Recognising and evaluating the effectiveness of extortion in the Iterated Prisoner's Dilemma

Vincent A. Knight Nikoleta E. Glynatsi Jonathan Gillard Marc Harper

December 19, 2018

Abstract

Since the introduction of zero-determinant strategies, extortionate strategies have received considerable interest. While an interesting class of strategies, the definitions of extortionate strategies are algebraically rigid, apply only to memory-one strategies, and require complete knowledge of a strategy (memory-one cooperation probabilities). We describe a method to detect extortionate behaviour from the history of play of a strategy. When applied to a corpus of 204 strategies this method detects extortionate behaviour in well-known extortionate strategies as well others that do not fit the algebraic definition. The highest performing strategies in this corpus are able to exhibit selectively extortionate behavior, cooperating with strong strategies while exploiting weaker strategies, which no memory-one strategy can do. These strategies emerged from an evolutionary selection process and their existence contradicts widely-repeated folklore in the evolutionary game theory literature: complex strategies can be extraordinarily effective, zero-determinant strategies can be outperformed by non-zero determinant strategies, and longer memory strategies are able to outperform short memory strategies. Moreover, while resistance to extortion is critical for the evolution of cooperation, the extortion of weak opponents need not prevent cooperation between stronger opponents, and this adaptability may be crucial to maintaining cooperation in the long run.

1 Introduction

The Iterated Prisoner's Dilemma is a model for rational and evolutionary interactive behaviour, having applications in biology, the study of human social behaviour, and many other domains. Since the introduction of zero-determinant strategies in [**press2012**], extortionate strategies have received considerable interest in the literature [10]. These strategies "enforce" a difference in stationary payouts between themselves and their opponents. The definition requires a precise algebraic relationship between the probabilities of cooperation given the outcome of the previous round of play and slight alterations to these probabilities can cause a strategy to no longer satisfy the necessary equations.

In [1, 8, 9, 10, 11, 16] the true effectiveness of these strategies in an evolutionary setting was discussed. For example [1] showed that ZD strategies were not evolutionarily stable. Furthermore, in that work it was also postulated that 'evolutionarily successful ZD strategies could be designed that use longer memory to distinguish self from non-self'.

The algebraic relationships of extortion define a submanifold of $p \in \mathbb{R}^4$ which can be used broaden the definition of an extortionate strategy by requiring only that the defining cooperation probabilities of a strategy are close to an algebraically extortionate strategy, by the usual technique of orthogonal projection. Moreover, given the history of play of a strategy in an empirical matchup, we can empirically observe its four cooperation probabilities, measure the distance to the subspace of extortionate strategies, and use this distance as a measure of the extortionality of a strategy. This method can be applied to any strategy regardless of the memory depth and avoids the algebraic rigidity issues.

We apply this method to the largest known corpus of strategies for the iterated prisoner's dilemma (the Axelrod Python library [15, 17]) and show empirically that the method in fact detects extortionate strategies. A large tournament with 204 strategies was run, with which it will be demonstrated that sophisticated strategies do in fact recognise extortionate behaviour and adapt to their opponents. Further, statistical analysis of these strategies in the context of evolutionary dynamics demonstrates the importance of adaptability to achieve evolutionary stability. All of the code and data discussed in Section 3 is open sourced, archived, and written according to best scientific principles [28]. The data archive can be found at [13]. In Section 4, this large tournament is complemented with evolutionary dynamics that offer some insight in to the effectiveness of extortionate strategies.

Several theoretical insights emerge from this work. Infamously, extortionate strategies do not play well with themselves. In [**press2012**], Press and Dyson claim that a player with a "theory of mind" would rationally chose to cooperate against an opponent that also has knowledge of zero determinant strategies to avoid sustained mutual defection. While not possible for memory-one strategies, we show that this behavior is exhibited by relatively simple strategies of memory depth two or more via strategies that emerged from an evolutionary selection process. Similarly, in [1], Adami and Hintze suggest that there may exist strategies that are able to selectively behave extortionately to some opponents and cooperatively to others. We show that this is indeed the case for the same evolved strategies.

It seems that humans have trouble defining such strategies but evolution is able to simply by optimizing for total payoff in IPD interactions. Accordingly, while resistance to extortionate behavior appears critical to the evolution of cooperation, there is no prohibition on selectively extorting weaker opponents, even in population dynamics, and this behavior is evolutionarily advantageous.

2 Recognising Extortion

Zero Determinant (ZD) strategies are a special case of memory-one strategies, which are defined by elements of $\mathbb{R}^4$ mapping a state of $\{C, D\}^2$, corresponding to the prior round of play, to a probability of cooperating in the next round. A match between two such strategies creates a Markov chain with transient states $\{C, D\}^2$. The main result of [22] is that given two memory-one players $p, q \in \mathbb{R}^4$, a linear relationship between the players' scores can, in some cases, be forced by one of the players for specific choices of these probabilities.

Using the notation of [22], the utilities for player p are given by $S_x = (R, S, T, P)$ and for player q by $S_y = (R, T, S, P)$ and the stationary scores of each player are given by S_X and S_Y respectively. The main result of [22] is that if

$$\tilde{p} = \alpha S_x + \beta S_y + \gamma \tag{1}$$

or

$$\tilde{q} = \alpha S_x + \beta S_y + \gamma \tag{2}$$

where $\tilde{p} = (1 - p_1, 1 - p_2, p_3, p_4)$ and $\tilde{q} = (1 - q_1, 1 - q_2, q_3, q_4)$ then:

$$\alpha S_X + \beta S_Y + \gamma = 0 \tag{3}$$

Extortionate strategies are defined as follows. If this relationship is satisfied

$$\gamma = -P(\alpha + \beta) \tag{4}$$

then the player can ensure $(S_X - P) = \chi(S_Y - P)$ where:

$$\chi = \frac{-\beta}{\alpha} \tag{5}$$

Thus, if (4) holds and $\chi > 1$ a player is said to extort their opponent. First, the reverse problem is considered: given a $p \in \mathbb{R}^4$ can one determine if the associated strategy is attempting to act in an extortionate way?

2.1 Subspace of Extortionate Strategies

Constraints (1) and (4) correspond to:

$$\begin{align}
\tilde{p}_1 &= \alpha R + \beta R - P(\alpha + \beta) \tag{6}\\
\tilde{p}_2 &= \alpha S + \beta T - P(\alpha + \beta) \tag{7}\\
\tilde{p}_3 &= \alpha T + \beta S - P(\alpha + \beta) \tag{8}\\
\tilde{p}_4 &= \alpha P + \beta P - P(\alpha + \beta) = 0 \tag{9}
\end{align}$$

Equation (9) ensures that $p_4 = \tilde{p}_4 = 0$. Equations (6-8) can be used to eliminate α, β, giving:

$$\tilde{p}_1 = \frac{(R - P)(\tilde{p}_2 + \tilde{p}_3)}{S + T - 2P} \tag{10}$$

with:

$$\chi = \frac{\tilde{p}_2(P - T) + \tilde{p}_3(S - P)}{\tilde{p}_2(P - S) + \tilde{p}_3(T - P)} \tag{11}$$

Given a strategy $p \in \mathbb{R}^4$ equations (9-11) can be used to check if a strategy is extortionate. The conditions correspond to:

$$\begin{align}
p_1 &= \frac{(R - P)(p_2 + p_3) - R + T + S - P}{S + T - 2P} \tag{12}\\
p_4 &= 0 \tag{13}\\
1 &> p_2 + p_3 \tag{14}
\end{align}$$

The algebraic steps necessary to prove these results are available in the supporting materials, and note that an equivalent formulation was obtained in [1].

All extortionate strategies reside on a triangular (14) plane (12) in 3 dimensions (13). Using this formulation it can be seen that a necessary (but not sufficient) condition for an extortionate strategy is that it cooperates on average less than 50% of the time when in a state of disagreement with the opponent (14).

As an example, consider the known extortionate strategy $p = (8/9, 1/2, 1/3, 0)$ from [25] which is referred to as `Extort-2`. In this case, for the standard values of $(R, S, T, P) = (3, 0, 5, 1)$ constraint (12) corresponds to:

$$p_1 = \frac{2(p_2 + p_3) + 1}{3} = \frac{2(1/2 + 1/3) + 1}{3} = \frac{8}{9} \tag{15}$$

It is clear that in this case all constraints hold. As a counterexample, consider the strategy that cooperates 25% of the time: $p = (1/4, 1/4, 1/4, 1/4)$ obeys (14) but is not extortionate as:

$$p_1 \neq \frac{2(p_2 + p_3) + 1}{3} = \frac{2(1/4 + 1/4) + 1}{3} = \frac{2}{3} \tag{16}$$

2.2 Measuring Extortion from the History of Play

Not all strategies are memory-one strategies but it is possible to measure a given p from any set of interactions between two strategies. This approach can then be used to confirm that a given strategy is acting in an extortionate manner even if it is not a memory-one strategy. However, in practice, if an exact form for p is not known but measured from observed plays of the game then measurement and/or numerical error might lead to an extortionate strategy not being confirmed as such. [1]

As an example consider Table 1 which shows some actual plays of Extort-2 ($p = (8/9, 1/2, 1/3, 0)$) against an alternating strategy ($p = (0, 0, 1, 1)$). In this particular instance the measured value of p for the known extortionate strategy would be: $(2/2, 1/5, 3/8, 0/4)$ which does not fit the definition of a ZD strategy.

Turn	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
(8/9, 1/2, 1/3, 0)	C	C	D	D	D	C	D	D	D	D	D	C	C	C	D	D	D	C	D	D
Alternator	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C	D

Table 1: A seeded play of 20 turns of two strategies.

Note that measurement of behaviour might in some cases lead to missing values. For example the strategy $p = (8/9, 1/2, 1/3, 0)$ when playing against an opponent that always cooperates will in fact never visit any state which would allow measurement p_3 and p_4. To overcome this, it is proposed that if s is a state that is not visited than p_s is approximated using a sensible prior or imputation. In Section 3 the overall cooperation rate is used. Another approach to overcoming this measurement error would be to measure our strategies in a sufficiently noisy environment.

We can measure how close a strategy is to being extortionate with a bit of linear algebra. Essentially we attempt to find $x = (\alpha, \beta)$ and $p^* = (\tilde{p}_1 - 1, \tilde{p}_2 - 1, \tilde{p}_3, \tilde{p}_4)$ such that

$$Cx = p^* \tag{17}$$

where C corresponds to equations (6-8) and is given by:

$$C = \begin{bmatrix} R-P & R-P \\ S-P & T-P \\ T-P & S-P \\ 0 & 0 \end{bmatrix} \tag{18}$$

Note that in general, equation (17) will not necessarily have a solution. From the Rouché-Capelli theorem if there is a solution it is unique since $\text{rank}(C) = 2$ which is the dimension of the variable x. The best fitting x^* is defined by:

$$x^* = \text{argmin}_{x \in \mathbb{R}^2} \|Cx - p^*\|_2^2 \tag{19}$$

Known results [18, 24, 27] yield x^*, corresponding to the nearest extortionate strategy to the measured p. It is in fact a normal projection of p on to the plane defined by (12).

$$x^* = \left(C^T C\right)^{-1} C^T p^* \tag{20}$$

The squared norm of the remaining error is referred to as sum of squared errors of prediction (SSE):

[1] Comparing theoretic and actual plays of the IPD is not novel, see for example [23].

$$\text{SSE} = \|Cx^* - p^*\|_2^2 \tag{21}$$

This gives expressions for α, β as $\alpha = x_1^*$ and $\beta = x_2^*$ thus the conditions for a strategy to be acting extortionately becomes:

$$-x_2^* < x_1^* \tag{22}$$

A further known result [18, 24, 27] gives an expression for SSE:

$$\text{SSE} = {p^*}^T p^* - p^* C \left(C^T C\right)^{-1} C^T p^* = {p^*}^T p^* - p^* C x^* \tag{23}$$

Using this approach, the memory-one representation $p \in \mathbb{R}^4$ of any strategy against any other can can be measured and if (22) holds then (23) can be used to identify if a strategy is acting extortionately. For a measured p, SSE corresponds to the best fitting α, β. Suspicion of extortion then corresponds to a threshold on SSE and a comparison of the measured $\chi = \frac{-\beta}{\alpha}$.

3 Numerical experiments

[25] presents results from a tournament with 19 strategies with specific consideration given to ZD strategies. This tournament is reproduced here using the Axelrod-Python library [15]. To obtain a good measure of the corresponding transition rates for each strategy all matches have been run for 2000 turns and every match has been repeated 60 times. All of this interaction data is available at [13]. Note that in the interest of open scientific practice, [13] also contains interaction data for noisy and probabilistic ending interactions which are not investigated here.

Figure 1 shows the SSE values for all the strategies in the tournament, as reported in [25] the extortionate strategy Extort-2 gains a large number of wins. Notice that the mean SSE for Extort-2 is approximately zero, while for the always cooperating strategy Cooperator the SSE is far from zero.

Next we investigate a tournament with 204 strategies. The results of this analysis are shown in Figure 2. The top ranking strategies by number of wins act in an extortionate way (but not against all opponents) and it can be seen that a small subgroup of strategies achieve mutual defection. All the top ranking strategies according to score achieve mutual cooperation and do not extort each other, however they **do** exhibit extortionate behaviour towards a number of the lower ranking strategies.

Figure 3 shows the relationship between χ and SSE. Note that while a strategy may attempt to act extortionately, not all opponents can be extorted. For example, a strategy that always defects never receives a lower score than its opponent, and therefore $\chi < 1$ for the would-be extortionate opponent. Accordingly, a low SSE indicates extortionate behavior rather than successful extortion. This is why there is not a strong correlation between SSE and χ in Figure 2.

A detailed look at selected strategies is given in Table 2 and Figure 4. The high scoring strategies presented have a large variation in SSE whilst the ZD strategies have a low score but high probability of winning. A version of Figure 4 with all the strategies considered is available in the appendix and gives the same conclusion. This evidences an idea proposed in [1]: sophisticated strategies are able to recognise their opponent and defend themselves against extortion. The high ranking strategies were in fact trained to maximise score [7] which seems to have created strategies able to extort weaker strategies whilst cooperating with stronger ones. Indeed unconditional extortion is self defeating.

4 Evolutionary dynamics

4.1 Replicator Dynamics

From the large number of interactions a payoff matrix S can be measured where S_{ij} denotes the score (using standard values of $(R, S, T, P) = (3, 0, 5, 1)$) of the ith strategy against the jth strategy. This defines a fitness landscape for which the replicator equation describes the evolution of a population of strategies:

$$\frac{dx_i}{dt} = x_i((Sx)_i - x^T Sx) \tag{24}$$

Equation (24) is solved numerically through an integration technique described in [21] until a stationary vector $x = s$ is found. Figure 5 shows the stationary probabilities for each strategy ranked by score. It is clear to see that only the high ranking strategies survive the evolutionary process (in fact, only 39 have a stationary probability value greater than 10^{-2}).

Figure 6 plots the mean and variance of SSE against the stationary probabilities s of (24). Strategies that perform strongly according to equation (24) seem to be strategies that are able to modify their memory one representation depending on the opponent. While the specific memory-one representation might not be one that acts extortionately, a high SSE does imply that a strategy is not extortionate. A general linear model is shown in Table 3 with stronger predictive power and giving similar conclusions.

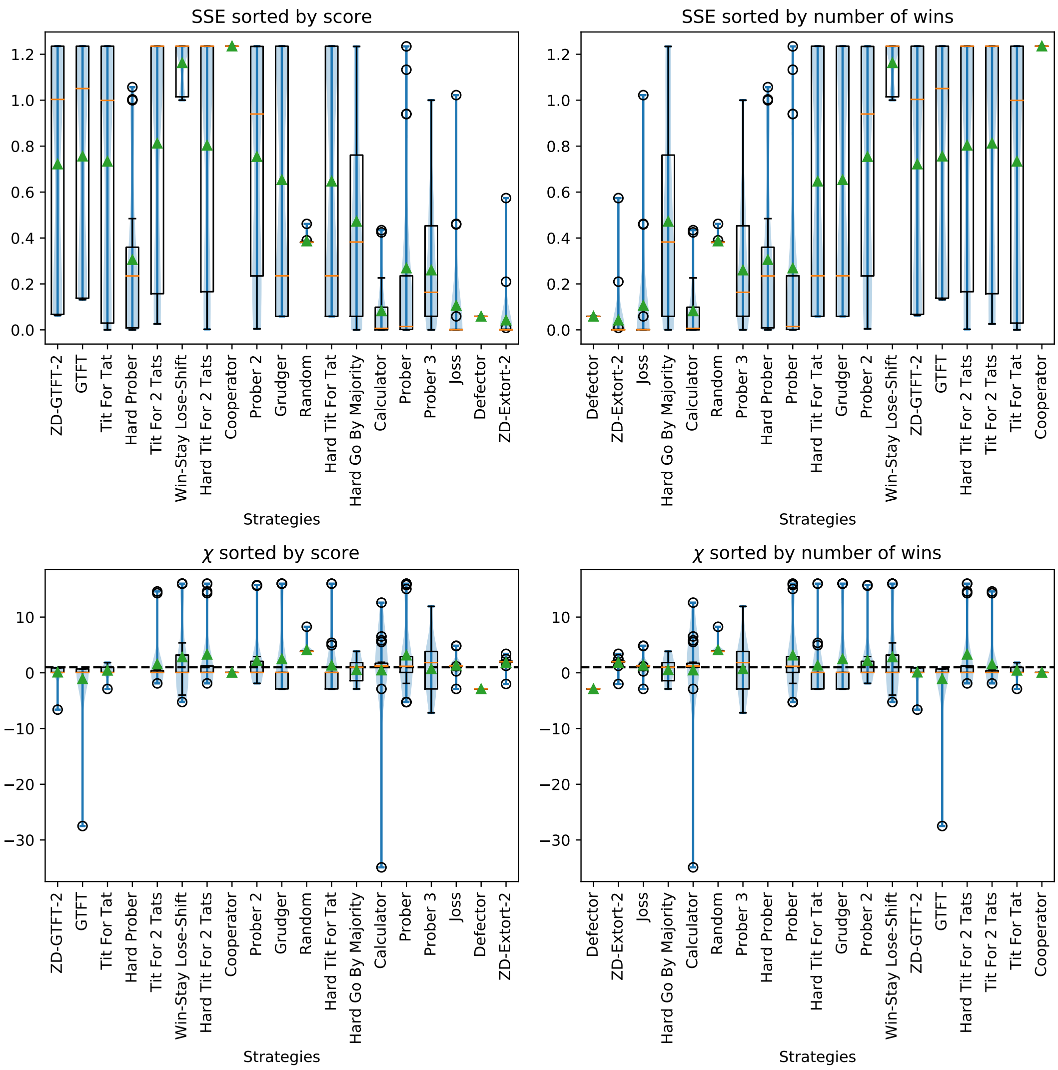


Figure 1: SSE and χ for [25], ordered both by number of wins and overall score. The dashed line shows the $\chi = 1$ boundary highlighting which strategies act in an extortionate manner. The strategies which a low variation in SSE and high χ win the most matches, although even the known extortionate strategy does not act in a perfectly extortionate manner in all matches. The strategies with a high score have a large variation in SSE.

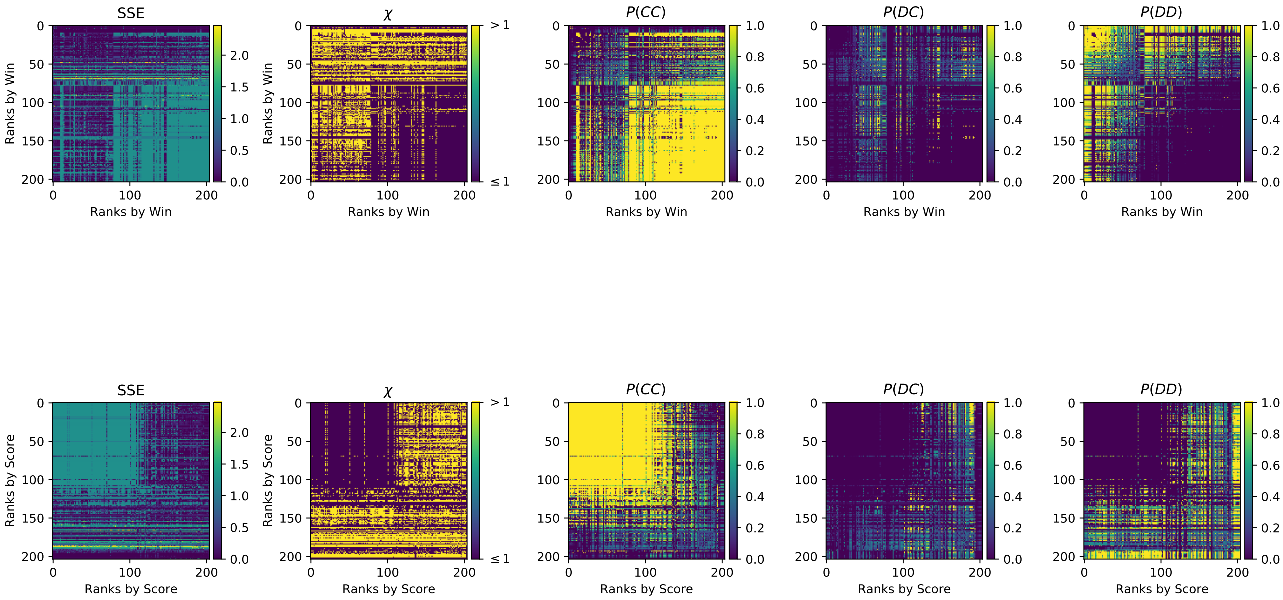


Figure 2: SSE, χ and state probabilities for the strategies for the full tournament. Note that $P(CD)$ is not shown as it corresponds to the transpose of $P(DC)$. The strategies with high number of wins have a low SSE and $\chi > 1$. The strategies with a high score have a high SSE against the other high scoring strategies but a low SSE and high χ against the low scoring strategies.

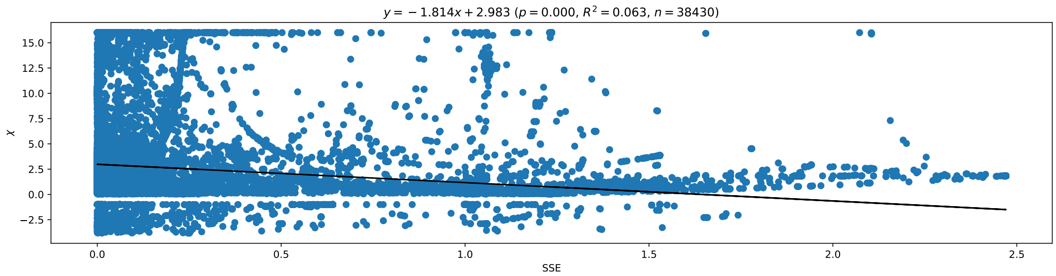


Figure 3: Linear relationship between SSE and χ, note that 5% and 95% confidence levels are considered to ignore extreme values of χ. Whilst a significant relationship between SSE and χ exists it is clear that having a low SSE does not guarantee extortionate behaviour.

Rank	Name	Score per turn	P(Win)	$P(CC)$	Median χ	5% CI SSerror	Mean SSerror	Std SSerror	95% CI SSerror
1	EvolvedLookerUp2_2_2	2.944	0.230	0.673	0.063	0.235	1.057	0.400	1.235
2	Evolved HMM 5	2.944	0.205	0.718	0.063	0.078	0.796	0.542	1.235
3	PSO Gambler 2_2_2	2.913	0.204	0.624	0.063	0.084	0.899	0.505	1.235
4	PSO Gambler Mem1	2.908	0.211	0.715	0.063	0.065	0.705	0.577	1.235
5	PSO Gambler 1_1_1	2.906	0.221	0.696	0.063	0.055	0.737	0.544	1.235
7	Evolved ANN 5	2.893	0.225	0.682	0.063	0.001	0.804	0.578	1.235
31	ZD-GTFT-2	2.721	0.000	0.806	0.063	0.064	0.786	0.537	1.235
45	ZD-GEN-2	2.689	0.016	0.801	0.063	0.015	0.694	0.599	1.235
47	Eatherley	2.682	0.000	0.828	0.063	0.007	0.895	0.474	1.235
69	Tit For Tat	2.638	0.000	0.723	0.063	0.000	0.773	0.549	1.235
75	Grumpy	2.630	0.075	0.793	0.063	0.000	0.978	0.495	1.235
88	Win-Stay Lose-Shift	2.616	0.099	0.649	0.063	1.000	1.172	0.164	1.235
103	Eventual Cycle Hunter	2.565	0.067	0.770	0.063	0.001	0.728	0.597	1.235
107	Tricky Level Punisher	2.537	0.062	0.828	0.063	0.000	0.710	0.595	1.235
127	Adaptive	2.272	0.500	0.363	-1.000	0.000	0.084	0.098	0.274
169	Bully	1.970	0.381	0.141	-1.000	0.200	1.373	0.375	1.529
179	Alternator	1.945	0.392	0.157	3.857	0.779	1.332	0.347	1.529
181	Negation	1.941	0.356	0.141	-1.000	1.130	1.470	0.288	1.529
183	Cycler DC	1.931	0.324	0.149	3.857	0.382	1.279	0.374	1.529
188	Hopeless	1.908	0.352	0.261	1.833	1.235	2.247	0.372	2.471
194	Gradual Killer	1.892	0.354	0.439	0.063	0.000	0.254	0.326	1.004
196	Aggravater	1.879	0.930	0.087	-2.889	0.059	0.163	0.256	1.023
200	ZD-Extort-2	1.821	0.851	0.179	2.005	0.000	0.019	0.094	0.010
201	ZD-Extort-4	1.820	0.865	0.106	4.003	0.000	0.021	0.069	0.204
202	ZD-Extort3	1.810	0.862	0.133	3.028	0.000	0.015	0.070	0.017
203	Defector	1.808	0.929	0.000	-2.889	0.059	0.059	0.000	0.059
204	Handshake	1.806	0.870	0.046	-2.888	0.000	0.126	0.288	1.200

Table 2: Summary of results for a selected list of strategies. Similarly to Figure 1, the high scoring strategies have a higher standard deviation of SSE. The strategies with a large number of wins have a low SSE and low variation of SSE. Note that a value of $\chi = 0.063$ and SSE $= 1.235$ corresponds to a vector $p = (1, 1, 1, 1)$ which highlights that the high scoring strategies, adapt and in fact cooperate often. A graphical representation of this table is given in Figure 4).

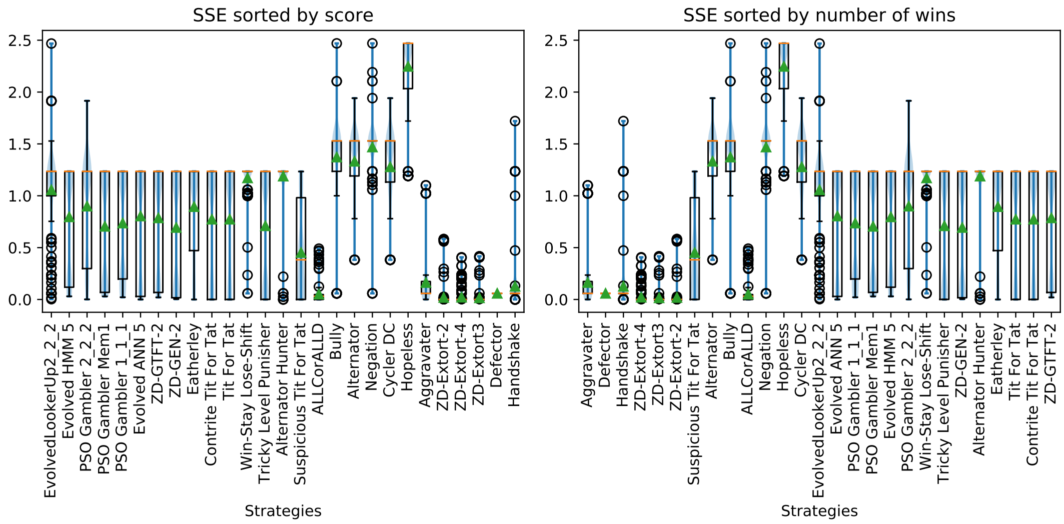


Figure 4: SSE and χ for the same selection of strategies as Table 2. A similar conclusion to that of Figure 1 can be made: the strategies that win often do not have a large variation of SSE whilst the strategies that score highly do.

Figure 5: Stationary distribution of the replicator dynamics (24): strategies are ordered by score. Note that strategies that make use of the knowledge of the length of the game are removed from this analysis as they have an evolutionary advantage.

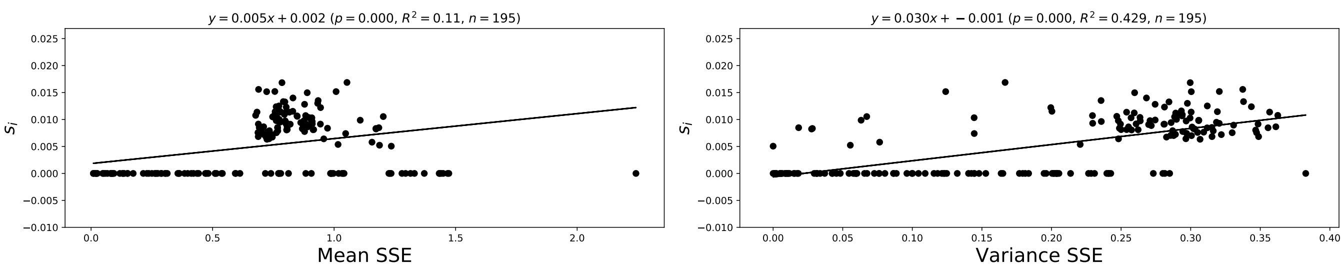


Figure 6: Linear regression analysis of stationary probabilities of (24) against the mean, and variance of SSE (for a given strategy).

Dep. Variable:	s_i	**R-squared:**	0.648
Model:	OLS	**Adj. R-squared:**	0.642
Method:	Least Squares	**F-statistic:**	117.0
Date:	Wed, 19 Dec 2018	**Prob (F-statistic):**	5.00e-43
Time:	11:06:00	**Log-Likelihood:**	851.41
No. Observations:	195	**AIC:**	-1695.
Df Residuals:	191	**BIC:**	-1682.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	0.0007	0.001	1.137	0.257	-0.000	0.002
('residual', 'mean')	-0.0134	0.002	-8.369	0.000	-0.017	-0.010
('residual', 'median')	0.0139	0.001	10.433	0.000	0.011	0.017
('residual', 'var')	0.0069	0.003	2.402	0.017	0.001	0.013

Omnibus:	17.190	**Durbin-Watson:**	1.664
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	25.453
Skew:	0.530	**Prob(JB):**	2.97e-06
Kurtosis:	4.418	**Cond. No.**	23.7

Table 3: General linear model. This shows that strategies with a low mean and high median are more likely to survive the evolutionary dynamics. This corresponds to negatively skewed distributions of SSE which again highlights the importance of adaptability.

4.2 Finite Population Dynamics: Moran Process

In [16] a large data set of pairwise fixation probabilities in the Moran process is made available at [14] Figure 7 shows linear models fitted to three summary measures of SSE and the mean (over population size N and opponents) value of $x_1 \cdot N$. This specific measure of fixation is chosen as x_1 is usually compared to the neutral fixation probability of $1/N$. As was noted in [16], the specific case of $N = 2$ differs from all other population sizes which is why it is presented in isolation. Similarly to the conclusions from Figure 6 we note that there is a significant relationship between the variability of SSE and the ability for a strategy to become fixed.

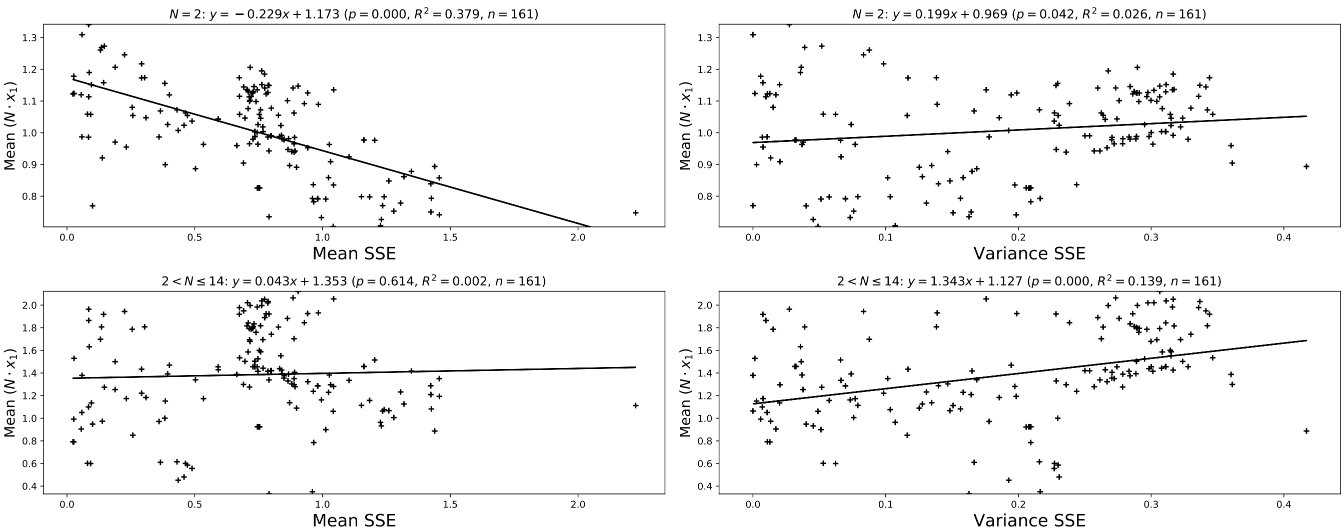


Figure 7: Linear regression analysis of pairwise fixation probabilities from [16] against the mean and variance of SSE (for a given strategy averaged over all opponents and population sizes).

Using recursive feature elimination a better linear model can be found that compliments these results. This is shown in Table 4.

These findings confirm the work of [16] in which sophisticated strategies resist evolutionary invasion of shorter memory strategies. This also confirms the work of [1, 10] which proved that ZD strategies where not evolutionarily stable due to the fact that they score poorly against themselves.

The work also provides strong evidence to the importance of adaptability: strategies that offer a variety of behaviours corresponding to a higher standard deviation of SSE are significantly more likely to survive the evolutionary process. This corresponds to the following quote of [4]:

> "It is not the most intellectual of the species that survives; it is not the strongest that survives; but the species that survives is the one that is able to adapt to and to adjust best to the changing environment in which it finds itself."

Dep. Variable:	mean	**R-squared:**	0.319
Model:	OLS	**Adj. R-squared:**	0.310
Method:	Least Squares	**F-statistic:**	36.53
Date:	Mon, 19 Nov 2018	**Prob (F-statistic):**	9.74e-14
Time:	16:08:50	**Log-Likelihood:**	-42.272
No. Observations:	159	**AIC:**	90.54
Df Residuals:	156	**BIC:**	99.75
Df Model:	2		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	1.2815	0.056	22.993	0.000	1.171	1.392
('residual', 'mean')	-1.0620	0.145	-7.323	0.000	-1.348	-0.776
('residual', 'median')	0.9037	0.106	8.535	0.000	0.695	1.113

Omnibus:	2.302	**Durbin-Watson:**	1.716
Prob(Omnibus):	0.316	**Jarque-Bera (JB):**	1.850
Skew:	-0.199	**Prob(JB):**	0.397
Kurtosis:	3.348	**Cond. No.**	11.2

Table 4: General linear model. This shows that strategies with a high mean and low median are likely to be evolutionarily stable. This corresponds to negatively skewed distributions of SSE which again highlights the importance of adaptability.

5 Conclusion

This work defines an approach to measure whether or not a player is playing a strategy that corresponds to an extortionate strategy as defined in [22]. All extortionate strategies have been classified as lying on a triangular plane. This rigorous classification fails to be robust to small measurement error, thus a statistical approach is proposed approximating the solution of a linear system. This method was applied to a large number of pairwise interactions.

The work of [22], while showing that a clever approach to taking advantage of another memory-one strategy exists, is not the full story. Though the elegance of this result is very attractive, just as the simplicity of the victory of Tit For Tat in Axelrod's original tournaments was, it is incomplete and in the author's opinions, has been oversimplified and overgeneralized in subsequent work. Extortionate strategies achieve a high number of wins but they do generally not achieve a high score and fail to be evolutionarily stable.

Rather more sophisticated strategies are able to adapt to a variety of opponents and act extortionately only against weaker strategies while cooperating with like-minded strategies that are not susceptible to extortion. This adaptability may be key to maintaining sustained cooperation, as some of these strategies emerged naturally from evolutionary processes trained to maximize payoff in IPD tournaments and fixation in population dynamics.

Following Axelrod's seminal work [2, 3], it was commonly thought that evolutionary cooperation required strategies that followed a simple set of rules. The discovery/definition of extortionate strategies [22] seemingly showed that complex strategies could be taken advantage of. In this manuscript it has been shown that not only is it possible to detect and prevent extortionate behaviour but that more complex strategies can be evolutionary stable. The complex strategies in question were obtained through reinforcement learning approaches [7, 16]. Thus, this demonstrates that it is possible to recognise extortion, both theoretically using SSE but also that this ability can develop through reinforcement learning. It seems human difficulty in directly developing effective complex strategies has been incorrectly generalized to a weakness in complex strategies themselves, which is demonstrable not the case. In fact, complex strategies can be the most effective against a diverse set of opponents.

In closing, the authors wish to emphasize the the role of comprehensive simulations to temper theoretical results from overgeneralization, and perhaps more importantly, the ability of simulations to provide insights that are difficult to obtain from theory.

Acknowledgements

The following open source software libraries were used in this research:

- The Axelrod [15, 17] library (IPD strategies and tournaments).
- The sympy library [19] (verification of all symbolic calculations).
- The matplotlib [6] library (visualisation).
- The pandas [26], dask [5] and NumPy [20] libraries (data manipulation).
- The SciPy [12] library (numerical integration of the replicator equation).

This work was performed using the computational facilities of the Advanced Research Computing @ Cardiff (ARCCA) Division, Cardiff University.

References

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

[2] Robert Axelrod. "Effective Choice in the Prisoner's Dilemma". In: *Journal of Conflict Resolution* 24.1 (Mar. 1980), pp. 3–25. DOI: 10.1177/002200278002400101.

[3] Robert Axelrod. "More Effective Choice in the Prisoner's Dilemma". In: *Journal of Conflict Resolution* 24.3 (Sept. 1980), pp. 379–403. DOI: 10.1177/002200278002400301.

[4] Charles Darwin. "ORIGIN OF SPECIES." In: *The Athenaeum* 2174 (1869), pp. 861–861.

[5] Dask Development Team. *Dask: Library for dynamic task scheduling*. 2016. URL: http://dask.pydata.org.

[6] Michael Droettboom et al. *Matplotlib/Matplotlib V2.2.2*. 2018. DOI: 10.5281/zenodo.1202077.

[7] Marc Harper et al. "Reinforcement learning produces dominant strategies for the Iterated Prisoner's Dilemma". In: *PLOS ONE* 12.12 (Dec. 2017). Ed. by Yong Deng, e0188046. DOI: 10.1371/journal.pone.0188046.

[8] C. Hilbe, M. A. Nowak, and K. Sigmund. "Evolution of extortion in Iterated Prisoner's Dilemma games". In: *Proceedings of the National Academy of Sciences* 110.17 (Apr. 2013), pp. 6913–6918. DOI: 10.1073/pnas.1214834110.

[9] Christian Hilbe, Martin A Nowak, and Arne Traulsen. "Adaptive dynamics of extortion and compliance". In: *PloS one* 8.11 (2013), e77886.

[10] Christian Hilbe, Arne Traulsen, and Karl Sigmund. "Partners or rivals? Strategies for the iterated prisoner's dilemma". In: *Games and economic behavior* 92 (2015), pp. 41–52.

[11] Genki Ichinose and Naoki Masuda. "Zero-determinant strategies in finitely repeated games". In: *Journal of theoretical biology* 438 (2018), pp. 61–77.

[12] Eric Jones, Travis Oliphant, Pearu Peterson, et al. *SciPy: Open source scientific tools for Python*. [Online; accessed ¡today¿]. 2001–. URL: http://www.scipy.org/.

[13] Vincent Knight. *Raw data for: "Suspicion: Recognising and evaluating the effectiveness of extortion in the Iterated Prisoner's Dilemma"*. June 2018. DOI: 10.5281/zenodo.1297075. URL: https://doi.org/10.5281/zenodo.1297075.

[14] Vincent Knight, Marc Harper, and Nikoleta E. Glynatsi. *Data for: Evolution Reinforces Cooperation with the Emergence of Self-Recognition Mechanisms: an empirical study of the Moran process for the iterated Prisoner's dilemma using reinforcement learning*. Nov. 2017. DOI: 10.5281/zenodo.1040129. URL: https://doi.org/10.5281/zenodo.1040129.

[15] Vincent Knight et al. "An Open Framework for the Reproducible Study of the Iterated Prisoner's Dilemma". In: *Journal of Open Research Software* 4 (Aug. 2016). DOI: 10.5334/jors.125.

[16] Vincent Knight et al. "Evolution Reinforces Cooperation with the Emergence of Self-Recognition Mechanisms: an empirical study of the Moran process for the iterated Prisoner's dilemma". In: 2017.

[17] Vince Knight et al. *Axelrod-Python/Axelrod: V4.2.0*. 2018. DOI: 10.5281/zenodo.1252994.

[18] Michael H Kutner, Chris Nachtsheim, and John Neter. *Applied linear regression models*. McGraw-Hill/Irwin, 2004.

[19] Aaron Meurer et al. "SymPy: symbolic computing in Python". In: *PeerJ Computer Science* 3 (Jan. 2017), e103. DOI: 10.7717/peerj-cs.103.

[20] Travis E. Oliphant. *Guide to NumPy: 2nd Edition*. CreateSpace Independent Publishing Platform, 2015. ISBN: 9781517300074. URL: https://www.amazon.com/Guide-NumPy-Travis-Oliphant-PhD/dp/151730007X?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=151730007X.

[21] Linda Petzold. "Automatic Selection of Methods for Solving Stiff and Nonstiff Systems of Ordinary Differential Equations". In: *SIAM Journal on Scientific and Statistical Computing* 4.1 (Mar. 1983), pp. 136–148. DOI: 10.1137/0904010.

[22] W. H. Press and F. J. Dyson. "Iterated Prisoner's Dilemma contains strategies that dominate any evolutionary opponent". In: *Proceedings of the National Academy of Sciences* 109.26 (May 2012), pp. 10409–10413. DOI: 10.1073/pnas.1206569109.

[23] David G. Rand and Martin A. Nowak. "Human cooperation". In: *Trends in Cognitive Sciences* 17.8 (Aug. 2013), pp. 413–425. DOI: 10.1016/j.tics.2013.06.003.

[24] Calyampudi Radhakrishna Rao. *Linear statistical inference and its applications*. Vol. 2. Wiley New York, 1973.

[25] 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 (June 2012), pp. 10134–10135. DOI: 10.1073/pnas.1208087109.

[26] Data Structures et al. “PROC. OF THE 9th PYTHON IN SCIENCE CONF. (SCIPY 2010)”. In: 2010.

[27] Jon Wakefield. *Bayesian and frequentist regression methods*. Springer Science & Business Media, 2013.

[28] Greg Wilson et al. “Best Practices for Scientific Computing”. In: *PLoS Biology* 12.1 (Jan. 2014). Ed. by Jonathan A. Eisen, e1001745. DOI: 10.1371/journal.pbio.1001745.

Proof of algebraic condition for extortionate strategies

The defining equations for an extortionate strategy are:

$$\tilde{p}_1 = \alpha(R - P) + \beta(R - P) \tag{1}$$
$$\tilde{p}_2 = \alpha(S - P) + \beta(T - P) \tag{2}$$
$$\tilde{p}_3 = \alpha(T - P) + \beta(S - P) \tag{3}$$
$$\tilde{p}_4 = 0 \tag{4}$$

Using equation (2), α is isolated

$$\alpha = \frac{-\beta(P - T) - \tilde{p}_2}{P - S} \tag{5}$$

Substituting this value in to equation (3), β is isolated:

$$\beta = -\frac{P\tilde{p}_1 - P\tilde{p}_2 + S\tilde{p}_2 - T\tilde{p}_1}{(S - T)(2P - S - T)} \tag{6}$$

Substituting this back in to (5) gives:

$$\alpha = \frac{-\tilde{p}_2 + (P - T)(P\tilde{p}_1 - P\tilde{p}_2 + S\tilde{p}_2 - T\tilde{p}_1)}{(S - T)(2P - S - T)(P - S)} \tag{7}$$

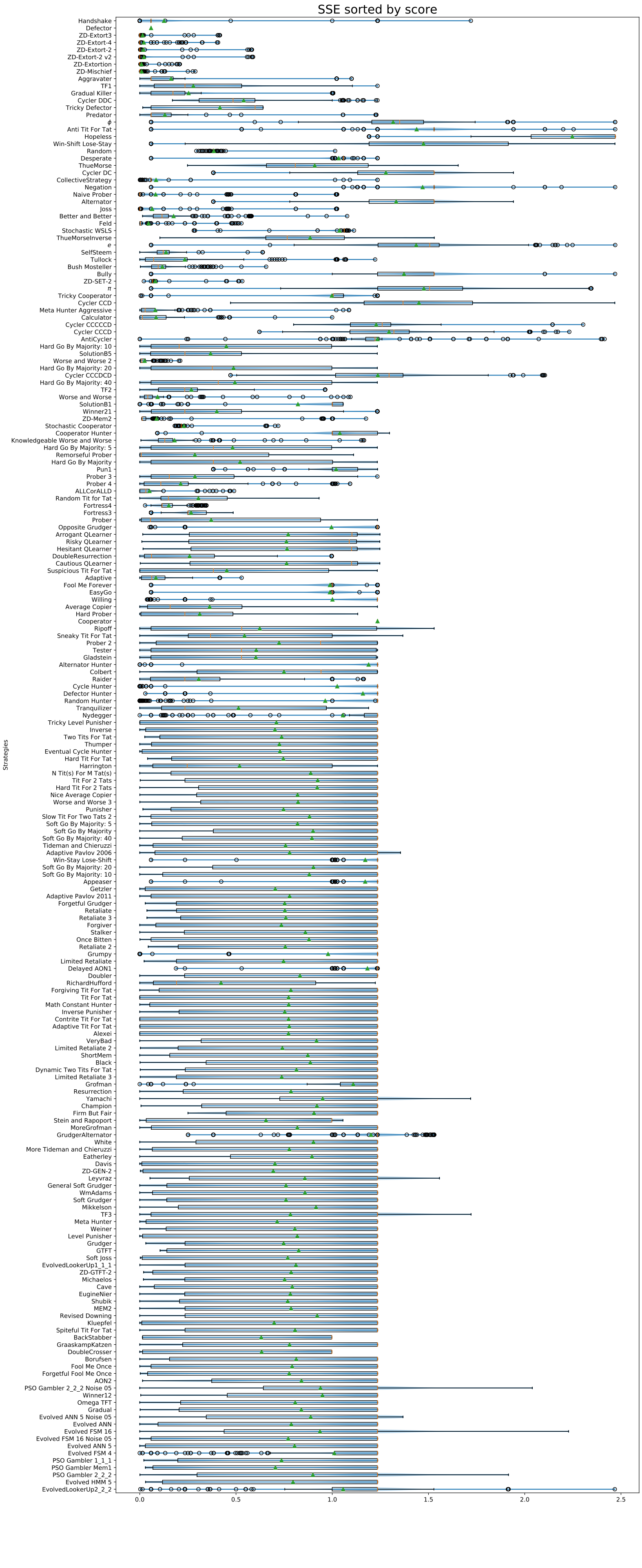
Substituting equations (6-7) in to equation (1) gives the required expression for p_1.
Taking the ratio of equations (6-7) gives the required expression for χ.
Finally, the condition $\chi > 1$ corresponds to:

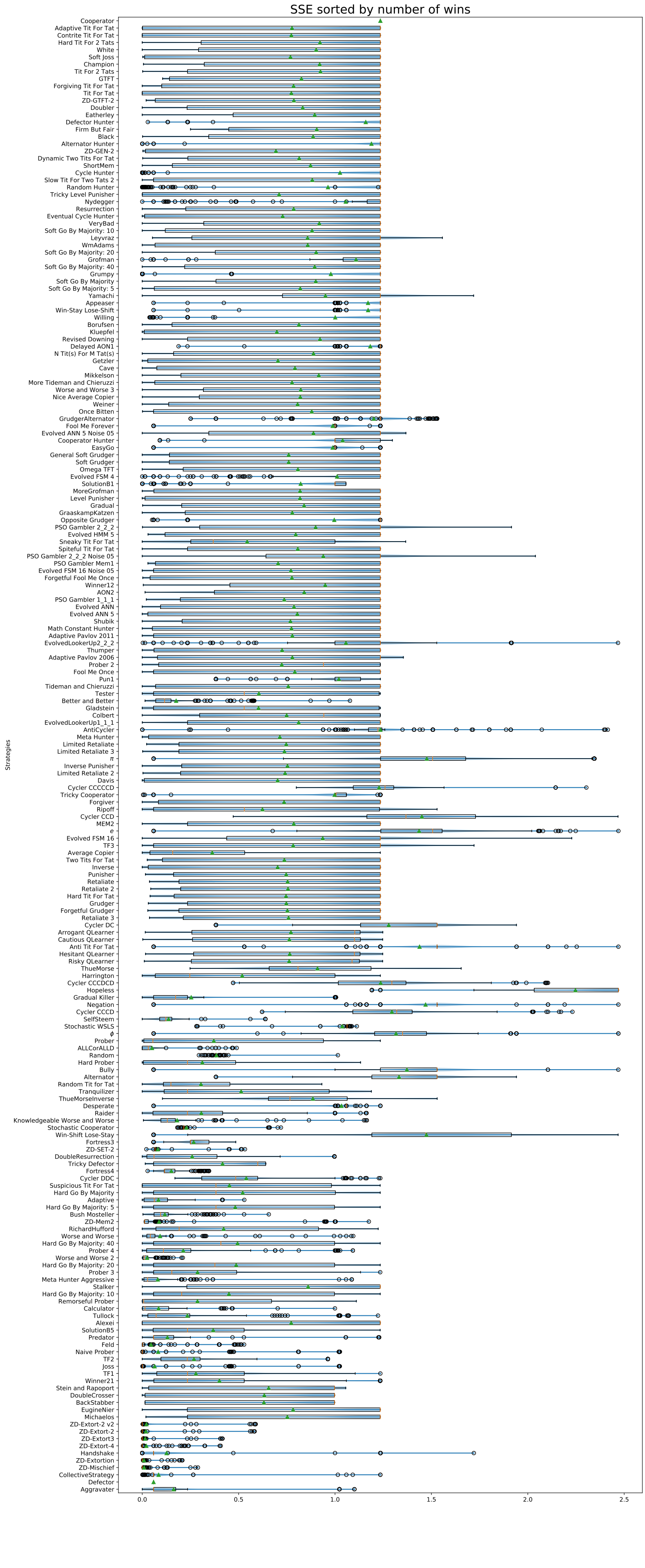
$$\tilde{p}_2(P - T) + \tilde{p}_3(S - P) > \tilde{p}_2(P - S) + \tilde{p}_3(T - P) \tag{8}$$

which can be simplified to:

$$\tilde{p}_2 < -\tilde{p}_3 \tag{9}$$

recalling that $\tilde{p}_2 = p_2 - 1$ and $\tilde{p}_3 = p_3$ gives the required result.


SSE sorted by score
Strategies
0.0
0.5
1.0
1.5
2.0
2.5


SSE sorted by number of wins
Strategies
0.0
0.5
1.0
1.5
2.0
2.5

List of all strategies used from [21]

1. Adaptive - Deterministic - Memory length: ∞ - [25]
2. Adaptive Tit For Tat: 0.5 - Deterministic - Memory length: ∞ - [39]
3. Aggravater - Deterministic - Memory length: ∞ - [21]
4. Alexei: (D,) - Deterministic - Memory length: ∞ - [43]
5. ALLCorALLD - Stochastic - Memory length: 1 - [2]
6. Alternator - Deterministic - Memory length: 1 - [31, 11]
7. Alternator Hunter - Deterministic - Memory length: ∞ - [21]
8. AntiCycler - Deterministic - Memory length: ∞ - [21]
9. Anti Tit For Tat - Deterministic - Memory length: 1 - [18]
10. AON2 - Deterministic - Memory length: 2 - [19]
11. Adaptive Pavlov 2006 - Deterministic - Memory length: ∞ - [24]
12. Adaptive Pavlov 2011 - Deterministic - Memory length: ∞ - [25]
13. Appeaser - Deterministic - Memory length: ∞ - [21]
14. Arrogant QLearner - Stochastic - Memory length: ∞ - [21]
15. Average Copier - Stochastic - Memory length: ∞ - [21]
16. BackStabber: (D, D) - Deterministic - Memory length: ∞ - [21]
17. Better and Better - Stochastic - Memory length: ∞ - [29]
18. Black - Stochastic - Memory length: 5 - [10]
19. Borufsen - Deterministic - Memory length: ∞ - [10]
20. Bully - Deterministic - Memory length: 1 - [32]
21. Bush Mosteller: 0.5, 0.5, 3.0, 0.5 - Stochastic - Memory length: ∞ - [20]
22. Calculator - Stochastic - Memory length: ∞ - [29]
23. Cautious QLearner - Stochastic - Memory length: ∞ - [21]
24. Cave - Stochastic - Memory length: ∞ - [10]
25. Champion - Stochastic - Memory length: ∞ - [10]
26. Colbert - Deterministic - Memory length: 4 - [10]
27. CollectiveStrategy - Deterministic - Memory length: ∞ - [26]
28. Contrite Tit For Tat - Deterministic - Memory length: 3 - [42]
29. Cooperator - Deterministic - Memory length: 0 - [31, 11, 34]
30. Cooperator Hunter - Deterministic - Memory length: ∞ - [21]
31. Cycle Hunter - Deterministic - Memory length: ∞ - [21]
32. Cycler CCCCCD - Deterministic - Memory length: 5 - [21]

33. Cycler CCCD - Deterministic - Memory length: 3 - [21]
34. Cycler CCD - Deterministic - Memory length: 2 - [31]
35. Cycler DC - Deterministic - Memory length: 1 - [21]
36. Cycler DDC - Deterministic - Memory length: 2 - [31]
37. Cycler CCCDCD - Deterministic - Memory length: 5 - [21]
38. Davis: 10 - Deterministic - Memory length: ∞ - [9]
39. Defector - Deterministic - Memory length: 0 - [31, 11, 34]
40. Defector Hunter - Deterministic - Memory length: ∞ - [21]
41. Desperate - Stochastic - Memory length: 1 - [41]
42. Delayed AON1 - Deterministic - Memory length: 2 - [19]
43. DoubleCrosser: (D, D) - Deterministic - Memory length: ∞ - [21]
44. Doubler - Deterministic - Memory length: ∞ - [29]
45. DoubleResurrection - Deterministic - Memory length: 5 - [15]
46. EasyGo - Deterministic - Memory length: ∞ - [25, 29]
47. Eatherley - Stochastic - Memory length: ∞ - [10]
48. EugineNier: (D,) - Deterministic - Memory length: ∞ - [43]
49. Eventual Cycle Hunter - Deterministic - Memory length: ∞ - [21]
50. Evolved ANN - Deterministic - Memory length: ∞ - [21]
51. Evolved ANN 5 - Deterministic - Memory length: ∞ - [21]
52. Evolved ANN 5 Noise 05 - Deterministic - Memory length: ∞ - [21]
53. Evolved FSM 4 - Deterministic - Memory length: 4 - [21]
54. Evolved FSM 16 - Deterministic - Memory length: 16 - [21]
55. Evolved FSM 16 Noise 05 - Deterministic - Memory length: 16 - [21]
56. EvolvedLookerUp1_1_1 - Deterministic - Memory length: ∞ - [21]
57. EvolvedLookerUp2_2_2 - Deterministic - Memory length: ∞ - [21]
58. Evolved HMM 5 - Stochastic - Memory length: 5 - [21]
59. Feld: 1.0, 0.5, 200 - Stochastic - Memory length: 200 - [9]
60. Firm But Fair - Stochastic - Memory length: 1 - [16]
61. Fool Me Forever - Deterministic - Memory length: ∞ - [21]
62. Fool Me Once - Deterministic - Memory length: ∞ - [21]
63. Forgetful Fool Me Once: 0.05 - Stochastic - Memory length: ∞ - [21]
64. Forgetful Grudger - Deterministic - Memory length: 10 - [21]
65. Forgiver - Deterministic - Memory length: ∞ - [21]
66. Forgiving Tit For Tat - Deterministic - Memory length: ∞ - [21]
67. Fortress3 - Deterministic - Memory length: 3 - [7]
68. Fortress4 - Deterministic - Memory length: 4 - [7]
69. GTFT: 0.33 - Stochastic - Memory length: 1 - [17, 33]

70. General Soft Grudger: n=1,d=4,c=2 - Deterministic - Memory length: ∞ - [21]
71. Getzler - Stochastic - Memory length: ∞ - [10]
72. Gladstein - Deterministic - Memory length: ∞ - [10]
73. Soft Go By Majority - Deterministic - Memory length: ∞ - [31, 11, 10]
74. Soft Go By Majority: 10 - Deterministic - Memory length: 10 - [21]
75. Soft Go By Majority: 20 - Deterministic - Memory length: 20 - [21]
76. Soft Go By Majority: 40 - Deterministic - Memory length: 40 - [21]
77. Soft Go By Majority: 5 - Deterministic - Memory length: 5 - [21]
78. ϕ - Deterministic - Memory length: ∞ - [21]
79. GraaskampKatzen - Deterministic - Memory length: ∞ - [10]
80. Gradual - Deterministic - Memory length: ∞ - [13]
81. Gradual Killer: (D, D, D, D, D, C, C) - Deterministic - Memory length: ∞ - [29]
82. Grofman - Stochastic - Memory length: ∞ - [9]
83. Grudger - Deterministic - Memory length: 1 - [12, 25, 13, 41, 9]
84. GrudgerAlternator - Deterministic - Memory length: ∞ - [29]
85. Grumpy: Nice, 10, -10 - Deterministic - Memory length: ∞ - [21]
86. Handshake - Deterministic - Memory length: ∞ - [35]
87. Hard Go By Majority - Deterministic - Memory length: ∞ - [31]
88. Hard Go By Majority: 10 - Deterministic - Memory length: 10 - [21]
89. Hard Go By Majority: 20 - Deterministic - Memory length: 20 - [21]
90. Hard Go By Majority: 40 - Deterministic - Memory length: 40 - [21]
91. Hard Go By Majority: 5 - Deterministic - Memory length: 5 - [21]
92. Hard Prober - Deterministic - Memory length: ∞ - [29]
93. Hard Tit For 2 Tats - Deterministic - Memory length: 3 - [38]
94. Hard Tit For Tat - Deterministic - Memory length: 3 - [40]
95. Harrington - Stochastic - Memory length: ∞ - [10]
96. Hesitant QLearner - Stochastic - Memory length: ∞ - [21]
97. Hopeless - Stochastic - Memory length: 1 - [41]
98. Inverse - Stochastic - Memory length: ∞ - [21]
99. Inverse Punisher - Deterministic - Memory length: ∞ - [21]
100. Joss: 0.9 - Stochastic - Memory length: 1 - [38, 9]
101. Kluepfel - Stochastic - Memory length: ∞ - [10]
102. Knowledgeable Worse and Worse - Stochastic - Memory length: ∞ - [21]
103. Level Punisher - Deterministic - Memory length: ∞ - [15]
104. Leyvraz - Stochastic - Memory length: 3 - [10]
105. Limited Retaliate: 0.1, 20 - Deterministic - Memory length: ∞ - [21]
106. Limited Retaliate 2: 0.08, 15 - Deterministic - Memory length: ∞ - [21]

107. Limited Retaliate 3: 0.05, 20 - Deterministic - Memory length: ∞ - [21]
108. Math Constant Hunter - Deterministic - Memory length: ∞ - [21]
109. Naive Prober: 0.1 - Stochastic - Memory length: 1 - [25]
110. MEM2 - Deterministic - Memory length: ∞ - [27]
111. Michaelos: (D,) - Stochastic - Memory length: ∞ - [43]
112. Mikkelson - Deterministic - Memory length: ∞ - [10]
113. MoreGrofman - Deterministic - Memory length: 8 - [10]
114. More Tideman and Chieruzzi - Deterministic - Memory length: ∞ - [10]
115. Negation - Stochastic - Memory length: 1 - [40]
116. Nice Average Copier - Stochastic - Memory length: ∞ - [21]
117. N Tit(s) For M Tat(s): 3, 2 - Deterministic - Memory length: 3 - [21]
118. Nydegger - Deterministic - Memory length: 3 - [9]
119. Omega TFT: 3, 8 - Deterministic - Memory length: ∞ - [37]
120. Once Bitten - Deterministic - Memory length: 12 - [21]
121. Opposite Grudger - Deterministic - Memory length: ∞ - [21]
122. π - Deterministic - Memory length: ∞ - [21]
123. Predator - Deterministic - Memory length: 9 - [7]
124. Prober - Deterministic - Memory length: ∞ - [25]
125. Prober 2 - Deterministic - Memory length: ∞ - [29]
126. Prober 3 - Deterministic - Memory length: ∞ - [29]
127. Prober 4 - Deterministic - Memory length: ∞ - [29]
128. Pun1 - Deterministic - Memory length: 2 - [6]
129. PSO Gambler 1_1_1 - Stochastic - Memory length: ∞ - [21]
130. PSO Gambler 2_2_2 - Stochastic - Memory length: ∞ - [21]
131. PSO Gambler 2_2_2 Noise 05 - Stochastic - Memory length: ∞ - [21]
132. PSO Gambler Mem1 - Stochastic - Memory length: 1 - [21]
133. Punisher - Deterministic - Memory length: ∞ - [21]
134. Raider - Deterministic - Memory length: 3 - [8]
135. Random: 0.5 - Stochastic - Memory length: 0 - [39, 9]
136. Random Hunter - Deterministic - Memory length: ∞ - [21]
137. Random Tit for Tat: 0.5 - Stochastic - Memory length: 1 - [21]
138. Remorseful Prober: 0.1 - Stochastic - Memory length: 2 - [25]
139. Resurrection - Deterministic - Memory length: 5 - [15]
140. Retaliate: 0.1 - Deterministic - Memory length: ∞ - [21]
141. Retaliate 2: 0.08 - Deterministic - Memory length: ∞ - [21]
142. Retaliate 3: 0.05 - Deterministic - Memory length: ∞ - [21]
143. Revised Downing: True - Deterministic - Memory length: ∞ - [9]

144. RichardHufford - Deterministic - Memory length: ∞ - [10]
145. Ripoff - Deterministic - Memory length: 2 - [5]
146. Risky QLearner - Stochastic - Memory length: ∞ - [21]
147. SelfSteem - Stochastic - Memory length: ∞ - [14]
148. ShortMem - Deterministic - Memory length: 10 - [14]
149. Shubik - Deterministic - Memory length: ∞ - [9]
150. Slow Tit For Two Tats 2 - Deterministic - Memory length: 2 - [29]
151. Sneaky Tit For Tat - Deterministic - Memory length: ∞ - [21]
152. Soft Grudger - Deterministic - Memory length: 6 - [25]
153. Soft Joss: 0.9 - Stochastic - Memory length: 1 - [29]
154. SolutionB1 - Deterministic - Memory length: 3 - [4]
155. SolutionB5 - Deterministic - Memory length: 5 - [4]
156. Spiteful Tit For Tat - Deterministic - Memory length: ∞ - [29]
157. Stalker: (D,) - Stochastic - Memory length: ∞ - [14]
158. Stein and Rapoport: 0.05: (D, D) - Deterministic - Memory length: ∞ - [9]
159. Stochastic Cooperator - Stochastic - Memory length: 1 - [1]
160. Stochastic WSLS: 0.05 - Stochastic - Memory length: 1 - [3]
161. Suspicious Tit For Tat - Deterministic - Memory length: 1 - [13, 18]
162. Tester - Deterministic - Memory length: ∞ - [10]
163. TF1 - Deterministic - Memory length: ∞ - [21]
164. TF2 - Deterministic - Memory length: ∞ - [21]
165. TF3 - Deterministic - Memory length: ∞ - [21]
166. ThueMorse - Deterministic - Memory length: ∞ - [21]
167. ThueMorseInverse - Deterministic - Memory length: ∞ - [21]
168. Thumper - Deterministic - Memory length: 2 - [5]
169. Tideman and Chieruzzi - Deterministic - Memory length: ∞ - [9]
170. Tit For Tat - Deterministic - Memory length: 1 - [9]
171. Tit For 2 Tats - Deterministic - Memory length: 2 - [11]
172. Tranquilizer - Stochastic - Memory length: ∞ - [9]
173. Tricky Cooperator - Deterministic - Memory length: 10 - [21]
174. Tricky Defector - Deterministic - Memory length: ∞ - [21]
175. Tricky Level Punisher - Deterministic - Memory length: ∞ - [15]
176. Tullock: 11 - Stochastic - Memory length: 11 - [9]
177. Two Tits For Tat - Deterministic - Memory length: 2 - [11]
178. VeryBad - Deterministic - Memory length: ∞ - [14]
179. Weiner - Deterministic - Memory length: ∞ - [10]
180. White - Deterministic - Memory length: ∞ - [10]

181. Willing - Stochastic - Memory length: 1 - [41]
182. Winner12 - Deterministic - Memory length: 2 - [30]
183. Winner21 - Deterministic - Memory length: 2 - [30]
184. Win-Shift Lose-Stay: D - Deterministic - Memory length: 1 - [25]
185. Win-Stay Lose-Shift: C - Deterministic - Memory length: 1 - [38, 33, 22]
186. WmAdams - Stochastic - Memory length: ∞ - [10]
187. Worse and Worse - Stochastic - Memory length: ∞ - [29]
188. Worse and Worse 2 - Stochastic - Memory length: ∞ - [29]
189. Worse and Worse 3 - Stochastic - Memory length: ∞ - [29]
190. Yamachi - Deterministic - Memory length: ∞ - [10]
191. ZD-Extortion: 0.2, 0.1, 1 - Stochastic - Memory length: 1 - [36]
192. ZD-Extort-2: 0.1111111111111111, 0.5 - Stochastic - Memory length: 1 - [38]
193. ZD-Extort3: 0.11538461538461539, 0.3333333333333333, 1 - Stochastic - Memory length: 1 - [34]
194. ZD-Extort-2 v2: 0.125, 0.5, 1 - Stochastic - Memory length: 1 - [23]
195. ZD-Extort-4: 0.23529411764705882, 0.25, 1 - Stochastic - Memory length: 1 - [21]
196. ZD-GTFT-2: 0.25, 0.5 - Stochastic - Memory length: 1 - [38]
197. ZD-GEN-2: 0.125, 0.5, 3 - Stochastic - Memory length: 1 - [23]
198. ZD-Mem2 - Stochastic - Memory length: 2 - [28]
199. ZD-Mischief: 0.1, 0.0, 1 - Stochastic - Memory length: 1 - [36]
200. ZD-SET-2: 0.25, 0.0, 2 - Stochastic - Memory length: 1 - [23]
201. *e* - Deterministic - Memory length: ∞ - [21]
202. Dynamic Two Tits For Tat - Stochastic - Memory length: ∞ - [21]
203. Meta Hunter: 6 players - Deterministic - Memory length: ∞ - [21]
204. Meta Hunter Aggressive: 7 players - Deterministic - Memory length: ∞ - [21]

References

[1] Christoph Adami and Arend Hintze. "Evolutionary instability of zero-determinant strategies demonstrates that winning is not everything." In: *Nature communications* 4.1 (2013), p. 2193. ISSN: 2041-1723. DOI: `10.1038/ncomms3193`. arXiv: `arXiv:1208.2666v4`. URL: `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3741637%7B%5C&%7Dtool=pmcentrez%7B%5C&%7Drendertype=abstract`.

[2] Ethan Akin. "What you gotta know to play good in the iterated prisoners dilemma". In: *Games* 6.3 (2015), pp. 175–190.

[3] Marco A Amaral et al. "Stochastic win-stay-lose-shift strategy with dynamic aspirations in evolutionary social dilemmas". In: *Physical Review E* 94.3 (2016), p. 032317.

[4] Daniel Ashlock, Joseph Alexander Brown, and Philip Hingston. "Multiple Opponent Optimization of Prisoners Dilemma Playing Agents". In: *IEEE Transactions on Computational Intelligence and AI in Games* 7.1 (2015), pp. 53–65.

[5] Daniel Ashlock and Eun-Youn Kim. "Fingerprinting: Visualization and automatic analysis of prisoner's dilemma strategies". In: *IEEE Transactions on Evolutionary Computation* 12.5 (2008), pp. 647–659.

[6] Daniel Ashlock, Eun-Youn Kim, and Nicole Leahy. "Understanding representational sensitivity in the iterated prisoner's dilemma with fingerprints". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 36.4 (2006), pp. 464–475.

[7] Wendy Ashlock and Daniel Ashlock. “Changes in prisoners dilemma strategies over evolutionary time with different population sizes”. In: *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*. IEEE. 2006, pp. 297–304.

[8] Wendy Ashlock, Jeffrey Tsang, and Daniel Ashlock. “The evolution of exploitation”. In: *Foundations of Computational Intelligence (FOCI), 2014 IEEE Symposium on*. IEEE. 2014, pp. 135–142.

[9] Robert Axelrod. “Effective choice in the prisoner’s dilemma”. In: *Journal of conflict resolution* 24.1 (1980), pp. 3–25.

[10] Robert Axelrod. “More Effective Choice in the Prisoner’s Dilemma”. In: *Journal of Conflict Resolution* 24.3 (1980), pp. 379–403. ISSN: 0022-0027. DOI: 10.1177/002200278002400301.

[11] Robert Axelrod. *The Evolution of Cooperation*. Basic Books, 1985. ISBN: 0-465-02121-2. URL: https://www.amazon.com/Evolution-Cooperation-Robert-Axelrod/dp/0465021212?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0465021212.

[12] Jeffrey S Banks and Rangarajan K Sundaram. “Repeated games, finite automata, and complexity”. In: *Games and Economic Behavior* 2.2 (1990), pp. 97–117.

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

[14] Andre LC Carvalho et al. “Iterated Prisoners Dilemma-An extended analysis”. In: (2013), pp. 1–6. DOI: 10.21528/CBIC2013-202.

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

[16] Marcus R Frean. “The prisoner’s dilemma without synchrony”. In: *Proceedings of the Royal Society of London B: Biological Sciences* 257.1348 (1994), pp. 75–79.

[17] Marco Gaudesi et al. “Exploiting evolutionary modeling to prevail in iterated prisoners dilemma tournaments”. In: *IEEE Transactions on Computational Intelligence and AI in Games* 8.3 (2016), pp. 288–300.

[18] C. Hilbe, M. A. Nowak, and K. Sigmund. “Evolution of extortion in Iterated Prisoner’s Dilemma games”. In: *Proceedings of the National Academy of Sciences* 110.17 (Apr. 2013), pp. 6913–6918. DOI: 10.1073/pnas.1214834110.

[19] Christian Hilbe et al. “Memory-n strategies of direct reciprocity”. In: *Proceedings of the National Academy of Sciences* 114.18 (2017), pp. 4715–4720.

[20] Luis R Izquierdo and Segismundo S Izquierdo. “Dynamics of the Bush-Mosteller learning algorithm in 2x2 games”. In: *Reinforcement Learning*. InTech, 2008.

[21] Vince Knight et al. *Axelrod-Python/Axelrod: V4.2.0*. 2018. DOI: 10.5281/zenodo.1252994.

[22] David Kraines and Vivian Kraines. “Pavlov and the prisoner’s dilemma”. In: *Theory and decision* 26.1 (1989), pp. 47–79. ISSN: 00405833. DOI: 10.1007/BF00134056.

[23] Steven Kuhn. “Prisoner’s Dilemma”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Spring 2017. Metaphysics Research Lab, Stanford University, 2017.

[24] Jiawei Li et al. “How to design a strategy to win an IPD tournament”. In: *The iterated prisoners dilemma* 20 (2007), pp. 89–104.

[25] Jiawei Li, Philip Hingston, and Graham Kendall. “Engineering design of strategies for winning iterated prisoner’s dilemma competitions”. In: *IEEE Transactions on Computational Intelligence and AI in Games* 3.4 (2011), pp. 348–360.

[26] Jiawei Li and Graham Kendall. “A strategy with novel evolutionary features for the iterated prisoner’s dilemma.” In: *Evolutionary Computation* 17.2 (2009), pp. 257–274. ISSN: 1063-6560. DOI: 10.1162/evco.2009.17.2.257. URL: http://www.ncbi.nlm.nih.gov/pubmed/19413490.

[27] Jiawei Li and Graham Kendall. “The effect of memory size on the evolutionary stability of strategies in iterated prisoner’s dilemma”. In: *IEEE Transactions on Evolutionary Computation* 18.6 (2014), pp. 819–826. DOI: 10.1109/TEVC.2013.2286492.

[28] Siwei Li. *Strategies in the Stochastic Iterated Prisoner’s Dilemma*. http://math.uchicago.edu/ may/REU2014/REUPapers/ 2014.

[29] LIFL. *PRISON*. http://www.lifl.fr/IPD/ipd.frame.html. 2008.

[30] Philippe Mathieu and Jean-Paul Delahaye. “New Winning Strategies for the Iterated Prisoner’s Dilemma (Extended Abstract)”. In: *14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)* (2015), pp. 1665–1666. ISSN: 15582914.

[31] Shashi Mittal and Kalyanmoy Deb. "Optimal strategies of the iterated prisoner's dilemma problem for multiple conflicting objectives". In: *IEEE Transactions on Evolutionary Computation* 13.3 (2009), pp. 554–565.

[32] John H Nachbar. "Evolution in the finitely repeated prisoner's dilemma". In: *Journal of Economic Behavior & Organization* 19.3 (1992), pp. 307–326.

[33] Martin A Nowak and Karl Sigmund. "A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game." In: *Nature* 364.6432 (1993), pp. 56–58. ISSN: 0028-0836. DOI: 10.1038/364056a0.

[34] W. H. Press and F. J. Dyson. "Iterated Prisoner's Dilemma contains strategies that dominate any evolutionary opponent". In: *Proceedings of the National Academy of Sciences* 109.26 (May 2012), pp. 10409–10413. DOI: 10.1073/pnas.1206569109.

[35] Arthur J Robson. "Efficiency in evolutionary games: Darwin, Nash and the secret handshake". In: *Journal of theoretical Biology* 144.3 (1990), pp. 379–396.

[36] Lars Roemheld. "Evolutionary Extortion and Mischief: Zero Determinant strategies in iterated 2x2 games". In: *arXiv preprint arXiv:1308.2576* (2013).

[37] Wolfgang Slany, Wolfgang Kienreich, et al. "On some winning strategies for the Iterated Prisoners Dilemma, or, Mr. Nice Guy and the Cosa Nostra". In: *The iterated prisoners dilemma* 20 (2007), p. 171.

[38] 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 (June 2012), pp. 10134–10135. DOI: 10.1073/pnas.1208087109.

[39] Elpida Tzafestas. "Toward adaptive cooperative behavior". In: *Proceedings of the Simulation of Adaptive Behavior Conference*. Citeseer. 2000, pp. 334–340.

[40] Unkwown. *www.prisoners-dilemma.com*. http://www.prisoners-dilemma.com/. 2017.

[41] Pieter Van den Berg and Franz J Weissing. "The importance of mechanisms for the evolution of cooperation". In: *Proc. R. Soc. B*. Vol. 282. 1813. The Royal Society. 2015, p. 20151382.

[42] Jianzhong Wu and Robert Axelrod. "How to cope with noise in the iterated prisoner's dilemma". In: *Journal of Conflict resolution* 39.1 (1995), pp. 183–189.

[43] *Zoo of strategies*. 2011. URL: http://lesswrong.com/lw/7f2/prisoners_dilemma_tournament_results/.