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.

School of Mathematics
Ysgol Mathemateg

Executive Summary

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Mae-

cenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetuer a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetuer. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Acknowledgements

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Dissemination of Work

Publications (4 Published & 5 In preparation)

1. 2018: **Reinforcement learning produces dominant strategies for the Iterated Prisoner's Dilemma.** Marc Harper, Vincent Knight, Martin Jones, Georgios Koutsovoulos, Nikoleta E. Glynatsi, Owen Campbell - PLOS One - Preprint arXiv:1707.06307
2. 2018: **An evolutionary game theoretic model of rhino horn devaluation.** Nikoleta E. Glynatsi, Vincent Knight, Tamsin Lee. Ecological Modelling - Preprint arXiv:1712.07640
3. 2017: **Evolution reinforces cooperation with the emergence of self-recognition mechanisms: an empirical study of the Moran process for the Iterated Prisoner's dilemma.** Vincent Knight, Marc Harper, Nikoleta E. Glynatsi, Owen Campbell - PLOS ONE - Preprint arXiv:1707.06920
4. 2016: **An open framework for the reproducible study of the Iterated prisoner's dilemma.** Vincent Knight, Owen Campbell, Marc Harper et al - Journal of Open Research Software

In preparation

1. 2019: **A meta analysis of tournaments and an evaluation of performance in the Iterated Prisoner's Dilemma.** Nikoleta E. Glynatsi and Vincent A. Knight - In preparation to be submitted - Preprint arXiv:2001.05911
2. 2019: **A bibliometric study of research topics, collaboration and influence in the field of the Iterated Prisoner's Dilemma.** Nikoleta E. Glynatsi and Vincent A. Knight - In preparation to be submitted - Preprint arXiv:1911.06128
3. 2019: **Game Theory and Python: An educational tutorial to game theory and repeated games using Python.** Nikoleta E. Glynatsi and Vincent A. Knight - Submitted to the Journal of Open Source Education - Available on GitHub Nikoleta-v3/Game-Theory-and-Python
4. 2019: **A theory of mind: Best responses to memory-one strategies. The limitations of extortion and restricted memory.** Nikoleta E. Glynatsi and Vincent A. Knight - Submitted to Scientific Reports Nature - Preprint arXiv:1911.12112
5. 2019: **Recognising and evaluating the effectiveness of extortion in the Iterated Prisoner's Dilemma.** Vincent Knight, Marc Harper, Nikoleta E. Glynatsi, Jonathan Gillard - Submitted to Nature Communications - Preprint arXiv:1904.00973

Talks & Posters

INVITED TALKS (KEYNOTES)

- How does a smile make a difference?, PyCon UK, Cardiff, 2018.
- The Fallacy of Meritocracy, PyCon Balkan, Belgrade, 2019.

OTHERS

- Accessing open research literature with Python - PyCon Namibia, Namibia 2017.
- Writing tests for research software - PyCon Namibia, Namibia 2017.
- Optimisation of short memory strategies in the Iterated Prisoners Dilemma - Wales Mathematics Colloquium, Gregynog Hall 2017.
- PIP INSTALL AXELROD (**poster**) - Euroscipy, Erlangen Germany 2017.
- Arcas: Using Python to access open research literature - Euroscipy, Erlangen Germany 2017.
- A trip to earth science with python as a companion - PyConUK, Cardiff 2017.
- The power of memory (**poster**) - SIAM UKIE Annual Meeting, Southampton 2018.
- Rhinos with a bit of Python - PyConNA, Namibia 2018.
- Memory size in the Prisoners Dilemma - Wales Mathematics Colloquium, Gregynog Hall 2018.
- Memory size in the Prisoners Dilemma - SIAM UKIE National Student Chapter, Bath University 2018.
- Stability of defection, optimisation of strategies and testing for extortion in the Prisoner's Dilemma (**poster**) - STEM for Britain, London 2019.
- Stability of defection, optimisation of strategies and testing for extortion in the Prisoner's Dilemma - 18th International Conference on Social Behaviour, Sedona, Arizona 2019.
- An introduction to Time Series - Joint workshop between CUBRIC & Mathematics Departments, Cardiff 2019.

Software Development

- Arcas, an open source package designed to help users collect academic articles' metadata from various prominent journals and pre print serves. **Contribution:** Main developer
- Axelrod-Python library, an open source framework decided to the study of the Iterated Prisoner's Dilemma. **Contributions:** Implementation of spatial tournaments functionality, implementation/addition of strategies (from the literature) to the library, reviewing of code contributed by other contributors
- SymPy, a Python library for symbolic mathematics. **Contributions:** Implementation of Dixon's and Macaulay's resultants which were developed for my 2019 publication "Sta-

bility of defection, optimisation of strategies and the limits of memory in the Prisoner's Dilemma"

- Pandas, an open source library providing high-performance, easy-to-use data structures and data analysis tools. **Contribution:** Fix bug which converted NaN values to strings

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Chapter 2

A systematic literature review of the Prisoner's Dilemma.

2.1 Introduction

Chapter 1 introduced the Prisoner's Dilemma as the main game theoretic model that will be used throughout this thesis, and presented a brief literature review of the research this thesis is building upon. This Chapter provides a more detailed systematic literature on the Prisoner's Dilemma. The aim of this Chapter is to provide a concrete summary of the existing literature and to identify research topics in the field of the PD. This is achieved by partitioning the literature in five different sections each reviewing a different aspect of research. The Chapter is structured as follows:

- section 2.2 presents the origin of the PD and reviews the early publications in the field and the use of human subject research.
- section 2.3 presents the pioneering computer tournaments of Robert Axelrod and reviews IPD strategies of intelligent design.
- section 2.4 discusses the emergence, or not, of cooperative behaviour in evolutionary dynamics.
- section 2.5 defines structured strategies in the IPD, the notion of training and discusses related papers.
- section 2.6 reports on educational and research software used for simulating the PD game.

2.2 Origins of the prisoner's dilemma

The origin of the PD goes back to the 1950s in early experiments conducted at RAND [40] to test the applicability of games described in [88]. The game received its name later the same year. According to [118], Albert W. Tucker (the PhD supervisor of John Nash [87]), 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, and 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 [39, 76, 79], the physical distance between the participants [108], the effect of their opening moves [117] and even how the experimenter, by varying the tone of their voice and facial expressions [44], 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 [101].

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. These tournaments and several strategies that were designed by researchers, such as Rapoport, are introduced in the following section.

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 [23], 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 and made the study of cooperation of critical interest. As described in [102], "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".

The first reported computer tournament took place in 1980 [19]. Axelrod asked researchers to design a strategy with the purpose of winning an IPD tournament. 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 [19] but no details were given.

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 [20]. 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' [24]) 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 [27], Grim Trigger [26] and Grim [120]. New entries in the second tournament included *Tit for Two Tats* submitted by John Maynard Smith and *KPavlovC*. KPavlovC, is also known as Simpleton [101], introduced by Rapoport or just Pavlov [90]. 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* [73].

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 [82]. For example, in [85] 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 [107]. The poor performance of the strategy in noisy environments was also demonstrated in tournaments. In [29, 36] 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 [122], 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 [27] 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* [119]. 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 [27] with now 13 strategies Adaptive Tit for Tat ranked first.

Another extension of strategies was that of teams of strategies [34, 35, 105] 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 [105]. 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" [95].

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 [91], 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 are 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* [93] $(1, 1, \frac{1}{3})$.

In [92], 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 $(1, 1, 0, 0, 1)$. Though reactive and memory-one strategies have to specify their move in the first round, the opening move is a transient effect and has no affect on the game in long run [?]. Consequently, reactive strategies can be described as elements $p \in R^2$ and memory-one strategies as $p \in R^4$.

Memory-one strategies made an impact when a specific subset of memory-one strategies were introduced called *Zero-determinant* strategies (ZDs) [99]. The American Mathematical Society's news section [56] stated that "the world of game theory is currently on fire" and in [112] it was stated that "Press and Dyson have fundamentally changed the viewpoint on the Prisoner's Dilemma". ZDs 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 that the ZDs were the dominant set of strategies in the IPD, and as memory did not benefit them then they argued that memory is not beneficial for any strategy. In [8, 48, 55, 54, 56, 68, 69, 72, 112] the effectiveness of ZDs is questioned. Namely, [113, 114] showed that memory-one strategies must be forgiving to be evolutionarily stable and [48, 58, 68, 69, 72, 98] demonstrated that longer-memory strategies have an advantage over short memory strategies. Chapter 5, studies the set of memory-one strategies, and more specifically, best response memory-one strategies and reinforces the discussion that the best action is adaptability and not manipulation, and short memory can be limiting.

This section of the literature covered the original computer tournaments of Axelrod, the early success of Tit For Tat in these tournaments and excessive amounts of IPD strategies. 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. The strategies covered in this section are all strategies of intelligent design. They have been designed by researchers and not surfaced from an indirect process, such strategies are covered in section 2.5.

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 (Chapter 1) and the IPD round robin tournaments (Sections 2.3). In the PD it is known 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 favour cooperation? Understanding the conditions under which natural selection can favour cooperative behaviour is important in understanding social behaviour amongst intelligent agents [31].

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.

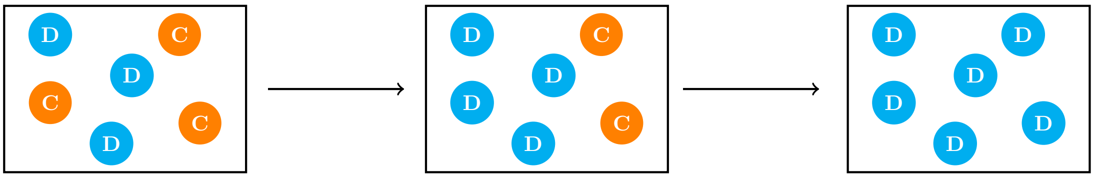


Figure 2.1: Natural selection favours defection in a mixed population of Cooperators and Defectors.

In the later sections of [20], 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 an 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. That was because the weaker strategies which were being exploitative were becoming extinct, and exploitative strategies were loosing their prey.

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 [109, 110, 111]. John M. Smith was a fundamental figure in evolutionary game theory and a participant of Axelrod's second tournament. The biological applications of the new evolutionary approach [21] won Axelrod and his co-author William Donald Hamilton the Newcomb-Cleveland prize of the American Association for the Advancement of Science. In [21] 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 w, where $0 < w < 1$, that the pair would interact again. This was referred to as the *importance of the future* of the game. 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 [31] 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 [100].

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 [77] 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.

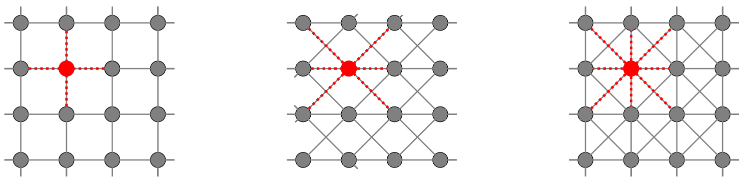


Figure 2.2: Spatial neighbourhoods

Limited/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 [97], whilst using the donation game (Equation (**??**)), 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.

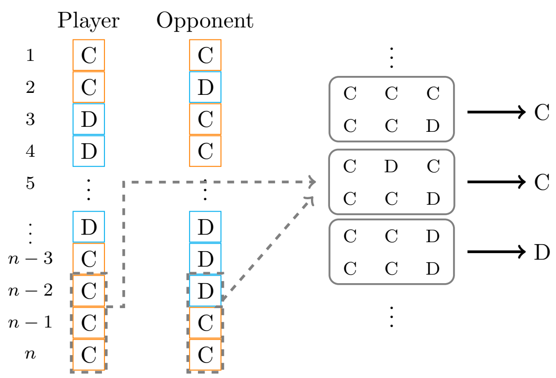
In structured populations local interactions that can dynamically change were considered in [123]. Graphs with a probability of rewiring connections were considered, and the rewire could be with any given node in the graphs and not just with immediate neighbours. Perc et al. concluded that "making new 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.

Another approach for increasing the likelihood of cooperation by increasing of assortative interactions among cooperative agents, include partner identification methods such as reputation [60, 94, 116], communication tokens [84] and tags [32, 47, 84, 104].

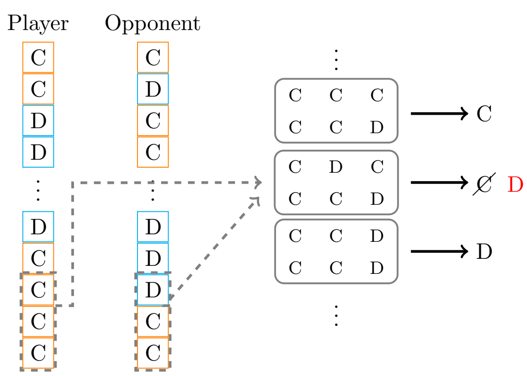
This section considered papers on evolutionary dynamics and mechanisms that ensure the emergence, or not, of cooperation. The following section focuses on strategy archetypes, training methods and strategies obtained from training.

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 through a *training process* using generic strategy archetypes. For example, in [22] Axelrod explored deterministic strategies that took into account the last 3 plays of both players. 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*. A graphical representation of the look up table strategy in [22] is given by Figure 2.3a. In [22] lookup tables were trained using a genetic algorithm [71]. A training process includes making random changes to a given instant of the lookup table, Figure 2.3b. The strategy which corresponds to the new altered instant is evaluated in a given setting set by the experiment, and if the utility of the strategy has increased this change is kept and its genes are passed on to a new generation of strategies. A genetic algorithm is not the only heuristic method which can be used for training strategies, realistically any heuristic method can be used.



(a) A graphical representation of a look up table player which considers 3 plays of both players.



(b) Training a look up table player includes making changes to the strategy's responses to a history.

Figure 2.3: A graphical representation of the lookup table strategy described in [22], and a demonstration of the changes a strategy exhibits during training.

In 1996 John Miller considered finite state automata as an archetype [83], more specifically, Moore machines [86]. The training process used a genetic algorithm and the strategies were evaluated in a tournament 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*. A training combination of finite state automata and a genetic algorithm was also considered in [18]. 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*.

Also, in 1996 the first structured strategies based on neural networks that had been trained using a genetic algorithm were introduced in [50] 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 [22]. Neural networks have broadly been used since 1996 to train IPD strategies [17, 33, 80, 41] with training methods such as genetic algorithms [17, 33, 80, 41] and particle swarm optimization [41]. Chapter 7 of this thesis discusses the training of strategies using neural network in more details, as the aim of the chapter is to use an extension of a neural network, a *recurrent neural network*, to train an IPD strategy.

In [48, 68] 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 [38] 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 [5] 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 [68] 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 2.4 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 2.4: 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 [13] by Daniel Ashlock. The method of fingerprinting is a technique for generating a functional signature for a strategy [14]. 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 [13] Tit for Tat is used as the probe strategy. In Figure 2.5 an example of Pavlov's fingerprint is given. Fingerprinting has been studied in depth in [14, 15, 16, 17]. Another type of fingerprinting is the *transitive fingerprint* [5]. 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 2.6.

This section covered a series of structured strategies based on different archetypes which have been trained via different training methods. The works discussed in this section has demonstrated that through these indirect training processes successful IPD strategies can emerge. This thesis explores both strategies of intelligent design (Chapter 5) and trained strategies

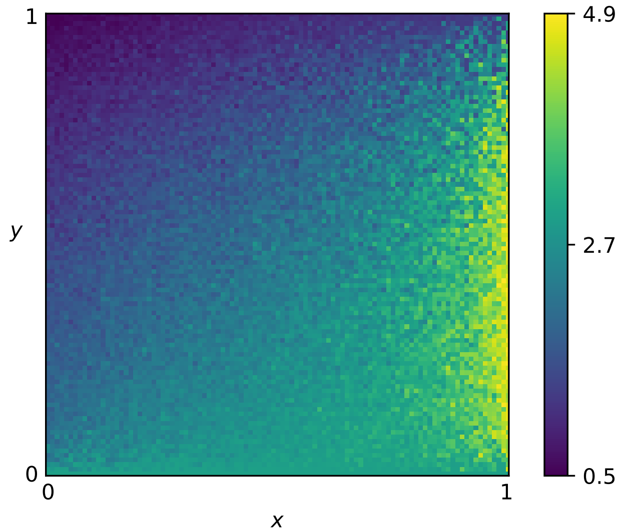


Figure 2.5: Pavlov fingerprinting with Tit for Tat used as the probe strategy. Figure was generated using [5].

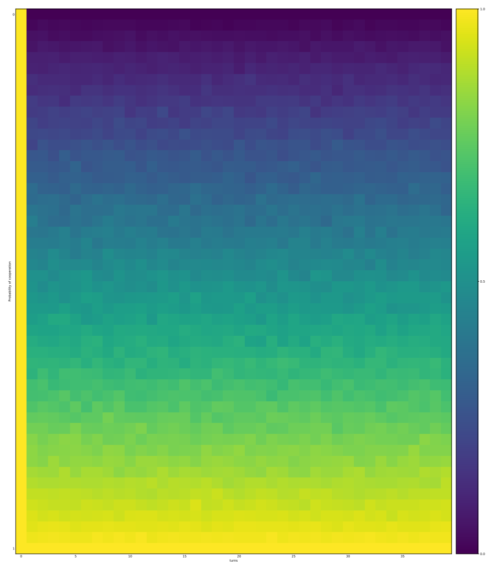


Figure 2.6: Transitive fingerprint of Tit for Tat against a set of 50 random opponents with varying cooperation rate.

(Chapter 7)in more details. The next section covers software that has been developed with the main aim of simulating the IPD interactions.

2.6 Software

Aside from human subject research 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 [1]).

Several projects, however, are open, available and have been used as research tools or educational platforms over the years. Two such tools include [2] and [59]. The "Game of Trust" [2] is an on-line, educational platform accessed through a graphical user interface, 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" [59] and "a whole heap of fun" [67]. Secondly, [3] 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.

Open source projects used for research include [4, 5]. PRISON [4] is written in the programming language Java and a preliminary version was launched in 1998. It was used by its authors in several publications, such as [27], which introduced Gradual, and [28]. 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 [5] is a software used in a number of works including [48, 68, 46, 121]. It is written in the programming language Python following best practice approaches [7, 30] and contains the largest collection of strategies, known to the author. The strategy list of the project has been cited by publications [11, 52, 89], and is used in this thesis for Chapter 4 and Chapter 6.

2.7 Chapter summary

This Chapter presented a literature review of 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 [96]. The game now serves as a model in a wide range of applications, for example in medicine and the study of cancer cells [12, 65], as well as in social situations and how they can be driven by rewards [37]. This thesis aims to contribute to the continued understanding of this well known and widely applied game theoretic model. Many of the papers reviewed in this Chapter have served as motivation to the research presented in the following Chapters. In Chapter 4 the performance of several of the strategies mentioned in this Chapter is evaluated in a large number of tournaments. Chapter 5 explores the set of memory-one strategies, and Chapters 6 and 7 explore trained strategies based on archetypes such as sequences and recurrent neural networks.

Chapter 3

A bibliometric study of research topics, collaboration and influence in the field of the Iterated Prisoner's Dilemma

The research reported in this Chapter has lead to a manuscript, entitled:
"A bibliometric study of research topics, collaboration and influence in the field of the Iterated Prisoner's Dilemma"
Available at: `arxiv.org/abs/1911.06128`
Associated data set: [?], [?], [?]

The manuscript's abstract is the following:

This manuscript explores the research topics and collaborative behaviour of authors in the field of the Prisoner's Dilemma using topic modelling and a graph theoretic analysis of the co-authorship network. The analysis identified five research topics in the Prisoner's Dilemma which have been relevant of the course of time. These are human subject research, biological studies, strategies, evolutionary dynamics on networks and modelling problems as a Prisoner's Dilemma game. Moreover, the results demonstrated the Prisoner's Dilemma is a field of continued interest, and although it is a collaborative field, it is not necessarily more collaborative than other scientific fields. The co-authorship network suggests that authors are focused on their communities and not many connections across the communities are made. The Prisoner Dilemma authors also do not influence or gain much information by their connections, unless they are connected to a "main" group of authors.

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

3.1 Introduction

This Chapter presents a bibliometric analysis of the data set "Articles' meta data on the Prisoner's Dilemma" [?]. Chapter 2 presented a review of published works on the PD, and manually assigned them to different topics. To complement that manual identification of topics this Chapter presents an automatic approach using natural language processing. More specifically, the 2422 articles' metadata in [?] are used to extract a list of research topics in the field. Moreover, the list of authors in [?] is also used to generate a co-authorship network and explore their collaborative behaviour. The Chapter is structured as follows:

- section 3.2, covers the data collection and an introduction to topic modelling and co-authorship networks.
- section 3.3, presents a preliminary analysis of the data set.
- section 3.4, identifies research topics in the field using natural language processing.
- section 3.5, evaluates the collaborative behaviour of the field.

3.2 Methodology

As discussed in [?], bibliometrics (the statistical analysis of published works originally described by [?]) has been used to support historical assumptions about the development of fields [?], identify connections between scientific growth and policy changes [?], develop a quantitative understanding of author order [?], and investigate the collaborative structure of an interdisciplinary field [?]. Most academic research is undertaken in the form of collaborative effort and as [?] points out, it is rational that two or more people have the potential to do better as a group than individually. Indeed this is the very premise of the PD itself. Collaboration in groups has a long tradition in experimental sciences and it has be proven to be productive according to [?]. The number of collaborations can be different between research fields and understanding how collaborative a field is not always an easy task. Several studies tend to consider academic citations as a measure for these things. A blog post published by Nature [?] argues that depending on citations can often be misleading because the true number of citations can not be known. Citations can be missed due to data entry errors, academics are influenced by many more papers than they actually cite and several of the citations are superficial.

A more recent approach to measuring collaborative behaviour, and to studying the development of a field is to use the co-authorship network, as described in [?]. The co-authorship network has many advantages as several graph theoretic measures can be used as proxies to explain author relationships. For example the average degree of a node corresponds to the average number of an authors' collaborators, and clustering coefficient corresponds to the extent that two collaborators of an author also collaborate with each other. In [?], the approach was applied to analyse the development of the field "evolution of cooperation", and in [?] to identify the subdisciplines of the interdisciplinary field of "cultural evolution" and investigate trends in collaboration and productivity between these subdisciplines. This Chapter builds on the works of [?] and [?] and extends their methodology. This is described in section 3.2.2.

Latent Dirichlet Allocation (LDA) is a topic modelling technique proposed in [?] as a generative probabilistic model for discovering underlying topics in collections of data. Applications of the technique include detection in image data [?, ?] and detection in video [?, ?]. Nevertheless, LDA has been applied by several works on publication data for identifying the topic structure of a subject area. In [?], it was applied to the publications on mathematical education of the journals "Educational Studies in Mathematics" and "Journal for Research in Mathematics Education", in [?] to the dissertations of the North American library and Information Science and in [?] to conference papers presented at EvoLang conferences. LDA is the topic modelling technique used in this thesis. An introduction to the technique is presented in section 3.2.3.

Several of the approaches of this Chapter have previously been carried out in [?, ?, ?, ?]. The novelty presented in thesis is the combination of these approaches and their application to a new data set. The data sets of [?] and [?] are from a single source, the Web of Science, whereas the data set [?] considered here has been collected from five sources using a bespoke open source software. This software and the data collection process are presented in section 3.2.1.

3.2.1 Data Collection

Academic articles are accessible through scholarly databases. Several databases and collections today offer access through an open application protocol interface (API). An API allows users to query directly a journal's database and bypass the graphical user interface. Interacting with an API has two phases: requesting and receiving. The request phase includes composing a url with the details of the request. For example, `http://export.arxiv.org/api/query?search_query=abs:prisoner'sdilemma&max_results=1` represents a request message. The first part of the request is the address of the API. In this example the address corresponds to the API of arXiv. The second part of the request contains the search arguments. In this example it is requested a single article that the word 'prisoners dilemma' exists within the article's title. The format of the request message is different from API to API. The receive phase includes receiving a number of raw metadata of articles that satisfies the request message. The raw metadata are commonly received in extensive markup language (xml) or Javascript object notation (json) formats [?]. Similarly to the request message, the structure of the received data differs from journal to journal.

The data collection is crucial to this chapter. To ensure that the research reported in this Chapter can be reproduced all code used to query the different APIs has been packaged as a Python library called *Arcas*. The source code of the library has been made available online and the package includes documentation of usage which is available at: `http://arcas.readthedocs.io/en/latest/`. Arcas allows users to communicate with a list of APIs by specifying a single keyword whilst not considering the differences between the requesting and receiving phases of the APIs. Consider the example of retrieving a single article with the word 'Prisoners Dilemma' in the title. Figure 3.1 demonstrates the Python code needed to query the publisher PLOS and Figure 3.2 demonstrates code for querying the API of Nature.

The only distinction between the two code snippets is their respective line 2 where the API is specified by creating an instance of a class corresponding to the publisher's API. The differences between querying the two APIs are visible from lines 6 and 12-onwards. Lines 6 show the requesting message and lines 12-onwards show the metadata of the article received by each source.

There are differences and similarities between the retrievable metadata of each API. Arcas

```
>>> import arcas
>>> api = arcas.Plos()
>>> parameters = api.parameters_fix(title="Prisoner's Dilemma", records=1)
>>> url = api.create_url_search(parameters)
>>> url
'http://api.plos.org/search?q=title:"Prisoner\'s Dilemma"&rows=1'

>>> request = api.make_request(url)
>>> root = api.get_root(request)
>>> article = api.parse(root)

>> article
[{'id': '10.1371/journal.pone.0028576',
'journal': 'PLoS ONE',
'eissn': '1932-6203',
'publication_date': '2011-12-14T00:00:00Z',
'article_type': 'Research Article',
'author_display': ['Irina Kareva'],
'abstract': ["As tumors outgrow their ..."],
'title_display': "Prisoner's Dilemma in Cancer Metabolism",
'score': 21,
'author': ['Irina Kareva'],
'date': 2011,
'provenance': 'PLOS',
'doi': '10.1371/journal.pone.0028576',
'url': 'https://doi.org/10.1371/journal.pone.0028576',
'title': "Prisoner's Dilemma in Cancer Metabolism",
'key': 'Kareva2011',
'unique_key': '0d56101113057d99fc6d83095812735a',
'category': 'Not available',
'open_access': 'Not available'}]
```

Figure 3.1: Example of using the library Arcas to communicate the API of the publisher PLOS. The query is for a single article with the word 'prisoners dilemma' in the title.

```
>>> import arcas
>>> api = arcas.Nature()
>>> parameters = api.parameters_fix(title="Prisoner's Dilemma", records=1)
>>> url = api.create_url_search(parameters)
>>> url
"http://www.nature.com/opensearch/request?&query=dc.title adj Prisoner's Dilemma&maximumRecords=1"

>>> request = api.make_request(url)
>>> root = api.get_root(request)
>>> article = api.parse(root)

>> article
[{'records': None,
'record': None,
'recordSchema': 'info:srw/schema/11/pam-v2.1',
'recordPacking': 'packed',
'recordData': None,
'message': None,
'article': None,
'head': None,
'identifier': 'doi:10.1057/ces.1994.6',
'title': """Survey Article: Cooperate or Defect? Russian and American Students
in a Prisoner's Dilemma""",
'creator': 'Michael Hemesath',
'productCode': 'ces',
'publicationName': 'Comparative Economic Studies',
'issn': '0888-7233',
'eIssn': '1478-3320',
'doi': '10.1057/ces.1994.6',
'publisher': 'Palgrave Macmillan',
'publicationDate': '1994-04',
'volume': '36',
'number': '1',
'startingPage': '83',
'endingPage': '93',
'url': 'http://dx.doi.org/10.1057/ces.1994.6',
'genre': 'Research',
'description': "<p>Do assumptions underlying the models of ...",
'copyright': '© 1994 Palgrave Macmillan Ltd',
'aggregationType': 'issue'}]
```

Figure 3.2: Example of using the library Arcas to communicate the API of the publisher Nature. The query is for a single article with the word 'prisoners dilemma' in the title.

includes a function which standarises the format of querying results. Figure 3.3 demonstrates the usage of the function.

```
>> meta_data = api.to_dataframe(article[0])
>> meta_data.columns
Index(['url', 'key', 'unique_key', 'title', 'author', 'abstract', 'doi',
       'date', 'journal', 'provenance', 'category', 'score', 'open_access'],
      dtype='object')
```

Figure 3.3: Python Code. Arcas includes a function which standarises the results of the queries regarding the API.

At the time of writing there are a total of five different APIs implemented within the project. These five include APIs of four prominent publishers in the field and a preprint server. Namely these are:

- arXiv [?]; a repository of electronic preprints. It consists of scientific papers in the fields of mathematics, physics, astronomy, electrical engineering, computer science, quantitative biology, statistics, and quantitative finance, which all can be accessed online.
- PLOS [?]; a library of open access journals and other scientific literature under an open content license. It launched its first journal, PLOS Biology, in October 2003 and publishes seven journals, as of October 2015.
- IEEE Xplore Digital Library (IEEE) [?]; a research database for discovery and access to journal articles, conference proceedings, technical standards, and related materials on computer science, electrical engineering and electronics, and allied fields. It contains material published mainly by the Institute of Electrical and Electronics Engineers and other partner publishers.
- Nature [?]; a multidisciplinary scientific journal, first published on 4 November 1869. It was ranked the world's most cited scientific journal by the Science Edition of the 2010 Journal Citation Reports and is ascribed an impact factor of 40.137, making it one of the world's top academic journals.
- Springer [?]; a leading global scientific publisher of books and journals. It publishes close to 500 academic and professional society journals.

Each APIs has a corresponding class implemented in Arcas. The classes include a series of methods which allow Arcas to communicate with the APIs. An example of an API class is given by both Figure 3.4 and Figure 3.5. These include the classes for the APIs of arXiv and IEEE. Note that IEEE is an example of an API which requires a user to have an access key (line 7 in Figure 3.5). An access key can be required from the publishers website and for the APIs of this work they can be acquired for free.

As mentioned in Chapter 1 the source code associated with the research projects of this thesis have been written following a set of best practices. These best practices include unit testing. There are a series of unit tests that test the functionality and correctness of each API class. For example, Figure 3.6 displays a test case for the method `to_dataframe` of the class Arxiv.

```python
class Arxiv(Api):
    def __init__(self):
        self.standard = 'http://export.arxiv.org/api/query?search_query='

    def to_dataframe(self, raw_article):
        """A function which takes a dictionary with structure of the arXiv results,
        transforms it to a standardized format and returns a dataframe."""
        raw_article['url'] = raw_article.get('id', None)

        for key_one, key_two in [['author', 'name'], ['category', 'category']]:
            raw_article[key_one] = raw_article.get(key_two, None)
            if raw_article[key_one] is not None:
                raw_article[key_one] = raw_article[key_one].split(',')

        raw_article['abstract'] = raw_article.get('summary', None)
        raw_article['date'] = int(raw_article.get('published', '0').split('-')[0])
        raw_article['journal'] = raw_article.get('journal_ref', None)
        if raw_article['journal'] is None:
            raw_article['journal'] = "arXiv"

        raw_article['provenance'] = 'arXiv'
        raw_article['title'] = raw_article.get('title', None)
        raw_article['doi'] = raw_article.get('doi', None)
        raw_article['key'], raw_article['unique_key'] = self.create_keys(raw_article)

        raw_article['open_access'] = True
        raw_article['score'] = 'Not available'
        return self.dict_to_dataframe(raw_article)

    def parse(self, root):
        """Removing unwanted branches."""
        branches = root.getchildren()
        raw_articles = []
        for record in branches:
            if 'entry' in record.tag:
                raw_articles.append(self.xml_to_dict(record))
        if not raw_articles:
            raw_articles = False
        return raw_articles

    @staticmethod
    def parameters_fix(author=None, title=None, abstract=None, year=None, records=None,
                       start=None, category=None, journal=None, keyword=None):
        parameters = []
        if author is not None:
            parameters.append('au:{}'.format(author))
        if title is not None:
            parameters.append('ti:{}'.format(title))
        if abstract is not None:
            parameters.append('abs:{}'.format(abstract))
        if category is not None:
            parameters.append('cat:{}'.format(category))
        if journal is not None:
            parameters.append('jr:{}'.format(journal))
        if keyword is not None:
            parameters.append('all:{}'.format(keyword))
        if records is not None:
            parameters.append('max_results={}'.format(records))
        if start is not None:
            parameters.append('start={}'.format(start))
        if year is not None:
            print('ArXiv does not support argument year.')

        return parameters

    @staticmethod
    def get_root(response):
        root = ElementTree.fromstring(response.text)
        return root
```

Figure 3.4: Class Arxiv is implemented in Arcas. It includes the code necessary for Arcas to query the API of arXiv.

```python
class Ieee(Api):
    """
        API argument is 'ieee'.
    """
    def __init__(self):
        self.standard = 'https://ieeexploreapi.ieee.org/api/v1/search/articles?'
        self.key_api = api_key

    def create_url_search(self, parameters):
        """Creates the search url, combining the standard url and various
        search parameters."""
        url = self.standard
        url += parameters[0]
        for i in parameters[1:]:
            url += '&{}'.format(i)
        url += '&apikey={}'.format(self.key_api)
        return url

    @staticmethod
    @ratelimit.rate_limited(3)
    def make_request(url):
        """Request from an API and returns response."""
        response = requests.get(url, stream=True, verify=False)
        if response.status_code != 200:
            raise APIError(response.status_code)
        return response

    def parse(self, root):
        """Parsing the xml file"""
        if root['total_records'] == 0:
            return False
        return root['articles']

    @staticmethod
    def parameters_fix(author=None, title=None, abstract=None, year=None,
                       records=None, start=None, category=None, journal=None,
                       keyword=None):
        parameters = []
        if author is not None:
            parameters.append('author={}'.format(author))
        if title is not None:
            parameters.append('article_title={}'.format(title))
        if abstract is not None:
            parameters.append('abstract={}'.format(abstract))
        if year is not None:
            parameters.append('publication_year={}'.format(year))
        if category is not None:
            parameters.append('index_terms={}'.format(category))
        if journal is not None:
            parameters.append('publication_title={}'.format(journal))
        if keyword is not None:
            parameters.append('querytext={}'.format(keyword))
        if records is not None:
            parameters.append('max_records={}'.format(records))
        if start is not None:
            parameters.append('start_record={}'.format(start))

        return parameters

    @staticmethod
    def get_root(response):
        root = response.json()
        return root
```

Figure 3.5: Class Ieee is implemented in Arcas. It includes the code necessary for Arcas to query the API of IEEE.

Moreover, Figure 3.7 shows several unit tests which ensure that the request url for IEEE, with different search arguments, is being generated correctly.

```
import arcas

def test_to_dataframe():
    dummy_article = {'entry': '\n', 'id': 'http://arxiv.org/abs/0000',
                     'updated': '2011', 'published': '2010', 'title': 'Title',
                     'summary': "Abstract", 'author': '\n', 'name': 'E Glynatsi, V Knight',
                     'doi': '10.0000', 'comment': 'This is a comment.',
                     'journal_ref': 'Awesome Journal', 'primary_category': 'Dummy',
                     'category': None}
    api = arcas.Arxiv()
    article = api.to_dataframe(dummy_article)

    assert isinstance(article, pandas.core.frame.DataFrame)
    assert list(article.columns) == api.keys()
    assert len(article['url']) == 2

    assert article['url'].unique()[0] == 'http://arxiv.org/abs/0000'
    assert article['key'].unique()[0] == 'Glynatsi2010'
    assert article['title'].unique()[0] == 'Title'
    assert article['abstract'].unique()[0] == 'Abstract'
    assert article['journal'].unique()[0] == 'Awesome Journal'
    assert article['primary_category'].unique()[0] == 'Dummy'
    assert article['category'].unique()[0] == None
    assert article['score'].unique()[0] == 'Not available'
    assert article['open_access'].unique()[0] == True
```

Figure 3.6: Unit tests for class Arxiv. Tests the functionality of the method `to_dataframe`.

```
import arcas

def test_setup():
    api = arcas.Ieee()
    assert api.standard == 'https://ieeexploreapi.ieee.org/api/v1/search/articles?'

def test_parameters_and_url_author():
    api = arcas.Ieee()
    parameters = api.parameters_fix(author='Glynatsi')
    assert parameters == ['author=Glynatsi']

    url = api.create_url_search(parameters)
    assert url == 'https://ieeexploreapi.ieee.org/api/v1/search/articles?author=Glynatsi&apikey=Your key here'

def test_parameters_and_url_title():
    api = arcas.Ieee()
    parameters = api.parameters_fix(title='Game')
    assert parameters == ['article_title=Game']

    url = api.create_url_search(parameters)
    assert url == 'https://ieeexploreapi.ieee.org/api/v1/search/articles?article_title=Game&apikey=Your key here'
```

Figure 3.7: Unit tests for class Ieee.

The 2422 articles metadata [?] explored in this Chapter has been collected using Arcas. More specifically, articles for which any of the terms "prisoner's dilemma", "prisoners dilemma", "prisoner dilemma", "prisoners evolution", "prisoner game theory" existed within the title, the abstract or the text are included in the analysis. The data set has been archived and is available at [?]. Note that the latest data collection was performed on the 30$^{\text{th}}$ November 2018.

3.2.2 Co-authorship Network

The relationship between the authors within a field will be modelled as a graph $G = (V_G, E_G)$ where V_G is the set of nodes and E_G is the set of edges. The set V_G represents the authors and an edge connects two authors if and only if those authors have written together. This co-authorship network is constructed using the data set [?] and the open source package [?]. The PD network is denoted as G where the number of unique authors $|V(G)|$ is 4226 and $|E(G)|$ is 7642 . All authors' names were formatted as their first name and last name (i.e. Martin A. Nowak to Martin Nowak). This was done to avoid errors such as Martin A. Nowak and Martin Nowak being treated as a different person. There are some authors for which only their first initial was found. These entries are left as such.

The collaborativeness of the authors will be analysed using measures such as, isolated nodes, connected components, clustering coefficient, communities, modularity and average degree. These measures show the number of connections authors can have and how strongly connected these people are. The number of isolated nodes is the number of nodes that are not connected to another node, thus the number of authors that have published alone. The average degree denotes the average number of neighbours for each nodes, i.e. the average number of collaborations between the authors. A connected component is a maximal set of nodes such that each pair of nodes is connected by a path [?]. The number of connected components as well as the size of the largest connected component in the network are reported. The size of the largest connected component represents the scale of the central cluster of the entire network, as will be discussed in later parts. Clustering coefficient and modularity are also calculated. The clustering coefficient, defined as 3 times the number of triangles on the graph divided by the number of connected triples of nodes, is a local measure of the degree to which nodes in a graph tend to cluster together in a clique [?]. It shows to which extent the collaborators of an author also write together.

In comparison, modularity is a global measure designed to measure the strength of division of a network into communities. The number of communities will be reported using the Clauset-Newman-Moore method [?]. Also the modularity index is calculated using the Louvain method described in [?]. The value of the modularity index can vary between $[-1, 1]$, a high value of modularity corresponds to a structure where there are dense connections between the nodes within communities but sparse connections between nodes in different communities. That means that there are many sub communities of authors that write together but not across communities.

Two further points are aimed to be explored in this thesis, (1) which people control the flow of information; as in which people influence the field the most and (2) which are the authors that gain the most from the influence of the field. To measure these concepts centrality measures are going to be used. Centrality measures are often used to understand different aspects of social networks [?]. The two centrality measures chosen here are closeness and betweenness centrality.

1. In networks some nodes have a short distance to a lot of nodes and consequently are able to spread information on the network very effectively. A representative of this idea is **closeness centrality**, where a node is seen as centrally involved in the network if it requires only few intermediaries to contact others and thus is structurally relatively

independent. Closeness centrality is interpreted as influence. Authors with a high value of closeness centrality, are the authors that spread scientific knowledge easier on the network and they have high influence.

2. Another centrality measure is the **betweenness centrality**, where the determination of an author's centrality is based on the quotient of the number of all shortest paths between nodes in the network that include the node in question and the number of all shortest paths in the network. In betweenness centrality the position of the node matters. Nodes with a higher value of betweenness centrality are located in positions that a lot of information pass through, this is interpreted as the gain from the influence, thus these authors gain the most from their networks.

3.2.3 Topic Modelling

The articles contained in the data set will be classified into research topics using LDA, a topic modelling technique designed to summarize large collections of documents by a small number of conceptually connected topics or themes [?, ?]. LDA is carried out using [?].

The input to an LDA is a collection of documents, and the collection of documents considered here are the articles' abstracts. The output of an LDA is an $N \times n$ matrix - N rows for N abstracts and n columns for n topics. The cells contain the percentage contributions for each topic for each abstract, c_i^j for $i \in \{1, 2, \dots, n\}$ for $j \in \{1, 2, \dots, N\}$. Thus each document/abstract is represented by a distribution over topics, and the topics themselves are represented by a distribution over words. More specifically, each topics is described by weights associated with words. For example assume two topics A and B were the words and their associated weights are:

- Topic A: $0.039\times$"cooperation", $0.028\times$"study" and $0.026\times$"human".
- Topic B: $0.020\times$"cooperation", $0.028\times$"agents" and $0.026\times$"strategies".

The percentage contribution for a document with abstract "The study of cooperation in humans" has a $c_A = 0.039 + 0.028 + 0.026 = 0.093$ and $c_B = .020 + 0.0 + 0.0 = 0.020$. In essence, LDA maps every paper to a vector. In this example the document is mapped to $[0.093, 0.020]$. Each document has a dominant topic to which is going to be assigned in. The dominant topic is the topic with the highest percentage contribution denoted as c^*. For the given example the dominant topic is Topic A $c^* = c_A$.

LAD requires that the number of topics is specified in advance before running the algorithm. The number of topics can be chosen using the coherence value [?] or through subjective minimisation of the overlapping keywords between two topics. Both these approaches will be used in this work. Preceding the analysis of research topics, the next the next section presents a preliminary analysis of the data set.

3.3 Preliminary Analysis

The data set [?] consists of 2422 articles with unique titles. In case of duplicates the preprint version of an article (collected from arXiv) was dropped. Similarly to [?], 76 articles have been manually added throughout the writing of Chapter 2 because they were of specific interest.

These papers include [40] the first publication on the PD, [97, 112] two well cited articles in the field, and a series of works from Robert Axelrod [19, 20, 22, 21, 104].

A more detailed summary of the articles' provenance is given by Table 3.1. Only 3% of the data set consists of articles that were manually added and 27% of the articles were collected from arXiv. The average number of publications is also included in Table 3.1. Overall an average of 43 articles are published per year on the topic. The most significant contribution to this appears to be from arXiv with 11 articles per year, followed by Springer with 9 and PLOS with 8.

	Number of Articles	Percentage %	Year of first publication	Average number of publications per year
IEEE	294	12.14%	1973	5
Manual	76	3.14%	1951	1
Nature	436	18.00%	1959	8
PLOS	477	19.69%	2005	8
Springer	533	22.01%	1966	9
arXiv	654	27.00%	1993	11
Overall	2470	100.00%	1951	43

Table 3.1: Summary of [?] per provenance.

The data handled here is in fact a time series from the 1950s, the formulation of the game, until 2018 (Figure 3.8). Two observations can be made from Figure 3.8.

1. There is a steady increase of the number of publications since the 1980s and the introduction of computer tournaments [21].
2. There is a decrease in 2017-2018. This is due to our data set being incomplete. Articles that have been written in 2017-2018 have either not being published or were not retrievable by the APIs at the time of the last data collection.

These observations can be confirmed by studying the time series. Using [?], an exponential distribution is fitted to the data. The fitted model can be used to forecast the behaviour of the field for the next 5 years. Even though the time series has indicated a slight decrease, the model forecasts that the number of publications will keep increasing, thus demonstrating that the field of the PD continues to attract academic attention.

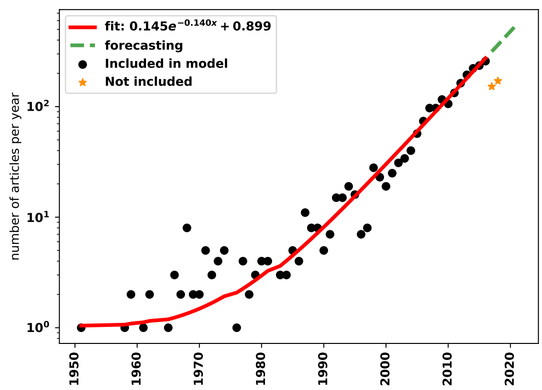


Figure 3.8: Number of articles published on the PD 1951-2018 (on a log scale), with a fitted exponential line, and a forecast for 2017-2022.

There are a total of 4226 authors in the data set and several of these authors have had multiple publications collected from the data collection process. The highest number of articles collected for an author is 83 publications for Matjaz Perc. The distribution of the number of papers per author is given by Figure 3.9, and it can be seen that Matjaz Perc is an outlier. More specifically, most authors have 1 to 6 publications in the data set.

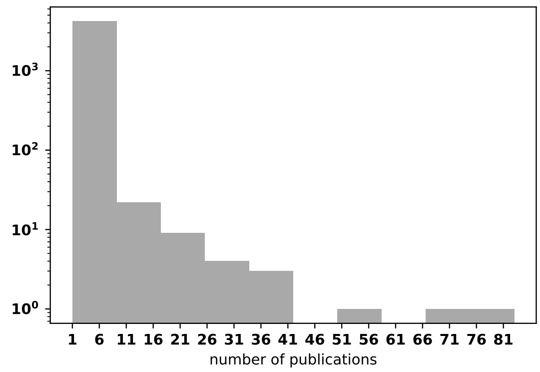


Figure 3.9: Distribution of number of papers per author (on a log scale).

The overall Collaboration Index (CI) or the average number of authors on multi-authored papers is 3.2, thus on average a non single author publication in the PD has 3 authors. This appears to be quite standard compared to other fields such as cultural evolution [?], Astronomy and Astrophysics, Genetics and Heredity, Nuclear and Particle Physics as reported by [?]. There are only a total of 545 publications with a single author, which corresponds to the 22% of the papers. It appears that academic publications tend to be undertaken in the form of collaborative effort, which is in line with the claim of [?]. From Figure 3.10 the trend of CI over the years is given. There are some peaks in the early years 1969 and 1980, however, a steady increase appears to happen after 2004. This could be an effect of better communication tools being introduced around that time which enabled more collaborations between researchers.

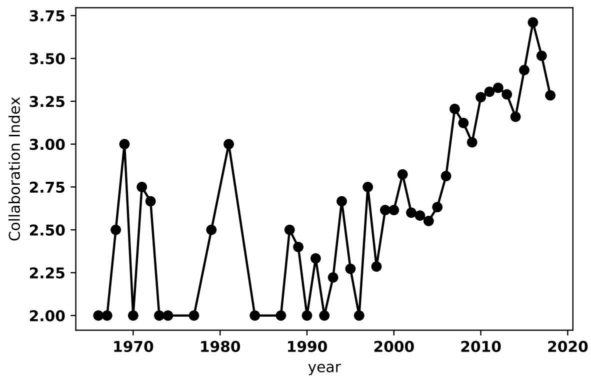


Figure 3.10: Collaboration index over time.

The collaborativeness of the authors is explored in more detail in Section 3.5 using the co-authorship network. The collaborative behaviour and relative influence of authors will also be explored in co-authorship networks which correspond to their publications research topics. These topics are presented in the next section.

3.4 Research topics in the Prisoner's Dilemma research

In order to identify the topics which are being discussed in the field of the PD, the LDA algorithm implemented in [?] is applied to the abstracts of the data set. As mentioned before, the number of topics, which will be denoted as n, needs to be specified before running the algorithm. The appropriate number of topics is chosen based on the coherence value [?]. Figure 3.11 gives the coherence values of 18 models where $n \in \{2, 3, \ldots, 19\}$, and it can be seen than the most appropriate number of topics is 6 with a coherence value of 0.418.

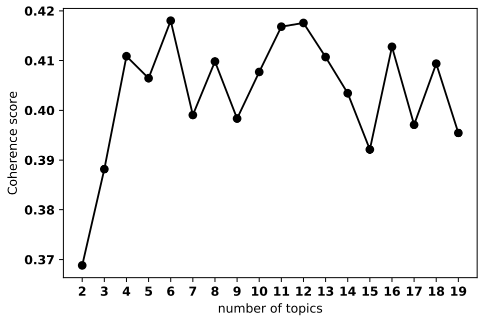


Figure 3.11: Coherence for LDA models over the number of topics.

The keywords associated with each topic for $n = 6$ are given by Table 3.2. Though $n = 6$ has the highest coherence score, from Table 3.2 it can be observed that there are overlapping keywords between the topics. Further manual investigation has revealed that the seperation of topics are the most clear when an n of 5 is considered. The LDA model for $n = 5$ has a coherence value 0.406 which is close to 0.418 (the score for $n = 6$. Thus, $n = 5$ is chosen to carry out the analysis of this work.

Topic	Topic Keywords
A	model, theory, system, base, paper, problem, propose, present, approach, provide, analysis, framework, method, develop, solution
B	behavior, social, human, decision, study, experiment, make, suggest, result, behaviour, effect, partner, participant, subject, experimental
C	individual, group, good, social, punishment, level, cost, mechanism, dilemma, cooperative, show, base, public, high, society
D	game, strategy, player, agent, play, dilemma, state, prisoner, payoff, equilibrium, result, iterate, set, probability, show
E	population, evolutionary, dynamic, model, selection, result, evolution, evolve, show, process, size, interaction, cooperator, change, system
F	cooperation, network, interaction, structure, study, evolution, find, behavior, cooperative, simulation, rule, spatial, cooperator, promote, result

Table 3.2: Keywords for each topic when $n = 6$. The highlighted keywords are overlapping keywords between topics.

For $n = 5$ the articles are clustered and assigned to their dominant topic, based on the highest percentage contribution. The keywords associated with a topic, the most representative article of the topic (based on the percentage contribution) and its academic reference are given by Table 3.3. The topics are labelled as A, B, C, D and E, and more specifically:

- Based on the keywords associated with Topic A, and the most representative article, Topic A appears to be about **human subject research**. Several publications assigned to the topic study the PD by setting experiments and having human participants simulate the game instead of computer simulations. These articles include [?] which showed that prosocial behaviour increased with the age of the participants, [74] which studied the difference in cooperation between high-functioning autistic and typically developing children, [?] explored the gender effect in highschool students and [?] explored the effect of facial expressions of individuals.

- Though it is not immediate from the keywords associated with Topic B, investigating the papers assigned to the topic indicate that it is focused on **biological studies**. Papers assigned to the topic include papers which apply the PD to genetics [?, ?], to the study of tumours [?, ?] and viruses [?]. Other works include how phenotype affinity can affect the emergence of cooperation [?] and modelling bacterial communities as a spatial structured social dilemma.
- Based on the keywords and the most representative article Topic C appears to include publications on PD **strategies**. Publications in the topic include the introduction of new strategies [?], the search of optimality in strategies [?] and the training of strategies [?] with different representation methods. Moreover, publications that study the evolutionary stability of strategies [8] and introduced methods of differentiating between them [14] are also assigned to C.
- The keywords associated with Topic D clearly show that the topic is focused on **evolutionary dynamics on networks**. Publications include [?] which explored the robustness of cooperation on networks, [?] which studied the effect of a strategy's neighbourhood on the emergence of cooperation and [?] which explored the fixation probabilities of any two strategies is spatial structures.
- The publication assigned to Topic E are on **modelling problems as a PD game**. Though Topic B is also concerned with problems being formulated as a PD, it includes only biological problems. In comparison, the problems in Topic E include decision making in operational research [?], information sharing among members in a virtual team [?], the measurement of influence in articles based on citations [?] and the price spikes in electric power markets [?], and not on biological studies.

Dominant Topic	Topic Keywords	Most Representative Article Title	Reference	# Documents	% Documents
A	social, behavior, human, study, experiment, cooperative, cooperation, suggest, find, behaviour	Facing Aggression: Cues Differ for Female versus Male Faces	[?]	496.0	0.2008
B	individual, group, good, show, high, increase, punishment, cost, result, benefit	Genomic and Gene-Expression Comparisons among Phage-Resistant Type-IV Pilus Mutants of Pseudomonas syringae pathovar phaseolicola	[?]	309.0	0.1251
C	game, strategy, player, agent, dilemma, play, payoff, state, prisoner, equilibrium	Fingerprinting: Visualization and Automatic Analysis of Prisoner's Dilemma Strategies	[?]	561.0	0.2271
D	cooperation, network, population, evolutionary, evolution, interaction, dynamic, structure, cooperator, study	Influence of initial distributions on robust cooperation in evolutionary Prisoner's Dilemma	[?]	556.0	0.2251
E	model, theory, base, system, problem, paper, propose, information, provide, approach	Gaming and price spikes in electric power markets and possible remedies	[?]	548.0	0.2219

Table 3.3: Keywords for each topic and the document with the most representative article for each topic.

Note that the whilst for the choice of 5 topics the actual clustering is not subjective (the algorithm is determining the output) the interpretation above is.

Thus, the five topics in the PD publications identified by the data set of using an LDA are:

1. human subject research,
2. biological studies,
3. strategies,
4. evolutionary dynamics on networks,
5. modelling problems as a PD.

These topics nicely summarise the PD research. They highlight the interdisciplinarity of the field; how it brings together applied modelling of real world situations (Topic B and E) and more theoretical notions such as evolutionary dynamics and optimality of strategies.

Figure 3.12 gives the number of articles per topic over time. The topics appear to have had a similar trend over the years, with topics B and D having a later start. Following the introduction of a topic the publications in that topic have been increasing. There is no decreasing trend in any of the topics. All the topics have been publishing for years and they still attract the interest of academics. Thus, there does not seem to be any given topic more or less in fashion.

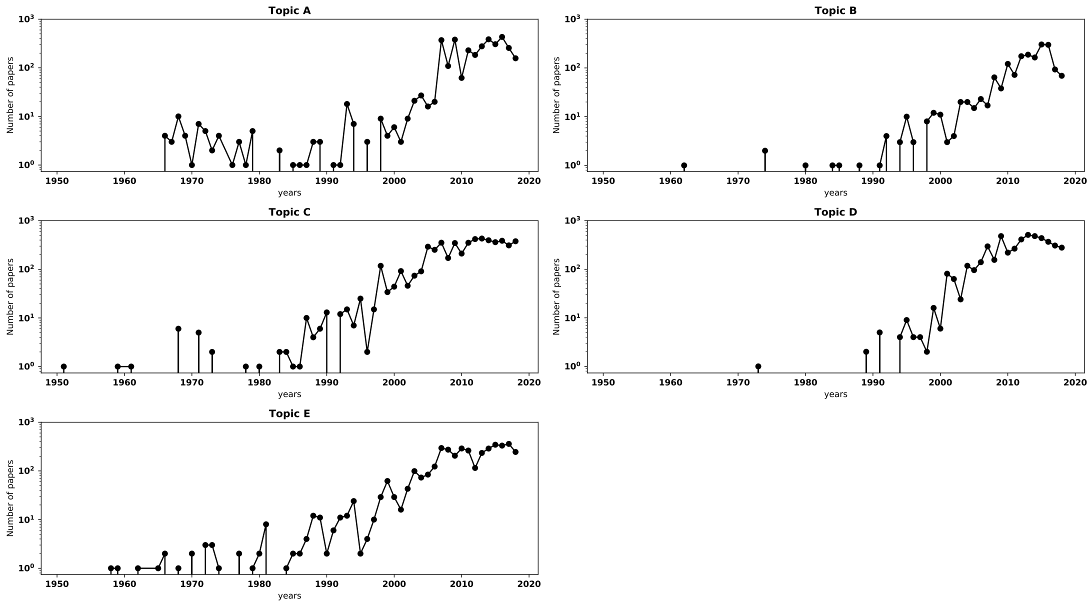


Figure 3.12: Number of articles per topic over the years (on a logged scale).

To gain a better understanding regarding the change in the topics over the years, LDA is applied to the cumulative data set over 8 time periods. These periods are 1951-1965, 1951-1973, 1951-1980, 1951-1988, 1951-1995, 1951-2003, 1951-2010, 1951-2018. The number of topics for each cumulative subset is chosen based on the coherence value and no objective approach is used. As a result, the period 1951-2018 has been assigned $n = 6$ which had the highest coherence value instead of 5. The chosen models for each period including the number of topics, their keywords and number of articles assigned to them are given by Table 3.4.

But how well do the five topics which were presented earlier fit the publications over time? This is answered by comparing the performance of three LDA models over the cumulative periods' publications. The three models are LDA models for the entire data set for n equal to 5, 6 and

Period	Topic	Topic Keywords	Num of Documents	Percentage of Documents
1951-1965	1	problem, technology, divert, euler, subsystem, requirement, trace, technique, system, untried	3	0.375
1951-1965	2	interpret, requirement, programme, evolution, article, increase, policy, system, trace, technology	2	0.25
1951-1965	3	equipment, agency, conjecture, development, untried, programme, trend, technology, weapon, technique	1	0.125
1951-1965	4	variation, celebrated, trend, untried, change, involve, month, technique, subsystem, research	1	0.125
1951-1965	5	give, good, modern, trace, technique, ambiguity, problem, trend, technology, system	1	0.125
1951-1973	1	study, shock, cooperative, money, part, vary, investigate, good, receive, equipment	12	0.3243
1951-1973	2	cooperation, level, significantly, sequence, reward, provoke, descriptive, principal, display, argue	4	0.1081
1951-1973	3	player, make, effect, triad, experimental, motivation, dominate, hypothesis, instruction, trend	3	0.0811
1951-1973	4	ss, sex, male, female, dyad, design, suggest, college, factor, tend	3	0.0811
1951-1973	5	result, research, format, change, operational, analysis, relate, understanding, decision, money	2	0.0541
1951-1973	6	condition, give, high, treatment, conflict, cc, real, original, replication, promote	2	0.0541
1951-1973	7	group, competitive, show, interpret, scale, compete, escalation, free, variable, individualistic	2	0.0541
1951-1973	8	outcome, strategy, choice, type, pdg, difference, dummy, conclude, compare, consistent	2	0.0541
1951-1973	9	game, difference, pair, approach, behavior, person, weapon, occur, advantaged, differential	2	0.0541
1951-1973	10	response, present, dilemma, influence, cooperate, bias, point, amount, participate, factor	2	0.0541
1951-1973	11	trial, problem, previous, involve, prisoner, experiment, follow, tit, increase, initial	1	0.027
1951-1973	12	matrix, behavior, rational, black, model, research, broad, distance, complex, trace	1	0.027
1951-1973	13	play, finding, individual, noncooperative, white, nature, race, ratio, represent, prisoner	1	0.027
1951-1980	1	play, trial, group, follow, white, interpret, scale, black, trend, small	14	0.25
1951-1980	2	outcome, level, effect, type, dyad, vary, pdg, participate, understanding, arise	9	0.1607
1951-1980	3	game, strategy, cooperation, significant, difference, sentence, text, occur, differential, hypothesis	4	0.0714
1951-1980	4	male, female, find, result, sex, subject, experimental, situation, treatment, computer	4	0.0714
1951-1980	5	research, problem, influence, matrix, format, model, analysis, year, crime, equipment	4	0.0714
1951-1980	6	condition, dilemma, bias, free, attempt, book, year, dummy, prison, design	4	0.0714
1951-1980	7	variable, result, factor, individual, ability, triad, half, migration, change, investigate	3	0.0536
1951-1980	8	show, present, suggest, rational, compete, approach, characteristic, examine, person, conduct	3	0.0536
1951-1980	9	behavior, high, finding, relate, obtain, assistance, ratio, good, weapon, competition	3	0.0536
1951-1980	10	ss, shock, money, competitive, part, difference, pair, amount, man, information	3	0.0536
1951-1980	11	player, conflict, theory, decision, determine, produce, maker, cooperate, specialist, programming	2	0.0357
1951-1980	12	study, prisoner, make, response, experiment, noncooperative, standard, separate, conclude, initial	2	0.0357
1951-1980	13	give, cooperative, choice, cognitive, real, operational, set, subject, ascribe, concern	1	0.0179
1951-1988	1	trial, difference, find, choice, significant, competitive, effect, triad, interact, occur	24	0.2553
1951-1988	2	ss, shock, money, pair, response, part, high, tit, receive, amount	13	0.1383
1951-1988	3	suggest, paper, case, debate, view, achieve, framework, natural, assumption, finitely	10	0.1064
1951-1988	4	prisoner, dilemma, behavior, model, present, involve, person, increase, trust, experiment	8	0.0851
1951-1988	5	game, player, show, approach, repeat, previous, move, tat, related, include	8	0.0851
1951-1988	6	cooperation, level, mutual, equilibrium, standard, provide, information, human, real, question	6	0.0638
1951-1988	7	play, result, male, subject, female, cooperative, sex, experimental, treatment, computer	5	0.0532
1951-1988	8	research, study, variable, ability, factor, conflict, matrix, year, student, interpret	4	0.0426
1951-1988	9	problem, group, small, scale, social, issue, large, base, bias, party	4	0.0426
1951-1988	10	game, strategy, outcome, type, cooperate, ethical, pdg, explain, dependent, separate	4	0.0426
1951-1988	11	give, condition, individual, major, dyad, behaviour, produce, conflict, assistance, collectively	3	0.0319
1951-1988	12	situation, iterate, statement, rational, card, side, paradox, true, consequence, front	2	0.0213
1951-1988	13	inflation, hypothesis, rate, run, change, demand, nominal, cost, output, growth	2	0.0213
1951-1988	14	theory, make, analysis, decision, system, examine, work, soft, lead, hard	1	0.0106
1951-1995	1	strategy, population, evolution, iterate, tit, opponent, evolve, dynamic, set, tat	31	0.1732
1951-1995	2	game, repeat, assumption, rule, person, equilibrium, general, finitely, indefinitely, analyze	24	0.1341
1951-1995	3	inflation, long, rate, hypothesis, run, policy, cost, nominal, demand, programming	20	0.1117
1951-1995	4	condition, outcome, trial, find, difference, cooperation, experiment, level, significant, response	15	0.0838
1951-1995	5	rational, result, receive, statement, money, paradox, shock, iterate, consequence, common	14	0.0782
1951-1995	6	cooperation, show, competitive, high, probability, conflict, simulation, altruism, yield, natural	14	0.0782
1951-1995	7	prisoner, dilemma, give, point, defect, form, cooperator, increase, relate, ethical	10	0.0559
1951-1995	8	player, give, decision, provide, cooperative, game, previous, pair, determine, interact	9	0.0503
1951-1995	9	play, cooperate, result, male, subject, female, time, relationship, suggest, student	8	0.0447
1951-1995	10	problem, group, theory, good, approach, society, large, scale, issue, level	8	0.0447
1951-1995	11	study, situation, behaviour, computer, argue, change, implication, characteristic, real, associate	8	0.0447
1951-1995	12	model, paper, behavior, examine, present, mutual, expectation, develop, type, variable	7	0.0391
1951-1995	13	make, research, system, analysis, choice, work, base, relation, world, wide	6	0.0335
1951-1995	14	individual, social, behavior, standard, choose, evolutionary, partner, payoff, defection, small	5	0.0279
1951-2003	1	game, player, dilemma, prisoner, theory, give, paper, make, group, problem	151	0.4266
1951-2003	2	cooperation, result, play, show, cooperate, condition, cooperative, high, level, time	106	0.2994
1951-2003	3	strategy, model, agent, study, behavior, individual, population, evolutionary, state, player	97	0.274
1951-2010	1	model, theory, paper, base, make, present, problem, provide, human, decision	325	0.3454
1951-2010	2	game, strategy, player, agent, play, dilemma, system, behavior, show, state	322	0.3422
1951-2010	3	cooperation, network, study, population, individual, evolutionary, social, evolution, interaction, structure	294	0.3124
1951-2018	1	model, theory, system, base, paper, problem, propose, present, approach, provide	556	0.2251
1951-2018	2	behavior, social, human, decision, study, experiment, make, suggest, result, behaviour	482	0.1951
1951-2018	3	individual, group, good, social, punishment, level, cost, mechanism, dilemma, cooperative	428	0.1733
1951-2018	4	game, strategy, player, agent, play, dilemma, state, prisoner, payoff, equilibrium	380	0.1538
1951-2018	5	population, evolutionary, dynamic, model, selection, result, evolution, evolve, show, process	351	0.1421
1951-2018	6	cooperation, network, interaction, structure, study, evolution, find, behavior, cooperative, simulation	273	0.1105

Table 3.4: Topic modelling result for the cumulative data set over the periods

the optimal number of topics over time. For each model the c^* is estimated for each document in the cumulative data sets. The performance of the models are then compared based on:

$$\bar{c^*} \times n \tag{3.1}$$

where $\bar{c^*}$ is the median highest percentage contribution and n is the number of topics of a given period. A model with more topics will have more difficulty to assign papers. Thus, equation (refeq:ratio) is a measure of confidence in assigning a given paper to its topic weighted by the number of topics. The performances are given by Figure 3.13.

The five topics of the PD presented in this manuscript appear to always be less good at fitting the publications compared to the six topics of LDA $n = 6$. Moreover, there are less good than the topics of the optimal number of topics from 1951 to 1995. The difference in the performance values, equation (3.1), however are small. The relevances of the five topics has been increasing over time, and though, the topics did not always fit the majority of published work over time, there were still papers being published on those topics.

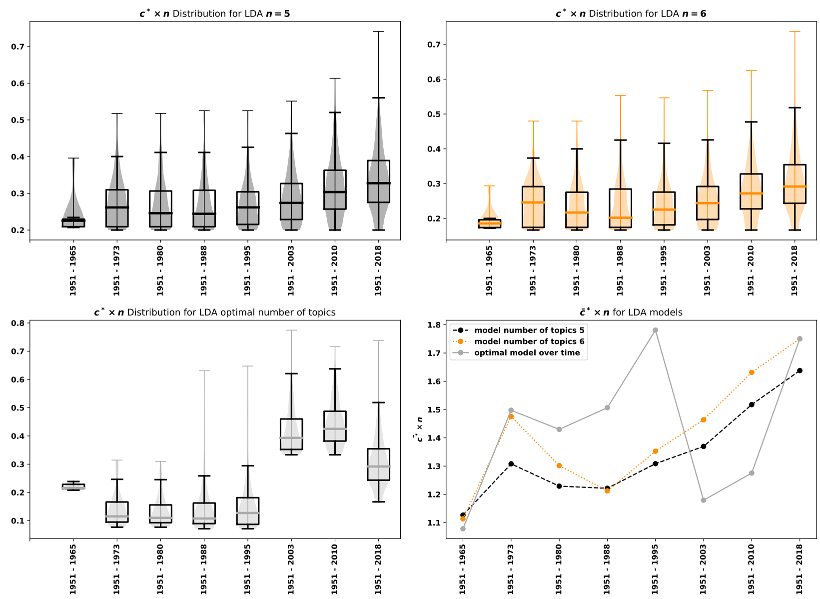


Figure 3.13: Maximum percentage contributions (c^*) over the time periods, for the LDA models for the entire data set for n equal to 5, 6 and the optimal number of topics over time.

In the following section the collaborative behaviour of authors in the field, and within the field's topics as were presented in this section, are explored using a network theoretic approach.

3.5 Analysis of co-authorship network

The collaborative behaviour of authors in the field of the PD is assessed using the co-authorship network, which as introduced in section 3.2 is denoted as G. There are a total of 947 connected components in G and the largest component has a size of 796 nodes. The largest connected component is going to be refereed to as the main cluster of the network and is denoted as $\bar{G}$. A graphical representation of both networks is shown in Figures 3.14-3.15 and a metrics summary is given by Table 3.5.

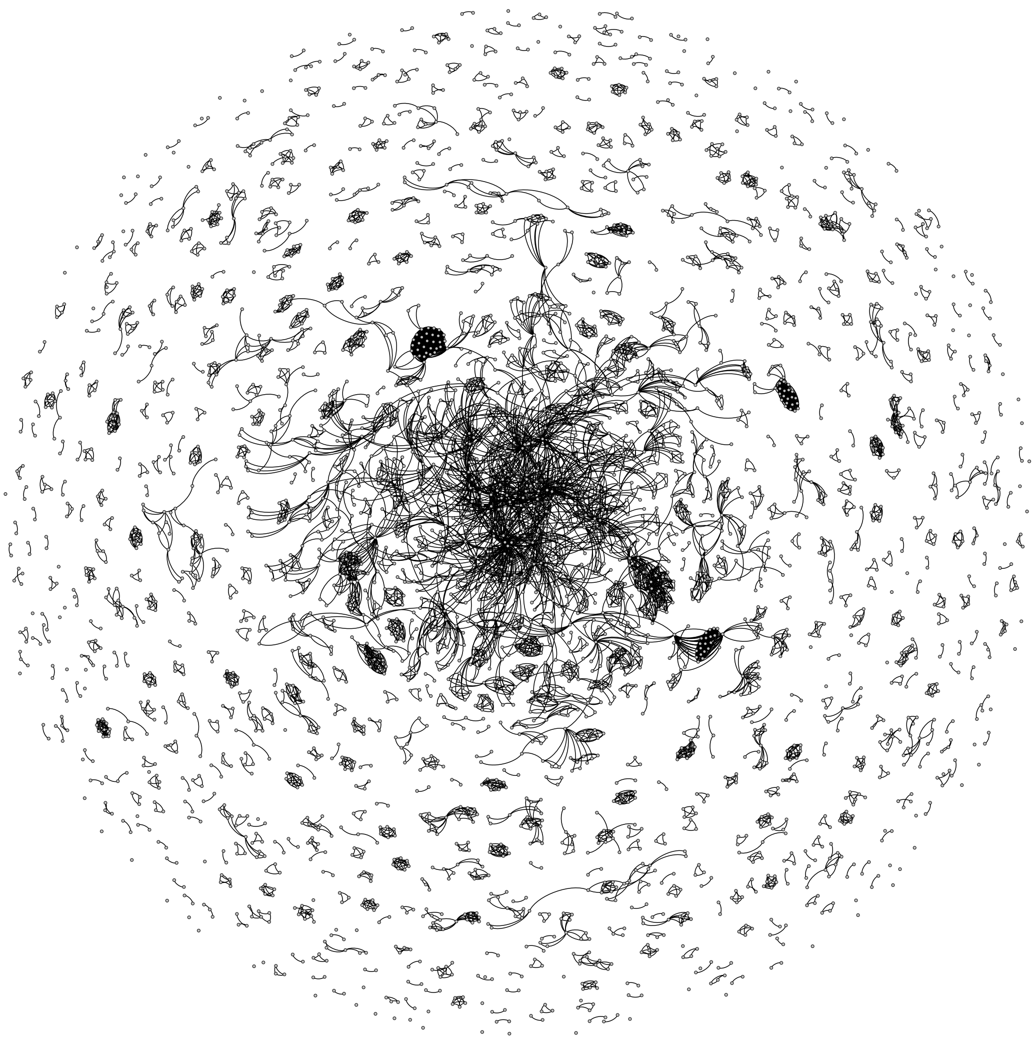


Figure 3.14: G the co-authorship network for the IPD.

Figure 3.15: $\bar{G}$ the largest connected component of G.

Based on Table 3.5 an author in G has on average 4 collaborators and a 70% probability of collaborating with a collaborator's co-author. An author of $\bar{G}$ on average is 7% more likely to write with a collaborator's co-author and on average has 2 more collaborators. Moreover, there are only 3.2 % of authors in the PD that has no connection to any other author.

	# Nodes	# Edges	% Isolated nodes	# Connected components	Size of largest component	Av. degree	# Communities	Modularity	Clustering coeff
G	4011	7642	3.2	947	796	3.811	967	0.96491	0.701
$\bar{G}$	796	2214	0.0	1	796	5.563	25	0.84406	0.773

Table 3.5: Network metrics for G and $\bar{G}$ respectively.

But how do these compare to other fields? Two more data sets for the topics "Price of Anarchy" and "Auction Games" have been collected in order to compare the collaborative behaviour of the PD to other game theoretic fields. A total of 3444 publications have been collected for Auction games and 748 for Price of Anarchy. Price of Anarchy is relatively a new field, with the first publication on the topic being [?] in 1999. This explains the small number of articles that have been retrieved. Both data sets have been archived and are available in [?, ?]. The networks for both data sets have been generated in the same way as G. A summary of the networks' metrics are given by Table 3.6.

	# Nodes	# Edges	# Isolated nodes	% Isolated nodes	# Connected components	Size of largest component	Av. degree	# Communities	Modularity	Clustering coeff
Auction Games	5165	7861	256	5.0	1272	1348	3.044	1294	0.957	0.622
Price of Anarchy	1155	1953	4	0.3	245	222	3.382	253	0.965	0.712

Table 3.6: Network metrics for auction games and price of anarchy networks respectively.

The average degrees for the Price of Anarchy and for Auction games are lower than the PD's. In Auction games an author is more likely to have no collaborators, and in the Price of Anarchy there are almost no authors that are not connected to someone. This could be an effect of the field being introduced in more modern days. Overall, an author in the PD has on average more collaborators and there are less isolated authors compared to another well established game theoretic field. These results seem to indicate that the PD is a *relatively* collaborative field.

However, both G and $\bar{G}$ have a high modularity (larger than 0.84) and a large number of communities (967 and 25 respectively). A high modularity implies that authors create their own publishing communities but not many publications from authors from different communities occur. Thus, author tends to collaborate with authors in their communities but not many efforts are made to create new connections to other communities and spread the knowledge of the field across academic teams. The fields of both Price of Anarchy and Auction games also have high modularity, and that could indicate that is in fact how academic publications are. Thus, the PD is indeed a collaborative field but perhaps it is not more collaborative than other fields, as there is no effort from the authors to write with people outside their community.

The evolution of the networks was also explored over time by constructing the network cumulatively over 51 periods. Except from the first period 1951-1966 the rest of the periods have a yearly interval (data for the years 1975 and 1982 were not retrieved by the collection data process). The metrics of each sub network are given by Tables 3.8 and 3.7.

The results, similarly to the results of [?], confirm that the networks grow over time and that

Period	# Nodes	# Edges	% Isolated nodes	Size of largest component	Av. degree	# Communities	Modularity	Clustering coeff
1951 - 1966	2	1	0.0	2	1.000	1	0.000	0.000
1951 - 1967	2	1	0.0	2	1.000	1	0.000	0.000
1951 - 1968	5	8	0.0	5	3.200	1	0.000	0.867
1951 - 2002	7	21	0.0	7	6.000	1	0.000	1.000
1951 - 2003	7	21	0.0	7	6.000	1	0.000	1.000
1951 - 2004	10	13	0.0	10	2.600	2	0.376	0.553
1951 - 2005	19	28	0.0	19	2.947	3	0.544	0.730
1951 - 2006	22	35	0.0	22	3.182	4	0.527	0.720
1951 - 2007	25	39	0.0	25	3.120	5	0.558	0.686
1951 - 2008	33	62	0.0	33	3.758	4	0.623	0.736
1951 - 2009	71	148	0.0	71	4.169	6	0.697	0.698
1951 - 2010	133	387	0.0	133	5.820	7	0.726	0.749
1951 - 2011	157	465	0.0	157	5.924	8	0.727	0.725
1951 - 2012	209	611	0.0	209	5.847	11	0.733	0.737
1951 - 2013	322	892	0.0	322	5.540	12	0.780	0.743
1951 - 2014	399	1109	0.0	399	5.559	15	0.794	0.742
1951 - 2015	504	1368	0.0	504	5.429	24	0.811	0.751
1951 - 2016	613	1677	0.0	613	5.471	21	0.819	0.761
1951 - 2017	706	1935	0.0	706	5.482	29	0.830	0.772
1951 - 2018	796	2214	0.0	796	5.563	25	0.845	0.773

Table 3.7: Collaborativeness metrics for cumulative graphs' main clusters, $\tilde{G} \subseteq \bar{G}$

the networks always had a high modularity. Since the first publications authors tend to write with people from their communities, and that is not an effect of a specific time period.

The networks corresponding to the topics of Section 3.3 have also been generated similarly to G. Note that authors with publications in more than one topic exist, and these authors are included in all the corresponding networks. A metrics' summary for all five topic networks is given by Table 3.9.

Topic B is the network with the highest average degree followed by Topic A. The topic with the smallest average degree, 2.5, is Topic C. In topics A and B the number of isolated nodes is very small $lessthan(0.2)$ compared to Topic E where the percentage of isolated nodes is approximately 6%. Moreover, in topics C and E an author is 10% more likely to collaborate with a collaborator's co-author. Thus, topics "human subject research" and "biological studies" tend to be more collaborative than the topic of "strategies", and an authors in these are less likely to have at least one collaborator compared to the topic of "modelling problems as a PD".

"Evolutionary dynamics on networks" also appear to be a collaborative topic. In fact the network of the topic is a sub graph of $\bar{G}$, the main cluster of G and it will be demonstrated in the following section that authors in this network are more like to gain from the influence of the network compared to any other topic network.

The two centrality measures reported in this thesis are closeness and betweenness centrality. Closeness centrality is a measure of how easy it is for an author to contact others, and consequently affect them; influence them. Thus closeness centrality is a measure of influence. Betweenness centrality is a measure of how many paths pass through a specific node, thus the amount of information this person has access to. Betweenness centrality is interpreted as a measure of how much an author gains from the field. The values of the centralities can range between 0 and 1. Influence and the amount of information an author has access to are proxies

Period	# Nodes	# Edges	% Isolated nodes	# Connected components	Size of largest component	Av. degree	# Communities	Modularity	Clustering coeff
1951 - 1966	6	3	0.0	3	2	1.000	3	0.667	0.000
1951 - 1967	8	4	0.0	4	2	1.000	4	0.750	0.000
1951 - 1968	19	15	0.0	8	5	1.579	8	0.684	0.228
1951 - 1969	20	17	0.0	8	6	1.700	8	0.630	0.250
1951 - 1970	22	18	0.0	9	6	1.636	9	0.667	0.227
1951 - 1971	33	28	0.0	13	6	1.697	13	0.827	0.424
1951 - 1972	39	34	0.0	15	6	1.744	15	0.867	0.513
1951 - 1973	42	35	2.4	17	6	1.667	17	0.873	0.476
1951 - 1974	42	35	2.4	17	6	1.667	17	0.873	0.476
1951 - 1976	42	35	2.4	17	6	1.667	17	0.873	0.476
1951 - 1977	44	36	2.3	18	6	1.636	18	0.880	0.455
1951 - 1978	44	36	2.3	18	6	1.636	18	0.880	0.455
1951 - 1979	47	40	2.1	18	6	1.702	18	0.884	0.454
1951 - 1980	47	40	2.1	18	6	1.702	18	0.884	0.454
1951 - 1981	50	46	2.0	18	6	1.840	18	0.889	0.497
1951 - 1983	51	46	3.9	19	6	1.804	19	0.889	0.487
1951 - 1984	53	47	3.8	20	6	1.774	20	0.894	0.469
1951 - 1985	53	47	3.8	20	6	1.774	20	0.894	0.469
1951 - 1986	53	47	3.8	20	6	1.774	20	0.894	0.469
1951 - 1987	56	48	5.4	22	6	1.714	22	0.898	0.443
1951 - 1988	62	52	6.5	25	6	1.677	25	0.909	0.449
1951 - 1989	75	62	6.7	31	6	1.653	31	0.926	0.424
1951 - 1990	79	64	6.3	33	6	1.620	33	0.930	0.403
1951 - 1991	87	69	6.9	37	6	1.586	37	0.937	0.400
1951 - 1992	95	72	10.5	42	6	1.516	42	0.941	0.367
1951 - 1993	106	81	11.3	47	6	1.528	47	0.947	0.366
1951 - 1994	124	95	12.9	56	6	1.532	56	0.955	0.394
1951 - 1995	135	102	12.6	61	6	1.511	61	0.960	0.384
1951 - 1996	142	105	12.7	65	6	1.479	65	0.962	0.365
1951 - 1997	155	115	12.9	71	6	1.484	71	0.966	0.392
1951 - 1998	191	140	11.0	87	6	1.466	87	0.973	0.367
1951 - 1999	221	169	11.3	99	6	1.529	99	0.977	0.397
1951 - 2000	250	195	10.8	110	6	1.560	110	0.979	0.418
1951 - 2001	287	235	10.5	125	7	1.638	125	0.977	0.419
1951 - 2002	335	278	10.7	146	7	1.660	146	0.979	0.428
1951 - 2003	381	310	10.5	168	7	1.627	168	0.982	0.413
1951 - 2004	437	370	9.2	185	10	1.693	185	0.983	0.424
1951 - 2005	532	476	7.7	214	19	1.789	214	0.985	0.458
1951 - 2006	640	603	6.7	246	22	1.884	246	0.987	0.486
1951 - 2007	793	877	5.8	283	25	2.212	283	0.985	0.532
1951 - 2008	948	1170	5.3	318	33	2.468	319	0.985	0.558
1951 - 2009	1108	1442	4.9	356	71	2.603	358	0.982	0.573
1951 - 2010	1300	1936	5.1	402	133	2.978	405	0.965	0.592
1951 - 2011	1560	2375	5.1	472	157	3.045	475	0.970	0.613
1951 - 2012	1837	2865	4.4	534	209	3.119	537	0.969	0.634
1951 - 2013	2149	3420	4.3	603	322	3.183	609	0.965	0.644
1951 - 2014	2481	3971	4.2	683	399	3.201	694	0.962	0.658
1951 - 2015	2938	4877	3.7	765	504	3.320	779	0.965	0.675
1951 - 2016	3469	6532	3.3	850	613	3.766	863	0.964	0.696
1951 - 2017	3735	7072	3.2	895	706	3.787	912	0.964	0.700
1951 - 2018	4011	7642	3.2	947	796	3.811	967	0.966	0.701

Table 3.8: Collaborativeness metrics for cumulative graphs, $\tilde{G} \subseteq G$

	# Nodes	# Edges	# Isolated nodes	% Isolated nodes	# Connected components	Size of largest component	Av. degree	# Communities	Modularity	Clustering coeff
Topic A	1124	2137	15	1.3	264	56	3.802	265	0.983	0.759
Topic B	695	1382	13	1.9	157	80	3.977	158	0.950	0.773
Topic C	900	1141	41	4.6	281	29	2.536	281	0.981	0.636
Topic D	880	1509	17	1.9	174	312	3.430	183	0.918	0.701
Topic E	1045	1964	59	5.6	354	31	3.759	354	0.926	0.664

Table 3.9: Network metrics for topic networks.

to understand if/which authors benefit more from their position.

For G and $\bar{G}$ the most central authors based on closeness and betweenness centralities are given by Table 3.10. The most central authors in G and $\bar{G}$ are the same. This implies that the results on centrality heavily rely on the main cluster (as expected). Matjaz Perc is an author with 83 publications in the data set and the most central authors based on both centrality measures. The most central authors are fairly similar between the two measures. The author that appear to be central based on one measure and not the other are Martin Nowak, Franz Weissing, Jianye Hao, Angel Sanchez and Valerio Capraro which have access to information due to their positioning but do not influence the network as much, and the opposite is true for Attila Szolnoki, Luo-Luo Jiang Sandro Meloni, Cheng-Yi Xia and Xiaojie Chen.

	G				$\bar{G}$			
	Name	Betweenness	Name	Closeness	Name	Betweenness	Name	Closeness
1	Matjaz Perc	0.015	Matjaz Perc	0.066	Matjaz Perc	0.373	Matjaz Perc	0.330
2	Zhen Wang	0.011	Long Wang	0.060	Zhen Wang	0.279	Long Wang	0.301
3	Long Wang	0.007	Yamir Moreno	0.059	Long Wang	0.170	Yamir Moreno	0.299
4	Martin Nowak	0.006	Attila Szolnoki	0.059	Martin Nowak	0.159	Attila Szolnoki	0.297
5	Angel Sanchez	0.004	Zhen Wang	0.059	Angel Sanchez	0.114	Zhen Wang	0.296
6	Yamir Moreno	0.004	Arne Traulsen	0.056	Yamir Moreno	0.110	Arne Traulsen	0.281
7	Arne Traulsen	0.004	Luo-Luo Jiang	0.055	Arne Traulsen	0.107	Luo-Luo Jiang	0.280
8	Franz Weissing	0.004	Sandro Meloni	0.055	Franz Weissing	0.101	Sandro Meloni	0.278
9	Jianye Hao	0.004	Cheng-Yi Xia	0.055	Jianye Hao	0.094	Cheng-Yi Xia	0.276
10	Valerio Capraro	0.004	Xiaojie Chen	0.055	Valerio Capraro	0.093	Xiaojie Chen	0.276

Table 3.10: 10 most central authors based on betweenness and closeness centralities for G and $\bar{G}$.

It is obvious that in G the centrality values are low which suggests that in the PD authors do not benefit from their positions. This could be an effect of information not flowing from one community to another as authors tend to write with people from their communities. Nevertheless, there are authors that do benefit from their position, but these are only the authors connected to the main cluster.

The centrality measures for the topic networks have also been estimated and are given in Tables 3.11-3.12. If information was flowing between the communities of the research topics then there would be an increase to the values of centralities for the sub networks. However, the only topic where authors gain from their positions are the authors of Topic D (topic on evolutionary dynamics on network). From the list of names it is obvious that these authors are part of $\bar{G}$, and that the network of Topic D is a sub network of $\bar{G}$. This confirms the results. The people benefiting from their position in the co-authorship networks corresponding to research topics of the PD are only the people from the main cluster of G.

The fact that most authors of the main cluster are primarily publishing in evolutionary dynamics on networks indicates that publishing in this specific topic differs from the other topics covered in this manuscript. There appears to be more collaboration and influence in the publications on evolutionary dynamics and authors are more likely to gain from their position, though it is not clear as to why.

The distributions of both centrality measures for all the networks of this work are given in the Appendix A.2.

	Topic A		Topic B		Topic C		Topic D		Topic E	
	Name	Betweeness	Name	Betweeness	Name	Betweeness	Name	Betweeness	Name	Betweeness
1	David Rand	0.002	Long Wang	0.006	Daniel Ashlock	0.001	Matjaz Perc	0.064	Zengru Di	0.0
2	Valerio Capraro	0.001	Luo-Luo Jiang	0.005	Matjaz Perc	0.000	Luo-Luo Jiang	0.037	Jian Yang	0.0
3	Angel Sanchez	0.001	Martin Nowak	0.004	Karl Tuyls	0.000	Yamir Moreno	0.031	Yevgeniy Vorobeychik	0.0
4	Feng Fu	0.001	Matjaz Perc	0.003	Philip Hingston	0.000	Christoph Hauert	0.027	Otavio Teixeira	0.0
5	Martin Nowak	0.000	Attila Szolnoki	0.003	Eun-Youn Kim	0.000	Long Wang	0.024	Roberto Oliveira	0.0
6	Nicholas Christakis	0.000	Christian Hilbe	0.002	Wendy Ashlock	0.000	Zhen Wang	0.024	M. Nowak	0.0
7	Pablo Branas-Garza	0.000	Yamir Moreno	0.002	Attila Szolnoki	0.000	Han-Xin Yang	0.023	M. Harper	0.0
8	Toshio Yamagishi	0.000	Xiaojie Chen	0.002	Seung Baek	0.000	Martin Nowak	0.020	Xiao Han	0.0
9	James Fowler	0.000	Arne Traulsen	0.002	Martin Nowak	0.000	Angel Sanchez	0.017	Zhesi Shen	0.0
10	Long Wang	0.000	Zhen Wang	0.002	Thore Graepel	0.000	Zhihai Rong	0.016	Wen-Xu Wang	0.0

Table 3.11: 10 most central authors based on betweenness centrality for topics' networks.

	Topic A		Topic B		Topic C		Topic D		Topic E	
	Name	Closeness	Name	Closeness	Name	Closeness	Name	Closeness	Name	Closeness
1	David Rand	0.027	Long Wang	0.043	Karl Tuyls	0.022	Matjaz Perc	0.123	Stefanie Widder	0.029
2	Valerio Capraro	0.023	Matjaz Perc	0.041	Thore Graepel	0.019	Zhen Wang	0.109	Rosalind Allen	0.029
3	Jillian Jordan	0.022	Attila Szolnoki	0.040	Joel Leibo	0.018	Long Wang	0.107	Thomas Pfeiffer	0.029
4	Nicholas Christakis	0.021	Martin Nowak	0.040	Edward Hughes	0.017	Yamir Moreno	0.105	Thomas Curtis	0.029
5	James Fowler	0.020	Olivier Tenaillon	0.038	Matthew Phillips	0.017	Luo-Luo Jiang	0.104	Carsten Wiuf	0.029
6	Martin Nowak	0.020	Xiaojie Chen	0.038	Edgar Duenez-Guzman	0.017	Attila Szolnoki	0.103	William Sloan	0.029
7	Angel Sanchez	0.019	Bin Wu	0.038	Antonio Castaneda	0.017	Gyorgy Szabo	0.102	Otto Cordero	0.029
8	Gordon Kraft-Todd	0.019	Yanling Zhang	0.037	Iain Dunning	0.017	Xiaojie Chen	0.102	Sam Brown	0.029
9	Akihiro Nishi	0.019	Feng Fu	0.037	Tina Zhu	0.017	Guangming Xie	0.101	Babak Momeni	0.029
10	Anthony Evans	0.019	David Rand	0.037	Kevin Mckee	0.017	Lucas Wardil	0.101	Wenying Shou	0.029

Table 3.12: 10 most central authors based on closeness centrality for topics' networks.

3.6 Chapter summary

This Chapter explored the research topics from a collection of 2422 publications of the Iterated Prisoner's Dilemma, and moreover, the authors' collaborative behaviour and their influence in the research field. This was achieved by applying network theoretic approaches and a LDA algorithm to the collection of publications. Both the software [?] and the main data set associated with the Chapter [?] have been archived and are available to be used by other researchers. In fact Arcas has been used by [?] and [?].

Arcas, it's development and the data collection were covered in section 3.2, as well as an introduction to the co-authorship network and to LDA. section 3.3 covered an initial analysis of the data set which demonstrated that the PD is a field that continues to attract academic attention and publications. In Section 3.4 LDA was applied to the data set to identify topics on which researchers have been publishing. The LDA analysis showed that the data could be classified into 5 topics associated with human subject research, biological studies, strategies, evolutionary dynamics on networks and modelling problems as a PD. These topics summarize the field of the PD well, as they demonstrate its interdisciplinarity and applications to a variety of problems. A temporal analysis explored how relevant these topics have been over the course of time, and it revealed that even though there were not the necessarily always the most discussed topics they were still being explored by researchers.

The collaborative behaviour of the field was explored in section 3.5 by constructing the co-authorship network. It was concluded that the field is a collaborative field, where authors are likely to write with a collaborator's co-authors and on average an author has 4 co-authors, however it not necessarily more collaborative than other fields. The authors tend to collaborate with authors from one community, but not many authors are involved in multiple communities. This however might be an effect of academic research, and it might not be true just for the field of the PD. Exploring the influence of authors and their gain from being in the network of the field demonstrated that authors do not gain much, and the authors with influence are only the ones connected to the main cluster, to a "main" group of authors. This 'main" group of authors consists of authors publishing in evolutionary dynamics on networks. Thus, an author would be aiming to publish on this topic if they were interested in gaining from their position in the publications of the PD.

The study of the PD is the study of cooperation and investigating the cooperative behaviours of authors is what this Chapter has aimed to achieve. The following Chapters focus on best responses in changing environments of the PD, and more specifically Chapter 4 studies best responses from a collection of strategies in a large number of IPD tournaments.

Chapter 4

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

Chapter 5

Stability of defection, optimisation of strategies and the limits of memory in the Prisoner's Dilemma.

The research reported in this Chapter has lead in a manuscript, entitled:
"A theory of mind: Best responses to memory-one strategies. The limitations of extortion and restricted memory"
Available at: `arxiv.org/abs/1911.12112`
Associated data set: [45]
Axerod-Python library version: 4.4.0

The manuscript's abstract is the following:

Memory-one strategies are a set of Iterated Prisoner's Dilemma strategies that have been praised for their mathematical tractability and performance against single opponents. This manuscript investigates a theory of mind: *best response* memory-one strategies, as a multidimensional optimisation problem. We add to the literature that has shown that extortionate play is not always optimal by showing that optimal play is often not extortionate. We also provide evidence that memory-one strategies suffer from their limited memory in multi agent interactions and can be out performed by optimised strategies with longer memory.

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

5.1 Introduction

This Chapter contributes to the question: what is the optimal behaviour an IPD strategy should adapt as a response to different environments? In [99] the authors stated that "Only a player with a theory of mind about his opponent can do better, in which case Iterated Prisoner's Dilemma is an Ultimatum Game". The purpose of this Chapter is to investigate the first part of this sentence, more specifically, to investigate the best response strategy with a theory of mind in an environment with memory-one opponents, and to understand the effects of extortion and restricted memory in those environments. Extortionate behaviour is explored using a linear algebraic approach presented in [69].

The outcomes of this Chapter reinforce and extend known results which were presented in Chapter 2. Namely that memory-one strategies must be forgiving to be evolutionarily stable [113, 114] and that longer-memory strategies have a certain form of advantage over short memory strategies [58, 98]. The Chapter is structured as follows:

- section 5.2, describes a closed form algebraic expression for the utility of a memory-one strategy to a given group of opponents.
- section 5.3, produces a compact method of identifying the best response memory-one strategy against a given set of opponents.
- section 5.4, explains best response reactive strategies and demonstrates the usage of resultant theory in explicitly finding a reactive best response.
- section 5.5, describes a series of numerical experiments and a well designed framework that allows the comparison of an optimal memory one strategy and a more complex strategy which has a larger memory.
- section 5.6, presents a compact method of identifying environments for which cooperation cannot occur.

5.2 The Utility

One specific advantage of memory-one strategies is their mathematical tractability. They can be represented completely as an element of $\mathbb{R}^4_{[0,1]}$. As previously discussed in Chapter 2, if a strategy is concerned with only the outcome of a single turn then there are four possible 'states' the strategy could be in;

- both players cooperated, denoted as CC
- first players cooperated whilst the second player defected, denoted as CD
- first players defected whilst the second player cooperated, denoted as DC
- both players defected, denoted as DD

Therefore, a memory-one strategy can be denoted by the probability vector of cooperating after each of these states; $p = (p_1, p_2, p_3, p_4) \in \mathbb{R}^4_{[0,1]}$.

In [91] it was shown that it is not necessary to simulate the play of a strategy p against a memory-one opponent q. Rather this exact behaviour can be modelled as a stochastic process,

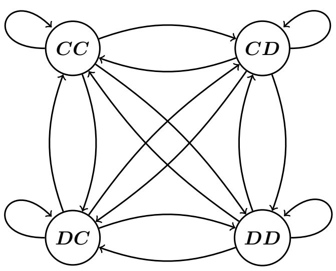


Figure 5.1: Markov Chain

and more specifically as a Markov chain (Figure 5.1) whose corresponding transition matrix M is given by (5.1). The long run steady state probability vector v, which is the solution to $vM = v$, can be combined with the payoff matrices of (**??**) to give the expected payoffs for each player. More specifically, the utility for a memory-one strategy p against an opponent q, denoted as $u_q(p)$, is given by (5.2).

$$M = \begin{bmatrix} p_1q_1 & p_1(-q_1+1) & q_1(-p_1+1) & (-p_1+1)(-q_1+1) \\ p_2q_3 & p_2(-q_3+1) & q_3(-p_2+1) & (-p_2+1)(-q_3+1) \\ p_3q_2 & p_3(-q_2+1) & q_2(-p_3+1) & (-p_3+1)(-q_2+1) \\ p_4q_4 & p_4(-q_4+1) & q_4(-p_4+1) & (-p_4+1)(-q_4+1) \end{bmatrix} \tag{5.1}$$

$$u_q(p) = v \cdot (R, S, T, P). \tag{5.2}$$

This work gives the form of $u_q(p)$, to the author's knowledge no previous work has done this, and Theorem 1 states that $u_q(p)$ is given by a ratio of two quadratic forms [66].

Theorem 1. *The expected utility of a memory-one strategy $p \in \mathbb{R}^4_{[0,1]}$ against a memory-one opponent $q \in \mathbb{R}^4_{[0,1]}$, denoted as $u_q(p)$, can be written as a ratio of two quadratic forms:*

$$u_q(p) = \frac{\frac{1}{2}pQp^T + cp + a}{\frac{1}{2}p\bar{Q}p^T + \bar{c}p + \bar{a}}, \tag{5.3}$$

where $Q, \bar{Q} \in \mathbb{R}^{4\times 4}$ are square matrices defined by the transition probabilities of the opponent q_1, q_2, q_3, q_4 as follows:

$$Q = \begin{bmatrix} 0 & -(q_1-q_3)(Pq_2-P-Tq_4) & (q_1-q_2)(Pq_3-Sq_4) & (q_1-q_4)(Sq_2-S-Tq_3) \\ -(q_1-q_3)(Pq_2-P-Tq_4) & 0 & (q_2-q_3)(Pq_1-P-Rq_4) & -(q_3-q_4)(Rq_2-R-Tq_1+T) \\ (q_1-q_2)(Pq_3-Sq_4) & (q_2-q_3)(Pq_1-P-Rq_4) & 0 & (q_2-q_4)(Rq_3-Sq_1+S) \\ (q_1-q_4)(Sq_2-S-Tq_3) & -(q_3-q_4)(Rq_2-R-Tq_1+T) & (q_2-q_4)(Rq_3-Sq_1+S) & 0 \end{bmatrix}, \tag{5.4}$$

$$\bar{Q} = \begin{bmatrix} 0 & -(q_1-q_3)(q_2-q_4-1) & (q_1-q_2)(q_3-q_4) & (q_1-q_4)(q_2-q_3-1) \\ -(q_1-q_3)(q_2-q_4-1) & 0 & (q_2-q_3)(q_1-q_4-1) & (q_1-q_2)(q_3-q_4) \\ (q_1-q_2)(q_3-q_4) & (q_2-q_3)(q_1-q_4-1) & 0 & -(q_2-q_4)(q_1-q_3-1) \\ (q_1-q_4)(q_2-q_3-1) & (q_1-q_2)(q_3-q_4) & -(q_2-q_4)(q_1-q_3-1) & 0 \end{bmatrix}. \tag{5.5}$$

c and $\bar{c} \in \mathbb{R}^{4\times 1}$ are similarly defined by:

$$c = \begin{bmatrix} q_1(Pq_2 - P - Tq_4) \\ -(q_3-1)(Pq_2 - P - Tq_4) \\ -Pq_1q_2 + Pq_2q_3 + Pq_2 - Pq_3 + Rq_2q_4 - Sq_2q_4 + Sq_4 \\ -Rq_2q_4 + Rq_4 + Sq_2q_4 - Sq_2 - Sq_4 + S + Tq_1q_4 - Tq_3q_4 + Tq_3 - Tq_4 \end{bmatrix}, \tag{5.6}$$

$$\bar{c} = \begin{bmatrix} q_1(q_2 - q_4 - 1) \\ -(q_3-1)(q_2-q_4-1) \\ -q_1q_2 + q_2q_3 + q_2 - q_3 + q_4 \\ q_1q_4 - q_2 - q_3q_4 + q_3 - q_4 + 1 \end{bmatrix}, \tag{5.7}$$

and the constant terms $a, \bar{a}$ are defined as $a = -Pq_2 + P + Tq_4$ and $\bar{a} = -q_2 + q_4 + 1$.

Proof. It was discussed that $u_q(p)$ it is the product of the steady state vector v and the PD payoffs,

$$u_q(p) = v \cdot (R, S, T, P).$$

The steady state vector which is the solution to $vM = v$ is given by

$$\begin{aligned} v = \Bigg[& \frac{p_2p_3(q_2q_4 - q_3q_4) + p_2p_4(q_2q_3 - q_2q_4 - q_3 + q_4) + p_3p_4(-q_2q_3 + q_3q_4) - p_3q_2q_4 + p_4q_4(q_2-1)}{\bar{v}}, \\ & \frac{p_1p_3(q_1q_4 - q_2q_4) + p_1p_4(-q_1q_2 + q_1 + q_2q_4 - q_4) + p_3p_4(q_1q_2 - q_1q_4 - q_2 + q_4) + p_3q_4(q_2-1) - p_4q_2(q_4+1) + p_4(q_4-1)}{\bar{v}}, \\ & \frac{-p_1p_2(q_1q_4 - q_3q_4) - p_1p_4(-q_1q_3 + q_3q_4) + p_1q_1q_4 - p_2p_4(q_1q_3 - q_1q_4 - q_3 + q_4) - p_2q_4(q_3+1) - p_4q_4(q_1+q_3) - p_4(q_3+q_4) - q_4}{\bar{v}}, \\ & \frac{p_1p_2(q_1q_2 - q_1 - q_2q_3 + q_3) + p_1p_3(-q_1q_3 + q_2q_3) - p_1q_1(q_2+1) + p_2p_3(-q_1q_2 + q_1q_3 + q_2 - q_3)}{\bar{v}} + \\ & \frac{p_2(q_3q_2 - q_2 - q_3 - 1) + p_3(q_1q_2 - q_3q_2 - q_2 - q_3) + q_2 - 1}{\bar{v}} \Bigg], \end{aligned}$$

where,

$$\begin{aligned} \bar{v} = \; & p_1p_2(q_1q_2 - q_1q_4 - q_1 - q_2q_3 + q_3q_4 + q_3) - p_1p_3(q_1q_3 - q_1q_4 - q_2q_3 + q_2q_4) - p_1p_4(q_1q_2 - q_1q_3 - q_1 - q_2q_4 + q_3q_4 + q_4) - \\ & p_1q_1(q_2 + q_4 + 1) + p_2p_3(-q_1q_2 + q_1q_3 + q_2q_4 + q_2 - q_3q_4 - q_3) + p_2p_4(-q_1q_3 + q_1q_4 + q_2q_3 - q_2q_4) + p_2q_2(q_3 - 1) - p_2q_3(q_4 - 1) + \\ & p_2(q_4 + 1) + p_3p_4(q_1q_2 - q_1q_4 - q_2q_3 - q_2 + q_3q_4 + q_4) + p_3q_2q_1(-p_3 - 1) + p_3(q_3 - q_4) - p_4(q_1q_4 + q_2 + q_3q_4 - q_3 + q_4 - 1) + \\ & q_2 - q_4 - 1 \end{aligned}$$

The dot product of $v \cdot (R, S, T, P)$ gives,

$$
\begin{aligned}
u_q(p) = {} & \frac{R\left(p_2p_3(q_2q_4 - q_3q_4) + p_2p_4(q_2q_3 - q_2q_4 - q_3 + q_4) + p_3p_4(-q_2q_3 + q_3q_4) - p_3q_2q_4 + p_4q_4(q_2 - 1)\right)}{\bar{v}} + \\
& \frac{S\left(p_1p_3(q_1q_4 - q_2q_4) + p_1p_4(-q_1q_2 + q_1 + q_2q_4 - q_4) + p_3p_4(q_1q_2 - q_1q_4 - q_2 + q_4) + p_3q_4(q_2 - 1) - p_4q_2(q_4 + 1) + p_4(q_4 - 1)\right)}{\bar{v}} + \\
& \frac{T\left(-p_1p_2(q_1q_4 - q_3q_4) - p_1p_4(-q_1q_3 + q_3q_4) + p_1q_1q_4 - p_2p_4(q_1q_3 - q_1q_4 - q_3 + q_4) - p_2q_4(q_3 + 1) - p_4q_4(q_1 + q_3) - p_4(q_3 + q_4) - q_4\right)}{\bar{v}} + \\
& \frac{P\left(p_1(p_2(q_1q_2 - q_1 - q_2q_3 + q_3) + p_3(-q_1q_3 + q_2q_3) - q_1(q_2 + 1)) + p_2p_3((-q_1q_2 + q_1q_3 + q_2 - q_3) + (q_3q_2 - q_2 - q_3 - 1))\right)}{\bar{v}} + \\
& \frac{P\left(p_3(q_1q_2 - q_3q_2 - q_2 - q_3) + q_2 - 1\right)}{\bar{v}} \implies
\end{aligned}
$$

$$
\begin{aligned}
u_q(p) = {} & \\
= {} & \left(\frac{\begin{array}{c}-p_1p_2(q_1 - q_3)(Pq_2 - P - Tq_4) + p_1p_3(q_1 - q_2)(Pq_3 - Sq_4) + p_1p_4(q_1 - q_4)(Sq_2 - S - Tq_3) + p_2p_3(q_2 - q_3)(Pq_1 - P - Rq_4) - \\ p_2p_4(q_3 - q_4)(Rq_2 - R - Tq_1 + T) + p_3p_4(q_2 - q_4)(Rq_3 - Sq_1 + S) + p_1q_1(Pq_2 - P - Tq_4) - p_2(q_3 - 1)(Pq_2 - P - Tq_4) + \\ p_3(-Pq_1q_2 + Pq_2q_3 + Pq_2 - Pq_3 + Rq_2q_4 - Sq_2q_4 + Sq_4) + p_4(-Rq_2q_4 + Rq_4 + Sq_2q_4 - Sq_2 - Sq_4 + S + Tq_1q_4 - Tq_3q_4 + Tq_3 - Tq_4) \\ -Pq_2 + P + Tq_4\end{array}}{\begin{array}{c}p_1p_2(q_1q_2 - q_1q_4 - q_1 - q_2q_3 + q_3q_4 + q_3) + p_1p_3(-q_1q_3 + q_1q_4 + q_2q_3 - q_2q_4) + p_1p_4(-q_1q_2 + q_1q_3 + q_1 + q_2q_4 - q_3q_4 - q_4) + \\ p_2p_3(-q_1q_2 + q_1q_3 + q_2q_4 + q_2 - q_3q_4 - q_3) + p_2p_4(-q_1q_3 + q_1q_4 + q_2q_3 - q_2q_4) + p_3p_4(q_1q_2 - q_1q_4 - q_2q_3 - q_2 + q_3q_4 + q_4) + \\ p_1(-q_1q_2 + q_1q_4 + q_1) + p_2(q_2q_3 - q_2 - q_3q_4 - q_3 + q_4 + 1) + p_3(q_1q_2 - q_2q_3 - q_2 + q_3 - q_4) + p_4(-q_1q_4 + q_2 + q_3q_4 - q_3 + q_4 - 1) + \\ q_2 - q_4 - 1\end{array}}\right).
\end{aligned}
$$

Let us consider the numerator of $u_q(p)$. The cross product terms p_ip_j are given by,

$$
\begin{aligned}
-p_1p_2(q_1 - q_3)(Pq_2 - P - Tq_4) + p_1p_3(q_1 - q_2)(Pq_3 - Sq_4) + p_1p_4(q_1 - q_4)(Sq_2 - S - Tq_3) + \\
p_2p_3(q_2 - q_3)(Pq_1 - P - Rq_4) - p_2p_4(q_3 - q_4)(Rq_2 - R - Tq_1 + T) + p_3p_4(q_2 - q_4)(Rq_3 - Sq_1 + S)
\end{aligned}
$$

This can be re written in a matrix format given by Equation 5.8.

$$
(p_1, p_2, p_3, p_4)\frac{1}{2}\begin{bmatrix} 0 & -(q_1 - q_3)(Pq_2 - P - Tq_4) & (q_1 - q_2)(Pq_3 - Sq_4) & (q_1 - q_4)(Sq_2 - S - Tq_3) \\ -(q_1 - q_3)(Pq_2 - P - Tq_4) & 0 & (q_2 - q_3)(Pq_1 - P - Rq_4) & -(q_3 - q_4)(Rq_2 - R - Tq_1 + T) \\ (q_1 - q_2)(Pq_3 - Sq_4) & (q_2 - q_3)(Pq_1 - P - Rq_4) & 0 & (q_2 - q_4)(Rq_3 - Sq_1 + S) \\ (q_1 - q_4)(Sq_2 - S - Tq_3) & -(q_3 - q_4)(Rq_2 - R - Tq_1 + T) & (q_2 - q_4)(Rq_3 - Sq_1 + S) & 0 \end{bmatrix}\begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} \tag{5.8}
$$

Similarly, the linear terms are given by,

$$
\begin{aligned}
p_1q_1(Pq_2 - P - Tq_4) - p_2(q_3 - 1)(Pq_2 - P - Tq_4) + p_3(-Pq_1q_2 + Pq_2q_3 + Pq_2 - Pq_3 + Rq_2q_4 - Sq_2q_4 + Sq_4) + \\
p_4(-Rq_2q_4 + Rq_4 + Sq_2q_4 - Sq_2 - Sq_4 + S + Tq_1q_4 - Tq_3q_4 + Tq_3 - Tq_4)
\end{aligned}
$$

and the expression can be written using a matrix format as Equation 5.9.

$$
(p_1, p_2, p_3, p_4)\begin{bmatrix} q_1(Pq_2 - P - Tq_4) \\ -(q_3 - 1)(Pq_2 - P - Tq_4) \\ -Pq_1q_2 + Pq_2q_3 + Pq_2 - Pq_3 + Rq_2q_4 - Sq_2q_4 + Sq_4 \\ -Rq_2q_4 + Rq_4 + Sq_2q_4 - Sq_2 - Sq_4 + S + Tq_1q_4 - Tq_3q_4 + Tq_3 - Tq_4 \end{bmatrix} \tag{5.9}
$$

Finally, the constant term of the numerator, which is obtained by substituting $p = (0, 0, 0, 0)$, is given by Equation 5.10.

$$
-Pq_2 + P + Tq_4 \tag{5.10}
$$

Combining Equation 5.8, Equation 5.9 and Equation 5.10 gives that the numerator of $u_q(p)$

can be written as,

$$\frac{1}{2}p\begin{bmatrix} 0 & -(q_1-q_3)(Pq_2-P-Tq_4) & (q_1-q_2)(Pq_3-Sq_4) & (q_1-q_4)(Sq_2-S-Tq_3) \\ -(q_1-q_3)(Pq_2-P-Tq_4) & 0 & (q_2-q_3)(Pq_1-P-Rq_4) & -(q_3-q_4)(Rq_2-R-Tq_1+T) \\ (q_1-q_2)(Pq_3-Sq_4) & (q_2-q_3)(Pq_1-P-Rq_4) & 0 & (q_2-q_4)(Rq_3-Sq_1+S) \\ (q_1-q_4)(Sq_2-S-Tq_3) & -(q_3-q_4)(Rq_2-R-Tq_1+T) & (q_2-q_4)(Rq_3-Sq_1+S) & 0 \end{bmatrix} p^T +$$

$$\begin{bmatrix} q_1(Pq_2-P-Tq_4) \\ -(q_3-1)(Pq_2-P-Tq_4) \\ -Pq_1q_2+Pq_2q_3+Pq_2-Pq_3+Rq_2q_4-Sq_2q_4+Sq_4 \\ -Rq_2q_4+Rq_4+Sq_2q_4-Sq_2-Sq_4+S+Tq_1q_4-Tq_3q_4+Tq_3-Tq_4 \end{bmatrix} p - Pq_2 + P + Tq_4$$

and equivalently as,

$$\frac{1}{2}pQp^T + cp + a$$

where $Q \in \mathbb{R}^{4\times4}$ is a square matrix defined by the transition probabilities of the opponent q_1, q_2, q_3, q_4 as follows:

$$Q = \begin{bmatrix} 0 & -(q_1-q_3)(Pq_2-P-Tq_4) & (q_1-q_2)(Pq_3-Sq_4) & (q_1-q_4)(Sq_2-S-Tq_3) \\ -(q_1-q_3)(Pq_2-P-Tq_4) & 0 & (q_2-q_3)(Pq_1-P-Rq_4) & -(q_3-q_4)(Rq_2-R-Tq_1+T) \\ (q_1-q_2)(Pq_3-Sq_4) & (q_2-q_3)(Pq_1-P-Rq_4) & 0 & (q_2-q_4)(Rq_3-Sq_1+S) \\ (q_1-q_4)(Sq_2-S-Tq_3) & -(q_3-q_4)(Rq_2-R-Tq_1+T) & (q_2-q_4)(Rq_3-Sq_1+S) & 0 \end{bmatrix},$$

$c \in \mathbb{R}^{4\times1}$ is similarly defined by:

$$c = \begin{bmatrix} q_1(Pq_2-P-Tq_4) \\ -(q_3-1)(Pq_2-P-Tq_4) \\ -Pq_1q_2+Pq_2q_3+Pq_2-Pq_3+Rq_2q_4-Sq_2q_4+Sq_4 \\ -Rq_2q_4+Rq_4+Sq_2q_4-Sq_2-Sq_4+S+Tq_1q_4-Tq_3q_4+Tq_3-Tq_4 \end{bmatrix},$$

and $a = -Pq_2 + P + Tq_4$.

The same process is done for the denominator. □

Numerical simulations have been carried out to validate the result. The simulated utility, which is denoted as $U_q(p)$, has been calculated with APL. For smoothing the simulated results the utility has been estimated in a tournament of 500 turns and 200 repetitions. Figure 5.2 shows two examples demonstrating that the formulation of Theorem 1 successfully captures the simulated behaviour.

Theorem 1 can be extended to consider multiple opponents. The IPD is commonly studied in tournaments and/or Moran Processes where a strategy interacts with a number of opponents. The payoff of a player in such interactions is given by the average payoff the player received against each opponent. More specifically the expected utility of a memory-one strategy against N opponents is given by Theorem 2.

Theorem 2. *The expected utility of a memory-one strategy $p \in \mathbb{R}^4_{[0,1]}$ against a group of*

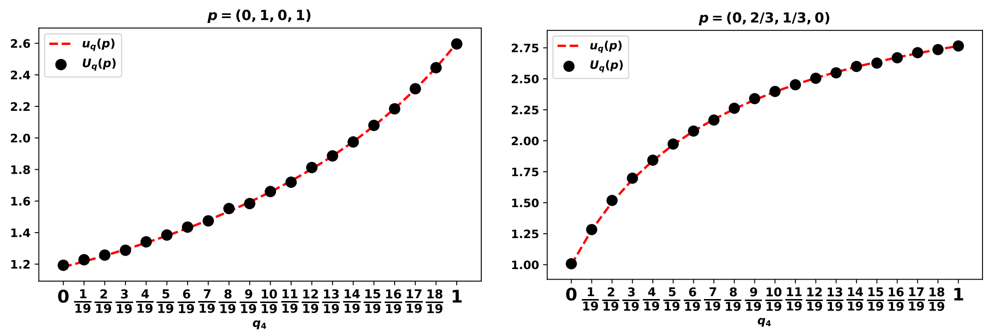


Figure 5.2: Simulated and empirical utilities for $p = (0, 1, 0, 1)$ and $p = (0, \frac{2}{3}, \frac{1}{3}, 0)$ against $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, q_4)$ for $q_4 \in \{0, \frac{1}{19}, \frac{2}{19}, \dots, \frac{18}{19}, 1\}$. $u_q(p)$ is the theoretic value given in Theorem 1, and $U_q(p)$ is simulated numerically.

opponents $\{q^{(1)}, q^{(2)}, \dots, q^{(N)}\}$, *denoted as* $\frac{1}{N}\sum\limits_{i=1}^{N} {u_q}^{(i)}(p)$, *is given by:*

$$\frac{1}{N}\sum_{i=1}^{N} {u_q}^{(i)}(p) = \frac{1}{N}\frac{\sum\limits_{i=1}^{N}\left(\frac{1}{2}pQ^{(i)}p^T + c^{(i)}p + a^{(i)}\right)\prod\limits_{\substack{j=1\\ j\neq i}}^{N}\left(\frac{1}{2}p\bar{Q}^{(j)}p^T + \bar{c}^{(j)}p + \bar{a}^{(j)}\right)}{\prod\limits_{i=1}^{N}\left(\frac{1}{2}p\bar{Q}^{(i)}p^T + \bar{c}^{(i)}p + \bar{a}^{(i)}\right)}. \tag{5.11}$$

Equation (5.11) is the average score (using Equation (5.3)) against the set of opponents.

Similar to the previous result, the formulation of Theorem 2 is validated using numerical simulations where the 10 memory-one strategies described in [112] have been used as the opponents. Figure 5.3 shows that the simulated behaviour has been captured successfully.

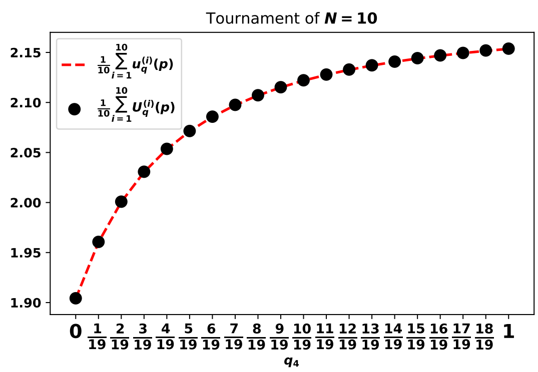


Figure 5.3: The utilities of memory-one strategies $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, p_4)$ for $p_4 \in \{0, \frac{1}{19}, \frac{2}{19}, \dots, \frac{18}{19}, 1\}$ against the 10 memory-one strategies described in [112]. $\frac{1}{10}\sum_{i=1}^{10} u_q^{(i)}(p)$ is the theoretic value given in Theorem 1, and $\frac{1}{10}\sum_{i=1}^{10} U_q^{(i)}(p)$ is simulated numerically.

The list of strategies from [112] was also used to check whether the utility against a group of strategies could be captured by the utility against the mean opponent. Thus whether condition

(5.12) holds. However condition (5.12) fails, as shown in Figure 5.4.

$$\frac{1}{N}\sum_{i=1}^{N} u_q{}^{(i)}(p) = u_{\frac{1}{N}\sum\limits_{i=1}^{N} q^{(i)}}(p), \tag{5.12}$$

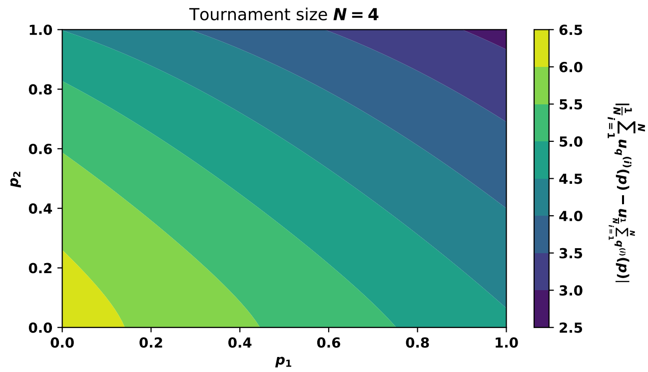


Figure 5.4: The difference between the average utility against the opponents from [112] and the utility against the average player of the strategies in [112] of a player $p = (p_1, p_2, p_1, p_2)$. A positive difference indicates that condition (5.12) does not hold.

Theorem 2 which allows for the utility of a memory-one strategy against any number of opponents to be estimated without simulating the interactions is the main result used in the rest of this Chapter. In section 5.3 it is used to define best response memory-one strategies, in section 5.4 to define best response reactive strategies and in section 5.6 to explore the conditions under which defection dominates cooperation.

5.3 Best responses to memory-one players

This section focuses on *memory-one best response* strategies. A best response is a strategy which corresponds to the most favourable outcome (Chapter 1), thus a memory-one best response to a set of opponents $q^{(1)}, q^{(2)}, \dots, q^{(N)}$ corresponds to a strategy p^* for which Equation (5.11) is maximised. This is considered as a multi dimensional optimisation problem given by:

$$\begin{aligned} \max_p : & \sum_{i=1}^{N} u_q{}^{(i)}(p) \\ \text{such that} : & \; p \in \mathbb{R}_{[0,1]} \end{aligned} \tag{5.13}$$

Optimising this particular ratio of quadratic forms is not trivial. It can be verified empirically for the case of a single opponent that there exists at least one point for which the definition of concavity does not hold, see Appendix B. The non concavity of $u(p)$ indicates multiple local optimal points. This is also intuitive. The best response against a cooperator, $q = (1, 1, 1, 1)$, is a defector $p^* = (0, 0, 0, 0)$. The strategies $p = (\frac{1}{2}, 0, 0, 0)$ and $p = (\frac{1}{2}, 0, 0, \frac{1}{2})$ are also best responses. The approach taken here is to introduce a compact way of constructing the discrete candidate set of all local optimal points, and evaluating the objective function Equation (5.11). This gives the best response memory-one strategy. The approach is given in Theorem 3.

Theorem 3. *The optimal behaviour of a memory-one strategy $p^* \in \mathbb{R}^4_{[0,1]}$ against a set of N* *opponents* $\{q^{(1)}, q^{(2)}, \dots, q^{(N)}\}$ *for* $q^{(i)} \in \mathbb{R}^4_{[0,1]}$ *is given by:*

$$p^* = \text{argmax} \sum_{i=1}^{N} u_q(p), \ p \in S_q.$$

The set S_q *is defined as all the possible combinations of:*

$$S_q = \left\{ p \in \mathbb{R}^4 \left| \begin{array}{l} \bullet \quad p_j \in \{0,1\} \quad \textit{and} \quad \dfrac{d}{dp_k} \displaystyle\sum_{i=1}^{N} u_q^{(i)}(p) = 0 \quad \textit{for all} \quad j \in J \quad \& \quad k \in K \quad \textit{for all} \quad J, K \\ \qquad\qquad \textit{where} \quad J \cap K = \emptyset \quad \textit{and} \quad J \cup K = \{1,2,3,4\}. \\ \bullet \quad p \in \{0,1\}^4 \end{array} \right. \right\}. \tag{5.14}$$

Proof. The optimal behaviour of a memory-one strategy player $p^* \in \mathbb{R}^4_{[0,1]}$ against a set of N opponents $\{q^{(1)}, q^{(2)}, \dots, q^{(N)}\}$ for $q^{(i)} \in \mathbb{R}^4_{[0,1]}$ is established by:

$$p^* = \text{argmax} \left(\sum_{i=1}^{N} u_q(p) \right), \ p \in S_q,$$

where S_q is given by (5.14).

The optimisation problem of (5.13) can be written as:

$$\begin{aligned} \max_p : \ & \sum_{i=1}^{N} {u_q}^{(i)}(p) \\ \text{such that} : p_i & \leq 1 \text{ for } \in \{1,2,3,4\} \\ -p_i & \leq 0 \text{ for } \in \{1,2,3,4\} \end{aligned} \tag{5.15}$$

The optimisation problem has two inequality constraints and regarding the optimality this means that:

- either the optimum is away from the boundary of the optimization domain, and so the constraints plays no role;
- or the optimum is on the constraint boundary.

Thus, the following three cases must be considered:

Case 1: The solution is on the boundary and any of the possible combinations for $p_i \in \{0,1\}$ for $i \in \{1,2,3,4\}$ are candidate optimal solutions.

Case 2: The optimum is away from the boundary of the optimization domain and the interior solution p^* necessarily satisfies the condition $\frac{d}{dp} \sum\limits_{i=1}^{N} u_q(p^*) = 0$.

Case 3: The optimum is away from the boundary of the optimization domain but some constraints are equalities. The candidate solutions in this case are any combinations of $p_j \in \{0,1\}$ and $\frac{d}{dp_k} \sum\limits_{i=1}^{N} u_q^{(i)}(p) = 0$ forall $j \in J$ & $k \in K$ forall J, K where $J \cap K = \emptyset$ and $J \cup K = \{1,2,3,4\}$.

Combining cases 1-3 a set of candidate solution is constructed as:

$$S_q = \left\{ p \in \mathbb{R}^4 \;\middle|\; \begin{array}{l} \bullet \quad p_j \in \{0,1\} \quad \text{and} \quad \dfrac{d}{dp_k}\displaystyle\sum_{i=1}^{N} u_q^{(i)}(p) = 0 \quad \text{for all} \quad j \in J \quad \& \quad k \in K \quad \text{for all} \quad J, K \\ \qquad\qquad\qquad\quad \text{where} \quad J \cap K = \emptyset \quad \text{and} \quad J \cup K = \{1,2,3,4\}. \\ \bullet \quad p \in \{0,1\}^4 \end{array} \right\}.$$

The derivative of $\sum\limits_{i=1}^{N} u_q^{(i)}(p)$ calculated using the following property (see [6] for details):

$$\frac{dxAx^T}{dx} = 2Ax. \tag{5.16}$$

Using property (5.16):

$$\frac{d}{dp}\frac{1}{2}pQp^T + cp + a = pQ + c \text{ and } \frac{d}{dp}\frac{1}{2}p\bar{Q}p^T + \bar{c}p + \bar{a} = p\bar{Q} + \bar{c}. \tag{5.17}$$

Note that the derivative of cp is c and the constant disappears. Combining these it can be proven that:

$$\begin{aligned} \frac{d}{dp}\sum_{i=1}^{N} u_q^{(i)}(p) &= \sum_{i=1}^{N} \frac{\frac{d}{dp}(\frac{1}{2}pQ^{(i)}p^T + c^{(i)}p + a^{(i)})(\frac{1}{2}pQ^{\bar{(i)}}p^T + c^{\bar{(i)}}p + a^{\bar{(i)}}) - \frac{d}{dp}(\frac{1}{2}pQ^{\bar{(i)}}p^T + c^{\bar{(i)}}p + a^{\bar{(i)}})(\frac{1}{2}pQ^{(i)}p^T + c^{(i)}p + a^{(i)})}{(\frac{1}{2}pQ^{\bar{(i)}}p^T + c^{\bar{(i)}}p + a^{\bar{(i)}})^2} \\ &= \sum_{i=1}^{N} \frac{(pQ^{(i)} + c^{(i)}+)(\frac{1}{2}pQ^{\bar{(i)}}p^T + c^{\bar{(i)}}p + a^{\bar{(i)}})}{(\frac{1}{2}pQ^{\bar{(i)}}p^T + c^{\bar{(i)}}p + a^{\bar{(i)}})^2} - \frac{(pQ^{\bar{(i)}} + c^{\bar{(i)}})(\frac{1}{2}pQ^{(i)}p^T + c^{(i)}p + a^{(i)})}{(\frac{1}{2}pQ^{\bar{(i)}}p^T + c^{\bar{(i)}}p + a^{\bar{(i)}})^2} \end{aligned}$$

For $\frac{d}{dp}\sum\limits_{i=1}^{N} u_q(p)$ to equal zero then:

$$\sum_{i=1}^{N} \left(pQ^{(i)} + c^{(i)}\right)\left(\frac{1}{2}p\bar{Q}^{(i)}p^T + \bar{c}^{(i)}p + \bar{a}^{(i)}\right) - \left(p\bar{Q}^{(i)} + \bar{c}^{(i)}\right)\left(\frac{1}{2}pQ^{(i)}p^T + c^{(i)}p + a^{(i)}\right) = 0, \quad \textit{while} \tag{5.18}$$

$$\sum_{i=1}^{N} \frac{1}{2}p\bar{Q}^{(i)}p^T + \bar{c}^{(i)}p + \bar{a}^{(i)} \neq 0. \tag{5.19}$$

The optimal solution to Equation 5.13 is the point from S_q for which the utility is maximised.

□

Note that there is no immediate way to find the zeros of $\frac{d}{dp}\sum\limits_{i=1}^{N} u_q(p)$;

$$\begin{aligned} &\frac{d}{dp}\sum_{i=1}^{N} {u_q}^{(i)}(p) = \\ &\qquad = \sum_{i=1}^{N} \frac{\left(pQ^{(i)} + c^{(i)}\right)\left(\frac{1}{2}p\bar{Q}^{(i)}p^T + \bar{c}^{(i)}p + \bar{a}^{(i)}\right) - \left(p\bar{Q}^{(i)} + \bar{c}^{(i)}\right)\left(\frac{1}{2}pQ^{(i)}p^T + c^{(i)}p + a^{(i)}\right)}{\left(\frac{1}{2}p\bar{Q}^{(i)}p^T + \bar{c}^{(i)}p + \bar{a}^{(i)}\right)^2} \end{aligned} \tag{5.20}$$

For $\frac{d}{dp}\sum\limits_{i=1}^{N} u_q(p)$ to equal zero then:

$$\sum_{i=1}^{N}\left(\left(pQ^{(i)}+c^{(i)}\right)\left(\frac{1}{2}p\bar{Q}^{(i)}p^{T}+\bar{c}^{(i)}p+\bar{a}^{(i)}\right)-\left(p\bar{Q}^{(i)}+\bar{c}^{(i)}\right)\left(\frac{1}{2}pQ^{(i)}p^{T}+c^{(i)}p+a^{(i)}\right)\right)=0, \quad while \tag{5.21}$$

$$\sum_{i=1}^{N}\frac{1}{2}p\bar{Q}^{(i)}p^{T}+\bar{c}^{(i)}p+\bar{a}^{(i)}\neq 0. \tag{5.22}$$

Finding best response memory-one strategies, more specifically constructing the subset S_q, can be done analytically. The points for any or all of $p_i \in \{0,1\}$ for $i \in \{1,2,3,4\}$ are trivial, and finding the roots of the partial derivatives which are a set of polynomial Equations (5.21) is feasible using resultant theory [63]; however, for large systems building the resultant quickly becomes intractable.

Nevertheless, there are constrained versions of problem (5.13) for which calculating the resultant is efficient and a best response strategy can be identified explicitly. A constrained version is that of reactive strategies. Section 5.4 presents best response reactive strategies, and demonstrates the usage of resultant theory in identifying best responses.

5.4 Reactive Strategies & Resultant Theory

Reactive strategies are a subset of memory-one strategies discussed in Chapter 2. Well known reactive strategies include Tit For Tat and Generous Tit for Tat. As a reminder, reactive strategies only take into account the opponent's previous moves, and thus can be described as $p = (p_1, p_2) \in R^2$.

Best response reactive strategies are incorporated in the formulation of this Chapter by adding two extra constraints to the optimisation problem of (5.13),

$$\begin{aligned}\max_{p}:&\ \sum_{i=1}^{N}u_q(p)\\ \text{such that}:&\ p_1=p_3\\ &\ p_2=p_4\\ &\ p_1,p_2\in\mathbb{R}_{[0,1]}.\end{aligned} \tag{5.23}$$

and a best response reactive strategy to a set of opponents N opponents $\{q^{(1)}, q^{(2)}, \dots, q^{(N)}\}$ is given by Lemma 4.

Lemma 4. *The optimal behaviour of a reactive strategy $p^* \in \mathbb{R}^2_{[0,1]}$ against a set of N opponents $\{q^{(1)}, q^{(2)}, \dots, q^{(N)}\}$ for $q^{(i)} \in \mathbb{R}^2_{[0,1]}$ is given by:*

$$p^* = \text{argmax}\sum_{i=1}^{N}u_q(p),\ \ p \in S_q.$$

The set S_q is defined as all the possible combinations of:

$$S_q = \left\{ p \in \mathbb{R}^2 \;\middle|\; \begin{array}{l} \bullet \quad p_j \in \{0,1\} \quad \textit{and} \quad \dfrac{d}{dp_k}\sum_{i=1}^{N} u_q^{(i)}(p) = 0 \quad \textit{for all} \quad j \in J \quad \& \quad k \in K \quad \textit{for all} \quad J, K \\ \qquad\qquad \textit{where} \quad J \cap K = \emptyset \quad \textit{and} \quad J \cup K = \{1,2\}. \\ \bullet \quad p \in \{0,1\}^2 \end{array} \right\}. \tag{5.24}$$

Note that $\frac{d}{dp}\sum\limits_{i=1}^{N} u_q^{(i)}(p) = 0$ corresponds to a system of 2 polynomials of 2 variables, corresponding to the partial derivatives over p_1 and p_2. Solving systems of 2 polynomials of 2 variables can be done analytically. The approach taken here to extract the roots from the partial derivatives is to use resultants.

5.4.1 Resultant Theory

The *resultant* of two polynomials is a polynomial expression of their coefficients which is equal to zero if and only if the polynomials have a common root.

More specifically given a polynomial,

$$f(x) = a_n x^n + a_{(n-1)} x^{(n-1)} + \cdots + a_1 x + a_0$$

of degree n with roots $\alpha_i, i = 1, \ldots, n$ and a polynomial

$$g(x) = b_m x^m + b_{(m-1)} x^{(m-1)} + \cdots + b_1 x + b_0$$

of degree m with roots $\beta_j, j = 1, \ldots, m$, the resultant denoted as $R(f,g)$ and also called the eliminant [106], is defined by

$$R(f,g) = a_n^m b_m^n \prod_{(i=1)}^{n} \prod_{(j=1)}^{m} (\alpha_i - \beta_j). \tag{5.25}$$

Interestingly, the resultant can also be expressed as the determinant of matrices such as Sylvester's, Bezout's and Macaulay's. For systems of 2 polynomials the resultant is commonly expressed as the determinant of Sylvester's matrix [9]. The Sylvester matrix associated with f

and g is the $(n+m)\times(n+m)$ matrix constructed as described in Algorithm 1.

Algorithm 1: Construction of Sylvester matrix [9]

if $n > 0$ **then**
- the 1^{st} row is $\begin{pmatrix} a_m & a_{m-1} & \cdots & a_1 & a_0 & 0 & \cdots & 0 \end{pmatrix}$;
- **for** $i \leftarrow 1$ **to** $n-1$ **do**
  - the i^{th} is the previous row shifted one column to the right; the other entries of the row are 0
- ;

if $m > 0$ **then**
- the $(n+1)^{\text{th}}$ is: $\begin{pmatrix} b_m & b_{m-1} & \cdots & b_1 & b_0 & 0 & \cdots & 0 \end{pmatrix}$;
- **for** $i \leftarrow n+1$ **to** $(m+n)-1$ **do**
  - the i^{th} is the previous row shifted one column to the right; the other entries of the row are 0

As an example consider the case of $m = 4$ and $n = 3$. The Sylvester matrix denoted as $S_{f,g}$ is given by,

$$S_{f,g} = \begin{pmatrix} a_4 & a_3 & a_2 & a_1 & a_0 & 0 & 0 \\ 0 & a_4 & a_3 & a_2 & a_1 & a_0 & 0 \\ 0 & 0 & a_4 & a_3 & a_2 & a_1 & a_0 \\ b_3 & b_2 & b_1 & b_0 & 0 & 0 & 0 \\ 0 & b_3 & b_2 & b_1 & b_0 & 0 & 0 \\ 0 & 0 & b_3 & b_2 & b_1 & b_0 & 0 \\ 0 & 0 & 0 & b_3 & b_2 & b_1 & b_0 \end{pmatrix} \tag{5.26}$$

and,

$$|S_{f,g}| = R(f,g).$$

The resultant can verify that the system has a root, but also can be used to extract the roots. In a system of 2 polynomial equations in 2 variables, the resultant can be defined over one variable whereas the second one is kept as a coefficient. It is used in this Chapter to analytically solve for the roots of the derivative of the utility, and thus explicitly identify the best response reactive strategy against a given set of memory-one opponents.

The open source package [81] is used to calculate the Sylvester matrix and subsequently it's determinant. An example is demonstrated in Figure 5.5.

The approach demonstrated in Figure 5.5 is used to find the roots of the partial derivatives of $\frac{d}{dp}\sum\limits_{i=1}^{N} u_q^{(i)}(p)$ and the candidate set of solutions S_q is constructed as defined in Equation (5.24).

```
>>> import sympy as sym
>>> from sympy.polys import subresultants_qq_zz
>>> p_1, p_2 = sym.symbols('p_1, p_2')

>>> f = p_1 ** 2 + p_1 * p_2 + 2 * p_1 + p_2 - 1
>>> g = p_1 ** 2 + 3 * p_1 - p_2 ** 2 + 2 * p_2 - 1
>>> matrix = subresultants_qq_zz.sylvester(f, g, p_2)
>>> matrix
Matrix([
[p_1 + 1, p_1**2 + 2*p_1 - 1,                  0],
[      0,            p_1 + 1, p_1**2 + 2*p_1 - 1],
[     -1,                  2, p_1**2 + 3*p_1 - 1]])
>>> matrix.det().factor()
-p_1*(p_1 - 1)*(p_1 + 3)
```

Figure 5.5: Example code for calculating the Sylvester matrix associated with $f = p_1^2 + p_1p_2 + 2p_1 + p_2 - 1$ and $g = p_1^2 + 3p_1 - p_2^2 + 2p_2 - 1$ using [81]. The matrix is calculated for p_2 whilst p_1 is handled as a coefficient, and thus the determinant is expressed in p_1. In order for the system to have a common root, p_1 must be $\in \{-3, 0, 1\}$. By substituting these values of p_1, each at a time, in f and g gives the roots for p_2.

A bespoke package has been developed to carry out the calculations for this Chapter. The package is called `opt_mo` and an example of how S_q is calculated against a given opponent $q = (0.513, 0.773, 0.870, 0.008)$ is given by Figure 5.6.

```
>>> import opt_mo
>>> import numpy as np
>>> import axelrod as axl

>>> axl.seed(14)
>>> opponents = [np.random.random(4)]
>>> opponents
[array([0.51394334, 0.77316505, 0.87042769, 0.00804695])]

>>> candinate_set = opt_mo.reactive_best_response.get_candinate_reactive_best_responses(
...     opponents
... )
>>> candinate_set
{(0.913428410721382+0j), 0.6964731896521483, 0.2775453690890986, 0, 1}
```

Figure 5.6: Code example of calculating S_q for a given opponent. The function `reactive_best_response.get_candidate_reactive_best_responses` retrieves the set S_q for a reactive strategy against a list of opponents. The set includes 0, 1 and to roots of the partial derivatives $0.2775453690890986, 0.6964731896521483$. An imaginary solution has also been calculated, however, it is ignore in the next step which calculates the best response.

Once S_q is calculated then defining the best response is trivial. Figure 5.7 demonstrates how this is done using `opt_mo`, and the result is validated by Figure 5.8.

Sylvester's formulation can only handle systems of 2 polynomials, however, the multivariate resultants can be calculated for n homogeneous polynomials in n variables. A number of multivariate resultants can be found in the literature such as Dixon's [103] resultant and Macaulay's [78] resultant.

Project [81] which was used to construct Sylvester's resultant is called SymPy and it is the Pythonic package for symbolic mathematics. However, the project did not include the feature to calculate multivariate resultants. As part of this Chapter the source code for constructing

```
>>> import numpy as np
>>> import opt_mo

>>> opponents = [np.array([0.51394334, 0.77316505, 0.87042769, 0.00804695])]
>>> candinate_set = opt_mo.reactive_best_response.get_candinate_reactive_best_responses(
...     opponents
... )

>>> opt_p1, opt_p2, score = opt_mo.reactive_best_response.get_argmax(
...     opponents, candinate_set
... )
>>> opt_p1, opt_p2
(0, 0.6964731842832972)
```

Figure 5.7: Code example of estimating the best response reactive strategy from a given S_q set and a given list of opponents.

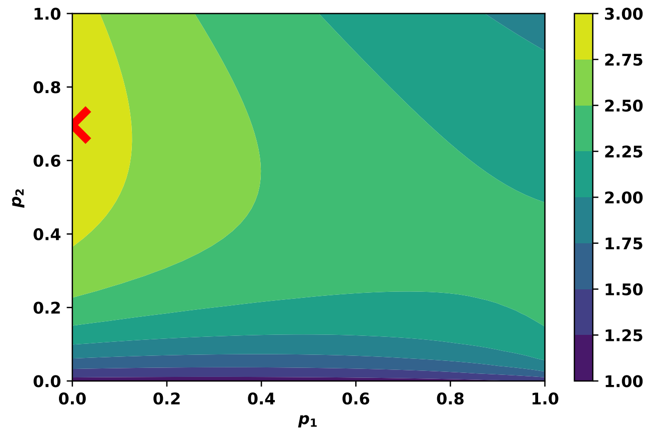


Figure 5.8: The utility of a $p = (p_1, p_2)$ reactive player against $q = (0.513, 0.773, 0.870, 0.008)$ for changing values of p_1 and p_2. The point marked with X is the point identified as the best response.

both Dixon's and Macaulay's resultants was developed and was integrated into project [81]. Figure 5.9 shows the pull request made to SymPy for integrating the source code to their codebase.

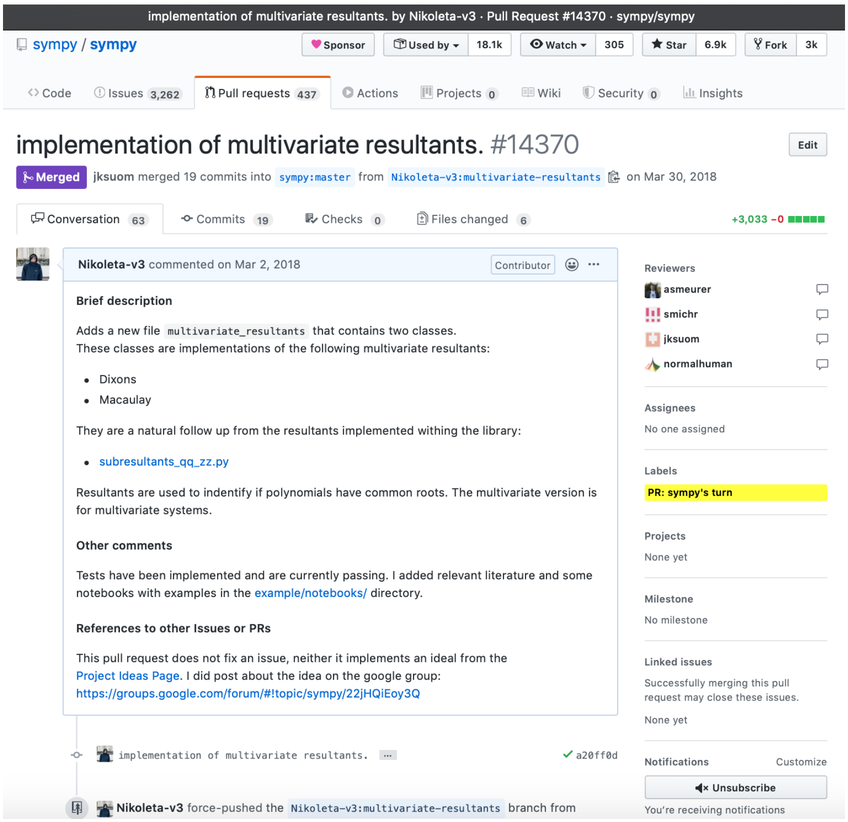


Figure 5.9: Screenshot of the pull request made to SymPy for integrating the source code of the multivariate resultants to the project's codebase. The details of the pull request as well as the conversation with the project's main contributor can be found at: `https://github.com/sympy/sympy/pull/14370`.

Figure 5.10 demonstrates an example of using [81] to calculate Dixon's resultant.

Multivariate resultants theoretically can be used to explicitly identify best response memory one strategies, but solving a system of 4 polynomials. However, as previously stated for large systems building the resultant quickly becomes intractable. As a result in section 5.5 a numerical approach was considered instead.

5.5 Numerical experiments

As briefly discussed in section 3.1, ZDs have been praised for their robustness against a single opponent. ZDs are evidence that extortion works in pairwise interactions. Their behaviour ensures that the strategies will never lose a game. However, this thesis argues that in multi opponent interactions, where the payoffs matter, strategies trying to exploit their opponents will suffer. Compared to ZDs, best response memory-one strategies which have a theory of mind of their opponents, utilise their behaviour in order to gain the most from their interactions. The question that arises then is whether best response strategies are optimal because they behave in an extortionate way (section 5.5.3).

```
>>> from sympy.polys.multivariate_resultants import DixonResultant
>>> p_1, p_2 = sym.symbols('p_1, p_2')

>>> f = p_1 + p_2
>>> g = p_1 ** 2 + p_2 ** 3
>>> h = p_1 ** 2 + p_2

>>> dixon = DixonResultant(variables=[p_1, p_2], polynomials=[f, g, h])
>>> poly = dixon.get_dixon_polynomial()
>>> matrix = dixon.get_dixon_matrix(polynomial=poly)
>>> matrix
Matrix([
[ 0,  0, -1,  0, -1],
[ 0, -1,  0, -1,  0],
[-1,  0,  1,  0,  0],
[ 0, -1,  0,  0,  1],
[-1,  0,  0,  1,  0]])

>>> matrix.det()
0
```

Figure 5.10: Code example of using [81] to calculate Dixon's resultant. f, g and h have a common root $(x = 1, y = -1)$. The determinant of Dixon's matrix falls to zero which confirms that the system has a common root.

The other main finding presented in [99] was that short memory of the strategies was all that was needed. This thesis argues that the second limitation of ZDs in multi opponent interactions is that of their restricted memory. To demonstrate the effectiveness of memory in the IPD a best response longer-memory strategy against a given set of memory-one opponents is explored, and it's performance is compared to that of a memory-one best response in section 5.5.4.

The results of this section rely on estimating best response memory-one strategies and understanding whether they behave in an extortionate way. Best responses will be estimated heuristically using Bayesian optimisation, which is described in section 5.5.1, and in order to investigate whether best responses behave in an extortionate matter the SSE method described in section 5.5.2 is used.

5.5.1 Bayesian optimisation

Bayesian optimisation is a global optimisation algorithm that has proven to outperform many other popular algorithms [61]. The algorithm builds a bayesian understanding of the objective function which is well suited to the potential multiple local optimas in the described search space of this Chapter. Differential evolution [115] was also considered, however, it was not selected due to Bayesian optimisation being computationally more efficient.

As described in [43] Bayesian optimisation consists of two main components: a Bayesian statistical model for modelling the objective function, and an acquisition function for deciding where to sample next. The algorithm initially evaluates the objective according to a space-filling experimental design, often consisting of points chosen uniformly at random. They are used iteratively to allocate the remainder of a budget of I function evaluations, as shown in

Algorithm 2.

Algorithm 2: Basic pseudo-code for Bayesian optimization. As given in [43]

Place a Gaussian process prior on f;
Observe f at i_0 points according to an initial space-filling experimental design, set $i = i_0$;
while $i \leq I$ **do**
    Update the posterior probability distribution on f using all available data;
    Let x_i be a maximiser of the acquisition function over x, where the acquisition function is computed using the current posterior distribution;
    Observe $y_i = f(x_i)$;
    Increment i;
return either the point evaluated with the largest $f(x)$, or the point with the largest posterior mean.

The statistical model is invariably a Gaussian process. It provides a Bayesian posterior probability distribution that describes potential values for the objective function at a candidate point. Each time the objective function is observed at the new point the posterior distribution is updated.

The acquisition function measures the value that would be generated by evaluation of the objective function at a new point, based on the current posterior distribution over f. The most commonly used acquisition function,

- Expected Improvement [62]
- Knowledge Gradient [42]
- Entropy Search and Predictive Entropy Search [53]

As an example of the algorithm's usage consider the optimisation problem of (5.13). Figure 5.11 illustrates the change of the utility function over I. The algorithm is set to run for $I = 60$. After 60 function evaluations if the utility has changed in the last 10% evaluations then algorithm runs for a further 20 evaluations. This is repeated until there is no change to the utility in the last 10% of evaluations.

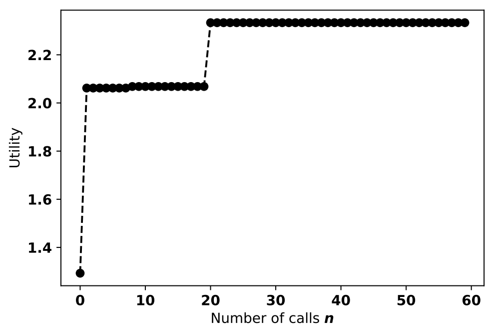


Figure 5.11: Utility over time of calls using Bayesian optimisation. The opponents are $q^{(1)} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ and $q^{(2)} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. The best response obtained is $p^* = (0, \frac{11}{50}, 0, 0)$

5.5.2 SSE Method

In order to investigate whether best responses behave in an extortionate matter the SSE method is used. The SSE method defines a point x^* in the space of memory-one strategies as the nearest extortionate strategy to a given strategy p. x^* is given by,

$$x^* = \left(C^T C\right)^{-1} C^T \bar{p} \tag{5.27}$$

where $\bar{p} = (p_1 - 1, p_2 - 1, p_3, p_4)$ and

$$C = \begin{bmatrix} R-P & R-P \\ S-P & T-P \\ T-P & S-P \\ 0 & 0 \end{bmatrix}. \tag{5.28}$$

Once this closest ZDs is found, the squared norm of the remaining error is referred to as sum of squared errors of prediction (SSE):

$$\text{SSE} = \bar{p}^T \bar{p} - \bar{p} C \left(C^T C\right)^{-1} C^T \bar{p} = \bar{p}^T \bar{p} - \bar{p} C x^* \tag{5.29}$$

Thus, SSE is defined as how far a strategy is from behaving as a ZD. A high SSE implies a non extortionate behaviour.

5.5.3 Best response memory-one strategies for $N = 2$

The results of this section use Bayesian optimisation to generate a data set of best response memory-one strategies for $N = 2$ opponents. This is done in tournaments with and without self interactions.

In several evolutionary settings such as Moran Processes self interactions are key. Previous work has identified interesting results such as the appearance of self recognition mechanisms when training strategies using evolutionary algorithms in Moran processes [70]. This aspect of reinforcement learning can be done for best response memory-one strategies by incorporating the strategy itself in the objective function as shown in Equation 5.13. K is the number of self interactions that will take place.

$$\begin{aligned} \max_p : & \ \frac{1}{N} \sum_{i=1}^{N} {u_q}^{(i)}(p) + K u_p(p) \\ \text{such that} : & \ p \in \mathbb{R}_{[0,1]} \end{aligned} \tag{5.30}$$

For determining the memory-one best response with self interactions, an algorithmic approach is considered, called *best response dynamics*. The best response dynamics approach used in this manuscript is given by Algorithm 3.

Algorithm 3: Best response dynamics Algorithm

$p^{(t)} \leftarrow (1, 1, 1, 1)$;

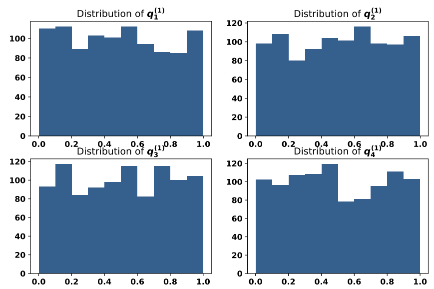
while $p^{(t)} \neq p^{(t-1)}$ **do**

$$p^{(t+1)} = \text{argmax} \frac{1}{N} \sum_{i=1}^{N} {u_q}^{(i)}(p^{(t)}) + K u_{p^{(t)}}(p^{(t)});$$

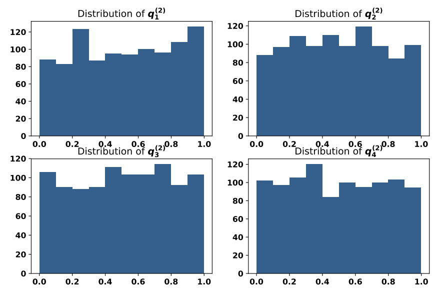
Using this approach it would be possible to create a memory-one best response strategy that updates on every generation of a Moran process to recalculate the optimal behaviour given the population.

For determining the memory-one best response without self interactions, Bayesian algorithm is used to numerically solve the problem of (5.13).

The data set contains a total of 1000 trials corresponding to 1000 different instances of a best response strategy in tournaments with and without self interactions. For each trial a set of 2 opponents is randomly generated and the memory-one best responses against them are found. The probabilities q_i of the opponents are randomly generated and Figures 5.12a and 5.12b, show that they are uniformly distributed over the trials. Thus, the full space of possible opponents has been covered.



(a) Distributions of first opponents' probabilities.



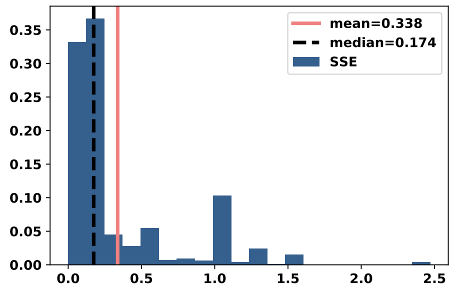
(b) Distributions of second opponents' probabilities.

The SSE method has been applied to the data set. The distributions of SSE for the best response in tournaments ($N = 2$) with and without self interactions with ($K = 1$) are given in Figure 5.13. Moreover, a statistical summary of the SSE distributions is given in Table 5.1.

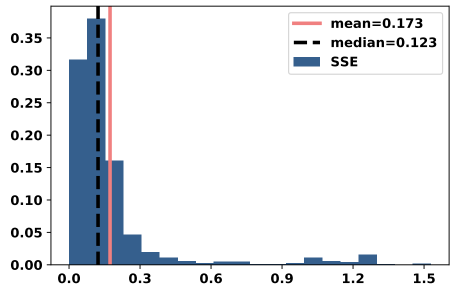
	mean	standard deviation	5%	50%	95%	max	median	skewness	kurtosis
Tournament without self interactions	0.34	0.40	0.028	0.17	1.05	2.47	0.17	1.87	3.60
Tournament with self interactions	0.17	0.23	0.01	0.12	0.67	1.53	0.12	3.42	1.92

Table 5.1: SSE of best response memory-one when $N = 2$

For the best response in tournaments that do not include self interactions the distribution of SSE is skewed to the left, indicating that the best response does exhibit ZDs behaviour and so could be extortionate, however, the best response is not uniformly a ZDs. A positive measure of skewness and kurtosis, and a mean of 0.34 indicate a heavy tail to the right. Therefore, in several cases the strategy is not trying to extort its opponents. In [57] a similar behaviour is refereed to as the *partner strategy*. The partner strategy aims to share the payoff for mutual



(a) SEE distribution for best response in tournaments without self interactions.



(b) SEE distribution for best response in tournaments with self interactions.

Figure 5.13: SEE distributions for best response in tournaments without and with self interactions.

cooperation, but it is ready to fight back when being exploited. The partner strategy was designed, but the best responses which are defined by their opponents seem to exhibit the same behaviour.

Similarly, when considering self interactions, the distribution of SSE for the best response strategy has skewness and kurtosis that indicate a heavy tail to the right. This indicates that evolutionary stable memory-one strategies need to more adaptable than a ZDs, and aim for mutual cooperation as well as exploitation which is in line with the results of [57] where their strategy was designed to adapt and was shown to be evolutionary stable. The findings of this work show that an optimal strategy acts in the same way.

The difference between best responses in tournaments without and with self interactions is further explored by Figure 5.14. Though, no statistically significant differences have been found, from Figure 5.14, it seems that the best response that incorporate self interactions has a higher median p_2; which corresponds to the probability of cooperating after receiving a defection. Thus, they are more likely to forgive after being tricked. This is due to the fact that they could be playing against themselves, and they need to be able to forgive so that future cooperation can occur.

5.5.4 Longer memory best response

This section focuses on the memory size of strategies. The effectiveness of memory in the IPD has been previously explored in the literature, as discussed in section 3.1, however, none of the previous works has compared the performance of longer-memory strategies to memory-one best responses.

The strategy used in this Chapter is one of the archetypes described in Chapter 2 called *Gambler* introduced in [49]. As a reminder, Gambler is a stochastic version of a lookup table. It makes probabilistic decisions based on the opponent's n_1 first moves, the opponent's m_1 last moves and the player's m_2 last moves was introduced. In this Chapter Gambler with parameters: $n_1 = 2, m_1 = 1$ and $m_2 = 1$ is used as a longer-memory strategy.

By considering the opponent's first two moves, the opponents last move and the player's last move, there are only 16 $(4 \times 2 \times 2)$ possible outcomes that can occur, furthermore, Gambler also

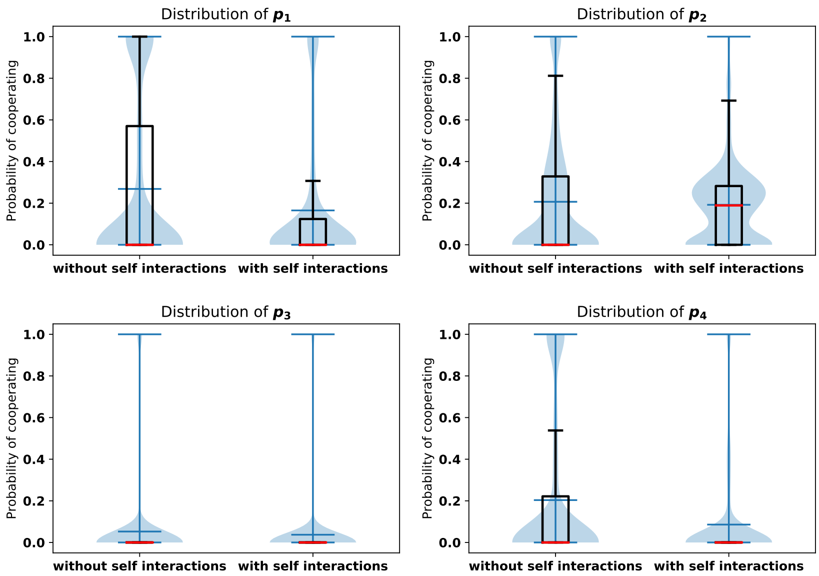


Figure 5.14: Distributions of p^* for best responses in tournaments and evolutionary settings. The medians, denoted as $\bar{p}^*$, for tournaments are $\bar{p}^* = (0, 0, 0, 0)$, and for evolutionary settings $\bar{p}^* = (0, 0.19, 0, 0)$.

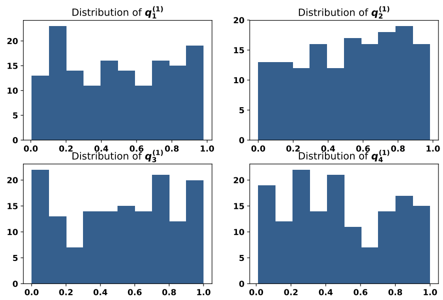
makes a probabilistic decision of cooperating in the opening move. Thus, Gambler is a function $f : \{\text{C, D}\} \to [0, 1]_{\mathbb{R}}$. This can be hard coded as an element of $[0, 1]_{\mathbb{R}}^{16+1}$, one probability for each outcome plus the opening move. Hence, compared to (5.13), finding an optimal Gambler is a 17 dimensional problem given by:

$$\begin{aligned} \max_p : & \sum_{i=1}^{N} {U_q}^{(i)}(f) \\ \text{such that} : & \ f \in \mathbb{R}_{[0,1]}^{17} \end{aligned} \tag{5.31}$$

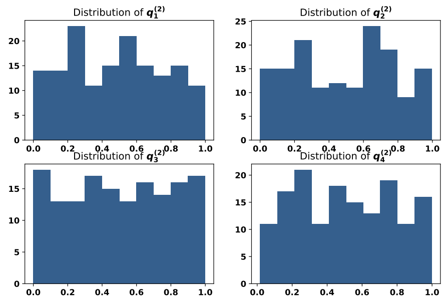
Note that (5.11) can not be used here for the utility of Gambler, and actual simulated players are used. This is done using [5] with 500 turns and 200 repetitions, moreover, (5.31) is solved numerically using Bayesian optimisation.

Similarly to the previous section, a large data set has been generated with instances of an optimal Gambler and a memory-one best response, available at [45]. Estimating a best response Gambler (17 dimensions) is computational more expensive compared to a best response memory-one (4 dimensions). As a result, the analysis of this section is based on a total of 130 trials. For each trial two random opponents have been selected. The 130 pair of opponents are a sub set of the opponents used in section 5.5.3. The distributions of their transition probabilities are given in Figures 5.15a and 5.15a.

The ratio between Gambler's utility and the best response memory-one strategy's utility has been calculated and its distribution in given in Figure 5.16. It is evident from Figure 5.16 that Gambler always performs as well as the best response memory-one strategy and often performs better. There are no points where the ratio value is less than 1, thus Gambler never performed



(a) Distributions of first opponents' probabilities for longer memory experiment.



(b) Distributions of second opponents' probabilities for longer memory experiment.

less than the best response memory-one strategy and in places outperforms it. This seems to be at odd with the result of [99] that against a memory-one opponent having a longer memory will not give a strategy any advantage. However, against two memory-one opponents Gambler's performance is better than the optimal memory-one strategy. This is evidence that in the case of two opponents having a shorter memory is limiting.

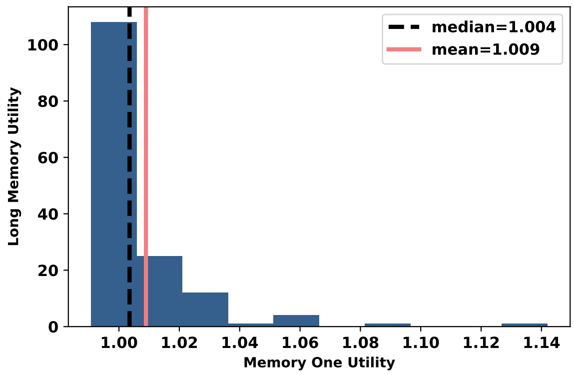


Figure 5.16: Utilities of Gambler and best response memory-one strategies for 130 different pair of opponents.

5.6 Stability of defection

An additional theoretical result that is possible to obtain due to Theorem 2, is a condition for which in an environment of memory-one opponents defection is the stable choice, based only on the coefficients of the opponents.

This result is obtained by evaluating the sign of Equation (5.11)'s derivative at $p = (0, 0, 0, 0)$. If at that point the derivative is negative, then the utility of a player only decreases if they were to change their behaviour, and thus defection at that point is stable.

Lemma 5. *In a tournament of N players $\{q^{(1)}, q^{(2)}, \dots, q^{(N)}\}$ for $q^{(i)} \in \mathbb{R}^4_{[0,1]}$ defection is stable if the transition probabilities of the opponents satisfy conditions Equation 5.32 and Equation (5.33).*

$$\sum_{i=1}^{N} (c^{(i)T} \bar{a}^{(i)} - \bar{c}^{(i)T} a^{(i)}) \leq 0 \tag{5.32}$$

while,

$$\sum_{i=1}^{N} \bar{a}^{(i)} \neq 0 \tag{5.33}$$

Proof. For defection to be stable the derivative of the utility at the point $p = (0, 0, 0, 0)$ must be negative.

Substituting $p = (0, 0, 0, 0)$ in Equation (5.20) gives:

$$\left.\frac{d \sum\limits_{i=1}^{N} {u_q}^{(i)}(p)}{dp}\right|_{p=(0,0,0,0)} = \sum_{i=1}^{N} \frac{(c^{(i)T}\bar{a}^{(i)} - \bar{c}^{(i)T}a^{(i)})}{(\bar{a}^{(i)})^2} \tag{5.34}$$

The sign of the numerator $\sum\limits_{i=1}^{N}(c^{(i)T}\bar{a}^{(i)} - \bar{c}^{(i)T}a^{(i)})$ can vary based on the transition probabilities of the opponents. The denominator can not be negative, and otherwise is always positive. Thus the sign of the derivative is negative if and only if $\sum\limits_{i=1}^{N}(c^{(i)T}\bar{a}^{(i)} - \bar{c}^{(i)T}a^{(i)}) \leq 0$. □

Consider a population for which defection is known to be stable. In that population all the members will over time adopt the same behaviour; thus in such population cooperation will never take over. This is demonstrated in Figure 5.17. These have been simulated using [5] an open source research framework for the study of the IPD.

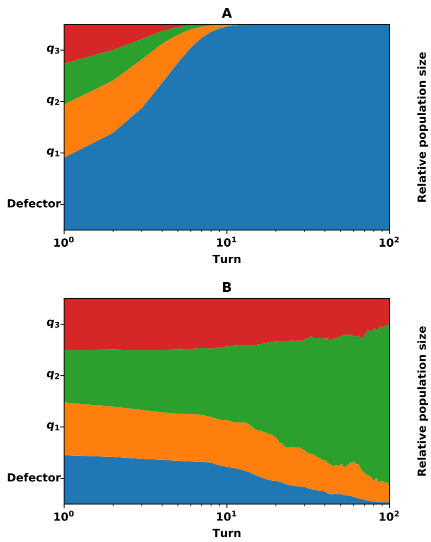


Figure 5.17: A. For $q_1 = (0.22199, 0.87073, 0.20672, 0.91861)$, $q_2 = (0.48841, 0.61174, 0.76591, 0.51842)$ and $q_3 = (0.2968, 0.18772, 0.08074, 0.73844)$, Equation 5.32 and Equation 5.33 hold and Defector takes over the population. B. For $q_1 = (0.96703, 0.54723, 0.97268, 0.71482)$, $q_2 = (0.69773, 0.21609, 0.97627, 0.0062)$ and $q_3 = (0.25298, 0.43479, 0.77938, 0.19769)$, Equation 5.32 fails and Defector does not take over the population.

5.7 Chapter Summary

This Chapter has considered best response strategies in the IPD game, and more specifically, memory-one best responses. It has proven that:

- The utility of a memory-one strategy against a set of memory-one opponents can be written as a sum of ratios of quadratic forms (Theorem 2).
- There is a compact way of identifying a memory-one best response to a group of opponents through a search over a discrete set (Theorem 3).

Note that Theorem 3 which does not only have game theoretic novelty, but also the mathematical novelty of solving quadratic ratio optimisation problems where the quadratics are non concave. Moreover Theorem 2, allowed for a condition for which in an environment of memory-one opponents defection is the stable choice, based only on the coefficients of the opponents to be obtained..

The empirical results have shown that the performance and the evolutionary stability of memory-one strategies rely on adaptability and not on extortion, and that memory-one strategies' performance is limited by their memory in cases where they interact with multiple opponents.

These results were mainly to investigate the behaviour of memory-one strategies and their limitations. A large data set which contained best responses in tournaments whilst including or not self interactions for $N = 2$ was generated and is archived in [45]. Their respective behaviours were investigated, and whether it was extortionate acts that made them the most favourable strategies. It was shown that it was not extortion but adaptability that allowed the strategies to gain the most from their interactions. In settings with self interactions there is some evidence that it is more likely to forgive after being tricked.

By specifically exploring the entire space of memory-one strategies to identify the best strategy for a variety of situations, this work adds to the literature casting doubt on the effectiveness of ZDs, highlights the importance of adaptability and provides a framework for the continued understanding of these important questions.

Chapter 6

Best Response Sequences in the Iterated Prisoner's Dilemma

The research reported in this Chapter has been carried out with:

Axerod-Python library version: 4.2.0
Associated data set: [?]

6.1 Introduction

In this Chapter best response strategies are explored in the form of static sequences of moves, in order to generate a large data set of best response sequences to a collection of opponents.

The data set is generated by considering best response sequences in finite IPD matches of 205 turns against 192 strategies available in the APL. These best response sequences are not obtained explicitly but instead are estimated heuristically using a genetic algorithm devised for this purpose.

The purpose of a large collection of best response sequences is to serve as training data in Chapter 7 which aims to train a recurrent neural network as an IPD strategy. In Chapter 7 the usage of the bespoke data set, which has been archived and made publicly available [?], is discussed in more details. This Chapter is structured as follow:

- section 6.2 formalises the use of sequences to express a player in a finite IPD match.
- section 6.3 describes the genetic algorithm used to estimate best response sequences.
- section 6.4 details the process of generating best response sequences to a collection of 192 strategies.

6.2 Iterated Prisoner Dilemma Strategies as Sequences

In a finite N round IPD match a player that does not react to their opponent can be defined by a sequence,

$$S \in \{C, D\}^n, \text{where } 1 \leq n \leq N. \tag{6.1}$$

Strategies that base their actions on sequences are already established in the literature [27], such as Periodic Player CD, Periodic Player DC, Periodic Player CCD and Periodic Player DDC [?, ?], or as referred to in [?] Cycler CD, Cycler DC, Cycler CCD and Cycler DDC. These are strategies that play a given sequences periodically, however, the strategies concerned with here play a given sequence only once, thus $n = N$.

As an example consider a match of 10 turns between the strategy $S = \{D, D, D, C, C, C, D, D, C, C\}$ and Cooperator. The match between the two strategies is captured by Table 6.1 where $U(s_1, s_2) \in \mathbb{R}^2$ is the average score per turn scored by strategies s_1 and s_2.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	$U(S, \text{Cooperator})$
S	D	D	D	C	C	C	D	D	C	C	4.0
Cooperator	C	C	C	C	C	C	C	C	C	C	1.5

Table 6.1: The interactions of a 10 turns match between $S = \{D, D, D, C, C, C, D, D, C, C\}$ and Cooperator as well as the average score per turn achieved by each strategy.

A sequence strategy S can play against strategies that react to the history, for example against Tit Tor Tat as demonstrated by Table 6.2,

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	$U(S, \text{Tit For Tat})$
S	D	D	D	C	C	C	D	D	C	C	2.2
Tit For Tat	C	D	D	D	C	C	C	D	D	C	2.2

Table 6.2: The interactions and average score per turn of a 10 turns match between $S = \{D, D, D, C, C, C, D, D, C, C\}$ and Tit For Tat.

and against stochastic strategies such as Random. Random cooperates with a probability of 0.5 at each turn, and thus the actions of the strategy are not deterministic. Random is a strategy that does react to the history, however, the actions of stochastic strategies that react to the history also differ between repetitions even when the history of the match is the same. Tables 6.3 and 6.4 both capture a match between S and a Random player. For a match of 10 turns Random has a total of 2^{10} possible plays. In order to capture several different plays of stochastic strategies computer seeding, further details of this are given in section 6.4.

As discussed in Chapters 1 and 5, a best response strategy is a strategy that achieves that most favourable outcome. Thus a best response sequence against a given opponent Q corresponds to a sequence S^* for which the average score per turn is maximised, as given in (6.2).

$$\max_{S^*} : U(S^*, Q) \tag{6.2}$$

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	$U(S, \text{Random})$
S	D	D	D	C	C	C	D	D	C	C	2.2
Random	D	D	C	C	D	C	D	C	C	D	2.2

Table 6.3: The interactions and average score per turn of a 10 turns match between $S = \{D, D, D, C, C, C, D, D, C, C\}$ and Random.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	$U(S, \text{Random})$
S	D	D	D	C	C	C	D	D	C	C	2.4
Random	C	D	D	C	C	C	D	D	C	C	1.9

Table 6.4: The interactions and average score per turn of a 10 turns match between $S = \{D, D, D, C, C, C, D, D, C, C\}$ and Random. The actions make by Random are different to that of Table 6.3.

Identifying best responses to some opponents can be a trivial problem. The optimal sequence against a Cooperator is in an all D sequence. In fact the sequence $\{\underbrace{D, \dots, D}_{N}\}$ is a best response against any sequence player whose plays are independent of the history. However, for some strategies identifying best responses is a complex problem.

Additionally, there are strategies that have multiple best response sequences. For instance the strategy Adaptive introduced in [**?**]. Adaptive opens by playing a sequence of $\{\underbrace{C, \dots, C}_{6}\}$ followed by a sequence of $\{\underbrace{D, \dots, D}_{5}\}$. The strategy then proceeds to play either C or D depending on which action had a higher total score for the strategy (the total score is recalculated at each turn). A sequence maximises its average score against Adaptive by locking the strategy into unconditional cooperations following its opening 11 turns sequence while the sequence defects. In order for cooperation to be the most favourable action for Adaptive the strategy needs to achieve two mutual cooperations at its opening sequence. That is because the score achieved by cooperating $2 \times 3 + 4 \times 0 = 6$ is greater than the score achieved by defecting $1 \times 5 = 5$. Any sequence which incorporates two cooperations in the first 6 turns and defects thereafter is a best response sequence to Adaptive. Thus, there can be 2^6 best response sequences to the strategy, for example S_1^* and S_2^*,

$$S_1^* = \{C, D, D, D, D, C, D, D, D, D, D, D, D, D, D\}$$

$$S_2^* = \{D, D, C, C, D, D, D, D, D, D, D, D, D, D, D\}$$

where $U(S_1^*, \text{Adaptive}) = U(S_2^*, \text{Adaptive}) = 3.4$.

Due to identifying best response sequences to some opponents being a complex problem, and moreover, multiple best response sequences existing for some opponents the best response sequences are not manually identified. Instead a genetic algorithm is used to estimate them. A background on genetic algorithms as well as the details of the specific genetic algorithm devised for this Chapter are presented in the following section.

6.3 Genetic Algorithm

A genetic algorithm (GA) is a heuristic inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms. As stated in [?] GAs encode a potential solution to a specific problem on a simple chromosome-like data structure, and apply recombination operators to these structures in such a way as to preserve critical information. GAs are often viewed as function optimisers, although the range of problems to which they have been applied is quite broad [?, ?, ?].

An implementation of a GA begins with a *population* P of potential solutions, a number of *generations* $G \in \mathbb{N}$ and a cut-off or *bottleneck* $b < |P|$. At each generation the algorithm scores and potentially removes each member of the population $p_i \in P$. This is done by using a mapping from a member of the population to an ordered set based on an evaluation function f, usually $f(p_i) \to \mathbb{R}$, and by only keeping the top b ranking members (or proportion of members) by score at the end of each generation. The rest of the member are discarded. By keeping the top ranked individuals critical information regarding the successful candidates is preserved and the population rebuilds on it by using a series of *crossovers* and *mutations*.

- During a crossover 2 members of the population are selected, and a new member is created based on combination of their "genes". The 2 selected members are commonly referred to as parents.
- Mutation is a probabilistic change that occurs to an individual, Mutation is commonly associated with a probability, denoted as p_m. p_m is the probability of a mutation happening, either to the individual or to each gene of the individual.

A diagrammatical representation of a generic GA is given in Figure 6.1.

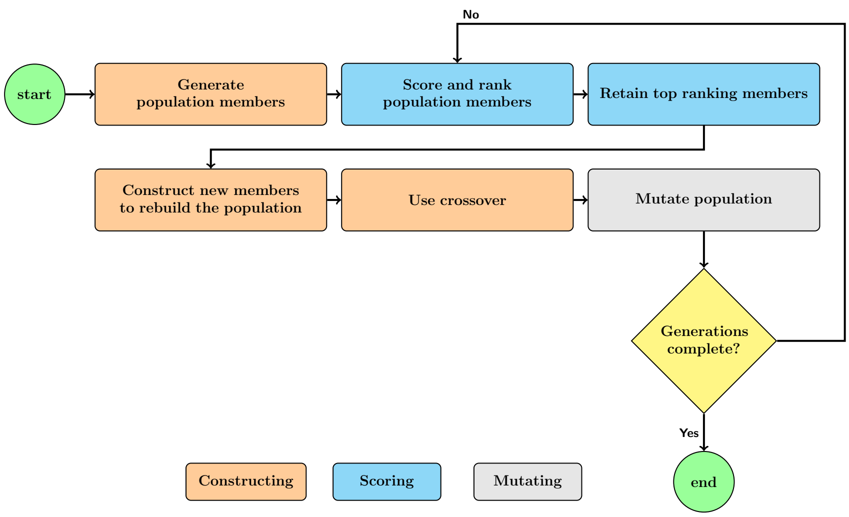


Figure 6.1: Generic flow diagram of a GA.

The purpose of a GA here is to estimate a best response sequence to a given opponent Q. Consequently, the members of the population correspond to sequences,

$$S_i \in P \text{ where } |S_i| = N$$

and the evaluation function corresponds to the average score per turn of a sequence against Q,

$$U(S_i, Q) \rightarrow \mathbb{R} \text{ for } S_i \in P.$$

More specifically, the exact GA used in this Chapter is given by Algorithm 4.

Algorithm 4: GA for estimating best response sequences to a given opponent Q.

```
Input: Q, N, b, p_m, G, K
Output: The populations at each generation and the members' scores
begin
    create initial population (Algorithm 5) of members S, where |S| = N and |P| = K
    while g_i < G do
        score each member based on U(S_i, Q) for S_i ∈ P
        sort population based on scores
        keep b top members
        while |P| < K do
            select 2 random members
            use members to create new member through crossover
            for gene in new member do
                mutate gene with probability p_m
            end
            add new member to population
        end
    end
end
```

The initial population is created using Algorithm 5.

Algorithm 5: Create initial population of individuals S

Input: K, N
Output: A population of size K.
begin
  set of cuts $\leftarrow$ K evenly spaced numbers over $[1, N]$
  for $c \in$ *set of cuts* **do**
    first new member $\leftarrow \{\underbrace{C, \dots, C}_{c}, \underbrace{D, \dots, D}_{N-c}\}$
    second new member $\leftarrow \{\underbrace{D, \dots, D}_{c}, \underbrace{C, \dots, C}_{N-c}\}$
    add both members to population
  end
end

Using a starting population of random guesses is a generally common approach in the GA literature [?]. However, there is efficiency in using non random starting populations [?, ?]. As discussed in section 6.2 the best response sequence to any strategy that does not react to the history is a Defector. Moreover, the best response sequence to several strategies include sequences mainly dominated by cooperation expect in the last turns. These are sequences that have been incorporated in the initial population. More specifically, Algorithm 5 consider all the possible combinations of:

$$\bullet\{\underbrace{C,\dots,C}_{c},\underbrace{D,\dots,D}_{N-c}\} \text{ for } c \in \text{evenly spaced numbers over } [1,N] \text{ and}$$

$$\bullet\{\underbrace{D,\dots,D}_{c},\underbrace{C,\dots,C}_{N-c}\} \text{ for } c \in \text{evenly spaced numbers over } [1,N].$$

The GA of Algorithm 4 has been implemented in the programming language Python and it has been organised into a open source package called `sequence_sensei` available at. The properties of creating an initial population, crossover and mutation have been implemented as individual functions. The implementation of Algorithm 5 in the package is given by Figure 6.2.

```
import numpy as np
def get_initial_population(half_size_of_population, sequence_length):
    """
    Generates an initial population of sequences. Note that the length
    of the population which is being generated is 2 * half_size_of_population.
    """
    cuts = np.linspace(1, sequence_length, half_size_of_population, dtype=int)
    sequences = []
    for cut in cuts:
        sequences.append(
            [1 for _ in range(cut)] + [0 for _ in range(sequence_length - cut)]
        )
        sequences.append(
            [0 for _ in range(cut)] + [1 for _ in range(sequence_length - cut)]
        )

    return sequences
```

Figure 6.2: Source code for the function `get_initial_population` implemented in `sequence_sensei` which is used to create an initial population of a given size.

Figure 6.3 gives an example of creating an initial population using the package. Note that the sequences are of 0s and 1s and not IPD actions. The APL project can map binary number to actions such that $0 \to D$ and $1 \to C$. This is also demonstrated in Figure 6.3.

In the GA, as given by Algorithm 4, the crossover occurs by randomly selecting two member of the population, while $|P| < K$, and randomly selecting a crossover point. Note that the crossover point is smaller than N. The new member initially inherits the genes to the left of the crossover point of the first parent, and to the right of the crossover point of the second parent.

For instance, given two member of the population $S_1 = \{C, C, C, C, C, C, C, C, C, C\}$ and $S_2 = \{C, D, C, D, C, D, C, D, C, D\}$ and given that the crossover point is 4, this gives a new member S_3:

$$S_3 = \{\underbrace{C, C, C, C}_{\text{from } S_1}, \underbrace{C, D, C, D, C, D}_{\text{from } S_2}\}.$$

The implementation of crossover in `sequence_sensei` is given by Figure 6.4, and Figure 6.5 demonstrates the usage of the `crossover` function to crossover S_1 and S_2.

```
>>> import sequence_sensei as ss
>>> import numpy as np

>>> initial_population = ss.get_initial_population(
...     half_size_of_population=5, sequence_length=8
... )
>>> np.matrix(initial_population)
matrix([[1, 0, 0, 0, 0, 0, 0, 0],
        [0, 1, 1, 1, 1, 1, 1, 1],
        [1, 1, 0, 0, 0, 0, 0, 0],
        [0, 0, 1, 1, 1, 1, 1, 1],
        [1, 1, 1, 1, 0, 0, 0, 0],
        [0, 0, 0, 0, 1, 1, 1, 1],
        [1, 1, 1, 1, 1, 1, 0, 0],
        [0, 0, 0, 0, 0, 0, 1, 1],
        [1, 1, 1, 1, 1, 1, 1, 1],
        [0, 0, 0, 0, 0, 0, 0, 0]])

>>> import axelrod as axl
>>> np.matrix([[axl.Action(gene) for gene in member] for member in initial_population])
matrix([[C, D, D, D, D, D, D, D],
        [D, C, C, C, C, C, C, C],
        [C, C, D, D, D, D, D, D],
        [D, D, C, C, C, C, C, C],
        [C, C, C, C, D, D, D, D],
        [D, D, D, D, C, C, C, C],
        [C, C, C, C, C, C, D, D],
        [D, D, D, D, D, D, C, C],
        [C, C, C, C, C, C, C, C],
        [D, D, D, D, D, D, D, D]], dtype=object)
```

Figure 6.3: Example of using `get_initial_population` to generate a population of $s = 10$ and $N = 8$.

```
import random

def crossover(sequence_one, sequence_two):
    sequence_length = len(sequence_one)
    crossover_point = random.randint(0, sequence_length)

    return sequence_one[:crossover_point] + sequence_two[crossover_point:]
```

Figure 6.4: Source code of the `crossover` function.

```
>>> import random
>>> import sequence_sensei as ss

>>> turns = 10
>>> s_one = [1 for _ in range(turns)]
>>> s_two = [i % 2 for i in range(turns)]

>>> random.seed(0)
>>> new_member = ss.crossover(s_one, s_two)
>>> new_member
[1, 1, 1, 1, 1, 1, 0, 1, 0, 1]
```

Figure 6.5: An example of using `crossover` function to crossover S_1 and S_2

Following the crossover between two members, a mutation is applied to new member before it is added to the population. Mutation has been implemented as a given probability p_m that each gene of the new member is flipped. A total of N random numbers between $[0, 1]$ are sampled. If the sampled probability at time i in less than p_m then the i^{th} gene of the individual is flipped, as demonstrated by Figure 6.6.

$$S_3 = \{C, C, C, C, C, C, D, C, D, C\}$$

$$\begin{array}{cccccccccc} 0.134\downarrow & 0.847\downarrow & 0.764\downarrow & 0.255\downarrow & 0.495\downarrow & 0.449\downarrow & 0.652\downarrow & 0.789\downarrow & 0.094\downarrow & 0.028\downarrow \end{array}$$

$$\text{mutated } S_3 = \{C, C, C, C, C, C, D, C, D, D\}$$

Figure 6.6: Mutation example of S_3.

The implementation of mutation in `sequence_sensei` is given by Figure 6.7, and an example of mutating S_3 using the function is given in Figure 6.8.

```
def mutation(gene, mutation_probability):
    if random.random() < mutation_probability:
        return abs(gene - 1)
    return gene
```

Figure 6.7: Source code of the `mutation` function.

```
>>> new_member = [1, 1, 1, 1, 1, 1, 0, 1, 0, 1]

>>> random.seed(1)
>>> [ss.mutation(gene, mutation_probability=0.05) for gene in new_member]
[1, 1, 1, 1, 1, 1, 0, 1, 0, 0]
```

Figure 6.8: An example of using the `mutation` function to mutate S_3.

The main function implemented in `sequence_sensei` for performing a GA is the `evolved` function. The function has several input arguments which correspond to the inputs of Algorithm 4. In the following section the `evolved` function is used to run several trials and estimate best response sequence. The parameters' values for each run will be presented there. Moreover, the details of the best response sequence collection against the 192 strategies are presented in the following section.

6.4 Data Collection

The data set generated in this Chapter was created using the GA of Algorithm 4 and the APL project. The GA of Algorithm 4 estimates the best response sequence for a given opponent, and in order to generate a collection of best responses a list of opponents is obtained from APL. More specifically 192 strategies are used in this Chapter. These can be found in the Appendix.

The APL project is also used to calculate the average score per turn, $U(S, Q)$, player S can achieve against an opponent Q. The project contains a specific player class that can simulate the play of any given sequence of Cs and Ds. The player class is called Cycler and it takes as an input argument a series of actions as a string. An example of creating and using such a player in a match is given by Figure 6.9. The average score per turn is obtained using an in built method of APL once a match has been simulated.

```
>>> import axelrod as axl

>>> players = [axl.Cycler('DDDCCCDDCC'), axl.Cooperator()]
>>> match = axl.Match(players, turns=10)
>>> match.play()
[(D, C), (D, C), (D, C), (C, C), (C, C), (C, C), (D, C), (D, C), (C, C), (C, C)]

>>> match.final_score_per_turn()
(4.0, 1.5)

>>> players = [axl.Cycler('DDDCCCDDCC'), axl.TitForTat()]
>>> match = axl.Match(players, turns=10)
>>> match.play()
[(D, C), (D, D), (D, D), (C, D), (C, C), (C, C), (D, C), (D, D), (C, D), (C, C)]

>>> match.final_score_per_turn()
(2.2, 2.2)
```

Figure 6.9: Simulating a match between Cycler and Cooperator and Cycler and Tit For Tat. The class Cycler takes a given sequence as an input argument in a string format ('DDDCCCDDCC'). Once a match has been simulate with the `play` method the average score per turn is obtained using the `final_score_per_turn` method.

From the 192 strategies, 62 are stochastic and 130 are deterministic. In section 6.2 it was explained that the outcome of a match between two deterministic strategies is always the same. In comparison, the outcome of a match with a stochastic strategy can differ, because the actions of a stochastic opponent are not deterministic. The actions of a stochastic opponent can be repeated by using *computer seeding* for seeding the pseudo random number generator (PRNG) that creates the parameters that define what moves the strategy will take. Seeds are set before generating a random number, and if the same seed is used on initialisation then the random output remains the same. Thus, as long as a match is seeded the behaviour of a stochastic strategy can be reproduced, and different seeds lead to different plays of stochastic strategies.

Consider the match between Random and $S = \{D, D, D, C, C, C, D, D, C, C\}$ presented in section 6.2. Random has a total of 2^{10} possible plays for the given match. By using different computer seeds a number of these plays can be simulated as shown in Figure 6.10.

A total of 10 different plays are captured for each of the stochastic strategies of this Chapter. Thus, a total of 10 different seeds are used for each stochastic strategy. The data collection process of best response sequences in more details is given by Figure 6.11.

From the collection of opponents a strategy is selected at each trial. If the strategy is deterministic a set of GAs with different parameters values are performed for 2000 generations. The summary of the GA trials which contain information for each generation is exported to a single csv file, then the next opponent is selected. If the opponent is stochastic the above process is repeated 10 times. Each time the stochastic strategy is accompanied by a different seed value which is used to initialise the pseudo random generator. For each combination of stochastic

```
>>> players = [axl.Cycler('DDDCCCDDCC'), axl.Random()]
>>> for seed in range(5):
...     axl.seed(seed)
...     match = axl.Match(players, turns=10)
...     actions = match.play()
...     print(actions, match.final_score_per_turn())
...     print("================================================================================")
[(D, D), (D, D), (D, C), (C, C), (C, D), (C, C), (D, D), (D, C), (C, C), (C, D)] (2.2, 2.2)
================================================================================
[(D, C), (D, D), (D, D), (C, C), (C, C), (C, C), (D, D), (D, D), (C, C), (C, C)] (2.4, 1.9)
================================================================================
[(D, D), (D, D), (D, C), (C, C), (C, D), (C, D), (D, D), (D, C), (C, D), (C, D)] (1.6, 2.6)
================================================================================
[(D, C), (D, D), (D, C), (C, D), (C, D), (C, C), (D, C), (D, D), (C, C), (C, C)] (2.6, 2.1)
================================================================================
[(D, C), (D, C), (D, C), (C, C), (C, C), (C, C), (D, D), (D, D), (C, D), (C, C)] (2.9, 1.9)
================================================================================
```

Figure 6.10: Example code of using seeding to generate different plays of Random. The value of seed changes to $\{0, 1, 2, 3, 4, 5\}$ and the seed is set with the command axl.seed(seed) before simulating the game. This initialises the pseudo random generator that define what moves Random will take. The above code snipped will always have the same output each time it is repeated.

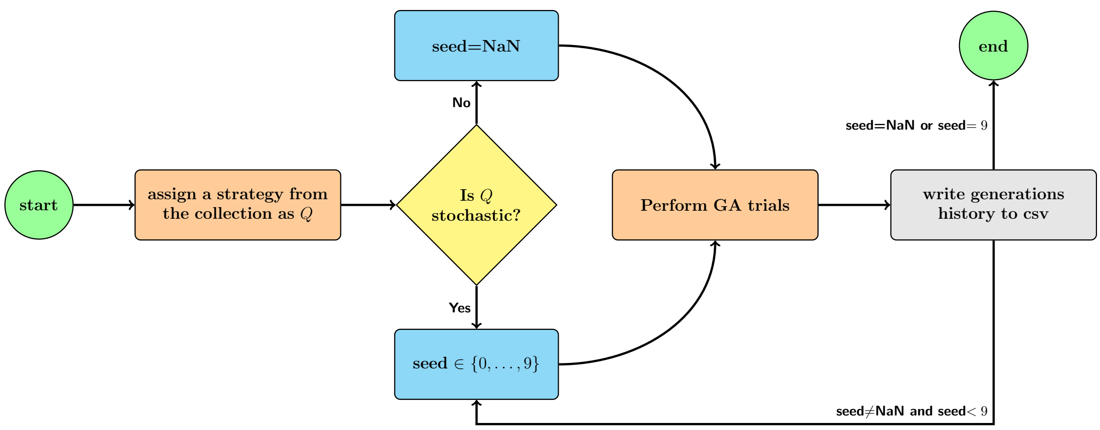


Figure 6.11: Diagrammatical representation of the best response sequences collection process.

opponent and seed the GAs summary is exported to a csv file.

For each opponent, or opponent-seed combination a total of 18 GAs are performed. The different values for each parameter are given by Table 6.5.

Parameter	Explanation	Values
N	number of turns	205
G	number of generations	2000
b	bottleneck	10, 20
K	size of a population	20, 30, 40
p_m	probability of gene mutating	0.01, 0.05, 0.1

Table 6.5: The parameters of the GA. The GA is performed a total of 18 times for each opponent. More specifically, it is performed for each possible combination of the parameters' values.

All the best response sequences that are generated in this Chapter are best response sequences of 205 turns. Moreover, they are best responses not to 192 strategies, but to a total of $(130 + 62 \times 10)$ 750 opponents. Thus, a total of 750 trials of Figure 6.11 have been performed.

An example of an exported summary for the deterministic opponent Alternator is given by Table 6.6.

	opponent	seed	b	p_m	$K/2$	g_i	index	score	gene 0	gene 1	...	gene 202	gene 203	gene 204
0	Alternator	NaN	10	0.01	10	0	19	3.009756	0	0	...	0	0	0
1	Alternator	NaN	10	0.01	10	0	0	3.000000	1	0	...	0	0	0
2	Alternator	NaN	10	0.01	10	0	2	2.839024	1	1	...	0	0	0
3	Alternator	NaN	10	0.01	10	0	17	2.839024	0	0	...	1	1	1
4	Alternator	NaN	10	0.01	10	0	4	2.673171	1	1	...	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1080535	Alternator	NaN	20	0.10	20	2000	29	2.775610	1	0	...	0	0	0
1080536	Alternator	NaN	20	0.10	20	2000	24	2.770732	0	0	...	0	0	0
1080537	Alternator	NaN	20	0.10	20	2000	31	2.770732	1	0	...	0	0	0
1080538	Alternator	NaN	20	0.10	20	2000	32	2.770732	0	0	...	0	0	1
1080539	Alternator	NaN	20	0.10	20	2000	34	2.756098	0	0	...	0	0	0

Table 6.6: An exampled of an exported summary. The specific output is for the opponent Alternator. Alternator is a deterministic strategy, consequently, the value of seed in NaN. The values of the different GA parameters are recorded in the summary, as well as the details of each member of each generation. The sequences' genes were recorded in 0 and 1, where $0 \to D$ and $1 \to C$. The best responses sequences are the individuals that have the maximum score at $g_i = 2000$.

For a stochastic strategy there are a total of 9 exported summaries. An example of an exported summary for Champion with seed=9 is given by Table 6.7.

The best response sequences are collected from the last generation of the exported summaries.

In section 6.2 it was stated that the best response to a strategy that does not react to the history is to always defect. Such strategies include Cooperator, Defector, Alternator and the family of the Cycler strategies. Algorithm 4 successfully identified their best responses, as shown in Table 6.8.

	opponent	seed	b	p_m	$K/2$	g_i	index	score	gene 0	gene 1	…	gene 202	gene 203	gene 204
0	Champion	9	10	0.01	10	0	10	3.712195	1	1	…	0	0	0
1	Champion	9	10	0.01	10	0	12	3.663415	1	1	…	0	0	0
2	Champion	9	10	0.01	10	0	8	3.585366	1	1	…	0	0	0
3	Champion	9	10	0.01	10	0	14	3.448780	1	1	…	0	0	0
4	Champion	9	10	0.01	10	0	6	3.312195	1	1	…	0	0	0
…	…	…	…	…	…	…	…	…	…	…	…	…	…	
1080535	Champion	9	20	0.10	20	2000	25	3.634146	0	0	…	1	1	0
1080536	Champion	9	20	0.10	20	2000	34	3.629268	1	1	…	0	0	1
1080537	Champion	9	20	0.10	20	2000	31	3.604878	1	1	…	1	0	1
1080538	Champion	9	20	0.10	20	2000	21	3.443902	0	0	…	1	0	0
1080539	Champion	9	20	0.10	20	2000	20	3.351220	1	0	…	0	1	0

Table 6.7: An exampled of an exported summary for a stochastic strategy. The column seed does not have a value of NaN anymore but has captured the seed that was used to generate the specific play of the stochastic opponent. The members' genes are also recorded for each generation. Note that $0 \to D$ and $1 \to C$.

opponent	score	gene 0	gene 1	gene 2	gene 3	gene 4	gene 5	…	gene 200	gene 201	gene 202	gene 203	gene 204
Cooperator	5.000	0	0	0	0	0	0	…	0	0	0	0	0
Alternator	3.010	0	0	0	0	0	0	…	0	0	0	0	0
Cycler CCCCCD	4.337	0	0	0	0	0	0	…	0	0	0	0	0
Cycler CCCD	4.005	0	0	0	0	0	0	…	0	0	0	0	0
Cycler CCCDCD	3.673	0	0	0	0	0	0	…	0	0	0	0	0
Cycler CCD	3.673	0	0	0	0	0	0	…	0	0	0	0	0
Cycler DC	2.990	0	0	0	0	0	0	…	0	0	0	0	0
Cycler DDC	2.327	0	0	0	0	0	0	…	0	0	0	0	0
Defector	1.000	0	0	0	0	0	0	…	0	0	0	0	0

Table 6.8: Best response sequences to a number of opponents that do not react to the history of the match. The best response to such strategies are to always defect. This was successfully captured by the best sequence collection process of this Chapter.

The best response sequence to Tit For Tat and Grudger is the sequence that cooperates until before the last turn. That way the sequence receives the payoff for mutual cooperation for $N-1$ turns, and then betrays its opponent in the last turn to maximise its score, receiving a payoff of T. It only defects on the last turn because then it can not be punished. The sequence is also the best response to the strategy Hard Tit For Tat. The strategy is a variant of Tit For Tat that uses that uses a longer history for retaliation. The best response sequences were captured for all three strategies, Table 6.9.

opponent	score	gene 0	gene 1	gene 2	gene 3	gene 4	gene 5	...	gene 200	gene 201	gene 202	gene 203	gene 204
Tit For Tat	3.01	1	1	1	1	1	1	...	1	1	1	1	0
Grudger	3.01	1	1	1	1	1	1	...	1	1	1	1	0
Hard Tit For Tat	3.01	1	1	1	1	1	1	...	1	1	1	1	0

Table 6.9: Best response sequences to strategies Tit For Tat, Grudger and Hard Tit For Tat.

There are more sophisticated yet still established best response sequences. These include the best response to TF1 [70]. In [70] the strategy TF1 was trained using a 16 state finite state machine in a Moran process setting. The TF1 strategy developed a hand shake mechanism that allowed it to identify strategies that play like itself. Once two copies of TF1 identify each other they go into mutually cooperations until the end of their interactions. The best response to the strategy is the a sequence that performs the handshake, and goes into mutual cooperations until before the final turn. This best response sequence was also captured by the data collection process. The handshake is performed in the opening three turns, and it is the sequence $\{C, C, D\}$ as shown in Table 6.10.

opponent	score	gene 0	gene 1	gene 2	gene 3	gene 4	gene 5	...	gene 200	gene 201	gene 202	gene 203	gene 204	
TF1	3.0	1	1	0	1	1	1		...	1	1	1	1	0

Table 6.10: Best response sequence to TF1 introduced in [70]. The strategy performs a handshake in the first three moves. The hand shake is the sequence CDC. If the opponent plays that same then the strategies go into mutual cooperation.

6.4.1 Parallelisation and stochastic results

The data collection process of this Chapter was carried out using parallel processing. In parallel processing many calculations or executions of tasks are carried out simultaneously. In the case of the data collection here the tasks were scoring members of the population. More specifically, at most 10 members of the population were being scored at the same time.

Parallel programming was executed using *multi threading* [**?**]. Threads are "light-weight" processes, a unit of execution within a process. Threads are designed to have shared memory and can manipulate global variables of main thread. Scoring a member of the population corresponded to a single task which was executed on a single thread. For the stochastic opponents the task included seeding/setting the PRNG state before simulating the match. Seeding at the time of generating the data collection was not implemented in a safe thread way.

The data collection was implemented in a way that each thread was setting the global PRNG state. That was then shared across all the the threads without synchronisation. Since the threads are running in parallel, at the same time, and their access to this global PRNG is not synchronised between them, errors occur. In the case of the stochastic opponents this means that there are given instances that are not reproducible and for those instance the simulated behaviour does not reflect the opponent's behaviour for its given seed.

There are several ways that this error can be corrected. There is a thread safe way of implementing seeding which involves giving each thread its own local PRNG. Then there is no longer any state that's shared by multiple threads without synchronisation.

For the stochastic opponents 34% of the collected sequences are not reproducible. Recalling that the aim of this Chapter is to generate a collection of well performing sequences in the IPD, so that they can be used as the training data for the purposes of Chapter 7. A 34% percentage of non reproducible sequences is a reasonable ratio which also ensures more variability to the data for training.

6.4.2 The collection of best response sequences

The collections process was performed for 750 trials, and a total of 18 GAs were performed for each trial. The best response sequences are the sequences with the highest average score per turn in the final generation regardless the GA.

In order to understand whether the algorithms reached convergence over the 2000 generations, the highest score in a population over the generations for four different strategies is given in Figure 6.12.

There are trials for which the algorithms never converged to the best response sequences. There are two reason that as to why this happened:

- The bottleneck was equal to the population size. A value $K/2 = 10$ while $b = 20$ results to no new members being added to the population. These trials would have reached convergence only if the best response was in the initial generation.
- The mutation probability is too high. The earliest converged GA trials are the trials for which $p_m = 0.01$. As p_m is the probability that each gene of the new member is being flipped, higher values could potentially add to much variation to the new members. This could lead to the new members losing critical information they inherited from their parents.

Nevertheless, a sequence that has not converged is still useful. Even though it is not the highest scoring member, it is a sequence that was not arbitrarily generated but has some critical information regarding playing against given opponents.

There are several trials that have managed to reach convergence, and they did so in less than 200 generations. This is potentially the effect of a non random initial population. Figure 6.13 shows the highest score in a population over the generations for the four strategies of Figure 6.12 but only up to $g_i = 500$. All trials with a $p_m = 0.01$ reach convergence within the first 200 generations. The trial which reaches convergence first (in the four demonstrated cases) is the trial with the parameter values of $K/2 = 15, b = 10$ and $p_m = 0.1$.

There are a total of 130 deterministic strategies in the collection of opponents and best response sequences were estimated for each. Several of the known best response sequences have been manually checked and they have been successfully estimated by the algorithm. These

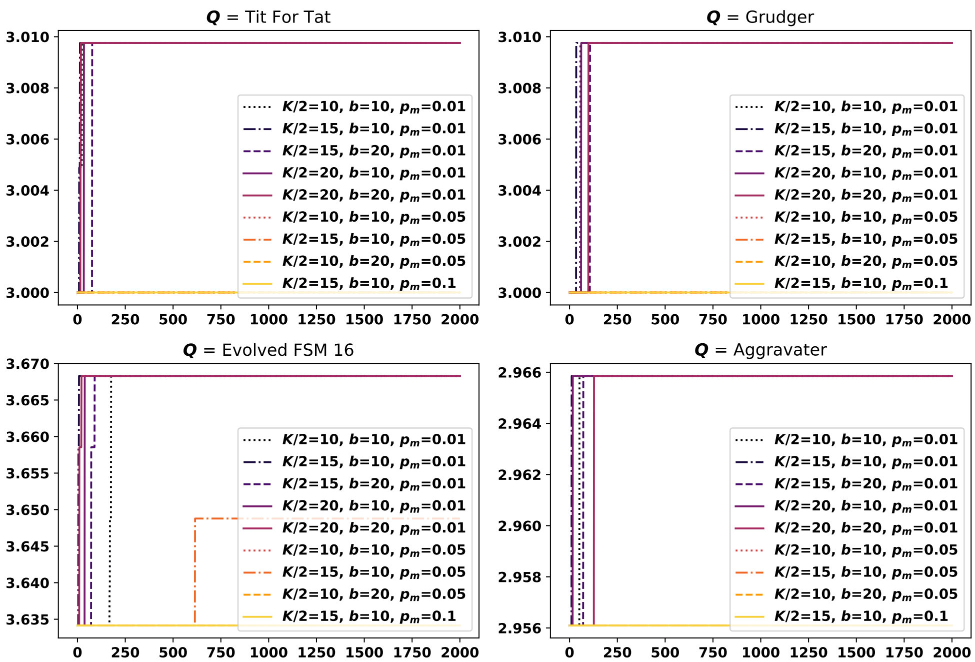


Figure 6.12: The highest score in a population over the generations for Tit For Tat, Grudger, FSM 16 and Aggravater. The selected trials capture the results of all the 18 trials for the given set of opponents.

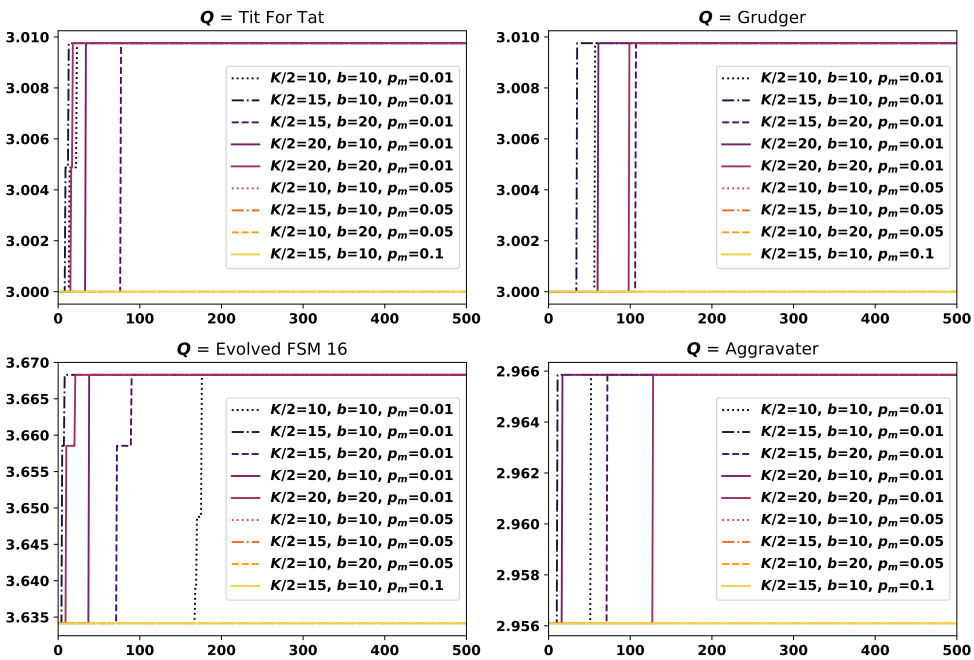


Figure 6.13: The highest score in a population over the generations for Tit For Tat, Grudger, FSM 16 and Aggravater up to $g_i = 500$.

include best response sequences to Tit For Tat, Grudger, Alternator, Pavlov and the Cycler strategies.

In section 6.2 is was explained that strategies can have more than a single best response sequence. In the case of the strategy Adaptive any strategy that cooperated twice in the opening 6 moves and defected thereafter is a best response. The data collection has managed to successfully identify multiple best responses to Adaptive, these are given by Table 6.11. Adaptive is not the only deterministic strategy with multiple best response sequences. More specifically, for the 130 deterministic opponents a total of 2949 best response sequences were collected.

index	gene 0	gene 1	gene 2	gene 3	gene 4	gene 5	gene 6	...	gene 202	gene 203	gene 204
0	1	0	0	0	0	1	0	...	0	0	0
1	0	0	1	1	0	0	0	...	0	0	0
2	1	1	0	0	0	0	0	...	0	0	0

Table 6.11: Best responses sequences estimated by the data collection process. Note that 0 corresponds to defection and 1 to cooperation.

An interesting questions that arises is: how diverse are the set of best response sequences? Out of the 2949 sequences 2309 are unique. A graphical representation of these sequences is given by Figure 6.16a. The two distinct colours represent genes of C and D. Overall, it can be seen that there is diversity in the best response sequences, and they are not just long sequences of either C or D. A common trend appears to be a series of defections at the last turns. In a finite IPD this is to expected. As it was mentioned in Chapter 4, as the likelihood of a match ending in the following turn increases the effectiveness of defecting.

A total of 2309 sequences have been estimated for the stochastic opponents. From these sequences, 66% did indeed play against the correct seeded opponent and achieved their respective scores at the final generation.

For instance for the strategy - seed combination of Champion - seed= 9 the highest score achieved by a member of the population over the generations is given by Figure 6.14. There is variation in the highest score occurring over the generations with several increasing and decreasing peeks. However, for most of the generations the highest score appears to be between $3.80 - 3.82$. The best response sequence retrieved by the data collection scored 3.82 against Champion, and it reflected the score of the sequence against Champion - seed= 9.

There are stochastic opponents for which more variation occurred over the generations. An example of that is Random. The highest score of the population in a single GA trial, for opponent - seed combination Random - seed= 1, is given by Figure 6.15. In the case of Random - seed= 1 the score of the best response sequence that was collected was not the actual score the sequence scores against Random - seed= 1.

From the 2309 sequences against stochastic strategies 2309 are unique. A graphical representation of 1000 of those sequences are given by Figure 6.16b. Similar to the results of the best response sequences against deterministic opponents the sequences are diverse.

In summary, from the list of 192 strategies examined in this Chapter, 750 different opponent

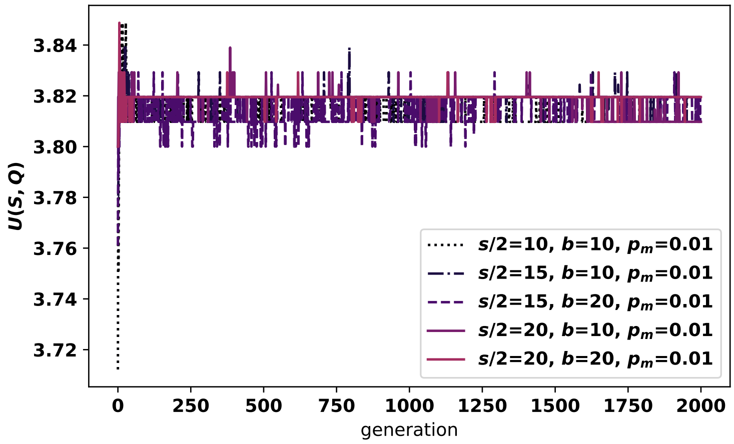


Figure 6.14: The maximum score a sequence achieved against Champion with seed 9 over the generations. Each line represents a different GA trial. Only the GAs with $p_m = 0.01$ have been included.

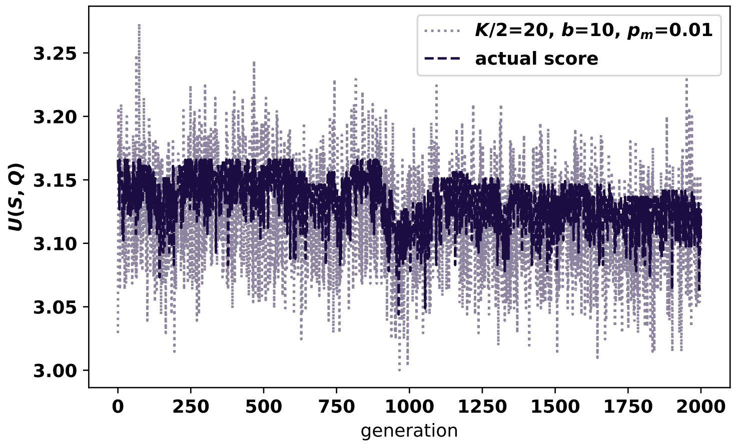
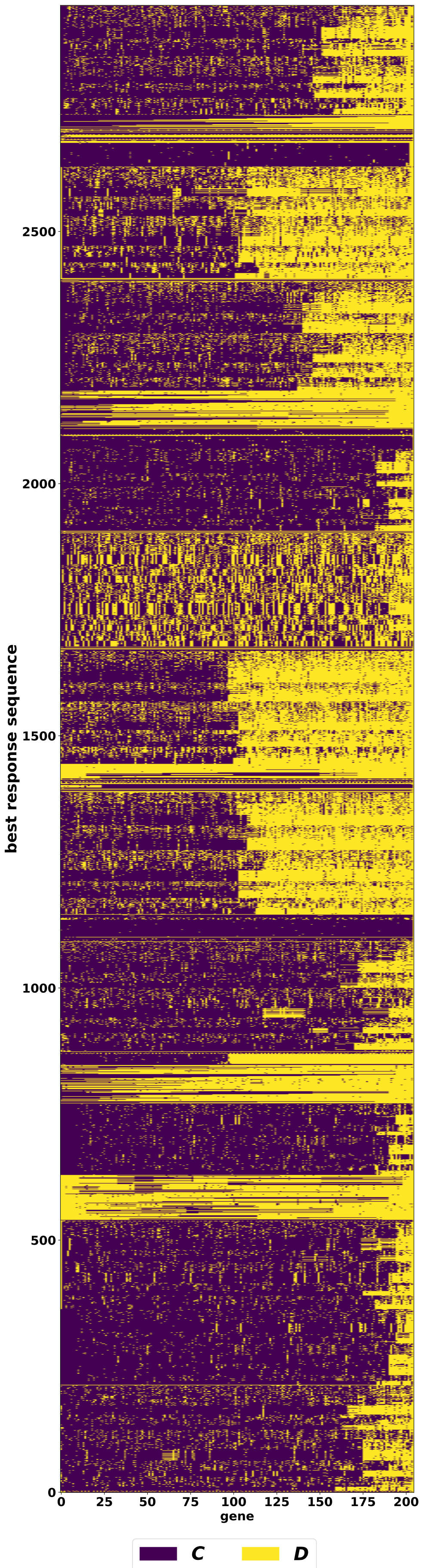
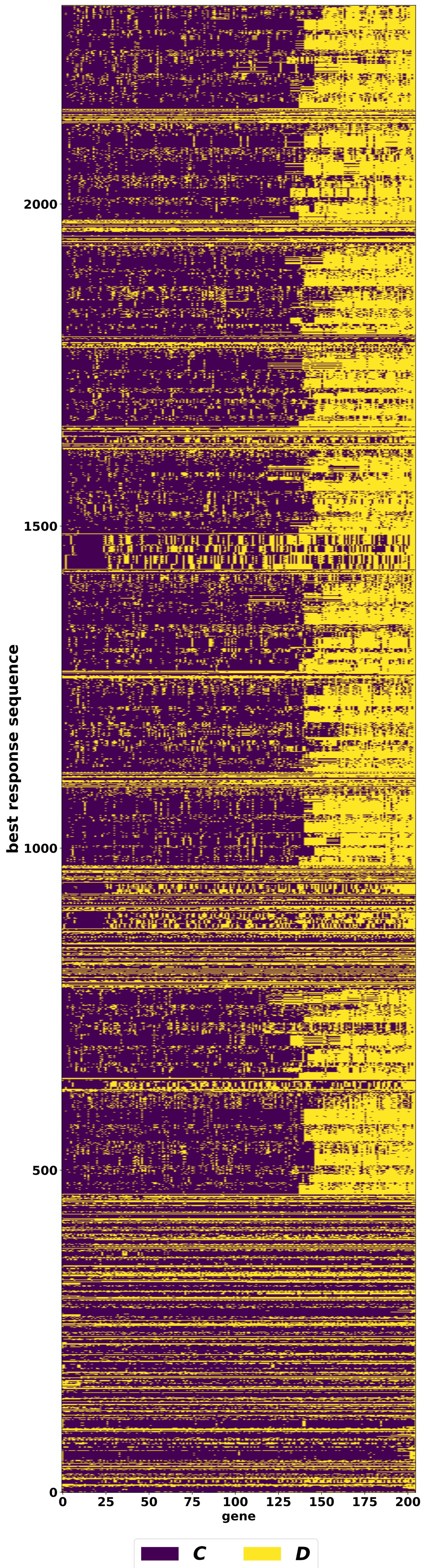


Figure 6.15: The maximum score a sequence achieved against Random - seed= 1 over the generations.



(a) A graphical representation of best 2949 response sequence. These have been estimated using deterministic opponents.



(b) A graphical representation of 2309 response sequence. These have been estimated using stochastic opponents.

instances were simulated and a total of 5258 best response sequences of length 205 were retrieved. The choice of 205 turns will be explained in the following Chapter. The best response sequences have been archived and made available at [?].

6.5 Chapter Summary

This Chapter has explored the concept of best responses in the IPD game in the form of static sequences of moves. It introduced an evolutionary algorithm, Algorithm 4, which can successfully identify best response sequences.

The algorithm was executed to estimate best response sequences to the majority of opponents listed in the APL. More specifically, a total of 192 opponents from APL were used. Several of the strategies in the project are stochastic and computer seeded versions of these strategies were used to explore their different behaviours. From the list of 192 opponents a total of 750 different behaviours were simulated.

For the 130 deterministic strategies a total of 2949 sequences, from which 2949 were unique, were estimated. These sequences were not just a set of trivial sequences of either C or D. A common trait in the best response sequences appeared to be a series of defection closer to the final turns. For the seeded versions of the 62 stochastic opponents the best response sequences are not guaranteed to have been captured due to issues related to PRNGs and multi threading. Nevertheless, a total of 2309 sequences from which 2309 are unique were collected. Similar, these sequences are more diverse than just a series of a single action. The 5258 sequences that were collected have been archived and are available at [?].

The main purpose of this Chapter has been to generate the bespoke data set, which contains a large number of unique and diverse sequences, so it can be used as training data set for Chapter 7.

Chapter 7

Training a recurrent neural network player

Chapter 8

Conclusions

This Chapter serves to summarise the work and contributions of this thesis. Each chapter contains a detailed chapter summary section, and so the summary here will be brief.

8.1 Research Summary

The fundamental research question of this thesis has been the same question that has troubled the scientific community since the formulation of the Iterated Prisoner's Dilemma in 1950. Namely, what is the optimal behaviour an Iterated Prisoner's Dilemma strategy should adapt as a response to different environments.

Chapter 1 introduced the Iterated Prisoner's Dilemma, carried out an initial literature review and outlined the research tasks of this thesis. A more detailed literature review was presented in Chapter 2. The literature reviewed in Chapter 2 was divided into different research topics. These included evolutionary dynamics, intelligently designed strategies, structured strategies and training and software that has been developed specifically for the game.

In Chapter 3 a bespoke research software tool called `arcas` was developed and used to collect a data set of articles' metadata on the Iterate Prisoner's Dilemma. A topic modelling technique, called Latent Dirichlet Allocation, was applied to the abstracts of these articles and allocated them into five different research topics. These were human subject research, biological studies, strategies, evolutionary dynamics on networks and modelling problems as a Prisoner's Dilemma.

The bespoke data set was further analysed to explore whether the academic field of the Prisoner's Dilemma is cooperative and whether there is influence between the authors. It was shown that the field of the Iterated Prisoner's Dilemma is a collaborative field, yet it is not necessarily more collaborative than other fields. Many authors tend to collaborate with authors from one community and are not involved in multiple communities. The collaborativeness was also explored over time, and it was shown that since the first publications authors tended to write only with a single community and that it is not an effect of a specific time period. Exploring the influence of authors in the field based on the specific publications showed that authors do not gain much influence, and the only ones with influence are the ones connected to a "main" group.

Chapter 4 examined the performance of a collection of 195 strategies in the largest collection of computer tournaments in the field. The results across the 45686 tournaments of various tournaments types deduced that there was not a single strategy that performs well in diverse Iterated Prisoner's Dilemma scenarios. The later parts of the Chapter analysed and extracted the salient features of the best performing strategies across the various tournament types and established that there are several properties that heavily influence the best performing strategies. There were: be nice, be provocable and forgiving, be a little envious, be clever, and adapt to the environment.

Chapter 5 investigated best response memory-one strategies with a theory of mind. It presented several theoretical and numerical results. More specifically, it proved that:

- The utility of a memory-one strategy against a set of memory-one opponents can be written as a sum of ratios of quadratic forms.
- There is a compact way of identifying a memory-one best response to a group of opponents through a search over a discrete set.
- There is a condition for which in an environment of memory-one opponents defection is the stable choice, based only on the coefficients of the opponents.

Additionally, the numerical results of Chapter 5 reinforced established result of the literature. Namely, they showed that extortionate play is not always optimal by showing that optimal play is often not extortionate, and that memory-one strategies suffer from their limited memory in multi agent interactions and can be out performed by optimised strategies with longer memory.

Chapter 7

8.2 Contributions

This thesis has made novel contributions across various themes. Numerous research software packages have been implemented as part of this thesis. These packages have been written following the highest standards of software development, and have been made available so that other researchers can contribute to and use them. The packages include `arcas` a tool designed for scraping academic articles from various APIs and `sequences-sensei` a project for performing genetic algorithms. Additionally, software contributions were made to well established Python libraries such as SymPy [81] and Axelrod-Python Library [5].

A total of six accompanying data sets have been generated as a result of this thesis, which include one of the largest collection of Iterated Prisoner's Dilemma tournaments known to the field:

1. Articles' meta data on the Prisoner's Dilemma [**?**].
2. Articles' meta data on the Price of Anarchy [**?**].
3. Articles' meta data on Auction Games [**?**].
4. Raw data for: "Stability of defection, optimisation of strategies and the limits of memory in the Prisoner's Dilemma" [45].

5. A data set of 45686 Iterated Prisoner's Dilemma tournaments' results [?]
6. Best response sequences in the Prisoner's Dilemma.

These have been archived and made available via Zenodo, and likewise, are available to other researchers. They can be used to conduct further analysis and provide new insights to the field.

A total of four scientific manuscripts presenting the methodology, analysis and results of this thesis have been prepared and are currently under submission to respective academic journals. The title of these manuscripts are:

1. A bibliometric study of research topics, collaboration and influence in the field of the Iterated Prisoner's Dilemma.
2. A meta analysis of tournaments and an evaluation of performance in the Iterated Prisoner's Dilemma.
3. Recognising and evaluating the effectiveness of extortion in the Iterated Prisoner's Dilemma [69].
4. A theory of mind: Best responses to memory-one strategies. The limitations of extortion and restricted memory

These manuscripts have been uploaded on the pre print server arXiv and are currently available and accessible to the scientific community.

Designing new strategies is an important type of research for the field. This thesis has introduced an abundant number of properties of successful strategies which can be of interest to researchers designing a new strategy for new environments, or just to understand the reasons behind some strategies being better than others. Complementing this, a new mathematical framework has been developed for the better understanding of memory-one strategies and an an initial understanding of using recurrent neural networks to train Iterate Prisoner's Dilemma strategies has been presented.

Finally, this thesis has contributed to the continuous understanding of the emergence of cooperation by providing a condition for which cooperation can not occur in memory-one environments. It has also has proven that constrained quadratic ratio optimisation problems that are non concave can be solved explicitly by using resultant theory.

8.3 Complementary Research

The results of this thesis are not the only scientific results to which I contributed during this doctoral research. The publications that will be discussed in this section are publications to which I am an author.

Two other projects which focused on the Iterated Prisoner's Dilemma have been [48, 49]. The works of [48, 49] focused on the usage of reinforcement learning algorithms (genetic algorithms and particle swarm optimisation algorithms) in training a series of strategies based on different structures such as finite state machines, hidden Markov models and neural networks. These strategies were trained in two settings:

- 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.
- A standard tournament.

The results of [48] were confirmed in Chapter 4. The trained strategies performed at the top of the standard tournament surpassing well established strategies such as Tit For Tat, Pavlov, Gradual and zero-determinant strategies. In [49] it was observed that the trained strategies (with no manual input) evolved the ability to have a handshake, to recognise themselves. This was particularly important in a Moran process of resisting invasion where a single individual of another type is introduced and the strategies need to resist the invasion.

Another undertaken project included exploring rhino poaching behaviour using evolutionary game theory [**?**]. Rhino populations are at critical level today and in protected areas devaluation approaches are used to secure the life of the animals. The effectiveness of these approaches, however, relies on poachers behaviour as they can be selective and not kill devalued rhinos or indiscriminate. Populations of differently behaving poachers were modelled using evolutionary game theory. The results demonstrated that full devaluation of all rhinos is likely to lead to indiscriminate poaching and that devaluating of rhinos can only be effective when implemented along with a strong disincentive framework. The paper aimed to contribute to the necessary research required for an informed discussion about the lively debate on legalising rhino horn trade.

Finally, providing science outreach workshops is a great way to gain a deeper understanding of science and its applications, and enhancing students interest in science. With that in mind I created an open source educational tutorial, called Game Theory and Python [**?**], aimed at introducing participants to game theory and more specifically to repeated games. The tutorial is aimed at two groups of individuals: individuals familiar with Python (programmers) who want to start to learn game theory and mathematicians with little or no programming knowledge as a pathway to programming through the interesting subject. The tutorial has gained much interest and is currently under submission at the Journal of Open Source Education.

A full list of the publications produced during the research presented in this section is:

1. Reinforcement Learning Produces Dominant Strategies for the Iterated Prisoner's Dilemma [48].
2. Evolution Reinforces Cooperation with the Emergence of Self-Recognition Mechanisms: an empirical study of the Moran process for the iterated Prisoner's dilemma [49].
3. An Evolutionary Game Theoretic Model of Rhino Horn Devaluation [**?**].
4. Game Theory and Python [**?**].

8.4 Future research directions

Each part of this thesis has given rise to further interesting questions and research directions that, although not in the scope of the current work, would improve or compliment it.

Future research - Meta tournament Analysis

In Chapter 4 during the data collection the probability of noise was allowed to vary between values of 0 and 1. However, it was established that large values of noise (> 0.1) caused an impactful variation to the environment. From the collection of 195 strategies considered in the Chapter there was not a single strategy that performed well in that spectrum of noise.

Strategies that have been trained specifically for noisy environments such as DBS, Evolved FSM 16 Noise 05, Evolved ANN 5 Noise 05, PSO Gambler 2 2 2 Noise 05 and Omega Tit For Tat, performed adequately only in tournaments with restricted noise. This indicates that possibly there is not a strategy in the literature trained to be effective for a broad spectrum of noise values. Training such a strategy would be an interesting avenue of further research. The analysis of the top performances would then be reproduced whilst including the new trained strategy.

Future research - Memory-one strategies

In Chapter 5 the empirical results supported that extortionate play is not always optimal and that memory-one strategies suffer from their limited memory in multi agent interactions. All the empirical results presented have been for the case of two opponents ($N = 2$). A future research direction would be to validate the empirical results of the Chapter for larger values of N.

Another restricted set of strategies on memory that have been studied in the literature are memory-two strategies. These are strategies that take into account the past two turns of the match. A compelling research question that arises is whether the current formulation of Chapter 5 can be expanded to include memory-two strategies, and whether the results still hold.

Future research - Training an LSTM strategy

Bibliography

[1] Complexity of cooperation web site. http://www-personal.umich.edu/~axe/research/Software/CC/CC2.html. Accessed: 2017-10-23.

[2] The evolution of trust. http://ncase.me/trust/. Accessed: 2017-10-23.

[3] The iterated prisoner's dilemma game. http://selborne.nl/ipd/. Accessed: 2017-10-23.

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

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

[6] Karim M Abadir and Jan R Magnus. *Matrix algebra*, volume 1. Cambridge University Press, 2005.

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

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

[9] Alkiviadis Akritas, Gennadi Malaschonok, and Panagiotis Vigklas. Sturm sequences and modified subresultant polynomial remainder sequences. *Serdica Journal of Computing*, 8(1):29–46, 2014.

[10] B. Allen, G. Lippner, Y. T. Chen, B. Fotouhi, N. Momeni, S. T. Yau, and M. A. Nowak. Evolutionary dynamics on any population structure. *Nature*, 544(7649):227, 2017.

[11] N. Anastassacos and M. Musolesi. Learning through probing: a decentralized reinforcement learning architecture for social dilemmas. *arXiv preprint arXiv:1809.10007*, 2018.

[12] M. Archetti and K. J. Pienta. Cooperation among cancer cells: applying game theory to cancer. *Nature Reviews Cancer*, page 1, 2018.

[13] D. Ashlock and E. Y. Kim. Techniques for analysis of evolved prisoner's dilemma strategies with fingerprints. 3:2613–2620 Vol. 3, Sept 2005.

[14] 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.

[15] D. Ashlock, E. Y. Kim, and W. Ashlock. Fingerprint analysis of the noisy prisoner's dilemma using a finite-state representation. *IEEE Transactions on Computational Intelligence and AI in Games*, 1(2):154–167, June 2009.

[16] D. Ashlock, E. Y. Kim, and W. Ashlock. A fingerprint comparison of different prisoner's dilemma payoff matrices. pages 219–226, Aug 2010.

[17] D. Ashlock, E. Y. Kim, and N. Leahy. Understanding representational sensitivity in the iterated prisoner's dilemma with fingerprints. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 36(4):464–475, July 2006.

[18] W. Ashlock and D. Ashlock. Changes in prisoner's dilemma strategies over evolutionary time with different population sizes. pages 297–304, 2006.

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

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

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

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

[23] R. Axelrod. Launching "the evolution of cooperation". *Journal of Theoretical Biology*, 299(Supplement C):21 – 24, 2012. Evolution of Cooperation.

[24] R. Axelrod and D. Dion. The further evolution of cooperation. *Science*, 242(4884):1385–1390, 1988.

[25] R. Axelrod and W. D. Hamilton. *The Evolution of Cooperation*. 1984.

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

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

[28] B. Beaufils, J. P. Delahaye, and P. Mathieu. Complete Classes of Strategies for the Classical Iterated Prisoner's Dilemma. 1447:33–41, 1998.

[29] 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.

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

[31] R. Boyd and J. P. Lorberbaum. No pure strategy is evolutionarily stable in the repeated prisoner's dilemma game. *Nature*, 327:58–59, 1987.

[32] J.K. Choi, J.S Yang, and H.H. Jo. The co-evolution of cooperation and trait distinction. In *Proceedings of the First Complexity Conference*, 2006.

[33] S. Y. Chong and X. Yao. Behavioral diversity, choices and noise in the iterated prisoner's dilemma. *IEEE Transactions on Evolutionary Computation*, 9(6):540–551, Dec 2005.

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

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

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

[37] S. Dridi and E. Akçay. Learning to cooperate: The evolution of social rewards in repeated interactions. *The American Naturalist*, 191(1):58–73, 2018. PMID: 29244562.

[38] S. R. Eddy. Hidden markov models. *Current opinion in structural biology*, 6(3):361–365, 1996.

[39] G. W. Evans and C. M. Crumbaugh. Payment schedule, sequence of choice, and cooperation in the prisoner's dilemma game. *Psychonomic Science*, 5(2):87–88, Feb 1966.

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

[41] N. Franken and A. P. Engelbrecht. Particle swarm optimization approaches to coevolve strategies for the iterated prisoner's dilemma. *IEEE Transactions on Evolutionary Computation*, 9(6):562–579, Dec 2005.

[42] Peter Frazier, Warren Powell, and Savas Dayanik. The knowledge-gradient policy for correlated normal beliefs. *INFORMS journal on Computing*, 21(4):599–613, 2009.

[43] Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[44] P. S. Gallo and I. A. Dale. Experimenter bias in the prisoner's dilemma game. *Psychonomic Science*, 13(6):340–340, Jun 1968.

[45] Nikoleta E. Glynatsi. Raw data for: "Stability of defection, optimisation of strategies and the limits of memory in the Prisoner's Dilemma.", September 2019.

[46] A. Goodman. Learning implicit communication strategies for the purpose of illicit collusion. *arXiv preprint arXiv:1802.06036*, 2018.

[47] D. Hales. Cooperation without memory or space: Tags, groups and the prisoner's dilemma. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 157–166. Springer, 2000.

[48] M. Harper, V. Knight, M. Jones, G. Koutsovoulos, N. E. Glynatsi, and O. Campbell. Reinforcement learning produces dominant strategies for the iterated prisoner's dilemma. *CoRR*, abs/1707.06307, 2017.

[49] 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.

[50] P. G. Harrald and D. B. Fogel. Evolving continuous behaviors in the iterated prisoner's dilemma. *Biosystems*, 37(1):135 – 145, 1996.

[51] D. Hathcock and S. H. Strogatz. Fitness dependence of the fixation-time distribution for evolutionary dynamics on graphs. *arXiv preprint arXiv:1812.05652*, 2018.

[52] V. Hayes. *The Evolution of Cooperation: A Recreation of Axelrod's Computer Tournament.* The University of North Carolina at Greensboro, 2017.

[53] Philipp Hennig and Christian J Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(Jun):1809–1837, 2012.

[54] C. Hilbe, M. A. Nowak, and K. Sigmund. Evolution of extortion in iterated prisoner's dilemma games. *Proceedings of the National Academy of Sciences*, page 201214834, 2013.

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

[56] C. Hilbe, A. Traulsen, and K. Sigmund. Partners or rivals? strategies for the iterated prisoner's dilemma. *Games and Economic Behavior*, 92:41 – 52, 2015.

[57] Christian Hilbe, Krishnendu Chatterjee, and Martin A Nowak. Partners and rivals in direct reciprocity. *Nature human behaviour*, 2(7):469–477, 2018.

[58] Christian Hilbe, Luis A Martinez-Vaquero, Krishnendu Chatterjee, and Martin A Nowak. Memory-n strategies of direct reciprocity. *Proceedings of the National Academy of Sciences*, 114(18):4715–4720, 2017.

[59] S. Horti. The evolution of trust is a cute explain-o-game about cooperation. `http://ncase.me/trust/`.

[60] M. Janssen. Evolution of cooperation when feedback to reputation scores is voluntary. *Journal of Artificial Societies and Social Simulation*, 9(1), 2006.

[61] Donald R Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21(4):345–383, 2001.

[62] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

[63] Gubjorn Jonsson and Stephen Vavasis. Accurate solution of polynomial equations using macaulay resultant matrices. *Mathematics of computation*, 74(249):221–262, 2005.

[64] M. Jurišić, D. Kermek, and M. Konecki. A review of iterated prisoner's dilemma strategies. pages 1093–1097, May 2012.

[65] A. Kaznatcheev, J. Peacock, D. Basanta, A. Marusyk, and J. G. Scott. Fibroblasts and alectinib switch the evolutionary games that non-small cell lung cancer plays. *bioRxiv*, 2017.

[66] Jeremy Kepner and John Gilbert. *Graph algorithms in the language of linear algebra.* SIAM, 2011.

[67] N. Kinch. Game theory and the evolution of trust. `https://medium.com/greater-than-experience-design/game-theory-and-the-evolution-of-trust-6da95b33407a`.

[68] V. A. Knight, M. Harper, N. E. Glynatsi, and O. Campbell. Evolution reinforces cooperation with the emergence of self-recognition mechanisms: an empirical study of the moran process for the iterated prisoner's dilemma. *CoRR*, abs/1707.06920, 2017.

[69] 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.

[70] Vincent Knight, Marc Harper, Nikoleta E. Glynatsi, and Owen Campbell. Evolution reinforces cooperation with the emergence of self-recognition mechanisms: An empirical study of strategies in the moran process for the iterated prisoner's dilemma. *PLOS ONE*, 13(10):1–33, 10 2018.

[71] J. R. Koza. Genetic programming. 1997.

[72] C. Lee, M. Harper, and D. Fryer. The art of war: Beyond memory-one strategies in population games. *PLOS ONE*, 10(3):1–16, 03 2015.

[73] J. Li. How to design a strategy to win an ipd tournament. pages 89–104, 04 2007.

[74] S. Li. Strategies in the stochastic iterated prisoner's dilemma. *REU Papers*, 2014.

[75] X.S. Liu, ZX. Wu, M. Z. Q. Chen, and JY. Guan. Evolutionary fate of memory-one strategies in repeated prisoner's dilemma game in structured populations. *The European Physical Journal B*, 90(7):138, Jul 2017.

[76] D. R. Lutzker. Sex role, cooperation and competition in a two-person, non-zero sum game. *Journal of Conflict Resolution*, 5(4):366–368, 1961.

[77] R. M. May M. A. Nowak. Evolutionary games and spatial chaos. *Letters to nature*, 359:826–829, 1992.

[78] Francis Sowerby Macaulay. Some formulae in elimination. *Proceedings of the London Mathematical Society*, 1(1):3–27, 1902.

[79] D. Mack, P. N. Auburn, and G. P. Knight. Sex role identification and behavior in a reiterated prisoner's dilemma game. *Psychonomic Science*, 24(6):280–282, Jun 1971.

[80] R. E. Marks and H. Schnabl. *Genetic Algorithms and Neural Networks: A Comparison Based on the Repeated Prisoners Dilemma*, pages 197–219. Springer US, Boston, MA, 1999.

[81] Aaron Meurer, Christopher Smith, Mateusz Paprocki, Ondřej Čertík, Sergey Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian Granger, Richard Muller, Francesco Bonazzi, Harsh Gupta, Shivam Vats, Fredrik Johansson, Fabian Pedregosa, and Anthony Scopatz. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3, 2017.

[82] M. Milinski. Tit for tat in sticklebacks and the evolution of cooperation. *Nature*, 325:433–435, January 1987.

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

[84] J. H. Miller, C. T. Butts, and D. Rode. Communication and cooperation. *Journal of Economic Behavior & Organization*, 47(2):179–195, 2002.

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

[86] E. F. Moore. Gedanken-experiments on sequential machines. *Automata studies*, 34:129–153, 1956.

[87] J. Nash. Non-cooperative games. *Annals of mathematics*, pages 286–295, 1951.

[88] J. Von Neumann and O. Morgenstern. Theory of games and economic behavior. *Princeton University Press*, page 625, 1944.

[89] S. Neumann, S. Sood, M. Hollander, F. Wan, A.W. Ahmed, and M. Hancock. Using bots in strategizing group compositions to improve decision–making processes. In Dylan D. Schmorrow and Cali M. Fidopiastis, editors, *Augmented Cognition: Users and Contexts*, pages 305–325, Cham, 2018. Springer International Publishing.

[90] 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.

[91] M. A. Nowak and K. Sigmund. Game-dynamical aspects of the prisoner's dilemma. *Applied Mathematics and Computation*, 30(3):191 – 213, 1989.

[92] M. A. Nowak and K. Sigmund. The evolution of stochastic strategies in the prisoner's dilemma. *Acta Applicandae Mathematica*, 20(3):247–265, Sep 1990.

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

[94] M. A. Nowak and K. Sigmund. The dynamics of indirect reciprocity. *Journal of theoretical Biology*, 194(4):561–574, 1998.

[95] University of Southampton. University of southampton team wins prisoner's dilemma competition. `https://www.southampton.ac.uk/news/2004/10/team-wins-competition.page`, 2004.

[96] H. Ohtsuki. Evolutionary dynamics of coordinated cooperation. *Frontiers in Ecology and Evolution*, 6:62, 2018.

[97] H. Ohtsuki, C. Hauert, E. Lieberman, and M. A. Nowak. A simple rule for the evolution of cooperation on graphs and social networks. *Nature*, 441(7092):502, 2006.

[98] Liming Pan, Dong Hao, Zhihai Rong, and Tao Zhou. Zero-determinant strategies in iterated public goods game. *Scientific reports*, 5:13096, 2015.

[99] 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.

[100] P. Pudaite. On the initial viability of cooperation. *Merriam Laboratory for Analytic ARTICLES 1389 Political Research*, 1985.

[101] A. Rapoport and A. M. Chammah. *Prisoner's Dilemma: A Study in Conflict and Cooperation, by Anatol Rapoport and Albert M. Chammah, with the Collaboration of Carol J. Orwant*. University of Michigan Press, 1965.

[102] A. Rapoport, D. A. Seale, and A. M. Colman. Is tit-for-tat the answer? on the conclusions drawn from axelrod's tournaments. *PLOS ONE*, 10(7):1–11, 07 2015.

[103] Dixon Resultant. The kapur-saxena-yang variant of the.

[104] R. L. Riolo, M. D. Cohen, and R. Axelrod. Evolution of cooperation without reciprocity. *Nature*, 414(6862):441, 2001.

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

[106] George Salmon. *Lessons introductory to the modern higher algebra*. Stechert, 1924.

[107] 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.

[108] J. Sensenig, T. E. Reed, and J. S. Miller. Cooperation in the prisoner's dilemma as a function of interpersonal distance. *Psychonomic Science*, 26(2):105–106, Feb 1972.

[109] J. Maynard Smith. The theory of games and the evolution of animal conflicts. *Journal of Theoretical Biology*, 47(1):209 – 221, 1974.

[110] J. Maynard Smith. Game theory and the evolution of behaviour. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 205(1161):475–488, 1979.

[111] J. Maynard Smith and G. R. Price. The logic of animal conflict. *Nature*, 246(5427):15–18, 1973.

[112] 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.

[113] Alexander J Stewart and Joshua B Plotkin. From extortion to generosity, evolution in the iterated prisoner's dilemma. *Proceedings of the National Academy of Sciences*, 110(38):15348–15353, 2013.

[114] Alexander J Stewart and Joshua B Plotkin. Small groups and long memories promote cooperation. *Scientific reports*, 6:26889, 2016.

[115] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[116] S. Suzuki and E. Akiyama. Reputation and the evolution of cooperation in sizable groups. *Proceedings of the Royal Society of London B: Biological Sciences*, 272(1570):1373–1377, 2005.

[117] J. T. Tedeschi, D. S. Hiester, S. Lesnick, and J. P. Gahagan. Start effect and response bias in the prisoner's dilemma game. *Psychonomic Science*, 11(4):149–150, 1968.

[118] A. W. Tucker. The mathematics of tucker: A sampler. *The Two-Year College Mathematics Journal*, 14(3):228–232, 1983.

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

[120] 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.

[121] S. Wang and F. Lin. Invincible strategies of iterated prisoner's dilemma. *arXiv preprint arXiv:1712.06488*, 2017.

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

[123] C. Zhang, J. Zhang, G. Xie, L. Wang, and M. Perc. Evolution of interactions and cooperation in the spatial prisoner's dilemma game. *PLOS ONE*, 6(10):1–7, 10 2011.

¡¡¡¡¡¡¡ HEAD

Appendix A

Centrality Measures Distributions

A.1 Distributions for G and $\bar{G}$

Betweeness and closeness centralities distributions for G and $\bar{G}$.

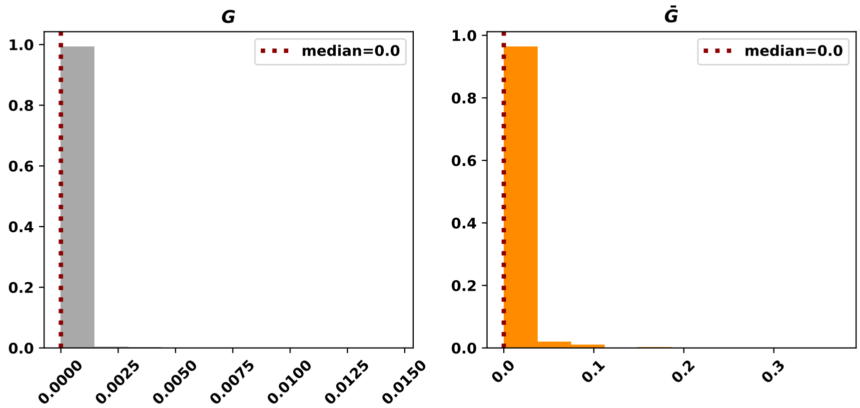


Figure A.1: Distributions of betweenness centrality in G and $\bar{G}$

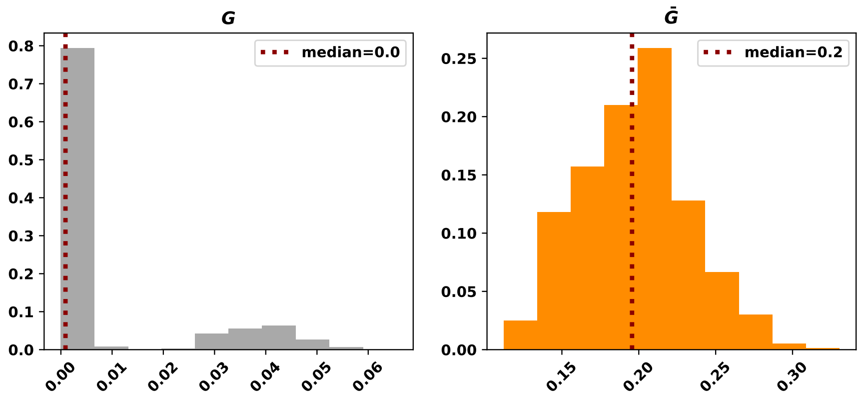


Figure A.2: Distributions of closeness centrality in G and $\bar{G}$

A.2 Distributions for Topic Networks

Betweeness and closeness centralities distributions for graphs of topics A to E.

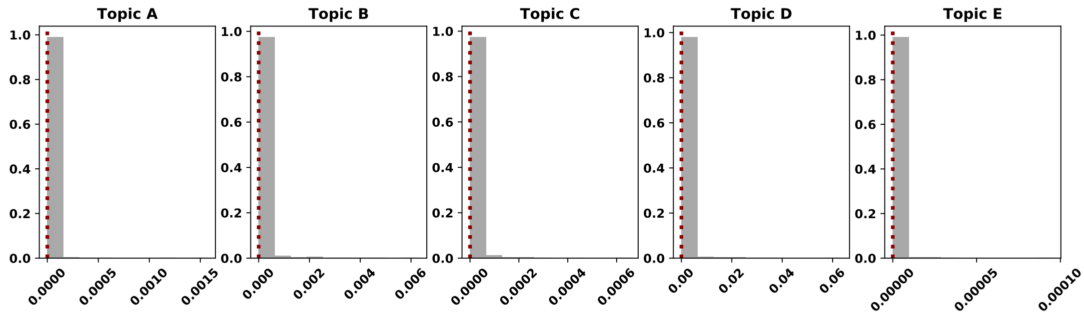


Figure A.3: Distributions of betweenness centrality in topics' networks.

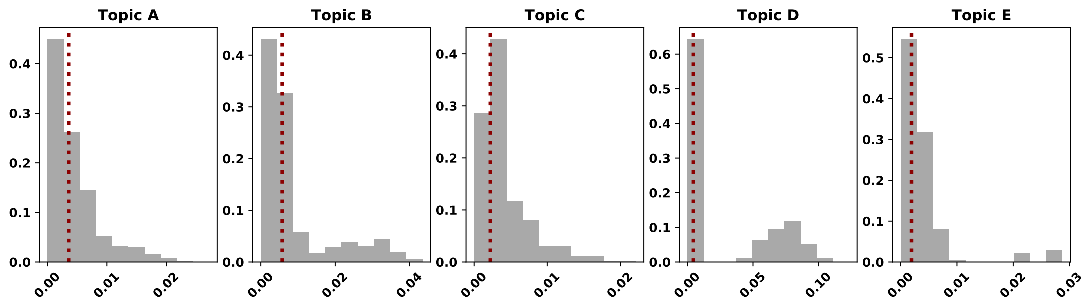


Figure A.4: Distributions of closeness centrality in topics' networks.

=======

Appendix B

Example of non concavity for $u(p)$

A function $f(x)$ is concave on an interval $[a, b]$ if, for any two points $x_1, x_2 \in [a, b]$ and any $\lambda \in [0, 1]$,

$$f(\lambda x_1 + (1-\lambda)x_2) \geq \lambda f(x_1) + (1-\lambda)f(x_2). \tag{B.1}$$

Let f be $u_{(\frac{1}{3},\frac{1}{3},\frac{1}{3},\frac{1}{3})}$. For $x_1 = (\frac{1}{4}, \frac{1}{2}, \frac{1}{5}, \frac{1}{2}), x_2 = (\frac{8}{10}, \frac{1}{2}, \frac{9}{10}, \frac{7}{10})$ and $\lambda = 0.1$, direct substitution in (B.1) gives:

$$\begin{aligned} u_{(\frac{1}{3},\frac{1}{3},\frac{1}{3},\frac{1}{3})}\left(0.1\left(\frac{1}{4},\frac{1}{2},\frac{1}{5},\frac{1}{2}\right)+0.9\left(\frac{8}{10},\frac{1}{2},\frac{9}{10},\frac{7}{10}\right)\right) \geq 0.1 \times u_{(\frac{1}{3},\frac{1}{3},\frac{1}{3},\frac{1}{3})}\left(\left(\frac{1}{4},\frac{1}{2},\frac{1}{5},\frac{1}{2}\right)\right) + 0.9 \times u_{(\frac{1}{3},\frac{1}{3},\frac{1}{3},\frac{1}{3})}\left(\left(\frac{8}{10},\frac{1}{2},\frac{9}{10},\frac{7}{10}\right)\right) \Rightarrow \\ 1.485 \geq 0.1 \times 1.790 + 0.9 \times 1.457 \Rightarrow \\ 1.485 \geq 1.490 \end{aligned}$$

which can not hold. Thus $u_{(\frac{1}{3},\frac{1}{3},\frac{1}{3},\frac{1}{3})}$ is not concave. ¿¿¿¿¿¿¿ initial draft of chapter