Iterated prisoner's dilemma and survival of the fittest from an ecological perspective

Martin Domajnko
martin.domajnko@student.um.si
Faculty of Electrical
Engineering and Computer Science,
University of Maribor
Koroška cesta 46
SI-2000 Maribor, Slovenia

ABSTRACT

The iterated prisoner's dilemma is a heavily studied concept in the field of game theory. It was popularized by Axelrod, and it can be used as a tool for modelling complex interactions between self-interested entities. The goal of this paper was to study the impact the environment has on the development of populations of prisoner's dilemma strategies in a simulation, where individuals interact with each other and play the iterated prisoner's dilemma game. This was done from an ecological perspective, meaning the behaviour of strategies stayed static and didn't evolve between generations, in a simulation extending the iterated prisoner's dilemma game. Additionally, the paper presents two new strategies, which are evaluated with Axelrod's original tournament and with our simulation. The implemented simulation uses Axelrod's tournament as a fitness function and fitness proportionate selection for choosing the next generation's strategies. Both of our strategies are based on the n-Pavlov strategy. They achieved average results, but none of them improved the original ones.

KEYWORDS

Prisoner's dilemma, Iterated prisoners's dilemma, n-Pavlov, Simulation, Game theory

1 INTRODUCTION

The prisoner's dilemma presents a situation in which two entities each need to choose between cooperation or defection, while being separated and unable to communicate. There are four possible outcomes: both entities cooperate, both entities defect, or one entity cooperates and one defects, and vice versa. In the first case, both entities get an equal payoff, normally called reward and marked as R. In the second case, both entities get the punishment payoff or P, and in the last case, the cooperating entity gets the sucker's payoff or S, and the defecting entity gets the temptation payoff or T. In the iterated prisoner's dilemma (IPD), two players play multiple rounds of the prisoner's dilemma game and are able to remember their opponent's previous actions [5].

In the classic example of the prisoner's dilemma game with classically rational players, defection always yields a better payoff regardless of the opponent's move. Defection is therefore a strictly dominant strategy for both players, and mutual defection presents the Nash equilibrium move [9] of the Prisoner's dilemma game. This means, that each player could only do worse by unilaterally changing their move [8]. On the other hand, in the iterated version of the game, there are two scenarios. In the first case, the game has a fixed number of rounds, which is known to both players in advance. Therefore, defection is still the dominant strategy and the proof can be constructed with the help of backward induction. In the second case, where the number of rounds is unknown or infinite, mutual defection presents still the Nash equilibrium, but there is no strictly dominant strategy. Studies on the iterated version of the game have shown that most top scoring strategies are nice. This means, that they defect only after their opponent has defected at least once.

Although the Prisoner's dilemma game is studied and applied in different fields, ranging from politics [12], economics [10] and biology [6], this paper focuses more on the implementation side of the problem. We present two new strategies for playing the iterated prisoner's dilemma game. The introduction of the paper gives a short explanation of the prisoner's dilemma game and a summary of the paper's structure. Section 2 presents two additional rules used in the iterated version of the game and gives a more detailed description of Axelrod's tournaments and experiments. This is followed up with Section 3, where we present the implemented strategies used in our experiments and explain the methods used for evaluating their performance. In Section 4, we present the interesting results and finish the paper with Section 5, with a short conclusion.

2 BASIC INFORMATION

To incentivize continuous cooperation over alternating cooperation and defection, the IPD game has two additional rules, defined by conditions $T > R > P > S$ and $2R > T + S$, where T, R, P and S present temptation, reward, punishment, and sucker's payoffs, respectively [5].

Axelrod got the idea for a computer tournament from his high school interest in artificial intelligence and also his interest in game theory starting in college. While pursuing his PhD in Political Science, the motivation for further research and understanding different ways of playing the game, were based on his desire to promote cooperation between players. Another factor was also the game's potential as a source of insight into international conflicts [3].

The IPD computer tournament consists of entrants, in this case programs, which select cooperation or defection on each move based on their decision rules and the history of the game so far. The programs for Axelrod's tournaments were provided from other experts in the field of game theory and researchers studying the Prisoner's Dilemma game [1].

2.1 The first Axelrod's tournament

In Axelrod's first tournament, the strategies were paired against each other in a round-robin style tournament. Additionally, each strategy played against itself and against a strategy which randomly cooperates and defects. The payoff matrix used in the tournament is presented in Table 1 and each game was played for 200 moves.

Table 1: Payoff matrix for a move in the Prisoner's Dilemma game

	Cooperate	Defect
Cooperate	3, 3	0, 5
Defect	5, 0	1, 1

In total, fourteen entries were submitted to the tournament [1]. The results of the tournament had many surprises. The most important results are:

(1) the strategy Tit for Tat won the tournamnet,
(2) none of the strategies succeeded to improve the Tit for Tat strategy, and
(3) all the top performing strategies had the property of being nice, which means, they defect only after their opponent has defected at least once.

Let us notice that the Tit for Tat strategy imitates its opponent's previous move, except in the first round when it cooperates.

2.2 The second Axelrod's tournament

The entrants providing programs for the second Axelrod's tournament were familiar with the analysis and discoveries of the first round. This is the main reason, why the results produced a better insight into the nature of effective choice in the Prisoner's Dilemma. Additionally, the size of the tournament increased from 14 to 62 entries. The rules of the tournament stayed similar to the first one, except the length of the games. The length was determined probabilistically with a 0.00346 chance of ending with each given move, instead of setting a fixed number of 200 moves to play. Again, the winner of the tournament was the Tit for Tat strategy [2]. The main finding from the paper presenting the second tournament is that the performance of the strategies is heavily dependent on the environment in which the game is played.

2.3 Axelrod's evolving new strategies

In addition to creating the previously described tournaments, Axelrod also looked at the problem from an evolutionary perspective [4]. He used the knowledge and methods from biological evolution, applying them, with the help of genetic algorithms, to a socially rich environment consisting of strategies submitted to the prisoner's dilemma tournament. The strategies, used in the simulation experiment, were represented as a string of genes on a chromosome on which genetic transformations can be applied [4]. In each step of the simulation, the strategies played against eight representatives, and based on their performance, the successful strategies were selected for a mating process in which they produced offspring. The offspring represents the next generation's strategies. Both mutation and crossover mechanisms were used for simulating evolution and discovering new strategies. The final results of his experiments have shown, that populations evolved members which were as good as the strategy Tit for Tat and also many of them reflected properties that make Tit for Tat successful [4].

3 IMPLEMENTATION OF EXPERIMENTS

The implementation of our solution was done in the Rust programming language. It heavily focuses on being extensible, adding new strategies, and customizable, changing the starting parameters of the tournament and simulation. The only condition for newly added strategy is that they need to implement the base Strategy trait we defined. For running the simulations, we also support reading the starting parameters from a file. The payoff matrix used in our experiments is the same as in Axelrod's first tournament, which is shown in Table 1.

3.1 The proposed simulation

The implemented simulation was inspired by genetic algorithms, albeit in our case, the chosen entities don't reproduce and aren't exposed to mutations but rather are cloned. This way, no new rules are introduced, and it presents a survival of the fittest simulation from a more ecological perspective rather than a strictly evolutionary one. A similar simulation was already presented in Axelrod's 1980 paper [2], which analyzes the results from his second tournament.

The proposed simulation starts with n different strategies, each having a population size of p. In every step of the simulation, each strategy plays t rounds of the prisoner's dilemma game with all other strategies. The total sum of scores achieved during those games is used as a fitness function, based on which the next generation's strategies are chosen. In the selection process, the fitness proportionate selection method, also known as roulette wheel selection, is used and through the simulation, the total number of entities stays constant. This produces a simulation where low performing strategies slowly die out, and high performing strategy's population sizes grow.

3.2 Strategies in the proposed simulation

In total, we implemented 20 strategies, 18 known ones and two of our own. All strategies are listed below with their name, a corresponding designation and a short description of their behaviour.

(1) Always Cooperate (ALLC): Always cooperates.
(2) Always Defect (ALLD): Always defects.
(3) CD: Alternates between cooperating and defecting.
(4) CCD: Alternates between cooperating twice and defecting once.
(5) DC: Same behaviour as CD, but starts with defecting.
(6) Grim: Cooperates until the opponent defects and then always defects thereafter.
(7) Prober: Plays D, C, C in the first three rounds and based on opponent's moves, chooses his further behaviour. If the opponent defected in round two and cooperated in round three, the strategy defects forever, else it applies the TFT strategy.
(8) Random (RAND): Cooperates with probability $P(C) = 0.5$.
(9) Probability p Cooperator: Cooperates with fixed probability $P(C) = p$.

(10) Reactive (with parameters $R(s, p, q)$): Cooperates with probability s in the first round and after that with probabilites p and q, based on the opponents decision in the previous round.
(11) Tit for Tat (TFT): Imitates its opponent's previous move, except in the first round when it cooperates.
(12) Tit for two Tats (TFTT or TF2T): Defects only if the opponent defected twice in a row.
(13) Two Tits for Tat (TTFT or 2TFT): Defects twice in a row if the opponent defected in the previous round.
(14) Imperfect Tit fot Tat (ImpTFT): Same behaviour as TFT, except that the probability for imitation is less than one.
(15) Suspicious Tit fot Tat (STFT): Same behaviour as TFT, except that it defects in the first round.
(16) Win-Stay-Lose-Shift (WSLS), also known as Pavlov: Cooperates if it and the opponents previous move were equal, else it defects [8, 11]
(17) n-Pavlov [7, 8]: Cooperates with probability p_n in round n. The probability in the first round is $p_1 = 1$ and the probability in round $n + 1$ is calculated based on the payoff in round n as:

$$p_{n+1} = \begin{cases} p_n + \frac{1}{n} & \text{payoff in the last round was R} \\ p_n + \frac{2}{n} & \text{payoff in the last round was T} \\ p_n - \frac{1}{n} & \text{payoff in the last round was P} \\ p_n - \frac{2}{n} & \text{payoff in the last round was S} \end{cases}$$

(18) PavlovD: Same behaviour as Pavlov but defects in the first round.
(19) My strategy simple (MSS): Cooperates the first 6 rounds, after that it either randomly cooperates every second or third move or plays as n-Pavlov.
(20) My strategy advanced (MSA): Plays as n-Pavlov. Additionally, if the probability drops below 0.3 it has a chance, equal to that probability, to forgive the opponent and cooperate the next 2 or 3 moves.

Both of our implemented strategies are based on the n-Pavlov strategy, and both exhibit the property of being nice.

4 EXPERIMENTS AND RESULTS

The implemented strategies were evaluated in a tournament similar to the Axelrod's first tournament and in a simulation, as explained in the previous section of this paper. Both evaluation methods were run multiple times, to avoid any bias introduced by strategies leveraging randomness. The impact of the tournament length on the final rankings of strategies was also tested. This section presents the results achieved.

4.1 Results of the implemented tournament

We run two tournaments with lengths 50 and 1000 moves, each 1000 times, and used the average of the scores achieved to rank the strategies. The results are collected in Table 2. In addition, we divided the scores from the long tournament by 20 for easier comparison with the short version scores.

[1] $s = 0.5, p = 0.7, q = 0.3$

Table 2: Ranking of strategies in the tournament evaluation. Rank #1 and score #1 present the results of the longer tournament with 1000 moves played, and the rank #2 and score #2 present the results from the shorter tournament with 50 moves played.

Rank #1	Rank #2	Score #1	Score #2	Name
1.	1.	2530	2532	Grim
2.	2.	2447	2459	TF2T
3.	3.	2414	2418	Pavlov
4.	4.	2407	2415	n-Pavlov
5.	5.	2405	2407	TFT
6.	7.	2317	2319	PavlovD
7.	9.	2314	2277	CCD
8.	6.	2303	2333	MSA
9.	12.	2270	2257	Random
10.	8.	2257	2280	DDC
11.	11.	2246	2258	CD
12.	10.	2230	2274	ALLD
13.	14.	2212	2222	$R(s, p, q)$[1]
14.	15.	2199	2184	$P(0.75)$
15.	13.	2194	2241	ImpTFT
16.	16.	2152	2149	ALLC
17.	17.	2150	2147	MSS
18.	18.	2055	2089	STFT
19.	19.	2022	2051	Prober
20.	20.	1947	1992	2TFT

From the results we can observe, that the winning strategy from Axelrod's tournaments landed only on place 5. The similar Tit for two Tats strategy performed better, placing second. As expected, both the Pavlov and n-Pavlov strategies placed in the top 5. The best score in both the short and long tournament, therefore placing first, was achieved by the Grim strategy. This was a small surprise, but after a more detailed look at the data, the results make sense and are also expected. Grim scores well against nice strategies, which are quite well represented in our tournament, and also takes advantage of strategies that heavily rely on randomness or forgiveness. Forgiving strategies are those that are still prepared to cooperate even when the opponent has defected against them [8]. Grim defects from the next move his opponent defected till the end, but the random and forgiving strategies will sometimes still cooperate, which brings Grim the extra score to place first. One of our strategies named My strategy advanced performed quite well, placing 8th in the short tournament and 6th in the long tournament. In comparison, My strategy simple placed only on the 17th place in both tournaments. While one of them performed rather well, none of them improved the n-Pavlov strategy on which they were based on.

4.2 Results of the proposed simulation

The first scenario we tested was the effect the length of the tournament has on the simulation's behaviour and its final outcome. The initial population size of each strategy was set to 100 members, and we tested tournament lengths of 10 and 200 moves. The performance was measured by the number of generations a specific strategy survived, before dying out and being completely removed from the simulation. The best performing strategies, as observed from the results, were the same as seen in Table 2 and there was very little difference discovered between the tournament lengths of 10 and 200 moves.

Our experiments continued with testing different initial population sizes for the strategies and its effect on the results. This also affected the total number of strategies playing against each other and therefore the running time of the simulation. We tested initial population sizes of 10 and 100 members for each strategy. The results were almost identical to the ones produced while testing different tournament lengths.

After the initial experiments, we have also done some smaller ones, where we tested the performance of our two strategies in different environments. One of the environments was a TFT based environment, consisting only of the strategies based on the original TFT strategy (TFT, TF2T, 2TFT, ImpTFT and STFT), and another one was a Pavlov based environment, consisting only of (Pavlov, n-Pavlov and PavlovD strategies). Although the scenarios looked quite interesting, they didn't produce any significant results.

4.3 Discussion

An interesting behaviour found in a population of TFT strategies is the ability to invade a population of ALLD strategies. Additionally, a population of TFT strategies can't be invaded by a population of ALLD strategies. This was the first scenario we tested, and the results have shown, that none of our strategies exhibit that behaviour. The same holds true for the n-Pavlov strategy which they are based on.

One thing that all the experiments had in common was, that most of the low performing strategies died out quickly. If the simulations were run for enough generations, we also observed that eventually one strategy took over and no other strategy was left. This led us to a more detailed examination of the scores produced throughout the simulation. We observed that initially the minimum scores start dropping, and after the low performers die out, the minimum scores start increasing, eventually converging with the average and maximum scores. The strategies that drop out early are mostly strategies that don't fall into the nice category. Eventually all the strategies left in the simulation exhibited the property of being nice and therefore the only possible move was cooperation. From this point onwards, the behaviour of the simulation was completely based on the random generator controlling the selection process.

5 CONCLUSION

In this study, we presented two new strategies for playing the iterated prisoner's dilemma game and evaluated them on an implementation similar to Axelrod's original tournament and on our own simulation. The results of the tournament have shown, that one strategy performed quite well, while the other one performed poorly, placing only on the 17th place out of 20. Additionally, none of the two presented strategies outperformed the n-Pavlov strategy which they were based on. Lastly, we observed that most strategies dropped out of our simulation quite early and only a small number of nice strategies were left. This lead to a scenario, where the only played move was cooperation and the final outcomes were only affected by the random generator controlling the selection process.

In the future, two possible improvements could be added to our solution. One would be to increase the number of the implemented strategies, and the second one would be to reduce the imbalance between the number of strategies exhibiting the property of being nice.

REFERENCES

[1] Robert Axelrod. 1980. Effective Choice in the Prisoner's Dilemma. *The Journal of Conflict Resolution* 24, 1 (1980), 3–25. http://www.jstor.org/stable/173932

[2] Robert Axelrod. 1980. More Effective Choice in the Prisoner's Dilemma. *The Journal of Conflict Resolution* 24, 3 (1980), 379–403. http://www.jstor.org/stable/173638

[3] Robert Axelrod. 2012. Launching "The Evolution of Cooperation". *Journal of Theoretical Biology* 299 (2012), 21–24. https://doi.org/10.1016/j.jtbi.2011.04.015 Evolution of Cooperation.

[4] Robert Axelrod et al. 1987. The evolution of strategies in the iterated prisoner's dilemma. *The dynamics of norms* 1 (1987), 1–16.

[5] Robert Axelrod and William D. Hamilton. 1981. The Evolution of Cooperation. *Science* 211, 4489 (1981), 1390–1396. http://www.jstor.org/stable/1685895

[6] R Dawkins. 1976. *The Selfish Gene.* Oxford University Press, Oxford, UK.

[7] David Kraines and Vivian Kraines. 1989. Pavlov and the prisoner's dilemma. *Theory and Decision* 26, 1 (Jan. 1989), 47–79. https://doi.org/10.1007/bf00134056

[8] Steven Kuhn. 2019. Prisoner's Dilemma. In *The Stanford Encyclopedia of Philosophy* (Winter 2019 ed.), Edward N. Zalta (Ed.). Metaphysics Research Lab, Stanford University.

[9] John Nash. 1951. Non-Cooperative Games. *Annals of Mathematics* 54, 2 (1951), 286–295. http://www.jstor.org/stable/1969529

[10] Abraham Neyman. 1985. Bounded complexity justifies cooperation in the finitely repeated prisoners' dilemma. *Economics Letters* 19, 3 (1985), 227–229. https://doi.org/10.1016/0165-1765(85)90026-6

[11] Martin Nowak and Karl Sigmund. 1993. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game. *Nature* 364, 6432 (jul 1993), 56–58. https://doi.org/10.1038/364056a0

[12] Glenn H. Snyder. 1971. "Prisoner's Dilemma" and "Chicken" Models in International Politics. *International Studies Quarterly* 15, 1 (03 1971), 66–103. https://doi.org/10.2307/3013593 arXiv:https://academic.oup.com/isq/article-pdf/15/1/66/5096260/15-1-66.pdf