse opponent populations. These are: be nice, be provocable and contrite, be a little envious, be clever, and adapt to the environment, which includes the parameters of the tournament (e.g. noise) and the population of opponents. More precisely, we find that strategies perform best when their probability of cooperation matches the total tournament population's aggregate cooperation probabilities, or

a proportion thereof in the case of noisy and probabilistically ending tournaments, and that the manner in which a strategy achieves the ideal cooperation rate is crucial. The features of high performing strategies reveal why strategies such as Tit For Tat performed historically well in tournaments and why zero-determinant strategies typically do not fare well in tournament settings.

The differences between the Chapter and the manuscript include ....

3.1 Introduction

As stated in Chapter 1 conceptualising strategies and understanding the best way of playing the game has been of interest to the scientific community since the formulation of the game, and in Chapter 2 it was established that following the computer tournaments of Axelrod in the 1980's, a strategy's performance in a round robin computer tournament became a common evaluation technique for newly designed strategies.

The winner of both of Axelrod's tournaments [12, 13] was the simple strategy Tit For Tat and Axelrod concluded that the strategy's robustness was due to four properties, which he adapted into four suggestions on doing well in an IPD:

- Do not be envious by striving for a payoff larger than the opponent's payoff
- Be "nice"; Do not be the first to defect
- Reciprocate both cooperation and defection; Be provocable to retaliation and forgiveness
- Do not be too clever by scheming to exploit the opponent

Compared to the works reviewed in Chapter 2, where typically a few selected or introduced strategies are evaluated on a small number of tournaments and/or small number of opponents, this Chapter evaluates the performance of 195 strategies in 45686 tournaments. Furthermore a large portion of the strategies used in this Chapter are drawn from the known and named strategies in IPD literature, including many previous tournament winners, in contrast to other work that may have randomly generated many essentially arbitrary strategies (typically restrained to a class such as memory-one strategies, or those of a certain structural form such as finite state machines or deterministic memory two strategies). Additionally, our tournaments come in a number of variations including standard tournaments emulating Axelrod's original tournaments, tournaments with noise, probabilistic match length, and both noise and probabilistic match length. This diversity of strategies and tournament types yields new insights and tests earlier claims in alternative settings against known powerful strategies.

The later part of the Chapter evaluates the impact of features on the performance of the strategies using modern machine learning techniques. These features include measures regarding a strategy's behaviour and measures regarding the tournaments. The outcomes of our work reinforce the discussion started by Axelrod, and it concludes that the properties of a successful strategy in the IPD are:

- ~~Do not be envious~~ Be a little bit envious

- Be "nice"; Do not be the first to defect
- Reciprocate both cooperation and defection; Be provocable and forgiving
- ~~Do not be too clever~~ It's ok to be clever
- Adapt to the environment; Adjust to the mean population cooperation

The Chapter is structured as follows:

- The different tournament types as well as the data collection, made possible due to an open source library called Axelrod-Python (APL), are covered in Section 3.2.
- Section 3.3, focuses on the best performing strategies for each type of tournament and overall.
- Section 3.4, explores the traits which contribute to good performance

3.2 Data collection

The data set generated for this manuscript was created with APL version 3.0.0. APL allows for different types of IPD computer tournaments to be simulated and contains a large list of strategies. Most of these are strategies described in the literature with a few exceptions of strategies that have been contributed specifically to the package. This paper makes use of 195 strategies implemented in version 3.0.0. A list of the strategies is given in the Appendix A.1. Although APL features several tournament types, this work considers standard, noisy, probabilistic ending, and noisy probabilistic ending tournaments.

Standard tournaments are tournaments similar to that of Axelrod's well-known tournaments [12]. There are N strategies which all play an iterated game of n number of turns against each other. Note that self-interactions are not included. Similarly, *noisy tournaments* have N strategies and n number of turns, but at each turn there is a probability p_n 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 p_e. Finally, *noisy probabilistic ending* tournaments have both a noise probability p_n and an ending probability p_e. For smoothing the simulated results a tournament is repeated for k number of times. This was allowed to vary in order to evaluate the effect of smoothing. The winner of each tournament is based on the average score a strategy achieved and not by the number of wins.

The process of collecting tournament results is described by Algorithm 1. For each trial a random size N is selected, and from the 195 strategies 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 3.1.

The source code for the data collection, as well as the source code for the analysis, which will be discussed in the following sections, have been written following best practices [5, 19] and is available here.

A total of 11420 trials of Algorithm 1 have been run. For each trial the results for 4 different

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_n	probability of flipping action at each turn	0	1
p_e	probability of match ending in the next turn	0	1

Table 3.1: Data collection; parameters' values

Algorithm 1: Data collection Algorithm

foreach *seed* $\in [0, 11420]$ **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 $\in [n_{min}, n_{max}]$;
  $p_n \leftarrow$ randomly select float $\in [p_{n\,min}, p_{n\,max}]$;
  $p_e \leftarrow$ randomly select float $\in [p_{e\,min}, p_{e\,max}]$;

  result standard $\leftarrow$ Axelrod.tournament(players$, n, k$);
  result noisy $\leftarrow$ Axelrod.tournament(players$, n, p_n, k$);
  result probabilistic ending $\leftarrow$ Axelrod.tournament(players$, p_e, k$);
  result noisy probabilistic ending $\leftarrow$ Axelrod.tournament(players$, p_n, p_e, k$);

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

tournaments were collected, thus a total of 45686 (11420×4) tournament results have been retrieved. Each tournament outputs a result summary in the form of Table 3.2. Each strategy have participated on average in 5154 tournaments of each type. The strategy with the maximum participation in each tournament type is Inverse Punisher with 5639 entries. The strategy with the minimum entries is EvolvedLookerUp 1 1 1 which was selected in 4693 trials.

A result summary (Table 3.2) has N number of rows because each row contains information for each strategy that participated in the tournament. The information includes the strategy's rank, median score, the rate with which the strategy cooperated (C_r), its match win count, and the probability that the strategy cooperated in the opening move. Moreover, the probabilities 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. The *normalised rank* feature that is manually added. The rank R of a given strategy can vary between 0 and $N-1$. Thus, the normalised rank, denoted as r, is calculated as a strategy's rank divided by $N-1$.

						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 3.2: Output result of a single tournament.

3.3 Top ranked strategies

The performance of each strategy is evaluated in four tournament types, as presented in Section 3.2, followed by an evaluation of their performance over all the 45686 simulated tournaments of this work. Each strategy participated in multiple tournaments of the same type (on average 5154). For example TIT FOR TAT participated in a total of 5114 tournaments of each type. The strategy's normalised rank distribution in these is given in Figure 3.1. A value of $r = 0$ corresponds to a strategy winning the tournament where a value of $r = 1$ corresponds to the strategy coming last. Because of the strategies' multiple entries their performance is evaluated based on the *median normalised rank* denoted as $\bar{r}$.

The top 15 strategies for each tournament type based on $\bar{r}$ are given in Table 3.3. The data collection process was designed such that the probabilities of noise and ending of the match varied between 0 and 1. However, commonly used values of these probabilities are values are often less than 0.1. Thus, Table 3.3 also includes the top 15 strategies in noisy tournaments with $p_n < 0.1$ and probabilistic ending tournaments with $p_e < 0.1$.

The r distributions for the top ranked strategies of Table 3.3 are given by Figure 3.2.

In standard tournaments 10 out of the 15 top strategies are introduced in [27]. These are strategies based on finite state automata (FSM), hidden markov models (HMM), artificial neural networks (ANN), lookup tables (LookerUp) and stochastic lookup tables (Gambler) that have been trained using reinforcement learning algorithms (evolutionary and particle swarm algorithms). They have been trained to perform well against the strategies in APL in a standard

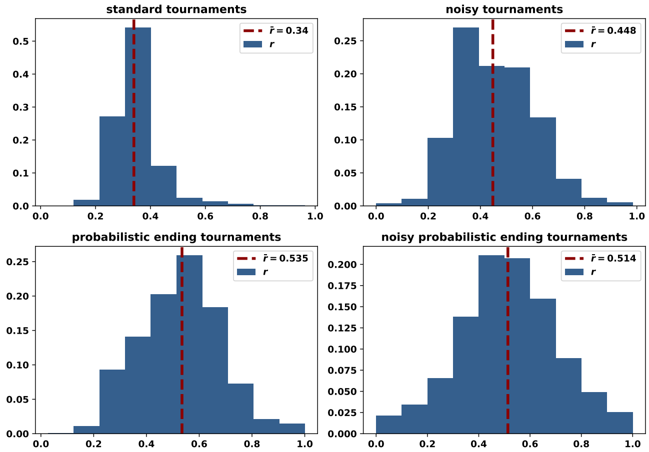
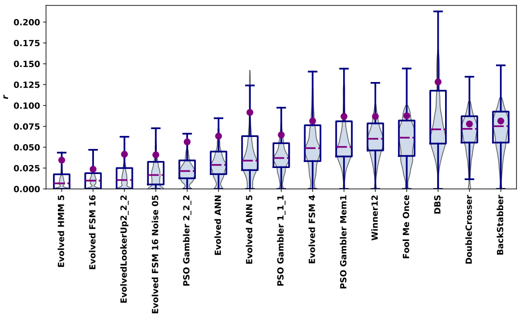


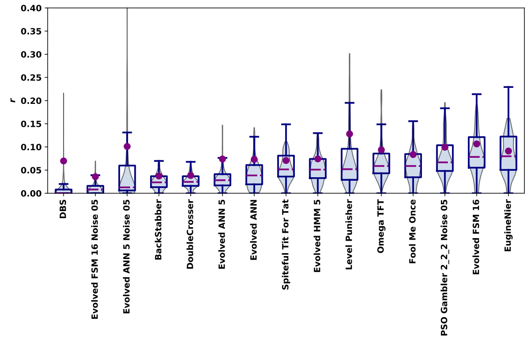
Figure 3.1: TIT FOR TAT's r distribution in tournaments. Lower values of r correspond to better performances. The best performance of the strategy has been in standard tournaments where it achieved a $\bar{r}$ of 0.34.

	Standard		Noisy		Noisy ($p_n < 0.1$)		Probabilistic ending		Probabilistic ending ($p_e < 0.1$)		Noisy probabilistic ending	
	Name	$\bar{r}$	Name	$\bar{r}$	Name	$\bar{r}$	Name	$\bar{r}$	Name	$\bar{r}$	Name	$\bar{r}$
0	Evolved HMM 5	0.007	Grumpy	0.140	DBS	0.000	Fortress4	0.013	Evolved FSM 16	0.000	Alternator	0.304
1	Evolved FSM 16	0.010	e	0.194	Evolved FSM 16 Noise 05	0.008	Defector	0.014	Evolved FSM 16 Noise 05	0.013	ϕ	0.310
2	EvolvedLookerUp2 2 2	0.011	Tit For 2 Tats	0.206	Evolved ANN 5 Noise 05	0.013	Better and Better	0.016	MEM2	0.027	e	0.312
3	Evolved FSM 16 Noise 05	0.017	Slow Tit For Two Tats	0.210	BackStabber	0.024	Tricky Defector	0.019	Evolved HMM 5	0.044	π	0.317
4	PSO Gambler 2 2 2	0.021	Cycle Hunter	0.215	DoubleCrosser	0.025	Fortress3	0.022	EvolvedLookerUp2 2 2	0.049	Limited Retaliate	0.353
5	Evolved ANN	0.029	Risky QLearner	0.222	Evolved ANN 5	0.028	Gradual Killer	0.025	Spiteful Tit For Tat	0.060	Anti Tit For Tat	0.354
6	Evolved ANN 5	0.034	Retaliate 3	0.229	Evolved ANN	0.038	Aggravater	0.028	Nice Meta Winner	0.068	Limited Retaliate 3	0.356
7	PSO Gambler 1 1 1	0.037	Cycler CCCCCD	0.235	Spiteful Tit For Tat	0.051	Raider	0.031	NMWE Finite Memory	0.069	Retaliate 3	0.356
8	Evolved FSM 4	0.049	Retaliate 2	0.239	Evolved HMM 5	0.051	Cycler DDC	0.045	NMWE Deterministic	0.070	Retaliate	0.357
9	PSO Gambler Mem1	0.050	Defector Hunter	0.240	Level Punisher	0.052	Hard Prober	0.051	Grudger	0.070	Retaliate 2	0.358
10	Winner12	0.060	Retaliate	0.242	Omega TFT	0.059	SolutionB1	0.060	NMWE Long Memory	0.074	Limited Retaliate 2	0.361
11	Fool Me Once	0.061	Hard Tit For 2 Tats	0.250	Fool Me Once	0.059	Meta Minority	0.061	Nice Meta Winner Ensemble	0.076	Hopeless	0.368
12	DBS	0.071	Limited Retaliate 3	0.253	PSO Gambler 2 2 2 Noise 05	0.067	Bully	0.061	EvolvedLookerUp1 1 1	0.077	Arrogant QLearner	0.407
13	DoubleCrosser	0.072	ShortMem	0.253	Evolved FSM 16	0.078	EasyGo	0.071	NMWE Memory One	0.080	Cautious QLearner	0.409
14	BackStabber	0.075	Limited Retaliate	0.257	EugineNier	0.080	Fool Me Forever	0.071	Winner12	0.085	Fool Me Forever	0.418

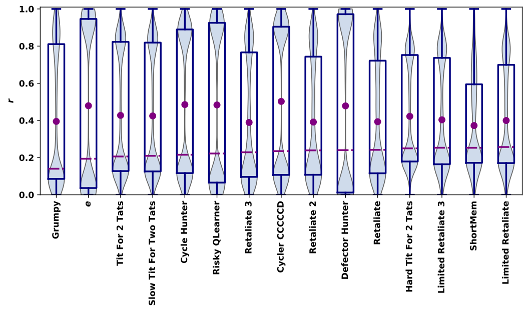
Table 3.3: Top performances for each tournament type based on $\bar{r}$. The results of each type are based on 11420 unique tournaments of each type. The results for noisy tournaments with $p_n < 0.1$ are based on 1151 tournaments, and for probabilistic ending tournaments with $p_e < 0.1$ on 1139. The top ranks indicate that trained strategies perform well in a variety of environments, but so do simple deterministic strategies. The normalised medians are close to 0 for most environments, except environments with noise not restricted to 0.1 regardless the number of turns. Noisy and noisy probabilistic ending tournaments have the highest medians. This implies that strategies from the collection of this work do not perform well in environments with high values of noise.



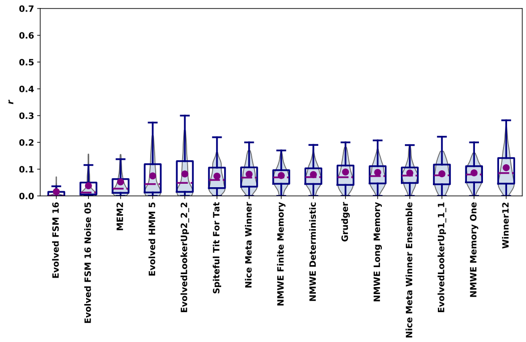
(a) r distributions of top 15 strategies in standard tournaments.



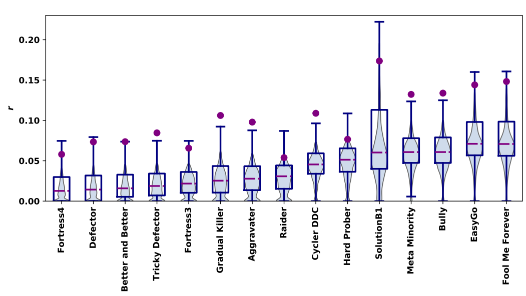
(b) r distributions of top 15 strategies in noisy tournaments with $p_n < 0.1$.



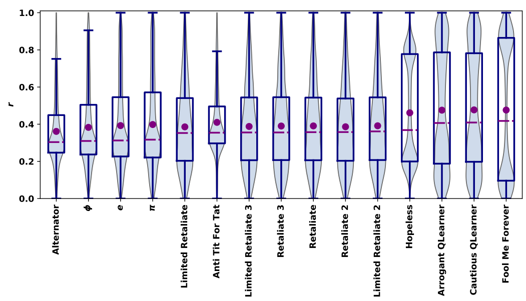
(c) r distributions of top 15 strategies in noisy tournaments.



(d) r distributions of top 15 strategies in 1139 probabilistic ending tournaments with $p_e < 0.1$.



(e) r distributions of top 15 strategies in probabilistic ending tournaments.



(f) r distributions of top 15 strategies in noisy probabilistic ending tournaments.

tournament, thus their performance in the specific setting was anticipated. DoubleCrosser, BackStabber and Fool Me Once, are strategies not from the literature but from the APL. DoubleCrosser is an extension of BackStabber and both strategies make use of the number of turns because they are set to defect on the last two rounds. It should be noted that these strategies can be characterised as "cheaters" because the source code of the strategies allows them to know the number of turns in a match (unless the match has a probabilistic ending). These strategies were expected to not perform as well in tournaments where the number of turns is not specified. Finally, Winner 12 [35] and DBS [11] are both from the literature. DBS is a strategy specifically designed for noisy environments, however, it ranks highly in standard tournaments as well. Similarly the fourth ranked player, Evolved FSM 16 Noise 05, was trained for noisy tournaments yet performs well in standard tournaments. Figure 3.2a shows that these strategies typically perform well in any standard tournament in which they participate.

In the case of noisy tournaments with $p_n < 0.1$ the top performed strategies include strategies specifically designed for noisy tournaments. These are DBS, Evolved FSM 16 Noise 05, Evolved ANN 5 Noise 05, PSO Gambler 2 2 2 Noise 05 and Omega Tit For Tat [29]. Omega TIT FOR TAT, another strategy designed to break the deadlocking cycles of CD and DC that TIT FOR TAT can fall into in noisy environments, places 10th. The rest of the top ranks are occupied by strategies which performed well in standard tournaments and deterministic strategies such as Spiteful Tit For Tat [1], Level Punisher [4], Eugine Nier [42]. Similarly to standard tournaments, the successful strategies in this given setting performed well overall in the tournaments they participated in, Figure 3.2b.

In comparison, the performance of the top ranked strategies in noisy environments when $p_n \in [0, 1]$ is bimodal. The top strategies include strategies which decide their actions based on the cooperation to defection ratio, such as ShortMem [20], Grumpy [3] and e [3], and the Retaliate strategies which are designed to defect if the opponent has tricked them more often than a given percentage of the times that they have done the same. The bimodality of the r distributions is explained by Figure 3.3 which demonstrates that the top 6 strategies were highly ranked due to the their performance in tournaments with $p_n > 0.5$, and that in tournaments with a noise probability lower than 0.5 they performed poorly. At a noisy level of 0.5 or greater, mostly cooperative strategies become mostly defectors and vice versa.

The new entrants to the most effective strategies list in probabilistic ending tournaments with $p_e < 0.1$ are a series of Meta strategies, trained strategies which performed well in standard tournaments, and Grudger [3] and Spiteful Tit for Tat [1]. The Meta strategies [3] create a team of strategies and play as an ensemble or some other combination of their team members. Figure 3.2d indicates that these strategies performed well in any probabilistic ending tournament they competed in.

In probabilistic ending tournaments with $p_e \in [0, 1]$ the top ranks are mostly occupied by defecting strategies such as Better and Better, Gradual Killer, Hard Prober (all from [3]), Bully (Reverse Tit For Tat) [39] and Defector, and a series of strategies based on finite state automata introduced by Daniel Ashlock and Wendy Ashlock; Fortress 3, Fortress 4 (both introduced in [9]), Raider [10] and Solution B1 [10]. The success of defecting strategies in probabilistic ending tournaments is due to larger values of p_e which lead to shorter matches (the expected number of rounds is $1/p_e$), so the impact of the PD being iterated is subdued.

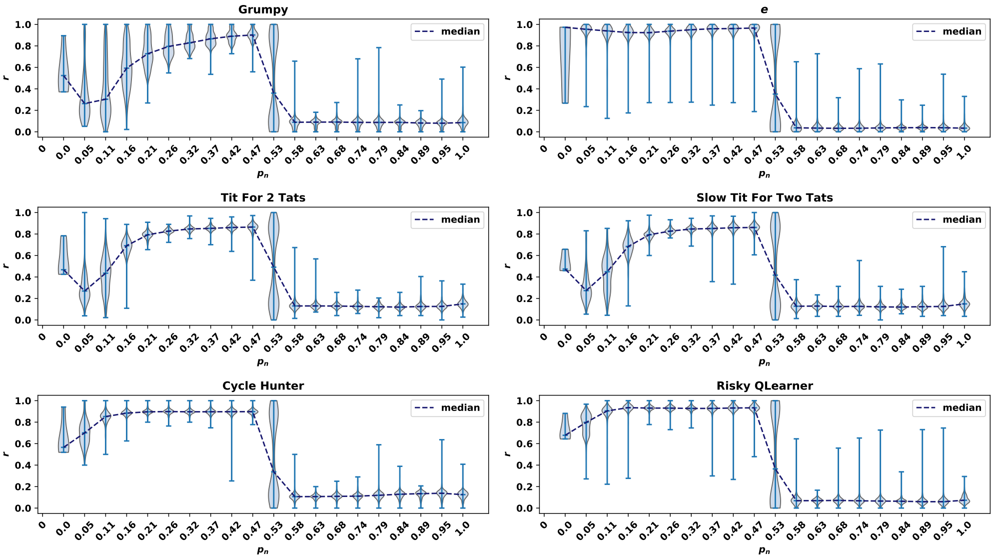


Figure 3.3: r distributions for top 6 strategies in noisy tournaments over the probability of noisy (p_n).

This is captured by the Folk Theorem [24] as defecting strategies do better when the likelihood of the game ending in the next turn increases. This is demonstrated by Figure 3.4, which gives the distributions of r for the top 6 strategies in probabilistic ending tournaments over p_e.

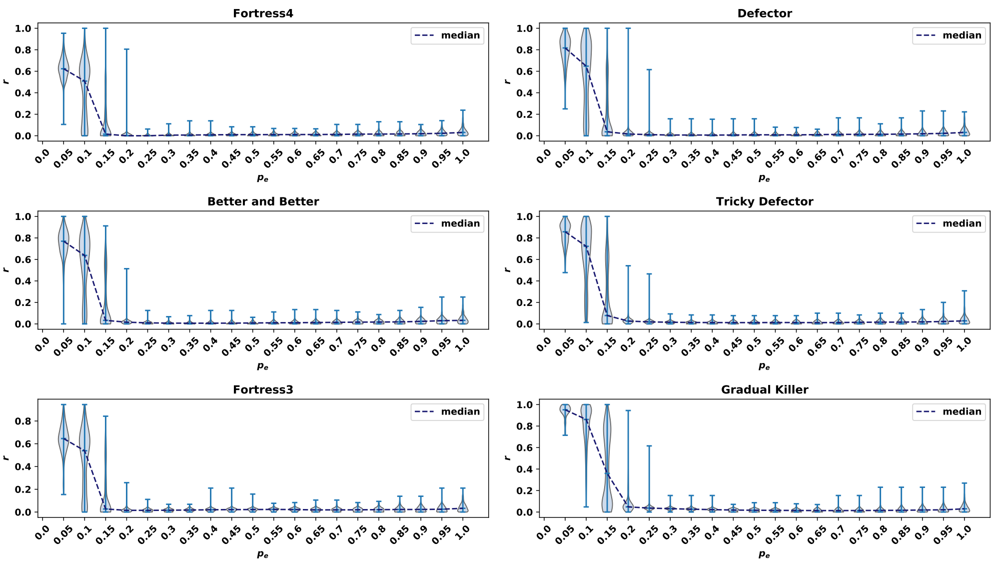


Figure 3.4: r distributions for top 6 strategies in probabilistic ending tournaments over p_e. The 6 strategies start of with a high median rank, however, their ranked decreased as the the probability of the game ending increased and at the point of $p_e = 0.1$.

The top performances in tournaments with both noise and a probabilistic ending and the top performances over the entire data set have the largest median values compared to the top rank strategies of the other tournament types, Figure 3.2f and Figure 3.5. The $\bar{r}$ for the top strategy is approximately at 0.3, indicating that the most successful strategy can on average just place at the top 30% of the competition.

On the whole, the analysis of this manuscript has shown that:

Name	$\bar{r}$
Limited Retaliate 3	0.286
Retaliate 3	0.296
Retaliate 2	0.302
Limited Retaliate 2	0.303
Limited Retaliate	0.310
Retaliate	0.317
BackStabber	0.324
DoubleCrosser	0.331
Nice Meta Winner	0.349
PSO Gambler 2 2 2 Noise 05	0.351
Grudger	0.352
Evolved HMM 5	0.357
NMWE Memory One	0.357
Nice Meta Winner Ensemble	0.359
Forgetful Fool Me Once	0.359

Table 3.4: Top performances over all the tournaments. The top ranks include strategies that have been previously mentioned. The set of Retaliate strategies occupy the top spots followed by BackStabber and DoubleCrosser. The distributions of the Retaliate strategies have no statistical difference. PSO Gambler and Evolved HMM 5 are trained strategies introduced in [27] and Nice Meta Winner and NMWE Memory One are strategies based on teams. Grudger is a strategy from Axelrod's original tournament and Forgetful Fool Me Once is based on the same approach as Grudger.

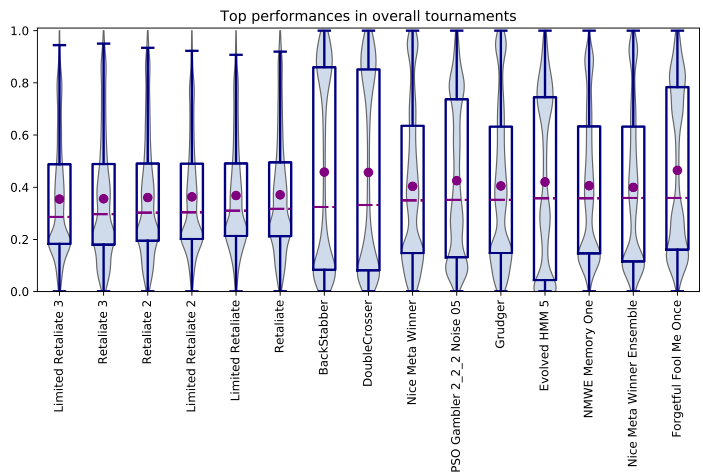


Figure 3.5: r distributions for best performed strategies in the data set [26]. A lower value of $\bar{r}$ corresponds to a more successful performance.

- In standard tournaments the dominating strategies were strategies that had been trained using reinforcement learning techniques.
- In noisy environments where the noise probability strictly less than 0.1 was considered, the successful strategies were strategies specifically designed for noisy environments.
- In probabilistic ending tournaments most of the highly ranked strategies were defecting strategies and trained finite state automata, all by the authors of [9, 10]. These strategies ranked high due to their performance in tournaments where the probability of the game ending after each turn was bigger than 0.1.
- In probabilistic tournaments with p_e less than 0.1 the highly ranked strategies were strategies based on the behaviour of others.
- From the collection of strategies considered here, no strategy can be consistently successful in noisy environments, except if the value of noise is constrained to less than a 0.1.

Though there is not a single strategy that repeatably outranks all others in any of the distinct tournament types, or even across the tournaments type, there are specific types of strategies have been repeatably ranked in the top ranks. These have been strategies that have been trained, strategies that retailiate, and strategies that would adapt their behaviour based on preassigned rules to achieve the highest outcome. These results contradict some of Axelrod's suggestions, and more specifically, the suggestions 'Do not be clever' and 'Do not be envious'. The features and properties contributing a strategy's success are further explored in Section 3.4.

3.4 Evaluation of performance

Now we examine performance of the strategies based on features of strategies described in Table 3.5. These features are measures regarding a strategy's behaviour from the tournaments the strategies competed in as well as intrinsic properties such as whether a strategy is deterministic or stochastic.

The memory usage of strategies with an infinite memory size, for example Evolved FSM 16 Noise 05, is equal to 1. Otherwise the memory usage is the number of rounds of play used by the strategy divided by the number of turns in each match. For example, Winner12 uses the previous two rounds of play, and if participating in a participated in a tournament where n was 100 the memory usage would be 2/100. Note that for tournaments with a probabilistic ending the number of turns was not collected, so the memory usage feature is not used for probabilistic ending tournaments.

The correlation coefficients between the features of Table 3.5 the median score and the median normalised rank are given by Table 3.6. The correlation coefficients between all features of Table 3.5 have been calculated and a graphical representation can be found in the Appendix B.

In standard tournaments the features CC to C, C_r, $C_r/C_{\max}$ and the cooperating ratio compared to C_{median} and C_{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

feature	feature explanation	source	value type	min value	max value
stochastic	If a strategy is stochastic	strategy classifier from APL	boolean	Na	Na
makes use of game	If a strategy makes used of the game information	strategy classifier from APL	boolean	Na	Na
makes use of length	If a strategy makes used of the number of turns	strategy classifier from APL	boolean	Na	Na
memory usage	The memory size of a strategy divided by the number of turns	memory size from APL	float	0	1
SSE	A measure of how far a strategy is from ZD behaviour	method described in [30]	float	0	1
max cooperating rate (C_{max})	The biggest cooperating rate in a given tournament	result summary	float	0	1
min cooperating rate (C_{min})	The smallest cooperating rate in a given tournament	result summary	float	0	1
median cooperating rate (C_{median})	The median cooperating rate in a given tournament	result summary	float	0	1
mean cooperating rate (C_{mean})	The mean cooperating rate in a given tournament	result summary	float	0	1
C_r / C_{max}	A strategy's cooperating rate divided by the maximum	result summary	float	0	1
C_{min} / C_r	A strategy's cooperating rate divided by the minimum	result summary	float	0	1
C_r / C_{median}	A strategy's cooperating rate divided by the median	result summary	float	0	1
C_r / C_{mean}	A strategy's cooperating rate divided by the mean	result summary	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
p_n	The probability of a player's action being flip at each interaction	trial summary	float	0	1
n	The number of turns	trial summary	integer	1	200
p_e	The probability of a match ending in the next turn	trial summary	float	0	1
N	The number of strategies in the tournament	trial summary	integer	3	195
k	The number of repetitions of a given tournament	trial summary	integer	10	100

Table 3.5: The features which are included in the performance evaluation analysis. Stochastic, makes use of length and makes use of game are APL classifiers that determine whether a strategy is stochastic or deterministic, whether it makes use of the number of turns or the game's payoffs. The memory usage is calculated as the number of turns the strategy considers to make an action (which is specified in the APL) divided by the number of turns. The SSE (introduced in [30]) shows how close a strategy is to behaving as a ZDs, and subsequently, in an extortionate way. The method identifies the ZDs closest to a given strategy and calculates the algebraic distance between them, defined as SSE. A SSE value of 1 indicates no extortionate behaviour at all whereas a value of 0 indicates that a strategy is behaving a ZDs. The rest of the features 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 (C_{min}), maximum (C_{max}), mean (C_{mean}) and median (C_{median}) cooperating ratios of each tournament.

	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	-0.501	0.501
CD to C rate	0.226	-0.199	0.456	-0.330	0.320	-0.017	0.205	-0.220	0.226	-0.199
C_r	-0.323	0.384	0.711	-0.678	0.714	-0.832	0.579	-0.135	-0.323	0.384
C_r / C_{max}	-0.323	0.381	0.616	-0.551	0.714	-0.833	0.536	-0.116	-0.323	0.381
C_r / C_{mean}	-0.331	0.358	0.731	-0.740	0.721	-0.861	0.649	-0.621	-0.331	0.358
C_r / C_{median}	-0.331	0.353	0.652	-0.669	0.712	-0.852	0.330	-0.466	-0.331	0.353
C_r / C_{min}	0.109	-0.080	-0.358	0.250	-0.134	0.150	-0.368	0.113	0.109	-0.080
C_{max}	-0.000	0.049	0.000	0.023	-0.000	0.046	0.000	-0.004	-0.000	0.049
C_{mean}	-0.000	0.229	-0.000	0.271	0.000	0.200	0.000	0.690	-0.000	0.229
C_{median}	0.000	0.209	-0.000	0.240	-0.000	0.187	-0.000	0.673	0.000	0.209
C_{min}	0.000	0.084	0.000	-0.017	-0.000	0.007	-0.000	0.041	0.000	0.084
DC to C rate	0.127	-0.100	0.509	-0.504	-0.018	0.033	0.341	-0.016	0.127	-0.100
DD to C rate	0.412	-0.396	0.533	-0.436	-0.103	0.176	0.378	-0.263	0.412	-0.396
N	0.000	-0.009	-0.000	0.002	-0.000	0.003	-0.000	0.001	0.000	-0.009
k	0.000	-0.002	-0.000	0.003	-0.000	0.001	-0.000	-0.008	0.000	-0.002
n	0.000	-0.125	-0.000	-0.024	-	-	-	-	0.000	-0.125
p_e	-	-	-	-	0.000	0.165	0.000	-0.058	-0.001	0.001
p_n	-	-	-0.000	0.207	-	-	-0.000	-0.650	0.002	-0.000
Make use of game	-0.003	-0.022	0.025	-0.082	-0.053	-0.108	0.013	-0.016	-0.003	-0.022
Make use of length	-0.158	0.124	0.005	-0.123	-0.025	-0.090	0.014	-0.016	-0.154	0.117
SSE	0.473	-0.452	0.463	-0.337	-0.156	0.223	0.305	-0.259	0.473	-0.452
memory usage	-0.082	0.095	-0.007	-0.017	-	-	-	-	-0.084	0.095
stochastic	0.006	-0.024	0.022	-0.026	0.002	-0.130	0.021	-0.013	0.006	-0.024

Table 3.6: Correlations table between the features of Table 3.5 the normalised rank and the median score.

effects. Thus, in standard tournaments behaving cooperatively corresponds to a more successful performance. Even though being nice pays off, that's not true against defective strategies. Being more cooperative after a mutual defection is associated to lesser overall success in terms of normalised rank. Figure 3.6 confirms that the winners of standard tournaments always cooperate after a mutual cooperation and almost always defect after a mutual defection.

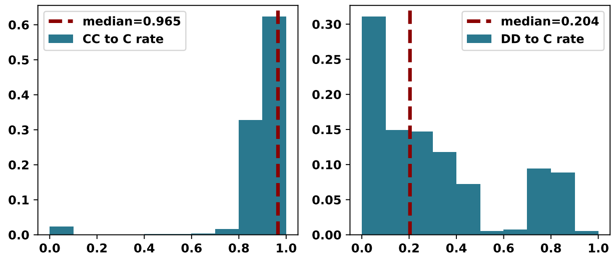


Figure 3.6: Distributions of CC to C and DD to C for the winners in standard tournaments.

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. A strategy would want to cooperate less than both the mean and median cooperator in such settings. In probabilistic ending tournaments the correlation coefficients have larger values, indicating a stronger effect. Thus a strategy will be punished more by its cooperative behaviour in probabilistic ending environments, supporting the results of Section 3.4 as well. The distributions of the C_r of the winners in both tournaments are given by Figure 3.7. It confirms that the winners in noisy tournaments cooperated less than 35% of the time and in probabilistic ending tournaments less than 10%. In noisy probabilistic ending tournaments and over all the tournaments' results, the only features that had a moderate effect are $C_r/C_{\text{mean}}, C_r/C_{\text{max}}$ and C_r. In such environments cooperative behaviour appears to be punished less than in noisy and probabilistic ending tournaments.

Moreover, the manner in which a strategy achieves a given cooperation rate relative to the tournament population average is important. Playing a strategy that randomly cooperates with C_{mean} is unlikely to be as effective as it will randomly have cooperations matched with defections. In contrast, TIT FOR TAT naturally achieves a cooperation rate near C_{mean} by virtue of copying its opponent's last move while also minimizing instances where it is exploited by an opponent (cooperating while the opponent defects), at least in non-noisy tournaments. [1] TIT FOR TAT forces the opponent to pay back the (C, D) round with a (D, C) round before returning to mutual cooperation. This explains why TIT FOR TAT performed well in Axelrod's original tournaments as most strategies submitted to those tournaments were typically cooperative and relatively few strategies used an easily exploitable pattern. TIT FOR TAT does not appear in the top ranks in these tournaments because it is too nice. Strategies like Grudger will always defect after a fixed number of opponent defections, which allows them to effectively exploit strategies like Alternator or stochastic strategies that have a non-zero chance of cooperating after mutual defection, which TIT FOR TAT will not do. Moreover in a noisy environment these strategies will naturally tend toward always defecting, leading them to exploit strategies like Cooperator. In such a scenario the noisy environment effectively voids Axelrod's rule to be nice, allowing strategies to attempt exploitation, whereas in a noise-free

[1] This also explains why Tit For N Tats does not fare well – it fails to achieve the proper cooperation ratio.

environment, exploitation is risky because several strategies exhibit a Grudger-like behavior, reducing the overall value in attempting to exploit strategies like Cooperator.

Similarly, these results suggest an explanation regarding the intuitively unexpected effectiveness of memory one strategies historically. Given that among the important features associated with success are the relative cooperation rate to the population average and the four memory-one probabilities of cooperating conditional on the previous round of play, all five features can be optimized by a memory one strategy such as TIT FOR TAT. Usage of more history becomes valuable when there are exploitable opponent patterns, indicated by the importance of SSE as a feature, that the first-approximation provided by a memory one strategy is no longer sufficient.

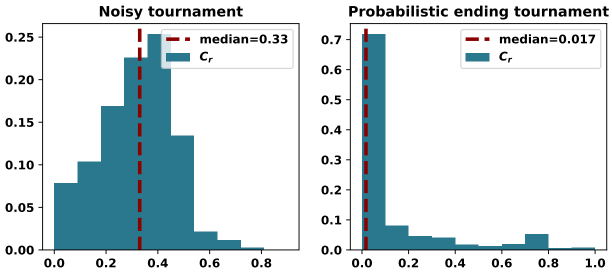


Figure 3.7: C_r distributions of the winners in noisy and in probabilistic ending tournaments.

A multivariate linear regression has been fitted to model the relationship between the features and the normalised rank. Based on the graphical representation of the correlation matrices given in Appendix B several of the features are highly correlated. Highly correlated features have been removed before fitting the linear regression model. The features included are given by Table 3.7 alongside their corresponding p values in the distinct tournaments and their regression coefficients.

	Standard		Noisy		Probabilistic ending		Noisy probabilistic ending		Overall	
	R adjusted: 0.541		R adjusted: 0.639		R adjusted: 0.587		R adjusted: 0.577		R adjusted: 0.242	
	Coefficient	p-value	Coefficient	p-value	Coefficient	p-value	Coefficient	p-value	Coefficient	p-value
CC to C rate	-0.042	0.000	-0.007	0.000	0.017	0.000	0.111	0.0	-0.099	0.0
CD to C rate	0.297	0.000	-0.068	0.000	0.182	0.000	0.023	0.0	0.129	0.0
C_r / C_{max}	-	-	1.856	0.000	-	-	1.256	0.0	-	-
C_r / C_{mean}	-0.468	0.000	-0.577	0.000	0.525	0.000	-0.120	0.0	0.300	0.0
C_{max}	-0.071	0.000	-	-	-0.022	0.391	1.130	0.0	-	-
C_{mean}	0.118	0.000	-2.558	0.000	-0.023	0.001	-1.489	0.0	-	-
C_{min}	-0.161	0.000	-1.179	0.000	-0.170	0.000	-	-	-	-
C_{min} / C_r	0.057	0.000	-0.320	0.000	0.125	0.000	-	-	-0.103	0.0
DC to C rate	0.198	0.000	0.040	0.000	-0.030	0.000	0.022	0.0	0.064	0.0
k	0.000	0.319	0.000	0.020	0.000	0.002	0.000	0.0	-	-
n	0.000	0.000	-	-	-	-	-	-	-	-
p_e	-	-	-	-	0.000	0.847	-0.083	0.0	-	-
p_n	-	-	-0.048	0.000	-	-	-	-	-	-
SSE	0.258	0.000	0.153	0.000	-0.041	0.000	0.100	0.0	0.056	0.0
constant	0.697	0.000	1.522	0.000	-0.057	0.019	-0.472	0.0	0.178	0.0
memory usage	-0.010	0.000	-0.000	0.035	-	-	-	-	-	-

Table 3.7: Results of multivariate linear regressions with r as the dependent variable. R squared is reported for each model.

A multivariate linear regression has also be fitted on the median score. The coefficients and p values of the features can be found in Appendix **??**. The results of the two methods are in agreement.

The feature C_r/C_{mean} has a statistically significant effect across all models and a high regression coefficient. It has both a positive and negative impact on the normalised rank depending on the environment. For standard tournaments, Figure 3.8 gives the distributions of several features for the winners of standard tournaments. The C_r/C_{mean} distribution of the winner is also given in Figure 3.8. A value of $C_r/C_{\text{mean}} = 1$ implies that the cooperating ratio of the winner was the same as the mean cooperating ratio of the tournament, and in standard tournaments, the median is 1. Therefore, an effective strategy in standard tournaments was the mean cooperator of its respective tournament.

The distributions of SSE and CC to D rate for the winners of standard tournaments are also given in Figure 3.8. The SSE distributions for the winners indicate that the strategy behaved in a ZD way in several tournaments, however, not constantly. The winners participated in matches where they did not try to extortionate their opponents. Furthermore, the CC to D distribution indicates that if a strategy were to defect against the winners they would reciprocate on average with a probability of 0.5.

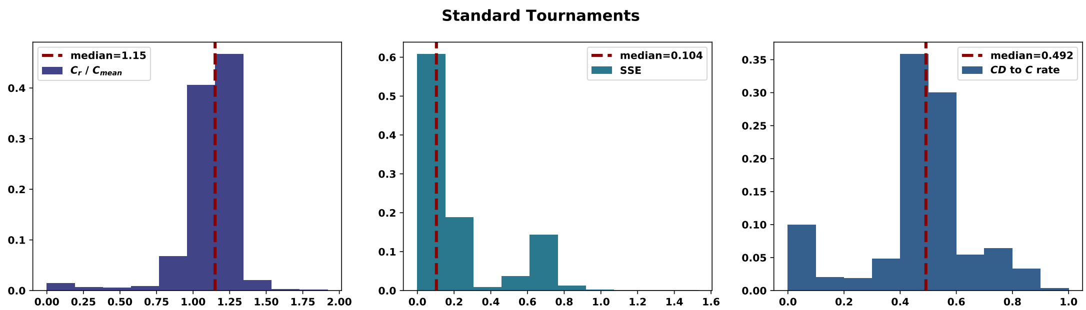


Figure 3.8: Distributions of C_r/C_{mean}, SSE and CD to C ratio for the winners of standard tournaments. A value of $C_r/C_{\text{mean}} = 1$ imply that the cooperating ratio of the winner was the same as the mean cooperating ratio of the tournament. An SSE distribution skewed towards 0 indicates a extortionate behaviour by the strategy.

Similarly for the rest of the different tournaments types, and the entire data set the distributions of C_r/C_{mean}, SSE and CD to C ratio are given by Figures 3.9, 3.11, 3.12 and 3.13.

Based on the C_r/C_{mean} distributions the successful strategies have adapted differently to the mean cooperator depending on the tournament type. In noisy tournaments where the median of the distribution is at 0.67, and thereupon the winners cooperated 67% of the time the mean cooperator did. In tournaments with noise and a probabilistic ending the winners cooperated 60%, whereas in settings that the type of the tournament can vary between all the types the winners cooperated 67% of the time the mean cooperator did. Lastly, in probabilistic ending tournaments above more defecting strategies prevail (Section 3.3), and this result is reflected here.

The probability of noise has been observed to excessively affect optimal behaviour. In environments with considerable values of noise no strategy from our collection managed to perform sufficiently. Figure 3.10 gives the ratio C_r/C_{mean} for the winners in tournaments with noise, over the probability of noise. From Figure 3.10a it is clear that the cooperating only 67% of

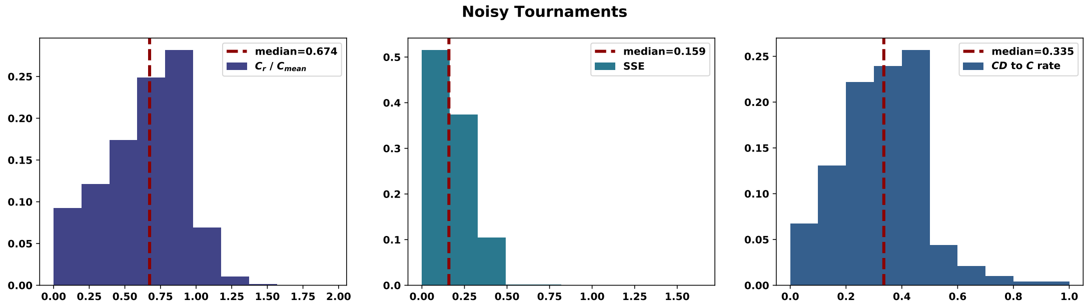
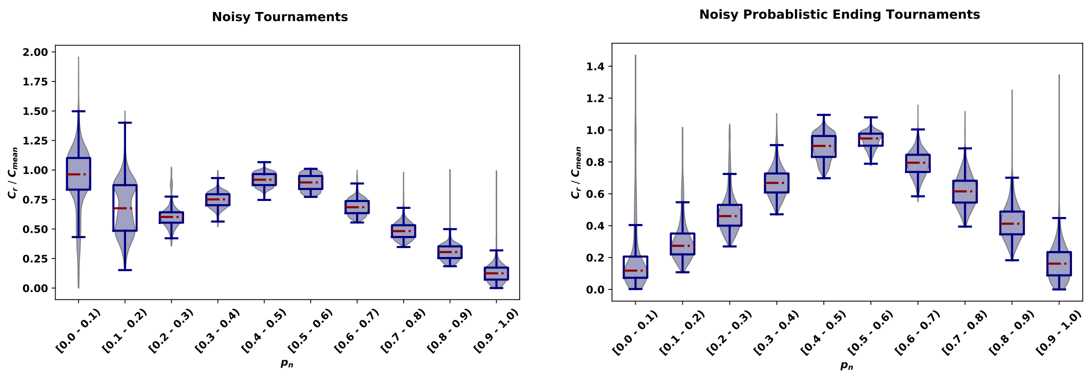


Figure 3.9: Distributions of C_r/C_{mean}, SSE and CD to C ratio for the winners of noisy tournaments.

the time the mean cooperator did is optimal only when $p_n \in [0.2, 0.4)$ and $p_n \in [0.6, 0.7]$. In environments with $p_n < 0.1$ the winners want to be close to the mean cooperator, similarly to standard tournaments, and as the probability of noise is exceeding 0.5 (the game becomes unreasonable) strategies should aim to be less and less cooperative.

Figure 3.10 gives C_r/C_{mean} for the winners over p_n in tournaments with noise and a probabilistic ending. The optimal proportions of cooperations are different now that the number of turns is not fixed, successful strategies want to be more defecting that the mean cooperator, that only changes when p_n approaches 0.5. Figure 3.10 demonstrates how the adjustments to C_r/C_{mean} change over the noise in the to the environment, and thus supports how important adapting to the environment is for a strategy to be successful.



(a) C_r/C_{mean} distribution for winners in noisy tournaments over p_n.

(b) C_r/C_{mean} distribution for winners in noisy probabilistic ending tournaments over p_n.

Figure 3.10: C_r/C_{mean} distributions over intervals of p_n. These distributions model the optimal proportion of cooperation compared to C_{mean} as a function of (p_n).

The distributions of the SSE across the tournament types suggest that successful strategies exhibit some extortionate behaviour, but not constantly. ZDs are a set of strategies that are envious as they try to exploit their opponents. The winners of the tournaments considered in this work are envious, but not as much as many ZDs. This highlights why TIT FOR TAT's early tournament success fails to generalize – it never attempts to defect against a cooperating or exploitable opponent (e.g. Alternator). Moreover, many of the strategies in the library will not tolerate exploitation attempts. A clever strategy can achieve mutual cooperation with stronger strategies while also being able to exploit weaker strategies. This is why ZDs fail to appear in the top ranks – they try to exploit all opponents and cannot actively adapt back to mutual cooperation against stronger strategies, which requires more depth of memory. [2]

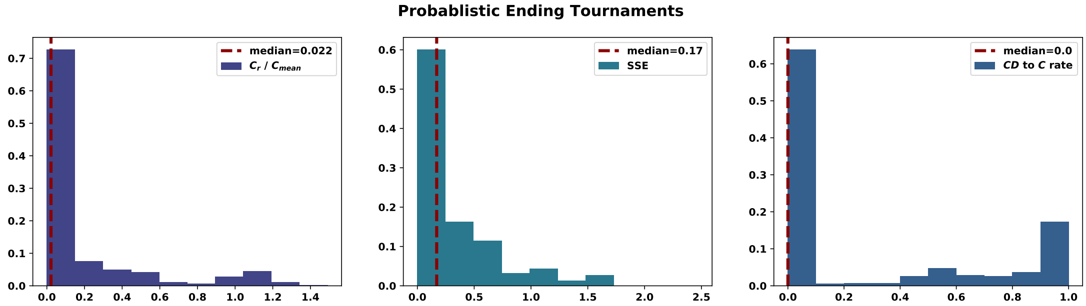


Figure 3.11: Distributions of C_r/C_{mean}, SSE and CD to C ratio for the winners of probabilistic ending tournaments.

The distributions of the CD to C rate evaluate the behaviour of a successful strategy after its opponent has defected against it. In standard tournaments it was observed that a successful strategy reciprocates with a probability of 0.5. This is distinct between the tournament types. In tournaments with noise a strategy is less likely to cooperate following a defection compared to standard tournaments, and in probabilistic ending tournaments a strategy will reciprocate a defection. In a setting that the type of the tournament can vary between all the examined types a winning strategy would reciprocate on average with a probability of 0.58.

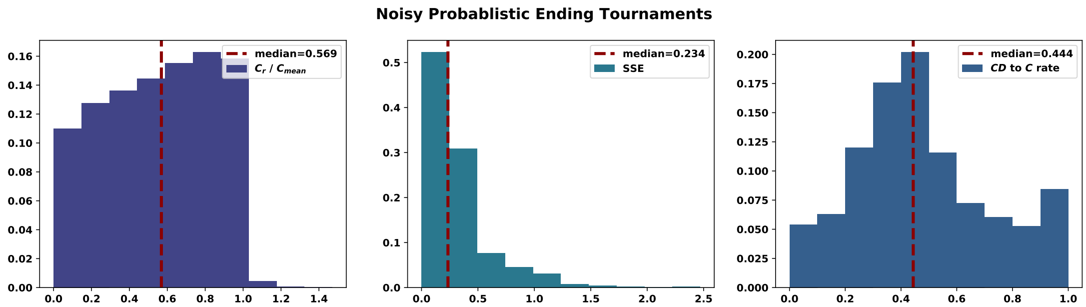


Figure 3.12: Distributions of C_r/C_{mean}, SSE and CD to C ratio for the winners of noisy probabilistic ending tournaments.

Further statistically significant features with strong effects include C_r/C_{min}, C_r/C_{max}, C_{min} and C_{max}. These add more emphasis on how important it is for a a strategy to adapt to its environment. Finally, the features number of turns, repetitions and the probabilities of noise and the game ending had no significant effects based on the multivariate regression models.

A third method that evaluates the importance of the features in Table 3.5 using clustering and random forests can be found in the Appendix **??**. The results uphold the outcomes of the correlation and multivariate regression. It also evaluates the effects of the classifiers stochastic, make use of game, and make use of length which have not been evaluated by the methods above because there are binary variables. The results imply that they have no significant effect on a strategy's performance.

[2]Note that ZDs also tend to perform poorly in population games for a similar reason: they attempt to exploit other players using ZDs, failing to form a cooperative subpopulation. This makes them good invaders but poor resisters of invasion.

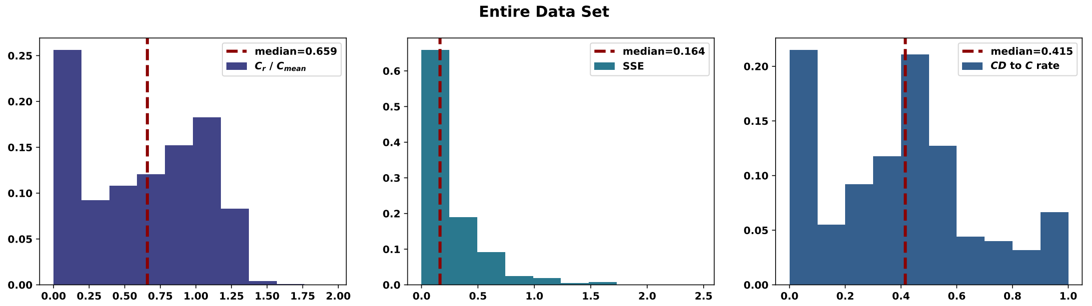


Figure 3.13: Distributions of C_r/C_{mean}, SSE and CD to C ratio for the winners over the tournaments of the entire data set.

3.5 Chapter Summary

This manuscript has explored the performance of 195 strategies of the Iterated Prisoner's Dilemma in a large number of computer tournaments. We analyzed and extracted the salient features of the best performing strategies across various tournament types, casting the results in terms of Axelrod's original suggested features of good IPD strategies. Moreover, our results shed light on the historic performance of TIT FOR TAT, zero-determinant strategies, and memory one strategies generally. Strategies need to match their play to the cooperativeness of the tournament population and do so in a way that prevents or minimizes exploitation. Overall we see that complex or clever strategies can be effective, whether trained against a corpus of possible opponents or purposely designed to mitigate the impact of noise such as in the strategy DBS. Further, we showed that while the type of exploitation attempted by ZDs is not typically effective in tournaments, more sophisticated strategies capable of selectively exploiting weaker opponents while mutually cooperating with stronger opponents can be highly successful. This fact was also indicated numerically by the importance of the strategy feature SSE in the analysis of strategy features. These results highlight a central idea in evolutionary game theory in this context: the fitness landscape is a function of the population (where fitness in this case is tournament performance). While that may seem obvious now, it shows why historical tournament results on small or arbitrary populations of strategies have so often failed to produce generalizable results.

Highly noisy or tournaments with short matches favor less cooperative strategies. These environments mitigate the value of being nice. Uncertainty enables exploitation, reducing the ability of maintaining or enforcing mutual cooperation, while triggering grudging strategies to switch from typically cooperating to typically defecting. Accordingly, we find that in noisy tournaments the best performing players cooperate a lower rate than the tournament population on average. Nevertheless we found some strategies designed or trained for noisy environments were also highly ranked in noise-free tournaments. This indicates that strategy complexity is not necessarily a liability, rather it can confer adaptability to a more diverse set of environments.

In Section 3.3, the tournaments results were used to present the top performances. The data set contained results from four different settings, and these were also studied individually. In standard tournaments complex strategies trained using reinforcement learning ranked in the top spots. Some of these strategies ranked again in the top spots in probabilistic ending tournaments when a p_e of less 0.1 was considered and in noisy tournaments when p_n was less than 0.1. In probabilistic ending tournaments p_e was designed to vary between 0 and 1. It was demonstrated that for values larger than 0.1, as stated in the Folk Theorem, defecting strategies were winning the tournaments because there was a high likelihood of the game ending in the next turn. In tournaments with noise the median ranks of the top 15 strategies had the highest values and the r distributions were bimodal. The top rank strategies were performing both well and bad, and this indicates that in noisy tournaments where the noise can vary substantially, there were no strategies that can guarantee winning across a range of noise. However, if the probability of noise was constrained at 0.1 then strategies designed for noisy tournaments indeed performed well.

So what is the best way of playing the IPD? And is there a single dominant strategy for the IPD? There was not a single strategy within the collection of the 195 strategies, that has managed to perform well in all the tournaments variations it competed in. Even if on average a strategy ranked highly in a specific environment it did not guarantee its success over the different tournament types. However, the results of sections 3.3 and 3.4 have demonstrated that there are properties associated with the success of strategies. A few of the properties that have been identified by this manuscript's analysis contradict the properties of Axelrod [14]. Namely, in Section 3.3 it was shown that trained strategies and strategies that decided their actions based on pre-designed strategies to maximise their utility dominated several tournaments across tournament types, hinting that successful IPD strategies are often clever or more complex than simple strategies like TIT FOR TAT. Most of the successful strategies highlighted in Section 3.3 were strategies that begin with cooperation.

Furthermore, in Section 3.3 and 3.4 it was shown that envious strategies performed well. Though these were not the most envious strategies in the tournaments (ZDs were included), these strategies benefited by being a bit envious. From Section 3.4 it was concluded that there is a significant importance in adapting to the environment, and more specifically in this work, to the mean cooperator. This section also demonstrated that a strategy should reciprocate, as suggested by Axelrod, but in some environments, such as standard and noisy, it should relax its readiness to do so.

Thus, the five properties successful strategies need to have in a IPD competition are: be nice, be provocable and contrite, be a little envious, be clever, and adapt to the environment (including the population of strategies).

The data set described in this work contains the largest number of IPD tournaments, to the authors knowledge, and it available at [26]. Further data mining could be applied and provide new insights in the field.

Bibliography

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

[2] The prisoner's dilemma. http://www.prisoners-dilemma.com/, 2017.

[3] The Axelrod project developers . Axelrod: 4.4.0, April 2016.

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

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

[6] C. Adami and A. Hintze. Evolutionary instability of zero-determinant strategies demonstrates that winning is not everything. *Nature communications*, 4:2193, 2013.

[7] D. Ashlock, J. A. Brown, and P. Hingston. Multiple opponent optimization of prisoner's dilemma playing agents. *IEEE Transactions on Computational Intelligence and AI in Games*, 7(1):53–65, 2015.

[8] D. Ashlock and E. Y. Kim. Fingerprinting: Visualization and automatic analysis of prisoner's dilemma strategies. *IEEE Transactions on Evolutionary Computation*, 12(5):647–659, Oct 2008.

[9] W. Ashlock and D. 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.

[10] W. Ashlock, J. Tsang, and D. Ashlock. The evolution of exploitation. In *2014 IEEE Symposium on Foundations of Computational Intelligence (FOCI)*, pages 135–142. IEEE, 2014.

[11] T. C. Au and D. 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.

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

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

[14] R. Axelrod. The emergence of cooperation among egoists. *American political science review*, 75(2):306–318, 1981.

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

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

[17] B. Beaufils, J. P. Delahaye, and P. Mathieu. Our meeting with gradual: A good strategy for the iterated prisoner's dilemma. 1997.

[18] 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.

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

[20] A. Carvalho, H. P. Rocha, F. T. Amaral, and F. G. Guimaraes. Iterated prisoner's dilemma-an extended analysis. 2013.

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

[22] Merrill M. Flood. Some experimental games. *Management Science*, 5(1):5–26, 1958.

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

[24] D. Fudenberg and E. Maskin. The folk theorem in repeated games with discounting or with incomplete information. In *A Long-Run Collaboration On Long-Run Games*, pages 209–230. World Scientific, 2009.

[25] M. Gaudesi, E. Piccolo, G. Squillero, and A. Tonda. Exploiting evolutionary modeling to prevail in iterated prisoner's dilemma tournaments. *IEEE Transactions on Computational Intelligence and AI in Games*, 8(3):288–300, 2016.

[26] N. E. Glynatsi. A data set of 45686 Iterated Prisoner's Dilemma tournaments' results. `https://doi.org/10.5281/zenodo.3516652`, October 2019.

[27] Marc Harper, Vincent Knight, Martin Jones, Georgios Koutsovoulos, Nikoleta E. Glynatsi, and Owen Campbell. Reinforcement learning produces dominant strategies for the iterated prisoner's dilemma. *PLOS ONE*, 12(12):1–33, 12 2017.

[28] C. Hilbe, M. A. Nowak, and A. Traulsen. Adaptive dynamics of extortion and compliance. *PLOS ONE*, 8(11):1–9, 11 2013.

[29] G. Kendall, X. Yao, and S. Y. Chong. *The iterated prisoners' dilemma: 20 years on*, volume 4. World Scientific, 2007.

[30] V. A. Knight, M. Harper, N. E. Glynatsi, and J. Gillard. Recognising and evaluating the effectiveness of extortion in the iterated prisoner's dilemma. *CoRR*, abs/1904.00973, 2019.

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

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

[33] J. Li, P. Hingston, S. Member, and G. Kendall. Engineering Design of Strategies for Winning Iterated Prisoner ' s Dilemma Competitions. 3(4):348–360, 2011.

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

[35] P. Mathieu and J. P. Delahaye. New winning strategies for the iterated prisoner's dilemma. *Journal of Artificial Societies and Social Simulation*, 20(4):12, 2017.

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

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

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

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

[40] M. Nowak and K. Sigmund. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game. *Nature*, 364(6432):56–58, 1993.

[41] M. A. Nowak and K. Sigmund. Tit for tat in heterogeneous populations. *Nature*, 355:250–253, January 1992.

[42] prase. Prisoner's dilemma tournament results. `https://www.lesswrong.com/posts/hamma4XgeNrsvAJv5/prisoner-s-dilemma-tournament-results`, 2011.

[43] W. H. Press and F. G Dyson. Iterated prisoner's dilemma contains strategies that dominate any evolutionary opponent. *Proceedings of the National Academy of Sciences*, 109(26):10409–10413, 2012.

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

[45] 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):115–116, 1984.

[46] D. W. Stephens, C. M. McLinn, and J. R. Stevens. Discounting and reciprocity in an iterated prisoner's dilemma. *Science*, 298(5601):2216–2218, 2002.

[47] A. J. Stewart and J. B. Plotkin. Extortion and cooperation in the prisoner's dilemma. *Proceedings of the National Academy of Sciences*, 109(26):10134–10135, 2012.

[48] E. Tzafestas. Toward adaptive cooperative behavior. 2:334–340, Sep 2000.

[49] 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.

[50] P. Van-Den-Berg and F. J. Weissing. The importance of mechanisms for the evolution of cooperation. In *Proc. R. Soc. B*, volume 282, page 20151382. The Royal Society, 2015.

Appendix A

List of Strategies

A.1 List of strategies considered in Chapter 3

The strategies considered in Chapter 3, which are from APL version 3.0.0.

1. ϕ [3]
2. π [3]
3. e [3]
4. ALLCorALLD [3]
5. Adaptive [33]
6. Adaptive Pavlov 2006 [29]
7. Adaptive Pavlov 2011 [33]
8. Adaptive Tit For Tat: 0.5 [49]
9. Aggravater [3]
10. Alexei [42]
11. Alternator [14, 37]
12. Alternator Hunter [3]
13. Anti Tit For Tat [28]
14. AntiCycler [3]
15. Appeaser [3]
16. Arrogant QLearner [3]
17. Average Copier [3]
18. Backstabber [3]
19. Better and Better [1]
20. Bully [39]
21. Calculator [1]
22. Cautious QLearner [3]
23. Champion [13]
24. CollectiveStrategy [34]
25. Contrite Tit For Tat [?]
26. Cooperator [14, 37, 43]
27. Cooperator Hunter [3]
28. Cycle Hunter [3]
29. Cycler CCCCCD [3]
30. Cycler CCCD [3]
31. Cycler CCCDCD [3]
32. Cycler CCD [37]
33. Cycler DC [3]
34. Cycler DDC [37]
35. DBS [11]
36. Davis [12]
37. Defector [14, 37, 43]
38. Defector Hunter [3]
39. Double Crosser [3]
40. Desperate [50]
41. DoubleResurrection [4]
42. Doubler [1]
43. Dynamic Two Tits For Tat [3]
44. EasyGo [33, 1]
45. Eatherley [13]
46. Eventual Cycle Hunter [3]
47. Evolved ANN [3]
48. Evolved ANN 5 [3]
49. Evolved ANN 5 Noise 05 [3]
50. Evolved FSM 16 [3]
51. Evolved FSM 16 Noise 05 [3]
52. Evolved FSM 4 [3]
53. Evolved HMM 5 [3]
54. EvolvedLookerUp1 1 1 [3]

55. EvolvedLookerUp2 2 2 [3]
56. Eugine Nier [42]
57. Feld [12]
58. Firm But Fair [23]
59. Fool Me Forever [3]
60. Fool Me Once [3]
61. Forgetful Fool Me Once [3]
62. Forgetful Grudger [3]
63. Forgiver [3]
64. Forgiving Tit For Tat [3]
65. Fortress3 [9]
66. Fortress4 [9]
67. GTFT [25, 40]
68. General Soft Grudger [3]
69. Gradual [17]
70. Gradual Killer [1]
71. Grofman[12]
72. Grudger [12, 16, 17, 50, 33]
73. GrudgerAlternator [1]
74. Grumpy [3]
75. Handshake [44]
76. Hard Go By Majority [37]
77. Hard Go By Majority: 10 [3]
78. Hard Go By Majority: 20 [3]
79. Hard Go By Majority: 40 [3]
80. Hard Go By Majority: 5 [3]
81. Hard Prober [1]
82. Hard Tit For 2 Tats [47]
83. Hard Tit For Tat [2]
84. Hesitant QLearner[3]
85. Hopeless [50]
86. Inverse [3]
87. Inverse Punisher [3]
88. Joss [12, 47]
89. Knowledgeable Worse and Worse [3]
90. Level Punisher [4]
91. Limited Retaliate 2 [3]
92. Limited Retaliate 3 [3]
93. Limited Retaliate [3]
94. MEM2 [?]
95. Math Constant Hunter [3]
96. Meta Hunter Aggressive [3]
97. Meta Hunter [3]
98. Meta Majority [3]
99. Meta Majority Finite Memory [3]
100. Meta Majority Long Memory [3]
101. Meta Majority Memory One [3]
102. Meta Minority [3]
103. Meta Mixer [3]
104. Meta Winner [3]
105. Meta Winner Deterministic [3]
106. Meta Winner Ensemble [3]
107. Meta Winner Finite Memory [3]
108. Meta Winner Long Memory [3]
109. Meta Winner Memory One [3]
110. Meta Winner Stochastic [3]
111. NMWE Deterministic [3]
112. NMWE Finite Memory [3]
113. NMWE Long Memory [3]
114. NMWE Memory One [3]
115. NMWE Stochastic [3]
116. Naive Prober [33]
117. Negation [2]
118. Nice Average Copier [3]
119. Nice Meta Winner [3]
120. Nice Meta Winner Ensemble [3]
121. Nydegger [12]
122. Omega TFT [29]
123. Once Bitten [3]
124. Opposite Grudger [3]
125. PSO Gambler 1 1 1 [3]
126. PSO Gambler 2 2 2 [3]
127. PSO Gambler 2 2 2 Noise 05 [3]
128. PSO Gambler Mem1 [3]
129. Predator [9]
130. Prober [33]
131. Prober 2 [1]
132. Prober 3 [1]
133. Prober 4 [1]
134. Pun1 [9]

135. Punisher [3]
136. Raider [10]
137. Random Hunter [3]
138. Random: 0.5 [12, 49]
139. Remorseful Prober [33]
140. Resurrection [4]
141. Retaliate 2 [3]
142. Retaliate 3 [3]
143. Retaliate [3]
144. Revised Downing [12]
145. Ripoff [8]
146. Risky QLearner [3]
147. SelfSteem [20]
148. ShortMem [20]
149. Shubik [12]
150. Slow Tit For Two Tats [3]
151. Slow Tit For Two Tats 2 [1]
152. Sneaky Tit For Tat [3]
153. Soft Go By Majority [14, 37]
154. Soft Go By Majority 10 [3]
155. Soft Go By Majority 20 [3]
156. Soft Go By Majority 40 [3]
157. Soft Go By Majority 5 [3]
158. Soft Grudger [33]
159. Soft Joss [1]
160. SolutionB1 [7]
161. SolutionB5 [7]
162. Spiteful Tit For Tat [1]
163. Stalker [**?**]
164. Stein and Rapoport [12]
165. Stochastic Cooperator [6]
166. Stochastic WSLS [3]
167. Suspicious Tit For Tat [17, 28]
168. TF1 [3]
169. TF2 [3]
170. TF3 [3]
171. Tester [13]
172. ThueMorse [3]
173. ThueMorseInverse [3]
174. Thumper [8]
175. Tit For 2 Tats (**Tf2T**) [14]
176. Tit For Tat (**TfT**) [12]
177. Tricky Cooperator [3]
178. Tricky Defector [3]
179. Tullock [12]
180. Two Tits For Tat (**2TfT**) [14]
181. VeryBad [20]
182. Willing [50]
183. Win-Shift Lose-Stay (**WShLSt**) [33]
184. Win-Stay Lose-Shift (**WSLS**) [31, 40, 47]
185. Winner12 [35]
186. Winner21 [35]
187. Worse and Worse[1]
188. Worse and Worse 2[1]
189. Worse and Worse 3[1]
190. ZD-Extort-2 v2 [32]
191. ZD-Extort-2 [47]
192. ZD-Extort-4 [3]
193. ZD-GEN-2 [32]
194. ZD-GTFT-2 [47]
195. ZD-SET-2 [32]

Appendix B

Correlation coefficients of features in Chapter 3

A graphical representation of the correlation coefficients for the features of Table ??.

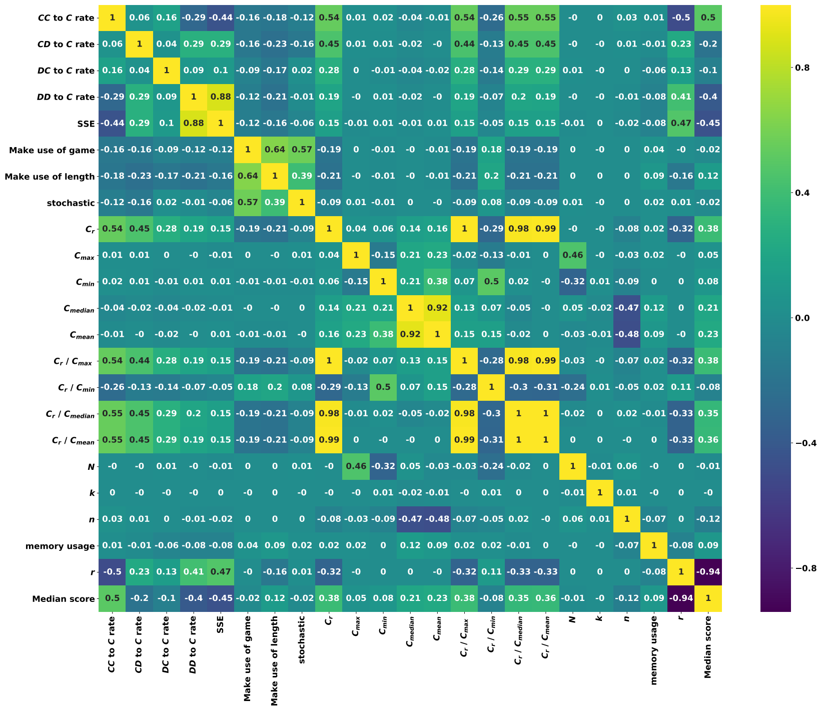


Figure B.1: Correlation coefficients of measures in Table ?? for standard tournaments

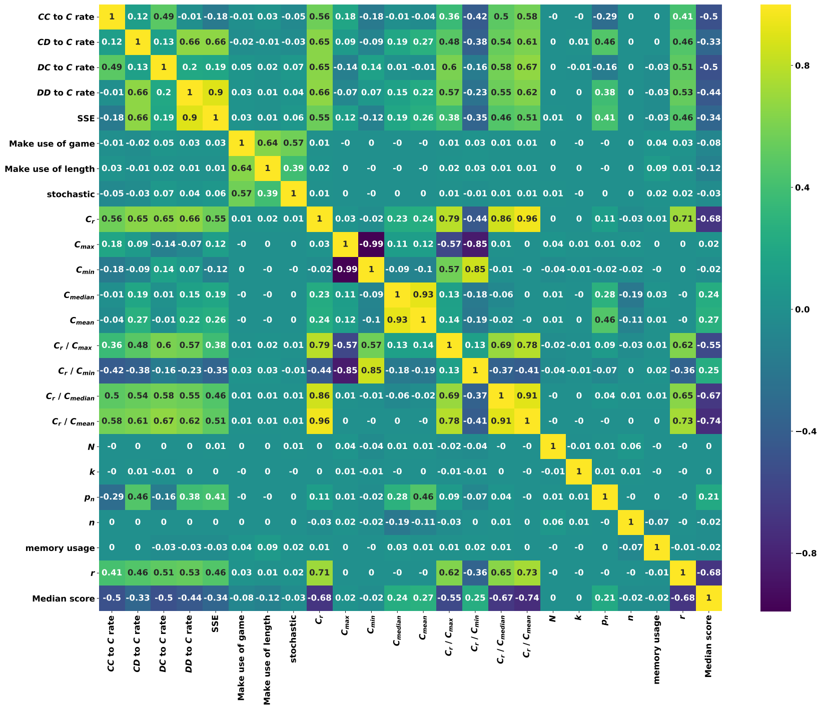


Figure B.2: Correlation coefficients of measures in Table ?? for noisy tournaments

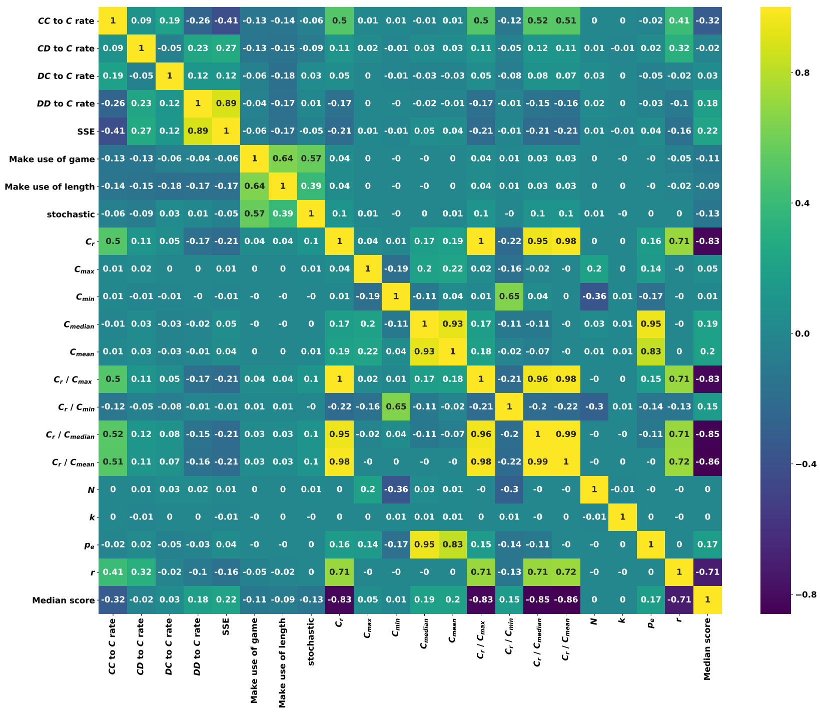


Figure B.3: Correlation coefficients of measures in Table ?? for probabilistic ending tournaments

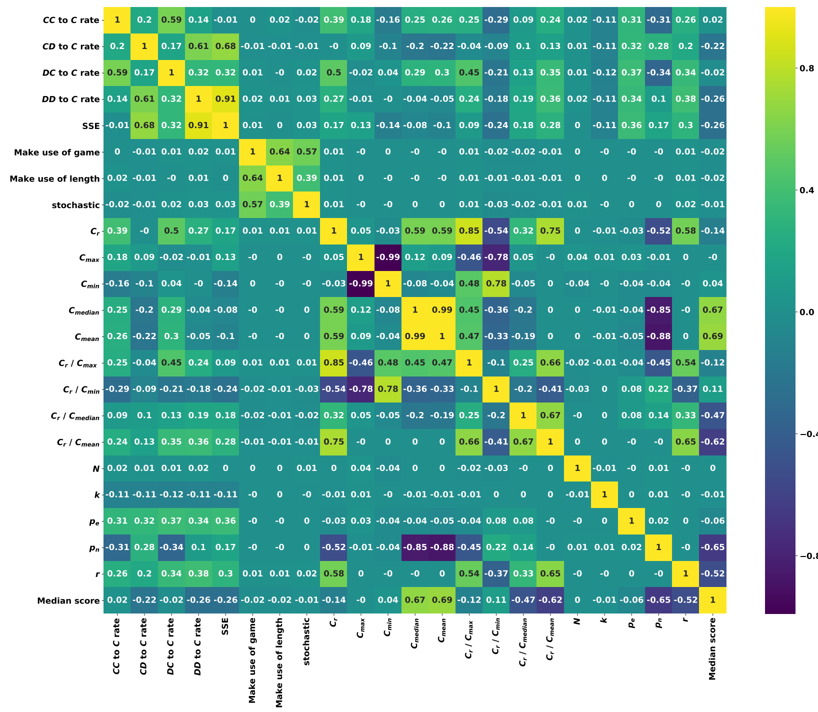


Figure B.4: Correlation coefficients of measures in Table ?? for noisy probabilistic ending tournaments

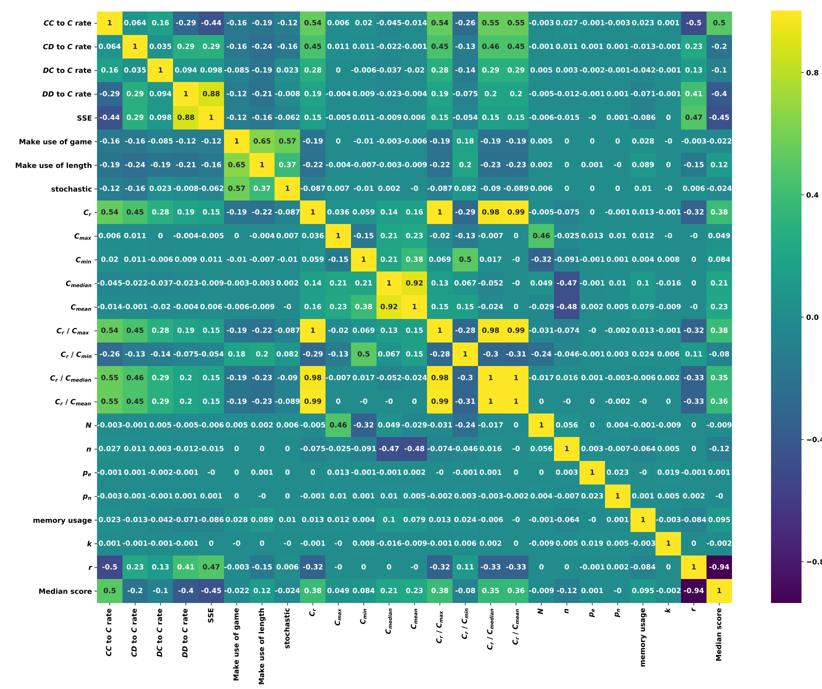


Figure B.5: Correlation coefficients of measures in Table ?? for data set