Computational analysis of trends in Evolutionary Iterated Prisoner's Dilemma

Shoyo Inokuchi

1W16BG02-3

Computational Modeling - Project Report

Waseda University

Table of Contents

Abstract

The "Prisoner's Dilemma" often posed within game theory involves two people who each make a decision to either cooperate for mutual gain or defect for a greater selfish gain. The optimal strategy for a rational, self-preserving individual in this dilemma is to always defect. However, when the Prisoner's Dilemma is iterated over numerous rounds and a natural selection mechanic is implemented, a model referred within this paper as the Evolutionary Iterated Prisoner's Dilemma (EIPD), the optimal strategy shifts according to initial conditions. The optimal strategy is no longer to defect every round, as the threat of retaliation introduces a negative incentive not present in the conventional scenario of the Prisoner's Dilemma. We implement this model of EIPD in the Python language to rigorously analyze outcomes for given various initial conditions. We address the implementation of code for investigation and the analysis of the data obtained as a result. This paper explores the mechanisms for why cooperative, retaliatory strategies such as "Tit for Tat" tend to reproduce with greater success than selfish, predatory strategies such as "Tester".

Background

The "Prisoner's Dilemma" is a hypothetical situation studied in the field of game theory. The situation is as follows. Two criminals, assumed to be partners-in-crime, are arrested and interrogated separately by authorities. There exists no way for the criminals to communicate with one another. Authorities do not possess enough evidence to convict either criminal for the suspected crime without a confession. With the current evidence, the two criminals would be charged for a lesser crime and go to prison for 1 year. Therefore, authorities grant each criminal one of two options: cooperate with their partner by staying silent and go to prison for 1 year, or defect against their partner and testify against them to avoid going to prison, while sending their partner to 3 years in prison. If both criminals defect, however, then both criminals are sentenced to 2 years in prison. The incentives of each scenario are visualized in Table 1.

Table 1: Visualization of incentives of Prisoner's Dilemma

PRISONER 1\PRISONER 2	COOPERATE	DEFECT
COOPERATE	1 year \ 1 year	3 years \ 0 years
DEFECT	0 years \ 3 years	2 years \ 2 years

The incentives can be generalized as follows in Table 2.

Table 2: Visualization of generalized incentives of Prisoner's Dilemma

PRISONER 1\PRISONER 2	COOPERATE	DEFECT
COOPERATE	B \ B	D \ A
DEFECT	A \ D	C \ C

where $A < B < C < D$ in terms of severity

Both criminals are assumed to be rational, and act in a manner of self-preservation and gain. The optimal payoff with regards to the criminals as a whole would be for both partners to cooperate, as that results in the minimum collective punishment (2 years in this case). However, from the perspective of each individual criminal, cooperating breeds a risk of an increased sentence, while defecting protects oneself from the worst possible outcome, and also breeds a chance of a decreased sentence. For each criminal, defecting always improves his own outcome regardless of his partner's action. It is in each criminal's best interest, therefore, to defect in this situation. In game theory, this optimal action is often referred to as the Nash equilibrium.

However, the optimal action becomes unclear when the Prisoner's Dilemma is iterated over a given number of times. This is due to the threat of retaliation. If, in a given iteration, a prisoner defects against a cooperative partner, the partner can then retaliate by defecting in successive iterations. This introduces a previously nonexistent negative incentive for defecting. The optimal action may no longer be to simply defect every iteration.

Suppose now, that this iterated model of the Prisoner's Dilemma was generalized to a game between two players, where each iteration of the Prisoner's Dilemma was considered to a be a "round". Each outcome can then be translated to each player gaining some number of points after each round, such as in Table 3.

Table 3: Points awarded for each outcome of Prisoner's Dilemma game

PLAYER 1\PLAYER 2	COOPERATE	DEFECT
COOPERATE	+1 point \ +1 point	+3 points \ +0 points
DEFECT	+3 points \ +0 points	+2 points \ +2 points

A player's final score would then indicate how well they performed against their opponent. If we then assume that each player acted strictly according to a predetermined strategy, such as "always cooperate", then we would obtain data on how a particular strategy performs against another particular strategy. If hypothetical Player A uses an "always cooperate" strategy, while hypothetical Player B uses an "always defect" strategy, and the game was iterated over 5 rounds, Player A would have a final score of +0 points/round × 5 rounds = 0 points, while Player B would have a final score of +3 points/round × 5 rounds = 15 points. This data would then reveal that an "always defect" strategy performs well against an "always cooperate" strategy, and vice versa.

To explore this concept even further, suppose there is a collection of players, each player with a predetermined strategy for how to play the iterated Prisoner's Dilemma game. If each player in the collection played against every other player in a round-robin styled tournament, each player would end up with a final score that indicates how well their strategy performs within that given group of players. To draw an analogy, if we assume the collection of players to be an initial "population", and each player's final score to indicate its "fitness" within the current population, and each player's fitness to determine the probability that it can "reproduce" and create copies of itself in the next generation, then we can observe how the population distribution would change from generation to generation. This paper refers to this model as the "Evolutionary Iterated Prisoner's Dilemma" (EIPD). Investigating EIPD would reveal trends for how a given initial population may favor certain "optimal" strategies over others.

Targets

With regards to the investigation of Evolutionary Iterated Prisoner's Dilemma, we will demonstrate the following.

1. We will demonstrate that in Evolutionary Iterated Prisoner's Dilemma, the optimal strategy changes according the initial population distribution, and is not simply to defect every round.
2. We will demonstrate how cooperative, retaliatory strategies, such as "Tit for Tat" described in the next section, perform particularly well among a diverse portfolio of strategies, but how even effective strategies such as these depend on the other strategies that are present.
3. We will demonstrate a fundamental issue with the reproductive success of selfish, predatory strategies and provide speculation for why such strategies are often outperformed by "Tit for Tat" given enough time.

Methodology

To execute and analyze trends in EIPD, computational methods were used. We constructed a framework in the Python language to run computational simulations of the Evolutionary Iterated Prisoner's Dilemma given a set of variable initial conditions (initial population distribution, number of generations to iterate, number of rounds members compete). First, we programmed the following strategies in Table 4 so that a given strategy can be modularly adopted by a population member.

Table 4: Implemented Strategies

Name	Nature	Description
Kantian	cooperative	Cooperates every round.
Tit for Tat	cooperative, retaliatory	Defects if opponent defected last round. Otherwise cooperates. (i.e. repeats opponent's last action)
Tit for 2 Tats	cooperative, retaliatory	Defects if opponent defected last 2 rounds. Otherwise cooperates.
Grudger	cooperative, retaliatory	Cooperates until opponent defects. Then defects for the rest of the game.
Defector	selfish	Defects every round.
Mean Tit for Tat	selfish	Plays like Tit for Tat, but occasionally defects at random* chance. *(1 in 6 in this case)
Conniver	selfish, predatory	An "always cooperate" exploiter. Plays like Tit for Tat, but randomly* tests opponent by defecting and following with 2 turns of apology. If opponent does not defect back within 2 turns, defects for rest of the game. *(1 in 6 in this case)
Tester	selfish, predatory	A "Tit for 2 Tats" exploiter. Plays like Tit for Tat, but randomly* tests opponent by defecting and following with a turn of apology. If opponent does not defect back within 1 turn, alternates between defecting and cooperating for rest of game. *(1 in 6 in this case)
Wary Tit for Tat	neutral	Plays like Tit for Tat, but starts by defecting.
Random	neutral	Cooperates or defects at 50% chance.

Secondly, we implemented the logic for the round-robin styled tournament that takes place in every generation, and a point system to keep track of the fitness of each member. The adopted point system is the same as shown previously in Table 3. We define a round-robin to be a tournament where every competitor plays against every other competitor. Thirdly, the logic for the reproduction, or the creation of a successive generation based on the members of the current generation, was programmed. Each member of the successive generation is determined probabilistically according to how each unique strategy in the current generation performed relative to the current generation's total score. An example is described below.

1. Generation *G(n)*, where *G* is the generation of the *n*-th iteration, is comprised of 4 members each with unique strategies: Player A, Player B, Player C, and Player D.
2. After the round-robin styled tournament, Player A finishes with a score of 35 points, Player B with 20 points, Player C with 45 points, and Player D with 0 points. (for a "population total score" of 100 points)
3. When *G(n+1)* is generated, each member of *G(n+1)* then has a 35% chance (35/100) of adopting the strategy of Player A, 20% (20/100) chance of adopting the strategy of Player B, and so on.

If multiple players in a given generation share the same strategy, their relative scores are totaled and treated as a cumulative score for that particular strategy. Note that for the implementation discussed in this paper, population size does not change from generation to generation. If the initial population contains *n* members, each successive generation will also contain *n* members. Another important point to address with this implementation is the factor of random chance. Due to the nature of chance, even a strategy that performed poorly in a given generation can possibly reproduce with more success (create more copies of itself in the successive generation) than a strategy that accumulated more points. This would potentially affect the obtained results greatly. This issue is handled by increasing the initial population size as much as is computational viable. By increasing the number of members in each generation, the likelihood of a poor performer overtaking a greater performer is decreased. With a greater number of population members, the proportion of each member's strategy in the successive generation approaches their actual proportion. (as population size $\rightarrow \infty$, Player A's strategy in next generation $\rightarrow$ 35%, Player B's strategy in next generation $\rightarrow$ 20%, etc.) To put this into perspective, for the previous example, the initial population can be instead comprised of 150 members with Player A's strategy, 150 members with Player B's strategy, and so on.

After generational logic was implemented, the population distributions listed in Table 5 were created to test various initial conditions. The framework built in Python was programmed so that these initial population profiles are modular and can be tested independently. These distributions contain a control group, which is an initial population (a few defectors among a population of Kantians) with an easily predictable behavior, and test groups, which are the focus of analysis in this experiment. The control group served to check that the constructed framework functioned as intended. Among the test groups, a base simulation was first conducted with an initial profile containing all of the strategies listed previously. All other simulations involved test groups which were created to observe in detail certain trends that were noticed in the base simulation. All test groups are listed with their population distribution, motive for selection, and which target they aim to demonstrate.

Initial population distributions implemented in experiment:

Table 5: Control group

Group	Initial distribution	Motive/Notes
Control 1	*Kantian (97%)* *Defector (3%)*	*Check that framework works as intended. Defectors expected to increase at exponential rate, eventually wipe out Kantian population.*

Table 6: Test groups

Group	Initial distribution	Motive/Notes
Test 1 (Targets 1, 2, and 3)	*Equal proportions of all strategies in Table 4.*	*Observe which strategy performs the best out of those listed in Table 4. Serves as a "base simulation" for the other tests listed below.*
Test 2 (Target 1)	*Defector (97%)* *Tit for Tat (3%)*	*Observe how a retaliatory, cooperative strategy performs among a population of non-cooperators.*
Test 3 (Target 2)	*Tit for Tat (25%)* *Tit for 2 Tats (25%)* *Wary Tit for Tat (25%)* *Mean Tit for Tat (25%)*	*Observe how a successful strategy for a diverse population can be outperformed in specific initial distributions. Being less retaliatory can be beneficial given the environment.*
Test 4 (Targets 2, 3)	*Equal proportions of strategies in Test 3 and Tester.*	*Observe how the introduction of a predatory, selfish strategy in a population improves the performance of retaliatory, cooperative strategies. Also observe a fundamental problem for the growth of predatory strategies.*

Finally, after creating the initial population distributions shown above, we implemented a way of visualizing the results obtained using the resources NumPy and Matplotlib. Every test conducted iterates EIPD with the initial populations mentioned above for 35 generations with 100 rounds each for every pairing in the round-robin tournament.

Data Analysis

Control Group 1:

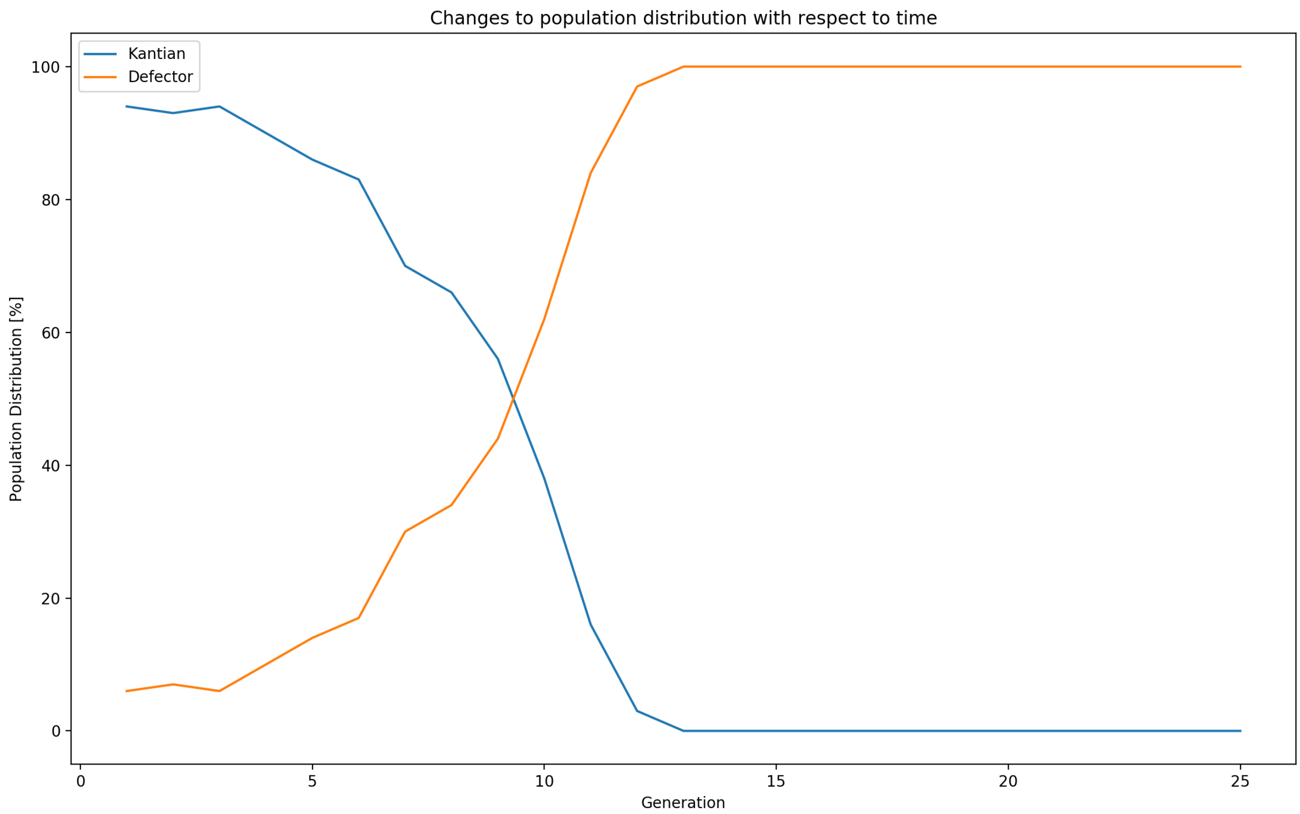

Figure 1: 97% Kantian, 3% Defector (Control 1)

The first initial population that was analyzed was the control group. This distribution includes a small population of Defectors among a large population of Kantians. The result can be observed above in Figure 1. As expected, the Defector population repeatedly wins the optimal selfish payout in the Prisoner's Dilemma and grows exponentially as generation increases. By the 13th generation, the Kantian population is completely eliminated. This model serves as a confirmation that the constructed framework works as intended.

Test Group 1:

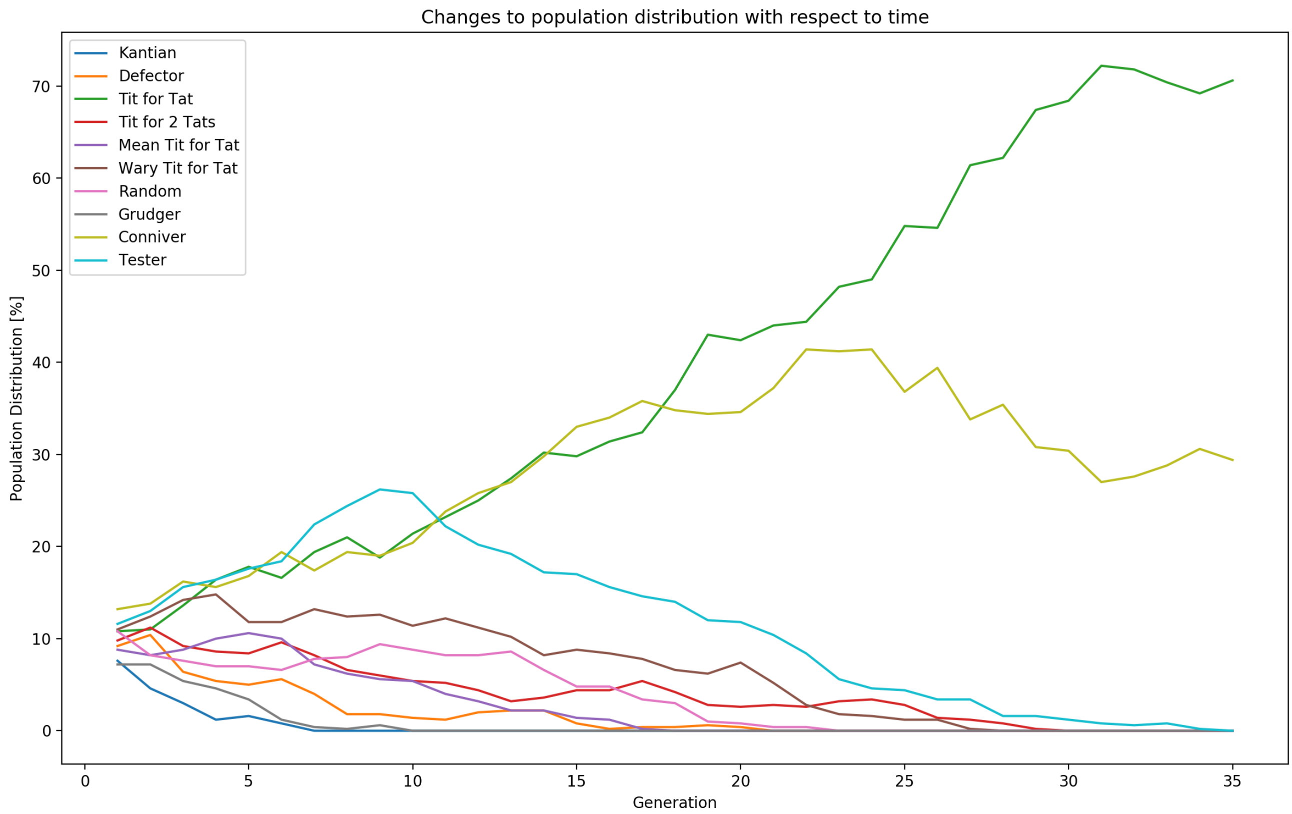

Figure 2: All strategies from Table 4 (Test 1)

The first test group allowed us to determine which strategy out of those listed in Table 4 reproduced most successfully. As the graph suggests, most cooperative strategies (Kantian, Tit for 2 Tats, Grudger) fall prey to predatory strategies in earlier generations and gradually decrease in population. On the other hand, predatory, selfish strategies such as Tester and Conniver perform exceptionally well in earlier generations, as the abundance of prey serve as a source of exploitation. However, we can observe that predatory strategies are difficult to sustain over a long period of time, as their growth primarily depends on the number of exploitable strategies in the population. Once such strategies die out, Tit for Tat is able to outpace the growth of predatory strategies. This graph also indicates that an "always defect" strategy is far from optimal in EIPD.

Test Group 2:

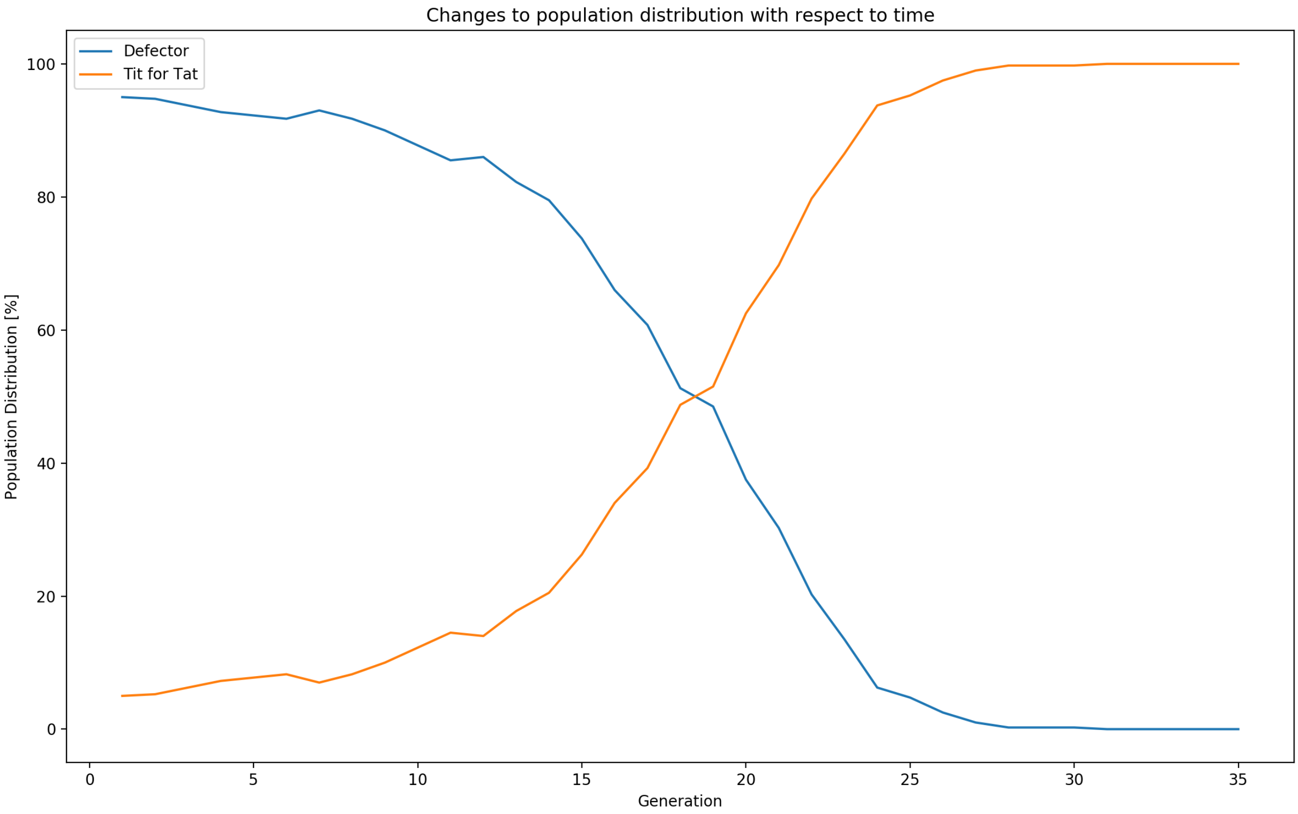

Figure 3: 97% Defector, 3% Tit for Tat (Test 2)

The previous test prompted further investigation of Tit for Tat, particularly to see whether it was an optimal strategy for all cases of EIPD. Before investigating cases where Tit for Tat could be suboptimal, however, Test 2 was created to demonstrate Tit for Tat's self-sustainability. Tit for Tat not only prevents itself effectively from being taken advantage of, its reproduction rate depends on how many cooperative strategies are in the population. Tit for Tat, despite being extremely simple in concept, minimizes loss when facing an uncooperative opponent, and maximizes gain when facing a cooperative opponent. Figure 3, therefore, contrary to the results obtained in Figure 1, shows that it is possible for a cooperative strategy to thrive among Defectors if it is sufficiently retaliatory to prevent being taken advantage of.

Test Group 3:

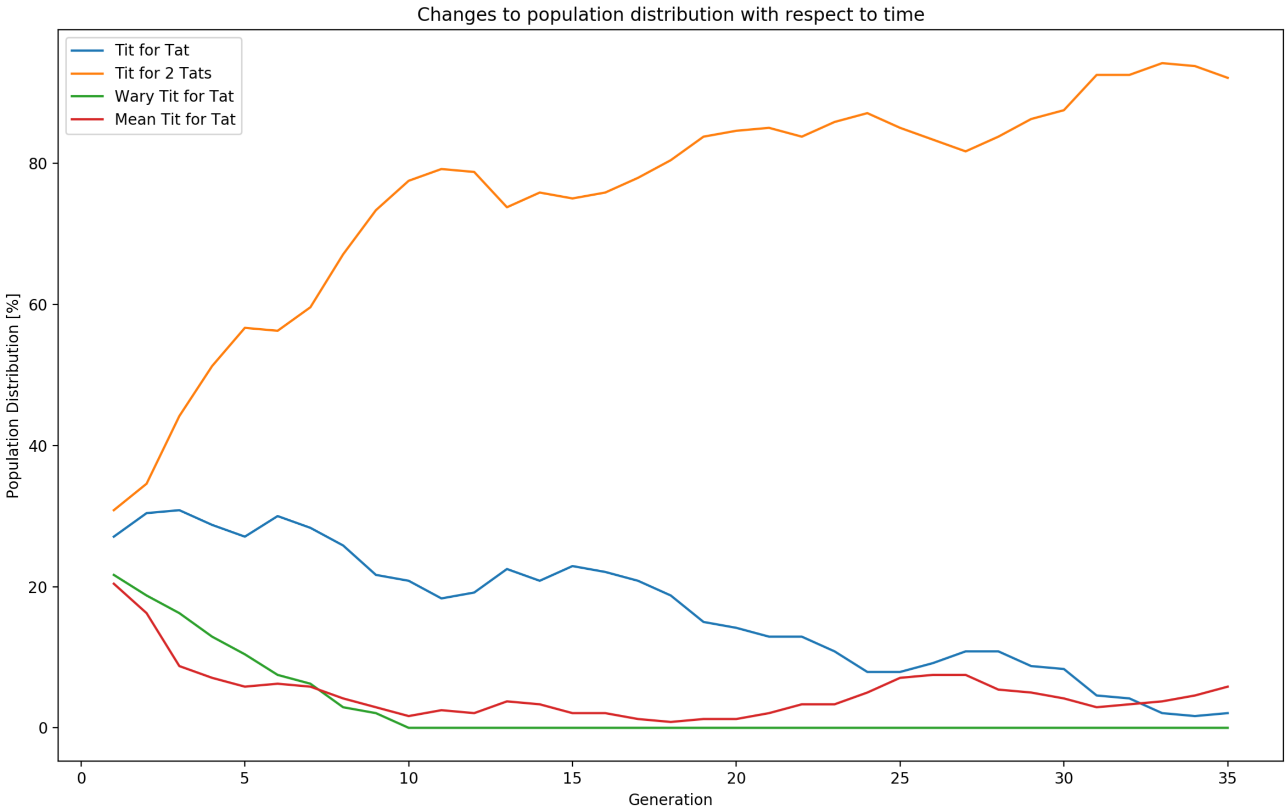

Figure 4: 25% Tit for Tat, 25% Tit for 2 Tats, 25% Wary TFT, 25% Mean TFT (Test 3)

Although Tit for Tat had appeared to be an optimal strategy thus far, Test Group 3 was created to demonstrate a suboptimal environment for regular Tit for Tat. Figure 4 demonstrates the optimal strategy in EIPD indeed changes according to its initial conditions. In this case, the lack of extremely selfish or predatory strategies (such as Defector, Conniver, Tester) allows more forgiving strategies such as Tit for 2 Tats to thrive. Both Wary Tit for Tat and Mean Tit for Tat are self-hindering in this environment, as most opponents in the population, including themselves, retaliate immediately. This makes defecting essentially a waste of time and a loss of potential points. Additionally, in this given population distribution, Tit for Tat is too unforgiving. This is visualized below in Figure 5.

Strategy \ Turn	1	2	3	4	5	6	7	8	9	10	11	12	13	...
Tit for Tat	C	C	C	C	D	C	D	C	D	D	D	D	D	...
Mean Tit for Tat	C	C	C	D	C	D	C	D	D	D	D	D	D	...

Figure 5: Hypothetical execution of Tit for Tat vs. Mean TFT

Figure 5 shows a simplified, hypothetical match between Tit for Tat and Mean Tit for Tat. As shown, both strategies begin cooperatively until Mean Tit for Tat randomly defects (on round 4 in this case). This defection triggers a period of "defection echo", in which the two strategies alternate between defecting and cooperating (rounds 4 - 8). After some time, Mean Tit for Tat randomly defects again (round 9), which triggers both strategies to defect for the rest of the game. As mentioned previously, defecting is simply a waste of time and a loss of potential points in a non-predatory environment such as Test Group 3. If Tit for Tat faces against Wary Tit for Tat, the latter's initial defection causes the entire game to be a defection echo. Due to these factors, the performance of Tit for Tat in this environment suffers greatly. A more forgiving strategy such as Tit for 2 Tats, on the other hand, plays out as follows.

Strategy \ Turn	1	2	3	4	5	6	7	8	9	10	11	12	13	...
Tit for 2 Tats	C	C	C	C	C	C	C	C	C	C	C	C	C	...
Mean Tit for Tat	C	C	C	D	C	C	C	C	D	C	C	C	C	...

Figure 6: Hypothetical execution of Tit for 2 Tats vs. Mean TFT

Although Tit for 2 Tats allows itself to be taken advantage of occasionally by a mildly selfish strategy such as Mean Tit for Tat, it is much more successful at reproduction in this environment because it does not lose any potential points of cooperation. While the other strategies perform poorly due to retaliatory and selfish behaviors, Tit for 2 Tats is able to grow exceptionally well.

Test Group 4:

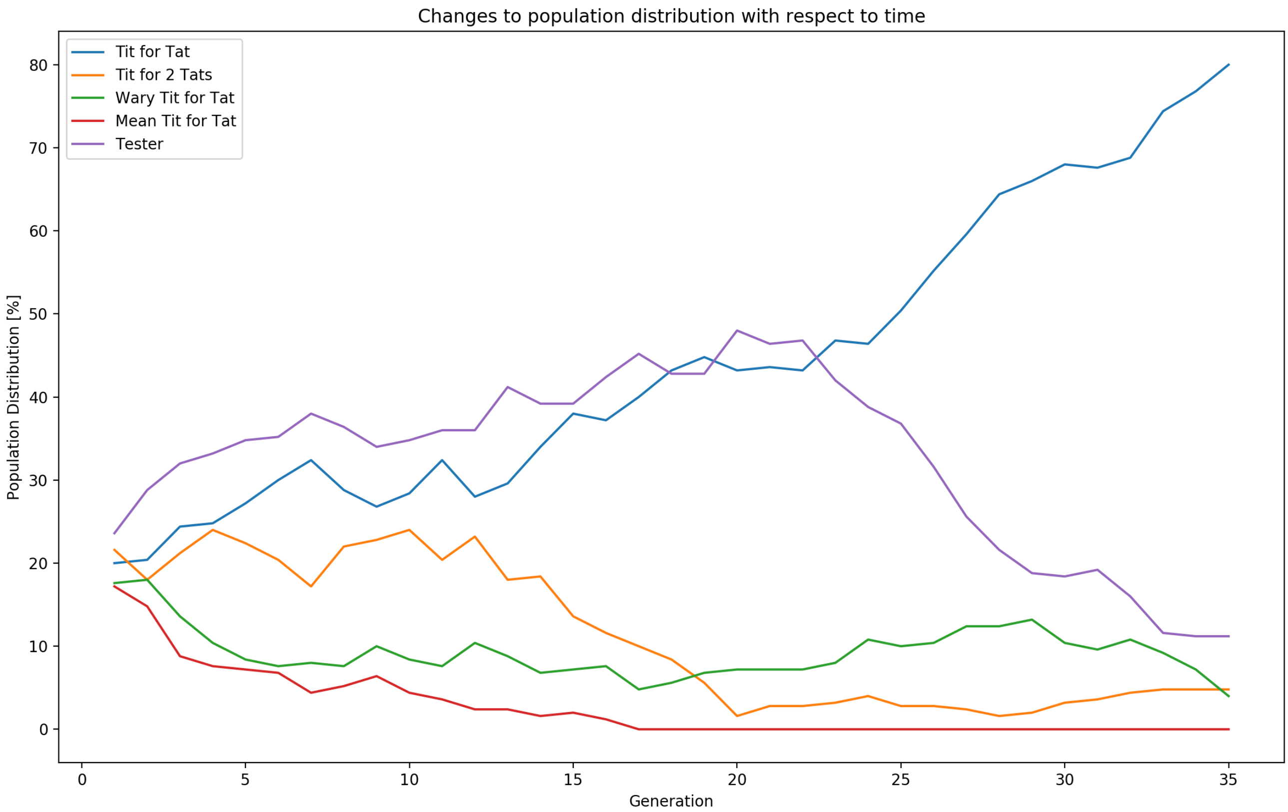

Figure 7: Various Tit for Tats with Tester (Test 4)

In Test Group 4, a predatory strategy is introduced to what was previously a fairly docile population distribution. As expected, Tester (which by design functions as a Tit for 2 Tats exploiter) initially finds much reproductive success in this environment, as the influx of Tit for 2 Tats from generations 1 to 10, for reasons discussed previously, grant Tester an exploitable source of growth. Tester continues to feed on Tit for 2 Tats until the population dwindles by generation 20. Consequently, after generation 20, we observe a sharp decline in the Tester population, which allows a self-sustaining strategy such as Tit for Tat to grow rapidly. This test group, along with previously discussed Test Group 1, reveals a fundamental obstacle for the growth of predatory strategies.

Suppose the population of predatory strategies and population of potential prey are represented by P_1 and P_2, respectively. The growth rate of predatory strategies can then be represented by $\frac{dP_1}{dt}$, which would affect $\frac{dP_2}{dt}$, the growth rate of potential prey. Intuitively, $\frac{dP_1}{dt}$ depends strongly on how many prey are present in the population. Being a zero-sum population, where an increase of one strategy implies the decrease of another, exploiting prey to increase population inherently unsustainable. As prey are exploited and $\frac{dP_1}{dt}$ increases, the resulting decrease of P_2 eventually causes $\frac{dP_1}{dt}$ to reach a critical point. The relationship between the predatory population growth and prey population growth can be represented with the Lokta-Volterra model shown below.

$$\frac{dP_1}{dt} = \alpha P_1 P_2 - \beta P_1$$

$$\frac{dP_2}{dt} = \gamma P_2 - \delta P_1 P_2$$

where α, β, γ, δ are positive constants.

The Lokta-Volterra model shows that in a zero-sum environment used in EIPD, predatory strategies are unsustainable over a long duration, and are inevitably going to decline in population. Aside from having an unsustainable model of growth, predatory strategies, at least the two investigated in this paper, do not play well against other members of the same strategy. Ideally, members of the same strategy are able to supplement each other's growth by always cooperating when facing off.

On the other hand, suppose the population of Tit for Tat was represented with function *P(t)*, where *P* is the population size of at time *t*. The growth of Tit for Tat with respect to time can then be represented as $\frac{dP}{dt}$. A defining trait of Tit for Tat is that it retaliates immediately, and thus the optimal strategy when facing a Tit for Tat is to always cooperate. This means that the Tit for Tat population performs well as a collective unit, because it plays its own optimal strategy. As a result, the growth of Tit for Tat is dependent on its own population size, and can be simplified to the equation below.

$$\frac{dP}{dt} = kP$$

where *k* is some constant.

This resembles the Malthusian growth model, where given enough time, the population grows exponentially based on a constant rate.

However, simply being compatible with one's own strategy does not imply success, as we observed with the lack of reproductive success of Kantians and Grudgers. Tit for Tat's strategy succeeds because it provides a way to immediately address uncooperative behavior (unlike Kantian) while having the enough forgiveness to cooperate with predators that are willing to apologize (unlike Grudger). In this way, Tit for Tat is able to maximize points gained with opponents who are willing to cooperate, and minimize losses against opponents who refuse to cooperate.

Conclusion

The data analyzed indeed demonstrated that in an EIPD model, contrary to the conventional Prisoner's Dilemma, the optimal strategy is not to always defect. We demonstrated that given a diverse distribution of strategies in an initial population, a cooperative, retaliatory strategy "Tit for Tat" performs extremely well. Our data suggests that the EIPD model does not have a universally effective strategy, as even "Tit for Tat" performs sub-optimally under certain initial conditions. We analyzed the mechanisms for population growth for predatory strategies and cooperative strategies by comparing them to the Lokta-Volterra model and Malthusian growth model, respectively. We used these models as a basis of understanding for why predatory strategies are inherently unsustainable, and are outperformed by "Tit for Tat" given enough time.

Footnotes

- The full Python framework that was constructed to conduct the research in this paper can be accessed on:

  https://github.com/shoyo-inokuchi/iterated-prisoners-dilemma

- Although not utilized in any manner for the research conducted for this paper, further tools for analysis on the topic of the iterated Prisoner's Dilemma can be accessed on:

  https://github.com/Axelrod-Python/Axelrod

Further research

Although the data analyzed in this paper seems to suggest that cooperative, retaliatory strategies find greater successive at reproducing than selfish, predatory strategies, this data is not necessarily conclusive. If a predatory strategy can identify itself and always cooperate with those of the its own strategy, it can possibly adopt the self-sustainability of "Tit for Tat" and maintain initial explosive growth of conventional predatory strategies. Additionally, although this is beyond the scope of this investigation, implementations of machine learning for the development of a strategy may drastically alter the conclusions made in this paper.