The study of replicator dynamics and iterated games

Danila Kurganov

January 1, 2022

Abstract

The study of various iterated prisoner's dilemma are discussed, with a deep analysis on the first such publicly held tournament. A different mathematical characterisation of such games is discussed through the lens of a replicator equation yielding Nash Equilibria with variously overlapping basins of attractions. These basins prove to have sub-cycles, non-transitivity properties, and significantly different basin structures even with the addition of miscommunication between opponents of the iterated game. The incredibly exciting zero determinant strategies are then introduced and discussed, with an added rigorous formalisation of several claims made by Press and Dyson, authors of the original paper. Finally, several computer simulations are run, running both the original and variant versions of the Axelrod tournament. A new set of hand-made strategies are tested in the prevailing environment, and a short discussion is made for possible further levels of abstraction.

Contents

1 The Iterated Prisoner's Dilemma and Axelrod's Tournament

The Prisoner's Dilemma (PD) game is first introduced, followed by a discussion of the Axelrod tournament interpreted from "Effective Choice in the Prisoner's Dilemma" by Robert Axelrod, an entry to a journal publication from 1980.

The Prisoner's Dilemma is a single-round game between two individuals (represented as Alice and Bob), each with two choices of play: cooperation or defection (c and d respectively). This is a 2x2 game yielding four possible game outcomes. Each outcome rewards each player in some way, say, with values T, R, P, S, representing Alice's defection with Bob's cooperation, mutual cooperation, mutual defection, and Alice cooperating with Bob defecting respectively. PD is a subset of an arbitrary 2x2 game, shown in table 1a, with the condition that $T > R > P > S$, as well as the condition of symmetry (each player has an identical rewards system from their point of view). The conventional version of the game, seen in table 1b, will be the game referred to as standard throughout this section of the text.

		Bob	
		C	D
Alice	C	R, R	S, T
	D	T, S	P, P

(a) Arbitrary PD Values with $T > R > P > S$

		Bob	
		c	d
Alice	c	$3, 3$	$0, 5$
	d	$5, 0$	$1, 1$

(b) Common PD values

Table 1: Payoff Matrices for the Prisoner's Dilemma Game

From Alice's perspective, without any empathetic connections to Bob, it is clear that defecting is her best strategy. In fact, the game has a unique Nash Equilibrium [link to its definition], that of mutual defection. However, if the PD game is tweaked so that multiple rounds of play occur (this will be denoted as IPD), then, along with the constraint that mutual cooperation is better than other forms of play in the long run ($2R > T + S$), it becomes clear that it pays to cooperate in the standard IPD. For example, if the game is 6 iterates long, a sequence of defections from both players yields either player a total payoff of 6. Yet, a sequence of collaborations yields each 18, arguably also better from the selfish point of view. Yet, if Alice defects in the last round whilst Bob doesn't, her payoff is 20. Selfishness, or maximal payoff, from the IPD perspective is therefore about yielding trust in such a way as to not be exploited by the other on a continual basis, yet also to benefit from mutual cooperation.

Due to IPD being able to frame interactions found in sociology, economics, warfare, trade agreements, biology, and more, its study was destined. Robert Axelrod was first to devise a computer tournament to pit IPD strategies against one another, with the first official tournament held in 1980.

The rules for the tournament given to all contestants were as follows; The standard IPD is to be played. Each "game" consists of 200 iterates of play, with 5 separate games played. Each strategy is then ranked by score, taken from the average score of the 5 games. In each round, each strategy plays against all other strategies, itself, and the RANDOM strategy - one which has an equal chance of cooperating as defecting for each round. There is no time limit on deciding a move to play, and each strategy, in forming a next-round decision, has access to only the previous moves it played with the other competitor of the game. Further information on the tournament is also given; contestants are all experts in a field of game theory, most with a great knowledge of PD. A copy of complete descriptions to two preliminary tournaments of similar style was also distributed. I now note that the TIT FOR TAT strategy previously won second and first place respectively.

The official tournament attracted 14 contestants sent in submissions, each sending a unique strategy. Even with all the given priors, the submitter of TIT FOR TAT won with 504 points. A complete description of tournament submissions and points are shown in Table 2.

It is odd to see, given the priors, how few submissions of TIT FOR TAT there were. However, upon closer inspection, most submissions actually had TIT FOR TAT as a sort of starting ground, adding further intuited complexities in hopes of stronger play. It is therefore interesting to see that TIT FOR TAT is seemingly unfazed by self-mutations.

Stepping outside the tournament, one might argue that TIT FOR TAT is simply the best IPD strategy there is. However, this is far from the case. In the very same tournament, a submission of TIT FOR TWO TATS (a strategy first cooperating, only defecting once the opponent defects twice in a row*) would have won the tournament. Similarly, the 10th place strategy, DOWNING, would have easily won, had it changed the first move from defection to contribution. It is therefore correct to say that the environment of strategies played plays a significant role in strategy dominion, in which case this and prior tournaments favoured TIT FOR TAT. With this context in mind, we discuss the attributes of the environment, and the effect this had on the strategies played.

This tournament had nice, not-nice, forgiving, non-forgiving, and exploitative attributes. Successful strategies shared two properties: niceness and forgiveness. We will discuss each and their implications. It is useful to note the top score was 504 points, in a possible range of tournament accepted scores of 0-1000.

The "niceness" attribute simply means that the strategy is not the first to defect. In this tournament, 8 of the 14 candidates were "nice", scoring roughly 600 points amongst themselves. Further, these strategies were no pushovers, being non-forgiving with strategies that defected first; non-nice strategies scoring rarely more than 400 points with them. Due to the sheer proportion of nice strategies, having this attribute was especially advantageous. The ranking of the nice strategies amongst themselves was determined by the rules they had with the non-nice.

The "forgiveness" attribute is the possibility of cooperation after the opponent has defected. One might assume that in a pool of nice and non-nice strategies, being forgiving is foolish, for then exploitation by the latter may occur. Although this is true in some extent, the tournament demonstrates that tactful forgiveness is better, and this is to be described by "echo phenomena". Suppose an unforgiving nice strategy is pitted against an unforgiving non-nice strategy. Due to non-nice strategies possessing the intent to noticeably defect within a string of cooperations, they will aim to mostly cooperate in such a sequence. However, the unforgiving opponent will instantly defect in such attempts, triggering a defection from the perpetrator! And so, one locally selfish defection triggers low scoring mutual defections for the rest of the game. Therefore, possessing skillful forgiveness to benefit amidst defections in a string of mutual cooperations is a far superior strategy!

In such an environment of nice and non-nice strategies, those that were nice and tactfully forgiving amongst the non-nice came out on top. Modified DOWNING and TIT FOR TWO TATS showing such attributes would have won the tournament. TIT FOR TAT was the winner, however, due to showing the least remorse compared to other entrants, therefore, not amplifying the mutually-defecting echo against non-nice play. There is a tug-of-war between the exploitative nature of non-nice play and forgiveness of those that are nice, so a set of different non-nice play would have changed the results of the top nice players. An equilibrium of actions and consequences if therefore crucial for play of IPD.

I lastly mention a specific game played between DOWNING and FELD. DOWNING estimates probabilistically what the other player does after it's own move throughout the match, and plays to maximise its own payoff correspondingly. FELD starts playing with TIT FOR TAT, slowly becoming more exploitative as rounds progress by defecting amidst mutual cooperation. Even with

FELDs exploits, DOWNING first was lured into thinking that it is best to cooperate, and was too late to consider that the opponent is now being exploitative. FELD was effectively able to successfully lure DOWNING to work harder for mutual cooperation compared to itself, specifically getting away with a 25% defection rate. This "lure" bears striking resemblance to that found in Zero-Determinant IPD strategies discovered 30 years later.

First Axelrod Tournament ranked competitors		
	Contestant	Strategy
1	TIT FOR TAT (504 pts)	Cooperate on the first move, then play what the opponent played in the previous round.
2	TIDEMAN and CHIERUZZI (500 pts)	Play TIT FOR TAT until the opponent defects more than once aggregately. In which case, defect the aggregate amount. Aggregates reset, two contributions are played, and strategy is reset if the opponent is ≥ 10 points behind, if he has not just started defecting, if it has been more than 20 moves since the last reset, if at least 10 moves remain, and if defections differ from a 50-50 random generator by 3 standard deviations. Defect on last two moves.
3	NYDEGGER (486 pts)	Play Tit for Tat for the first 3 moves, unless you were the only one to cooperate then defect in moves 1 and 2 resp.; in which case defect. Then, moves are determined by computing $16(2b_{i_3} + a_{i_3}) + 4(2b_{i_2} + a_{i_2}) + (2b_{i_1} + a_{i_1})$., where i_n are defection indicator variables for Alice (a) and Bob (b). Defect if this sum is 1, 6, 7, 17, 22, 23, 26, 29, 30, 31, 33, 38, 39, 45, 49, 54, 55, 58, or 61.
4	GROFMAN (482 pts)	If the player played differently previously, cooperate with probability 2/7, otherwise, cooperate.
5	SHUBIK (481 pts)	Cooperate until a defection, in which case defect once. If defection occurs after cooperation resumes, defect twice... etc.
6	STEIN and RAPOPORT (478 pts)	Cooperate first 4 moves, play TFT, defect last 2 moves. Check if the other is playing randomly every 15 moves via chi-squared test and alternating CD or DC moves.
7	FRIEDMAN (473 pts)	Cooperate until defection. Then defect.
8	DAVIS (472 pts)	Cooperate on the first 10 moves, and defect until the end of the game only if a defection occurs.
9	GRAASKAMP (401 pts)	Play TFT for the first 50 moves, defect on move 51, play TFT for 5 moves. Then, if the other player seems random, defect from then on. If the other player is TFT, ANALOGY, or itself, play TFT. Otherwise, randomly defect every 5 to 15 moves.
10	DOWNING (391 pts))	Maximise long-term payoff assuming the opponent cooperates with a fixed probability conditional to my previous move. Estimates are updated continuously, with starting assumption of (0.5, 0.5) cooperation.
11	FELD (328 pts)	Play with memory-1 strategy: (1 - .0025n, 0, 1, 0), where n is game's move iterate.
12	JOSS (304 pts)	Play with memory-1 strategy: (0.9, 0, 1, 0).
13	TULLOCK (301 pts)	Cooperate on the first 11 moves. Then, cooperate 10% less than the other player has in the preceding 10 moves.
14	UNNAMED (282 pts)	30% cooperation probability, and update every 10 moves depending on if the other is random, very cooperative, or very uncooperative. Adjust if after move 130 the rule is lower-scored than the other.
15	RANDOM (276 pts)	Play randomly, cooperating with the same probability as defecting.

2 The Calculus of Selfishness by Karl Sigmund

The content of chapter 3 "Direct Reciprocity: The Role of Repetition" of Sigmund's book is now discussed. This chapter provides an alternative analysis for the success of TIT FOR TAT strategies in IPD, and why their dominance is dependent on the environment of play. A fairly unimportant difference of Sigmund's analysis, in relation to the analysis of IPD, is that of the game analysed; the Donation Game (DG). This game doesn't hold to the PD condition of $2R > T + S$, and instead has the following payoff matrix:

$$P = \begin{pmatrix} b-c & -c \\ b & 0 \end{pmatrix} \qquad \text{satisfying } b > c > 0.$$

This difference in payoffs is unimportant, however, due to both payoff matrices' maintenance of the quo that in the long run, mutual cooperation is more beneficial for each player in the long run from other game sequences. This, however, assumes neither player is a pushover. The iterated version of the Donation Game will be called the Iterated Donation Game (IDG).

Before introducing Sigmund's machinery, I start with a disagreement on it's origins. I disagree with Sigmund on his statement that backwards inductions predicts that selfish players ought to defect in each round if the number of rounds is known to both players in IDG. His backwards induction argument, specifically the base case, assumes that only one future round of play will occur, which is not the case for IDG when more than several rounds are present. The inducting argument therefore only states the facts for the tactics for being locally selfish by considering a one-iterate game, as opposed to a properly iterated game, in which strategy plays a stronger role. These digressions are to be cast aside.

Let $w \in (0, 1)$, and $X \sim Geo(w)$ be a random variable with the Geometric distribution, denoting the number of rounds of IDG play. This model assumes the game master views each iterate of play independently from the others, that each further round can happen or not, and the probability of iterate continuation is constant at any iterate. Let 0 be the starting label for rounds of play, with n denoting the n-th iteration. Let $A(n)$ be the payoff for one player in iterate n.

$$E[X] = \frac{1}{1-w}$$
$$A(w) = \sum_{n=0}^{\infty} w^n(1-w)[A(0) + ... + A(n)]$$

$E[X]$ denotes the expected number of rounds played. $A(w)$ denotes the expected total payoff. The second series is convergent to $A(q)$ due to being monotonically decreasing and bounded from below.

Sigmund's first analysis considers IDG in the context of a round-robin tournament. The following important distinctions are made between this tournament and that of Axelrod's. Sigmund's tournament will only allow for 3 strategies of play from competitors, namely, always cooperate (AllC), always defect (AllD), and TIT FOR TAT (TFT), with x, y, and z representing the proportion of individuals with this respective strategy. There will be an infinite number of players, specifically, enough to allow for any proportion of each type of play to exist. Furthermore, each player is allowed to dynamically evolve their strategy, by switching from one strategy to the next,

with opportunity to do so in each round. This switching is specifically governed by the following replicator equation:

$$\dot{x} = x(P_x - \hat{P})$$
$$\dot{y} = y(P_y - \hat{P})$$
$$\dot{z} = z(P_z - \hat{P})$$

with P_x, P_y, P_z denoting $A(w)$ for corresponding player, and $\hat{P} = xP_x + yP_y + zP_z$ denoting the average payoff of the entire population. Therefore, from the tournament's perspective, each player plays each other "once", extrapolating payoffs with opponents with regards to each's probability of play. This is done by all players, and at the end of such extrapolated interactions, labelled by a single time-step, each player is left with access to their own score, how many others played their strategy, and the population's score. The proportion of each type of strategy then changes in accordance to how well their strategy did. Therefore, we see that Sigmund's tournament has many differences to that of Axelrod's. Further tournaments of Sigmund give generalisations on the above, including superpositions of noise and infinite iterates to the original Sigmund's tournament. What follows is an analysis on the dynamics of strategy from the perspective of TFT dominance, with figures referenced from Sigmund's book.

Figure 3.1 corresponds to Sigmund's original tournament in accordance to the replicator equation. We see that a relatively small TFT population is able to invade a population of defectors, and in their absence, TFT and AllC players are evenly matched and correspond to rest points. Sigmund's analysis becomes more insightful with thought to small perturbations introduced to the game's strategy ratio. If a random shock introduces a small number of defectors to the system, they will first be dominant and repopulate, however, with enough TFT players, they will ultimately vanish. During the invasion of AllD, they will exploit and deplete AllC, until enough TFT players reverse their victory. The only hope AllD has is to dominate the tournament is to allow for a large perturbation towards AllC to occur, removing the number of recipricators, and to then make their attack. Hence, if too few AllD invasions are found, potentially representing a society with too little conflict, immunity to invaders is lost.

Figure 3.2 corresponds to a variant tournament, one in which an intended cooperation or defection has some probability of misimplementation. Let $Y \sim Geo(\epsilon)$ and $Z \sim Geo(k\epsilon)$ with $k \geq 0$ represent the distribution of cooperation and defection misimplementations respectively. Compared to prior dynamics, TFT is no longer able to successfully rid invasions by AllD, instead resulting in a periodic orbit between all three strategies. Though a small number of TFT moves the tournament significantly away from AllD, TFT holds much looser control over enforcement of strategy as it did before. For AllD, y = 1 is a basin of attraction. It can be made arbitrarily small if $w \approx 1$, which is shown in the next figure.

Figure 3.3 corresponds to another variant of the tournament, in which defections fail. The limiting dynamics are studied, with limits occurring in a specific order. First, the rounds go to infinity, and then the proportion of move misimplementation goes to 0. The dynamics are similar to that of rock-paper-scissors. AllD beats TFT, TFT beats AllC, and AllC beats AllD. TFT shows no dominance in this environment.

In Figure 3.4, corresponds to the tournament of Figure 3.3, however limits occur in the reverse order. First proportion of move misimplementation goes to 0, followed by the number of rounds going to infinity. It's clear by viewing the resulting dynamics that the limits do not commute.

Separate to Sigmund's analysis, it can be said that TFT gets its dominance only after misimplementation no longer occur, otherwise, a forever-periodic nature of strategies is to exist, as found in Figure 3.3.

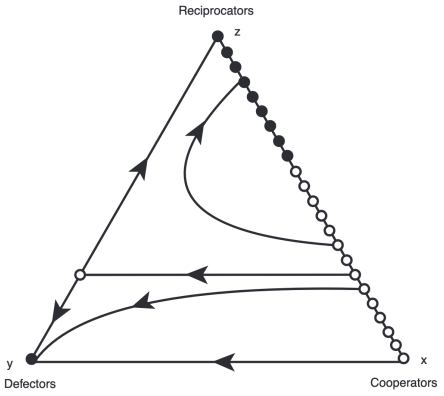

Figure 3.1 The good, the bad, and the reciprocator, in the absence of errors. A horizontal line $z = c/wb$ divides the state space. Below the line, defectors win; above the line, defectors are eliminated. Here and in all other figures, filled circles correspond to stable rest points, and empty circles to unstable rest points.

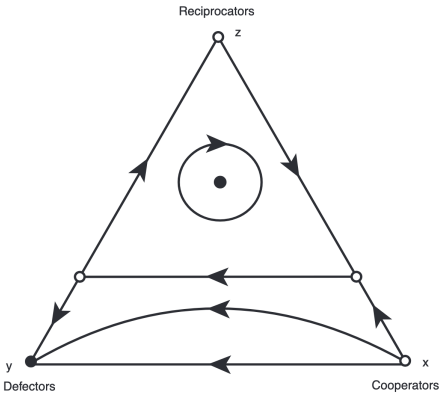

Figure 3.2 The good, the bad, and the reciprocator, with errors. If z is below a threshold, defectors win; if z is above the threshold, all three strategies co-exist, their frequencies oscillating periodically.

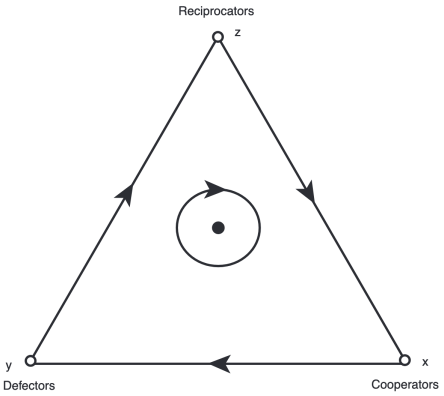

Figure 3.3 The infinitely iterated Donation game ($w = 1$), if there is a positive probability that intended moves (donation or refusal) are mis-implemented.

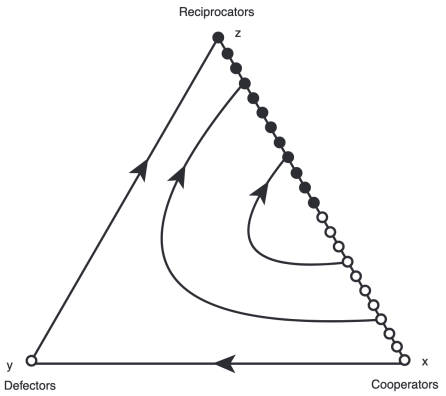

Figure 3.4 The infinitely iterated Donation game ($w = 1$), in the absence of errors (i.e., $\epsilon = 0$).

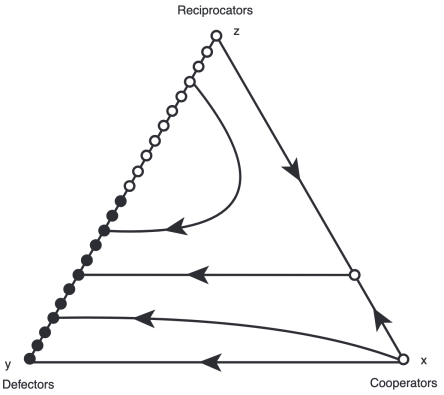

Figure 3.5 The replicator dynamics of the infinitely iterated Donation game, if only donations can be mis-implemented, but refusals are not. Cooperation vanishes in the long run.

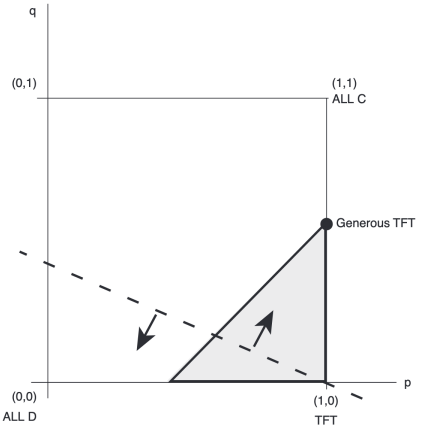

Figure 3.6 The cooperation-rewarding zone (shaded in grey) is a subset of the space of reaction norms (p, q) for the infinitely iterated Donation game. The arrows point in the direction of the most favorable adaptation. This direction is always orthogonal to the line connecting the norm with $(1, 0)$.

Figure 1: Replicator Dynamics for a variety of system rules.

Figure 3.5 describes an asymmetric variant to the prior Figure. Defections no longer fail, however cooperations do. In this case, most starting system conditions result in the dominance of AllD strategy.

So far, Sigmund's prior tournaments highlighted the dynamics of heterogeneous populations; those that were on the interior of the simplex. Studying the same tournament's dynamics on the boundary reveals new information. In this case, the starting population is of one strategy type, and occasionally a small perturbation of an invading strategy is introduced to the section. The evolution of such a system are then studied. Furthermore, Sigmund allows for a fourth strategy, Generous TIT FOR TAT (GTFT), to be played. Recall that GTFT could have won Axelrod's tournament had it been a submission. It is defined by playing TFT, however, defecting only after two opponent defections. Figure 3.6 shows the change in proportion of strategy of boundary IDG strategies in an environment with four strategies. In such an environment, TFT no longer held its dominance. In fact, TFT could be considered a pivot strategy moving game dynamics towards favouring GTFT. An analysis referenced by Sigmund suggested that with uniformly distributed memory-1 strategies not using the player's own moves, many iterates of play converge population proportions first to AllD, then TFT, then converging finally to GTFT. GTFT has no strategic dominance with non TFT strategy dominance, and TFT has no dominance with non AllD dominance. This reveals interesting overlapping 'basins of attraction' for specific strategies.

Considering true memory-1 strategies, Sigmund further the 16 possible binary options and pitted them against one another. This reveals a more complex web of basin of attractions between strategies. Within the environment of 16 players, TFT was a non-final basin of attraction, occasionally played the role of kingmaker, and in the case of inclusion of noise, reveals terrible performance against itself, whilst not being the case for other strategies such as PAVLOV. In summary, although Sigmund considered a different tournament to that of Axelrod, he implicitly came across the same conclusion that TIT FOR TAT reveals varying performance depending strongly on the other strategies played, and even more severely upon the introduction of noise.

3 The paper of Press and Dyson

In 2012, William H. Press and Freeman J. Dyson released the paper "Iterated Prisoner's Dilemma contains strategies that dominate any evolutionary opponent", containing a completely new class of IPD strategies: Zero-Determinant strategies. What was remarkable was coming up with such a novel and simple theory in a field so rich and mature of various iterated prisoner dilemma strategies. Not to take away from Press and Dyson, however, a subclass of zero-determinant strategies is analysed by Sigmund a few years prior. Remarks aside, these class of strategies enforce a linear relationship between the players' payoffs. Furthermore, by specifying further constraints, it's mathematically possible to enforce an extortionate payoff, set by only one player of the tournament. Best of all, these strategies are inescapable; an opponent using a strategy taking information from more than one prior round of play will always be at a disadvantage, with the only best response being that of continual defection (effectively, no longer playing the game). The strategy pre-induces a trap, which, when realised after these several rounds of 'decision' by the longer memory opponent, will have already set the extortion in motion. The mathematics for these claims now follows, first proving the linear payoff relationship between opponents.

3.1 Zero Determinant Strategies

Let vector $\mathbf{xy} = (cc, cd, dc, dd) \in \{0, 1\}^4$ represent an indicator for one of the four possibilities prior matches realised. For example, $\mathbf{xy} = (1, 0, 0, 0)$ denotes the event that both parties cooperated

previously. Let memory-n strategies be defined as strategies which rely on the previous n rounds of play to make a decision. All memory-n strategies will be considered as stochastic vectors. Let X, Y denote two players of the standard IPD game, as outlined in section 1. Let X and Y's memory-1 strategies be given by $\mathbf{p} = (p_1, p_2, p_3, p_4)$ and $\mathbf{q} = (q_1, q_2, q_3, q_4)$, respectively. Each index represents the probability of cooperating given the indices in $\mathbf{xy}$. For example, $\mathbf{p} = (0.5, 0, 1, 0)$ implies that X has a 50% chance of cooperating if both players cooperated previously, and a 100% chance of cooperating if X defected and Y cooperated previously. Let us for now assume both players have memory-1 strategies; therefore all probabilities of the next match's outcomes, given the previous match's outcomes can be given by the Markov matrix (M) shown below.

	cc	cd	dc	dd
cc	p_1q_1	$p_1(1-q_1)$	$(1-p_1)q_1$	$(1-p_1)(1-q_1)$
cd	p_2q_3	$p_2(1-q_3)$	$(1-p_2)q_3$	$(1-p_2)(1-q_3)$
dc	p_3q_2	$p_3(1-q_2)$	$(1-p_3)q_2$	$(1-p_3)(1-q_2)$
dd	p_4q_4	$p_4(1-q_4)$	$(1-p_4)q_4$	$(1-p_4)(1-q_4)$

Table 2: Markov Matrix for the memory-one game.

By knowing the distribution of outcome for each iterate, with the respective payoffs, it's possible to deduce the expected payoff for each player. It may first seem that the expected payoff can only be computed on an iterate-by-iterate basis, however, this is generally not the case. Markov matrices, once applied to a distribution in an iterated sense, obtain basins of attraction; one which cause a wide variety of starting distributions to converge to a single stationary distribution. Therefore, knowing this stationary distribution is enough to determine expected payoffs in the long-run. Press and Dyson specifically exploit stationary distributions to construct ZD strategies.

Markov matrices don't always have unique stationary distributions, however. It's possible that different starting distributions converge differently, or potentially not converge to anything at all. Therefore, the conditions for uniqueness and convergence are of importance when analysing Markov matrices such as those given by Table 2. These conditions are proved below.

Theorem 3.1. *An irreducible and aperiodic Markov matrix has a unique stationary distribution.*

A proof of this theorem requires the following machinery.

Definition 3.1. Let A be an n×n square matrix. A is *irreducible* if for every pair of indices $i, j = 1, ..., n$, there exists an $m \in \mathbb{N}$ such that $(A^m)_{i,j} \neq 0$. If A has non-negative entries, then the *period of index i*, (for $i = 1, ..., n$), is the GCD of all $m \in N$ such that $(A^m)_{i,i} > 0$. If the period is 1, A is *aperiodic*. A *row-stochastic* matrix is one in which the rows sum to 1.

Remark The Markov matrix is row-stochastic.

The following lemma holds many of the keys to deduce properties of stationary distributions.

Lemma 3.2 (Perron-Frobenius)**.** *Let A be an irreducible, non-negative n×n matrix with period α and spectral radius $\rho(A) = r$. Then*

1. *r is the unique eigenvalue of A.*
2. *A has left eigenvalue z with associated eigenvalue r, and z has all positive entries.*
3. *A has exactly α complex eigenvalues with modulus r and each is a simple root of the characteristic polynomial of A.*

Lemma 3.3. *A row-stochastic square matrix has a largest eigenvalue of 1. Furthermore, if the matrix is irreducible and aperiodic, 1 is its unique eigenvalue.*

Proof. Let A be a row-stochastic square matrix. Since $A\mathbf{1} = \mathbf{1}$, $\mathbf{1} \in \mathbb{R}^n$ is an eigenvalue. Suppose $Av = \lambda v$ for $\lambda \in \mathbb{R}$ and $v \in \mathbb{R}^n$. Let k denote the index of a maximal value in v. Hence, $|\lambda||v_k| = |\lambda v_k| = |(Av)_k| = |\sum_{i=1}^n A_{n,i}v_i| \leq |\sum_{i=1}^n A_{n,i}v_k| = |v_k||\sum_{i=1}^n A_{n,i}| \leq \sum_{i=1}^n(|A_{n,i}|)|v_k| = \sum_{i=1}^n(A_{n,i})|v_k| = |v_k| \Rightarrow \lambda \leq 1$. Therefore, 1 is the maximal eigenvalue.

By Perron-Frobenius, 1 is its unique eigenvalue. □

Lemma 3.4. *An nxn row-stochastic square matrix* Π *and its transpose have the same set of eigenvalues. Further, there exists an eigenvector* $v \in \mathbb{R}^n$ *such that* $v^T\Pi = \Pi$.

Proof. Let $\lambda \in \mathbb{R}$. Then, $det(\Pi - \lambda I) = det((\Pi - \lambda I)^T) = det(A\Pi^T - \lambda I)$. So Π and Π^T have the same characteristic polynomials, and so share the same eigenvalues.

By Lemma 5.3, there exists a $v \in \mathbb{R}^n$ such that $\Pi^T v = v \Rightarrow v^T\Pi = v^T$. Therefore, any row-stochastic matrix has a stationary distribution. □

We are now ready to prove the theorem.

Proof. By Lemma 5.4, a row-stochastic matrix which is irreducible and aperiodic must have a left-eigenvector. Perron-Frobenius implies that all eigenvector entries must be positive, and so this eigenvector can be scaled to be a stationary distribution. Further, by Perron-Frobenius, this stationary distribution is unique. □

Now supplied are the conditions for convergence.

Theorem 3.5. *If a markov matrix is diagonalisable, then any initial distribution converges to a stationary distribution.*

First, a lemma.

Lemma 3.6. *If A is diagonalisable, then so is* A^T.

Proof. $A = PDP^{-1} \Rightarrow A^T = (P^{-1})^T D^T P^T$. Hence, as A^T has n-linearly independent columns, it is diagonalisable. □

And now, the proof.

Proof. Let d_0 be an arbitrary real-valued row-vector, denoting an initial distribution. Let d_k denote the distribution after k rounds of play by the markov matrix Π. Π's eigenvectors form a basis (v_i), so there exist scalars a_i such that $d_0 = \sum_{i=1}^n c_i v_i$.

$d_k = \Pi^k d_0 = \Pi^k \sum_{i=1}^n c_i v_i = \sum_{i=1}^n c_i \Pi^k v_i = \sum_{i=1}^n c_i \lambda_i^k v_i$. By Lemma 5.2, the maximal eigenvalue is 1, so $\lim_{k\to\infty} d_k = \sum_p c_p v_p = d$, for some set $p \subset \{1, ..., n\}$. It's shown that the stationary distribution is given by a superposition of eigenvalues of the markov matrix found in the starting distribution. Depending on these eigenvectors present in the starting distribution, different stationary distributions are obtained. □

Theorem 3.7. *If a markov matrix is diagonalisable, irreducible, and aperiodic, then any initial distribution converges to the either the zero distribution or the unique stationary distribution.*

Proof. Equivalent to the proof of Theorem 5.5, further conditions of irreducibility and aperiodicity imply, by Lemma 5.1, the uniqueness of a single eigenvalue of size 1. Hence, following the variables from theorem 5.5, the stationary distribution is given by $d_k = c_p v_p = d$. If the initial distribution falls in the eigenspace of the markov matrix whose vector corresponds to an eigenvalue 1, this eigenvector becomes the stationary distribution. Otherwise, it is 0. □

It is assumed in Press and Dyson's analysis, that the memory-1 strategies induce irreducible, aperiodic, and diagonalisable markov matrices. Cases where this isn't the case, not found in the paper, are to be separately analysed.

The results for uniqueness and convergence to stationary distributions are now clear; however no such method is specified for finding the stationary distribution(s). Theorem 5.5 and 5.6, however, gives light on the possibility of finding the eigenspace of the markov matrix to determine convergence codomains. What follows is a rigorous extension of that argument, followed by a corresponding figure.

Lemma 3.8. *Let M be a row-stochastic, irreducible, and aperiodic matrix. Let M' = M - I. Then every row of the adjoint matrix M' is proportional to the stationary distribution of M.*

Proof. By our previous results, M has unit eigenvalue with unique stationary distribution v^T. Therefore, M' := M - I has 0 determinant.

The determinant of a matrix can be computed by expanding columns or rows by using the minors: $detM = \sum_{i=1}^{n} a_{ij}(-1)i + jdetA(i|j)$. The coefficients of the adjoint are defined by $(Adj(M))_{i,j} = (-1)^{i+j} detA(j|i)$. By matrix multiplication, $(M \cdot adj(M))_{k,l} = \sum_{i=1}^{n}(-1)^{i+l} a_{ki} det(A(l|i))$. Therefore, for $k = l$ we get $detA$, and for $k \neq l$, 0. Hence, $adj(A)A = detA \cdot I$. Since $detM' = 0$, and stationary distributions are unique for M, each row of M' is proportional to v^T, the unique stationary distribution of M. □

$$M' = \begin{matrix} p_1q_1-1 & p_1(1-q_1) & (1-p_1)q_1 & (1-p_1)(1-q_1) \\ p_2q_3 & p_2(1-q_3)-1 & (1-p_2)q_3 & (1-p_2)(1-q_3) \\ p_3q_2 & p_3(1-q_2) & (1-p_3)q_2-1 & (1-p_3)(1-q_2) \\ p_4q_4 & p_4(1-q_4) & (1-p_4)q_4 & (1-p_4)(1-q_4)-1 \end{matrix} \quad \Rightarrow \quad Adj(M')^T = \begin{matrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & M_{22} & M_{23} & M_{24} \\ M_{31} & M_{32} & M_{33} & M_{34} \\ M_{41} & M_{42} & M_{43} & M_{44} \end{matrix}$$

$$Adj(M')_{\text{last row}} = (-M_{14}, M_{24}, -M_{34}, M_{44}) \propto v^T$$

Hence, using adjoints, a quick expression is given for the values of the stationary distribution.

Now that the long run-time stationary distribution can be found, each player's payoff can be determined by taking the dot product of this distribution with the payoff values.

A neat shortcut for this dot product, however, can be found. The computation of adjoints, say by running through the fourth column of matrix M', requires finding determinants of the other three columns without a common row. The computation of determinant of M', however, is completely equivalent, by running through the fourth column, except for the need to multiply each sub-determinant by a corresponding element of the fourth column. Therefore, this dot product can be computed by finding the determinant of M', having first replaced the fourth column by a payoff value vector.

A formalisation of this shortcut now follows. Let $S_x = (R, S, T, P) \in \mathbb{R}^4$ (resp. $S_y = (R, T, S, P) \in \mathbb{R}^4$) denote the payoff vector for X (resp. Y), with indices corresponding to the same order of payoffs as specified by vector xy. For example, $S_y = (1, 2, 0, 0)$ implies player Y receives 1 point (respectively 2 points) given a mutual cooperation (resp. cooperation by X and

$$v \cdot f = det \begin{pmatrix} p_1q_1 - 1 & p_1(1-q_1) & (1-p_1)q_1 & f_1 \\ p_2q_3 & p_2(1-q_3) - 1 & (1-p_2)q_3 & f_2 \\ p_3q_2 & p_3(1-q_2) & (1-p_3)q_2 - 1 & f_3 \\ p_4q_4 & p_4(1-q_4) & (1-p_4)q_4 & f_4 \end{pmatrix} = det \begin{pmatrix} p_1q_1 - 1 & p_1 - 1 & q_1 - 1 & f_1 \\ p_2q_3 & p_2 - 1 & q_3 & f_2 \\ p_3q_2 & p_3 & q_2 - 1 & f_3 \\ p_4q_4 & p_4 & q_4 & f_4 \end{pmatrix}$$

Figure 2

defection by Y) as the past game's iterate. Let v denote the stationary distribution of the irreducible and aperiodic markov matrix. It's important to keep in mind that columns can be added from within a matrix to other columns without changing the value of the determinant. With this in mind, Figure 2 gives the computation of expected long run time payoffs per round.

Indirectly, by having found a shortcut to compute payoffs, Press and Dyson made a remarkable observation. Columns 2 and 3 of the matrix in Figure 2 are completely and independently under the control of players X and Y. Although this has little meaning for a player clueless about the opponent's moves if the column under control is not in the span of f, it has remarkable consequences for a playing knowing the opponent's payoff vector. Specifically, by choosing to play in the span of f, one player can choose to make $v \cdot f = 0$, regardless of what the opponent does. In the context of payoffs, this means X can fully control the Y's pay! This inescapable form of control a player has, by setting the determinant of figure 2 to be 0 creates a category of strategies called zero determinant strategies. Due to the interest in vectors S_x and S_y in IPD, along with an easy to compute vector $\vec{1}$, zero determinant strategies boil down to the following equation:

$$v \cdot (\alpha S_x + \beta S_y + \gamma \vec{1}) := \alpha s_x + \beta s_y + \gamma = 0.$$
$$\alpha, \beta, \gamma \in \mathbb{R}$$
$$s_x = v \cdot S_x \qquad s_y = v \cdot S_y$$

This equation guarantees a linear relationship between payoffs of X and Y, precisely for irreducible, aperiodic, and diagonalisable markov matrices.

Press and Dyson use this equation liberally, considering sub cases and implications. Before continuing with these, however, it's important to consider potential cases when markov matrices do not meet the above assumptions. To easily and visually consider such cases, a graph (shown in Figure 3) is drawn of the general memory-1 markov matrix. Reducibility is implied by the graph containing several separate, yet non-communicating subgraphs. Periodicity is given by a graph containing a node that only returns to itself in $n \cdot t$ iterates, where $n \in \mathbb{N}$, and $t \in \{2, 3, 4, ...\}$.

Due to the large number of cases when Figure 3 does not satisfy the prior criteria, I will be focusing on the special case that X is purposefully playing a ZD strategy, and Y, with no input from X, tries to 'get away' by causing the graph to be reducible or periodic. Therefore, graph changes are to solely to be forced by one player. First, reducibility is considered by ocnsidering potential graph topologies. This means considered one isolated nodes with 3 connected nodes, 2 pairs of connected nodes, a pair of connected nodes with two isolated nodes, or all isolated nodes. Suppose, WLOG CC is determined to be isolated along with the other nodes being connected in some way. For a node to be completely isolated, it must immediately return to itself at each iterate. However, with p_i arbitrary, this is not necessarily the case. Therefore this case is not possible. Furthermore,

no isolation of any node is possible, so the only topologically possible reducible graph is that of two pairs of connected nodes. Not all two pairs of connections are possible. If CC and CD stay connected, then for CC to disconnect from DC and DD q must both be equal to 0 and 1. The same goes for having CC and DD connected. Therefore the only possible reducible graph is that of the two-node sub graphs (CC, DC), and (CD, DD). The stationary distribution, therefore, is obtained by considering the initial distribution, and the proportion of tries in each of the two subgraphs.

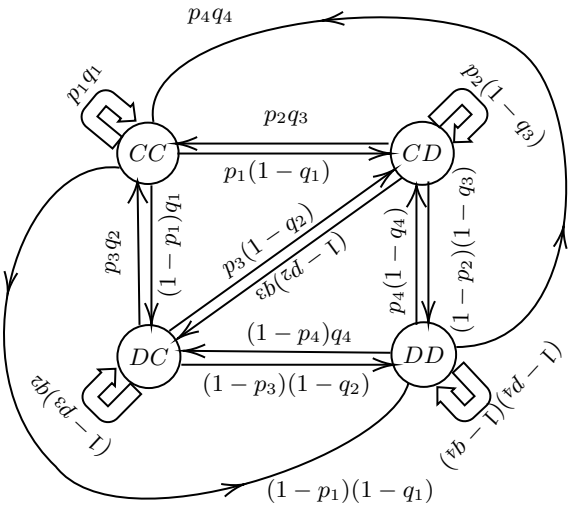

Figure 3

Now considered are periodic graphs. Periodicity is a class property, so if one element of a communicating class is aperiodic, so are the others. Considering the irreducible graph (having only one communicating class), for periodicity to occur, we must have no node being able to return to itself in a single iterate. This means only $q = (0, 0, 1, 1)$ can be played. Due to aperiodic graphs with two communicating classes only coming with the flavour $q = (1, 1, 0, 0)$, no other periodic graphs exist. Hence, to escape the linear relation of payoffs resulting from ZD strategies, $(1, 1, 0, 0), (0, 0, 1, 1)$ are the only strategies Y can play.

Now that the outlier cases that Y has full control over are fleshed out, implications of ZD strategies are now considered. First, an attempt at being in manual control of one's own score (X's) is tried, by considering the fixed strategy $\alpha S_x + \gamma \vec{1}$. Equating column values:

$$p_2 = \frac{(1 + p_4)(R - S) - p_1(P - S)}{R - P} \tag{1}$$

$$p_3 = \frac{-(1 - p_1)(T - P) - p_4(T - R)}{R - P} \tag{2}$$

$$\tag{3}$$

It is unfortunate to discover that $p_2 \geq 1$ and $p_3 \leq 0$ yields a strategy $\mathbf{p} = (1, 1, 0, 0)$. This does not guarantee X a payoff.

Alternatively, suppose X tries to be in command of Y's payoff.

X needs only to play a fixed strategy $\beta S_y + \gamma \mathbf{1}$. Equating columns:

$$p_2 = \frac{p_1(T-P) - (1+p_4)(T-R)}{R-P} \tag{4}$$

$$p_3 = \frac{(1-p_1)(P-S) + p_4(R-S)}{R-P} \tag{5}$$

$$s_Y = \frac{(1-p_1)P + p_4 R}{(1-p_1) + p_4} \tag{6}$$

Hence, as equation 3 is the weighted average of P and R, X can force Y's payoff to be any of $R \leq s_Y \leq P$, independent of the strategy Y chooses! This form of control is mathematically allowed.

Suppose X would like to enforce a ratio between her own payoffs and Y's payoffs. Specifically, Press and Dyson consider this ratio to hold for payoffs received above that of mutual noncooperation. The further restriction of $\mathbf{p} = \phi[(S_x - P\mathbf{1}) - \chi(S_y - P\mathbf{1})]$ does the trick, visibly enforcing $s_x - P = \chi(s_y - p)$. χ is the extortion factor, and ϕ a scaling constant ensuring probabilities are valid. Aside from Press and Dyson's analysis, other relationships can easily be made. For example, in the paper "Extortion and cooperation in the Prisoner's Dilemma" by Stewart and Plotkin, a relationship between payoffs below that of mutual cooperation is enforced: $(s_x - R = \chi(s_y - R)$. This results in a very successful strategy in a a tournament similar to that of Axelrod.

Returning focus to Press and Dyson, their extortionate strategy results in:

$$p_1 = 1 - \phi(\chi - 1)\frac{R-P}{P-S} \tag{7}$$

$$p_2 = 1 - \phi(1 + \chi\frac{T-P}{P-S}) \tag{8}$$

$$p_3 = \phi(\chi + \frac{T-P}{P-S}) \tag{9}$$

$$p_4 = 0 \tag{10}$$

Substituting the values found in the standard IPD, the probabilities become

$$\mathbf{p} = [1 - 2\phi(\chi - 1), 1 - \phi(4\chi + 1), \phi(\chi + 4), 0]$$

yielding a best respective score of

$$s_X = \frac{2 + 13\chi}{2 + 3\chi} - s_Y = \frac{12 + 3\chi}{2 + 3\chi}.$$

Interestingly enough, setting $\chi = 1, \phi = 0.2$ gives the well-known tit-for-tat strategy.

Press and Dyson's final remarks are about evolutionary play; that where an opponent tries to maximise payoff against an extortionate ZD strategy. Although not mentioned in the paper, analysis from simulated tournaments suggest ZD strategies to perform poorly against unforgiving players; therefore setting proper boundaries against extortionate individuals is ideal. Due to this, I re frame Press and Dyson's discussion to that of play against a ZD extortioner, where the player is unable to leave the tournament - either through reliance or naivety. The ZD player can be said to not have such characteristics, due to themselves being extortionate. In this case, the non ZD player (say, Y), tries to maximise payoff after a reactive non-cooperative string of play.

$$\frac{\partial s_Y}{\partial \mathbf{q}}|_{q=q_0} = (0, 0, 0, \frac{(T-S)(S+T-2P)}{(P-S) + \chi(T-P)})$$

It's clear that due to X (the ZD user) using an extortionate strategy, Y must cooperate amidst mutual defection. From X's perspective, once X sees Y has fallen into this trap, it can make this transition of score increase unnoticeable, by slowly changing the extortion factor. Y then must be normalised into slight extortion, turning into more extreme forms.

In the case when Y is no longer reliant on the support of X, Y can simply mutually defect. The only hope X has to re-trick Y is to convince Y to be relying on X, perhaps by show of great reward, or by playing fairly at first only to later extort. Y must hope X becomes fair, or simply refuse to play. ZD strategies therefore certainly result in some particularly scary scenarios in players of different strengths.

With the case discussions from Press and Dyson concluded, it is worth to follow up on a yet unproven assumption. Press and Dyson prove that memory $>$ 1-strategies playing against ZD strategies pose no greater threat than memory-1 strategies, and can therefore be ignored in their analysis. Intuitively, in the sense of $>$ 1-memory players playing against extortionate strategies, a player will need more time to figure out that the opponent is extortionate, before realising a best response. The extortionate player, already set in the mind of extortion, will have obtained disproportionate reward.

Mathematically this claim in now proved for arbitrary ZD strategies:

Let X and Y be random variables with realisations x, y - denoting X and Y's decision of play at any given moment. Probabilities depend solely on the joint realisation (x, y), and so expected probabilities (with respect to past histories) can be computed for each player. Once expected probabilities are found, so too can expected scores.

$$E[X = x, Y = y|H_0, H_1] = \sum_{H_0,H_1} P(x, y|H_0, H_1)P(H_0, H_1) \tag{11}$$

$$= \sum_{H_0,H_1} P(X = x|H_0)P(Y = y|H_0, H_1)P(H_0, H_1) \tag{12}$$

$$= \sum_{H_0} P(X = x|H_0)[\sum_{H_1} P(y|H_0, H_1)P(H_1|H_0)P(H_0)] \tag{13}$$

$$= \sum_{H_0} P(x|H_0)P(y|H_0)P(H_0) \tag{14}$$

$$= E[P(X = x, Y = y|H_0)] \tag{15}$$

Thus, the result is a game conditioned only on H_0 for both players, and so the game has the same results as if Y played longer memory strategies.

The mathematical intuition used in the proof is that the longer memory strategies can be realised as memory-1 strategies over the long-run by considering each scenario in the frequentist point of view.

I now present some concluding remarks from Press and Dyson's paper. The paper created so much excitement due to the novelty of mathematics used to enforce a relationship between payoffs of players, especially given the age of analysis of IPD. Special ZD cases, such as extortion, were also quite significant, giving a more mathematical basis for well-known strategies such as tit-for-tat, as well as opening eyes on abusive play, with some potential generalisations on IPD for non-symmetric situations of play.

It's clear that Nash equilibrium isn't a term appearing in Press and Dyson's paper, nor in most other discussions on the iterated prisoner dilemma. There are several reasons for this. This can be seen by contrasting Press and Dyson with Sigmund. Sigmund's replicator equations certainly gave

rise to NE specifying variously overlapping basins of attraction, taking into account noise, length of game play, and tournament environment; each of which took a leading role in shaping the analysis of the game. Press and Dyson on the other hand do not take these factors into account, they do not focus on multiple games of play, or tournament with many opponents, or in maximising their own payoffs, a criteria intricately linked with NE. Instead focus is made on a single player, with interest on bounding the opponents score with their own, regardless of whether this maximises the ZD protagonist. This concludes the discussion on Press and Dyson's paper.

4 Axelrod Tournament Simulation

The tournament of Axelrod is simulated. Several of my own strategies are intuited and placed to compete in Axelrod tournament simulations, along with the simulation of individual match ups between select strategies. A focus is then placed on tournaments of IPD on graphs; domains which direct which opponents players compete against, all the while simulating the normal IPD.

The first tournament of Axelrod, along with several variants, are simulated below in Figure 4. Due to some of the original strategies being too vaguely described, the precise Axelrod tournament could not be fully replicated. Apart from this, all other tournament rules are held equivalent, and scoring criteria is also identical.

We see that in our replicated original tournament, competition results vary. A distinction is still made between 'nice' strategies and 'non-nice' strategies as discussed by Axelrod, however, further complications are now less pronounced.

To discuss the other simulated variants, it is first useful to describe the other strategies. Found are several custom made strategies.

My custom strategies are heavily based on mimicking extortion, however, using TFT as a baseline, and they are all encoded in a memory-1 strategy fashion. This was decided as a result of seeing zero determinant strategies perform poorly, globally, in competition environments of many types of players, yet TFT being seen to use too little force when it could have extorted successfully. A middle ground was therefore explored. Progressive strategy versions take into consideration varying emotional automata, as well as manual improvement of strategy by way of viewing competition results. Due to the likeliness of overfitting a strategy by way of seeing it's performance, and changing it to improve score, for transparency all six strategies formulated are shown. I make clear that that intuited changes in strategy did not involve a humanised gradient descent, instead, changes were made by formulating a theory of cooperation and deception, and making numerical changes to benefit from such possibilities.

	Strategy & Score																				
	TFT	T&C	NYD.	GRO.	SHU.	S&R	FRI.	DAV.	GRA.	DOW.	FEL.	JOS.	TUL.	UNN.	RAN.	D1.	D2.	D3.	D4.	D5.	D6.	ZDE	ZD2
Original	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15								
Scores	504	500	486	482	481	478	473	472	401	391	328	304	301	282	276								
Simulated	4	6	5	2	3	1	7	8	9	10	11	13	12	14	15								
Scores	467	460	459	473	468	483	449	448	428	412	359	330	334	314	310								
Custom 1	7	8	2	1	6	3	9	10	11	12	13	17	16	19	18	14	4	5	15				
Scores	454	452	468	477	457	466	441	440	426	418	340	308	311	301	310	330	458	456	319				
Custom 2																5	4	3	7	1	2	6	
																256	347	347	255	371	367	255	
Custom 3	6	8	5	4	7	3	9	10	11	12	13	16	14	17	18					1	2		15
Scores	475	464	476	482	469	484	452	451	440	432	375	347	352	323	323					499	489		347
Custom 4	9	10	5	3	8	4	11	12	14	13	15	19	18	21	20	17	6	7	16	1	2		
Scores	469	463	480	484	472	480	456	455	431	432	351	327	330	316	321	336	472	471	337	489	489		

Figure 4: Axelrod Tournament results, with varying competitors.

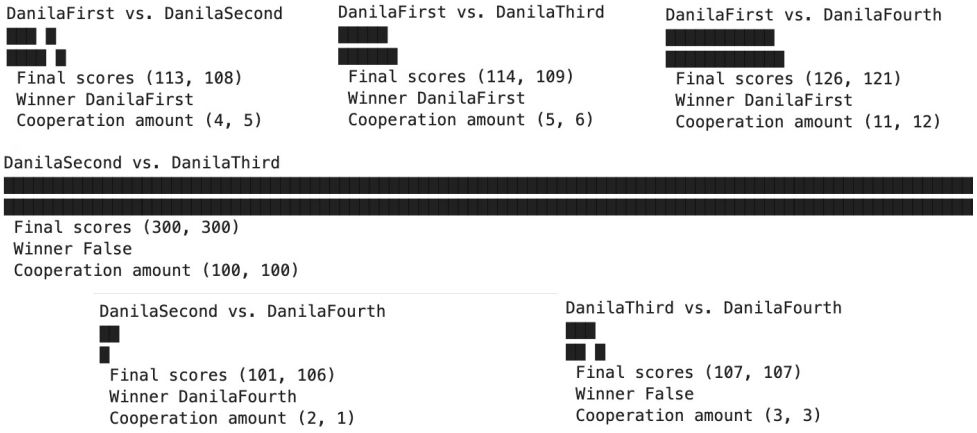

Figure 5: Fingerprints for individual IPD matches between my custom strategies, with 100 matches played.

CustomFirst represented my interpretation of extortion with probability response vector $p = (0.8, 0, 1/3, 0)$. The aim was to cause disproportionate extortion with quick to punish tactics. On average, every fifth move the strategy 'cheats' (defecting on a string of mutual cooperation). And, in a somewhat unfair way, makes the opponent work x3 as hard to regain 'trust', and move back to mutual cooperation.

CustomSecond represented my second attempt at CustomFirst, with response vector $p = (1, 0, 1/3, 0)$. This strategy was a response to my own intuition suggesting CustomFirst to be too harsh. Instead, in a non-noisy environment, it felt much more fair to continue to cooperate given a string of distortions, and disproportionately (by 3 times) punish defections. This strategy is somewhere in between TFT and the SHUBIK strategy.

CustomThird represented a natural followup to CustomSecond with response vector $p = (1, 0, 0.5, 0)$. The need for (on average) three contributions from the opponent to regain trust seemed like a far stretch, and too much work for the cleverly suspicious opponent to have a subroutine for. Due to this, a less severe punishment is instilled on defection.

CustomFourth represents a greedy version of CustomThird, which, not only is more empathetic towards the opponent as the prior, now occasionally defects. It's probability vector is $p = (0.8, 0, 1/2, 0)$.

To test the effectiveness of my first four strategies, a custom Axelrod tournament was made "Custom 1", where all four of the above strategies were submitted to the original simulated tournament. The CustomSecond and CustomThird strategies were surprisingly effective, whilst First and Fourth were not. A clear distinction between these pairs of strategies is in the fairness attribute of each. My poorly performing strategies were the very ones that sought to exploit, unfairly, from a string of mutual cooperation. The very characteristic of non-niceness, as described by Axelrod, determined the outcome of my strategies significantly.

Having remembered the 'echo into defection' effects from Axelrod's discussion of his first tournament, along with the need to make strategies nice, I formulated my last two strategies to be nice. I then decided to consider individual matches between strategies, specifically between the ones I had encoded. As shown in Figure 5, it was clear that many matches ended with long strings of mutual defections. I knew that my strategies would be relatively forgiving, along with the environment of Axelrod's original tournament, so I further made my last two strategies forgiving. To make sure these strategies were no 'pushovers', it was decided to be less forgiving against opponents who first defect, but more forgiving if previously both players defected - simulating the notion of 'let bygones be bygones'.

CustomFifth encoded the memory-1 response vector $p = (1, 0.2, 0.5, 0.5)$. It was very forgiving if both players defected in the prior round, and much less so if the other player first forgave. Due to the number of iterates per game, however, this still gave ample chances for the opponent to witness forgiveness, and for the game to restart from that of mutual defections.

CustomSixth was made directly after CustomFifth, in response to a personal vendetta that perhaps being too forgiving after a string of mutual defections would result in being too exploitative. The vector $p = (1, 0.2, 0.5, 0.3)$ its strategy.

The results of these improved strategies was remarkable! Against other Custom strategies and a Zero Determinant strategy ZDE, CustomFifth and CustomSixth score 1st and second place respectively, completely dominating all other opponents. I believe this to be the case due to the forgiveness and niceness attributes of 4 of the 7 candidates present, giving them scores approximately 100 greater than that of the non-nice players, yet still benefiting from occasional cooperations by the other.

In the case that all Custom strategies are submitted to the original Axelrod tournament, the new strategies still come out first and second. As Axelrod speculated, being nice and forgiving

pays. Axelrod's idea of echo-effects into mutual discrimination held true, too, as CustomFifth and CustomSixth strategies quickly fixed these occurrences.

Finally, the Custom 3 tournament shows, once again, CustomFifth and CustomSixth strategies dominating, if solely submitted to Axelrod's original tournament. This came as quite a delight.

I further took interested in studying optimal strategies in a variety of different environments. I noted that my Python library gave access to iterated game play on graphs. In this case, each node represents a strategy, and edges between nodes represented the limited number of other players this strategy can face. This gave the idea that Sigmund's replicator equation and analysis instantly gave great predictions towards the outcome of such games. In fact, by considering the direction of the tangent on some n-dimensional simplex with n strategies, one can mimic the proportion of strategies played by counting the proportion of strategies linked to a node of a graph. The tangent vector direction instantly shows which strategy yields the highest reward. Therefore, for studying iterated game play on graphs, one simply has to plot Sigmund's replicator equation, and for each of the graph's environments, find the direction of tangent. Iterating this through all the nodes, gives a player an advantage towards figuring out optimal strategy solely considering its neighbours. If the player is unaware of the strategies its neighbours play, yet knows the number of opponents it will play and the list of possible strategies that can be used, one may reduce the space of proportions found on the simplex to a discrete few. Each point in the reduction corresponding to an integer amount of neighbours to the player. A more rigorous analysis, although not written here, should be done.

A great number of interesting ideas using graphs can be discussed, for example simulating a disconnected graph with several communicating classes, with each 'state' being able to introduce noise to other communicating classes. Analysing robustness of the individual or state-wide strategy here would be fascinating, and would certainly be analysed in the future.

4.1 Python Code and Implementations

Python 3.8 was used, along with the packages from the 'requirements.txt' file of the "axelrod" python package. Code was exectuted on a Jupyter Notebook.

4.1.1 Imports and custom functions

```
import axelrod as axl

def resultings(results):
    import numpy as np
    j = 0
    for i in results.scores:
        print(results.ranked_names[j], np.mean(i)/len(results.scores),
        ↪  np.std(i)/len(results.scores))
        j+=1
```

4.1.2 Extracting 'axelrod' package classes to define memory-1 strategies

```
import warnings
from typing import Tuple

from axelrod.action import Action
from axelrod.player import Player
```

```python
import numpy as np

C, D = Action.C, Action.D

class MemoryOnePlayer(Player):
    name = "Generic Memory One Player"
    classifier = {
        "memory_depth": 1,  # Memory-one Four-Vector
        "stochastic": True,
        "long_run_time": False,
        "inspects_source": False,
        "manipulates_source": False,
        "manipulates_state": False,
    }

    def __init__(
        self,
        four_vector: Tuple[float, float, float, float] = None,
        initial: Action = C,
    ) -> None:

        super().__init__()
        self._initial = initial
        self.set_initial_four_vector(four_vector)

    def set_initial_four_vector(self, four_vector):
        if four_vector is None:
            four_vector = (1, 0, 0, 1)
            warnings.warn("Memory one player is set to default (1, 0, 0, 1).")

        self.set_four_vector(four_vector)

    def set_four_vector(self, four_vector: Tuple[float, float, float, float]):
        if not all(0 <= p <= 1 for p in four_vector):
            raise ValueError(
                "An element in the probability vector, {}, is not "
                "between 0 and 1.".format(str(four_vector))
            )
        self._four_vector = dict(
            zip([(C, C), (C, D), (D, C), (D, D)], four_vector)
        )

    def _post_init(self):
        # Adjust classifiers
        values = set(self._four_vector.values())
        self.classifier["stochastic"] = any(0 < x < 1 for x in values)
        if all(x == 0 for x in values) or all(x == 1 for x in values):
            self.classifier["memory_depth"] = 0

    def strategy(self, opponent: Player) -> Action:
        if len(opponent.history) == 0:
            return self._initial
        # Determine which probability to use
```

```
        p = self._four_vector[(self.history[-1], opponent.history[-1])]
        # Draw a random number in [0, 1] to decide
        try:
            return self._random.random_choice(p)
        except AttributeError:
            return D if p == 0 else C
```

4.1.3 Custom memory-1 strategies

```
class CustomFirst(MemoryOnePlayer):
    name = "CustomFirst"

    def __init__(self) -> None:
        four_vector = (0.8, 0, 0.333, 0)
        super().__init__(four_vector)
        self.set_four_vector(four_vector)


class CustomSecond(MemoryOnePlayer):
    name = "CustomSecond"

    def __init__(self) -> None:
        four_vector = (1, 0, 0.333, 0)
        super().__init__(four_vector)
        self.set_four_vector(four_vector)

class CustomThird(MemoryOnePlayer):
    name = "CustomThird"

    def __init__(self) -> None:
        four_vector = (1, 0, 0.5, 0)
        super().__init__(four_vector)
        self.set_four_vector(four_vector)

class CustomFourth(MemoryOnePlayer):
    name = "CustomFourth"

    def __init__(self) -> None:
        four_vector = (0.8, 0, 0.5, 0)
        super().__init__(four_vector)
        self.set_four_vector(four_vector)

class CustomFifth(MemoryOnePlayer):
    name = "CustomFifth"

    def __init__(self) -> None:
        four_vector = (1, 0.2, 0.5, 0.5)
        super().__init__(four_vector)
        self.set_four_vector(four_vector)

class CustomSixth(MemoryOnePlayer):
    name = "CustomSixth"
```

```
44
45      def __init__(self) -> None:
46          four_vector = (1, 0.2, 0.5, 0.3)
47          super().__init__(four_vector)
48          self.set_four_vector(four_vector)
```

4.1.4 Simulating Axelrod's original tournament

```
1 first_tournament_participants_ordered_by_reported_rank = [s() for s in
  ↪  axl.axelrod_first_strategies]
2 tournament = axl.Tournament(
3     players=first_tournament_participants_ordered_by_reported_rank, turns=200,
      ↪  repetitions=5, seed=1)
4 results = tournament.play()
5 resultings(results)
```

4.1.5 Simulating Axelrod's Custom 1 tournament

```
1 first_tournament = [s() for s in axl.axelrod_first_strategies]
2 first_tournament += [CustomFirst(), CustomSecond(), CustomThird(), CustomFourth()]
3 players = first_tournament
4 tournament = axl.Tournament(players, 100)
5 results = tournament.play()
6 resultings(results)
```

To simulate other custom tournaments, simply replace lines 1 and 2 with the needed strategies. Options used were 'CustomFifth()', 'CustomSixth()', 'axl.ZDExtortion()', and 'axl.ZDExtort2()'.

4.1.6 Simulating Custom matches

```
1 players = (CustomFirst(), CustomSecond())
2 match = axl.Match(players, 100)
3 match.play()
4 print('CustomFirst vs. CustomSecond')
5 print(match.sparklines(),'\n', 'Final scores', match.final_score(), '\n', 'Winner',
  ↪  match.winner(),'\n', 'Cooperation amount', match.cooperation())
```

For other match variants, lines 1 and 4 with the respective players.

References

[1] Robert Axelrod. "Effective Choice in the Prisoner's Dilemma". In: *Journal of Conflict Resolution* (1980), 3–25.

[2] William H Press and Freeman J Dyson. "Iterated Prisoner's Dilemma contains strategies that dominate any evolutionary opponent". In: *PNAS* 109.26 (2012).

[3] David G. Rand and Martin A. Nowak. "Evolutionary Dynamics in finite populations can explain the full range of cooperative behaviors observed in the centipede game". In: *Journal of Theoretical Biology* 300 (2012), 212–221. DOI: 10.1016/j.jtbi.2012.01.011.

[4] A. J. Stewart and J. B. Plotkin. "Extortion and cooperation in the prisoner's dilemma". In: *Proceedings of the National Academy of Sciences* 109.26 (2012), 10134–10135. DOI: 10.1073/pnas.1208087109.

[5] György Szabó and Gábor Fáth. "Evolutionary games on graphs". In: *Physics Reports* 446.4-6 (2007), 97–216. DOI: 10.1016/j.physrep.2007.04.004.

[6] Karl Sigmund. "Chapter 3, Direct Reciptrocity: The Role of Repetition". In: *The calculus of selfishness*. Princeton University Press, 2010, 49–81.

[7] URL: https://github.com/Axelrod-Python/Axelrod.