

# The Cooperation Space

## Abstract

To what degree do spatial dynamics affect the viability of the Tit-For-Tat rule in an iterated prisoner's dilemma? How are those dynamics affected by the ecosystem of rules evolved by genetic algorithms? In an effort to better understand some intricacies of cooperation theory and further solidify the foundations laid by Robert Axelrod, a spatially-constrained iterated prisoner's dilemma model was constructed. Using genetic algorithms to evolve rulesets in this model produces Tit-For-Tat rules that tend toward states of equilibrium. The abstract concepts of environmental harshness and mutation rates play important roles in optimizing for the emergence and success of the Tit-For-Tat rule.

## Background

### Motivation

This project began with an interest in evolutionary biology. Consequently, the work of Richard Dawkins was near the forefront of the pursuit of that interest.

In particular, the book *The Blind Watchmaker* inspired me to the possibility of performing a computer simulation relating to some element of evolutionary biology. In the book, Dawkins writes about a simulation he himself performed. It involved the production and reproduction of shapes on a screen - shapes produced by a set of randomized parameters. The reproduction of those shapes, however, was guided by Dawkins' selection of those shapes which were most aesthetically pleasing. The result was a myriad of fantastic looking 'creatures'. The 'creatures' were of shapes and designs which he did not believe he could have come up with himself.

Dawkins' description of his simulation is quite easily recollectable for me. I remembered it while Dr. Osgood was explaining to the class the option for undergraduate students to do a simulation project in lieu of assignments. Although Dr. Osgood stressed the importance of having external shareholders in the project, I stressed the importance of pursuing a topic that was not represented in his list of shareholders. Thankfully, it seemed as though his own interest in the topic was enough for him to make a small exception for me. Dr. Osgood pointed me towards the niche field of cooperation theory, and convinced me to pursue the prospects there.

Dr. Osgood then gave me the name of Robert Axelrod, a political scientist who pioneered the field of cooperation theory in the 80's with his book *The Evolution of Cooperation*. This book, along with Axelrod's *The Complexity of Cooperation*, explored how and why cooperation arises between individuals even when rationality and selfishness are assumed. These themes are explored with a game theory tool called the iterated prisoner's dilemma. Although Axelrod's conclusions are many and often quite complex, he asserts that

there is one behavioural rule in the Iterated Prisoner's Dilemma which is the most robust rule across environments. The rule will be explored later in this document.

It was this research by Axelrod which inspired me to try my hand at a simulation with similar content. However, the focus of my simulation would be the evolution and dynamics of rules within two-dimensional space. Jarrod Pas and began developing the design and implementation of a simulation which could capture evolutionary displacement and spatial dynamics within the context of cooperation and the iterated prisoner's dilemma.

## Context

### Prisoner's Dilemma

The prisoner's dilemma is a non-zero-sum game commonly analyzed in game theory to show why two rational actors may not cooperate even though it may be in the best interest of the pair. The general case is structured as follows:

Two actors compete for a resource. If they choose to cooperate with each other in obtaining the resource, each will receive a *Reward* payoff  $R$ . If they compete aggressively against each other for the resource, each will receive only the *Punishment* payoff  $P$ . However, if one cooperates while the other defects, the defector will receive the *Temptation* payoff,  $T$ , and the cooperator the payoff  $S$ , the *Sucker's payoff*. Although the values of  $T$ ,  $R$ ,  $P$ , and  $S$  may be quite abstract, some general rules must be observed:

$$1) T > R > P > S$$

$$2) R > ((T + S) / 2)$$

(Axelrod, R., 2006, p. 714a)

When these rules are obeyed, the game becomes a legitimate Prisoner's Dilemma. The game is often presented informally as such:

Two members of a criminal organization have committed a crime together and are being investigated separately. They are each given the opportunity to betray the other culprit, or keep quiet. The investigators also state what will happen in the each of the cases:

- If prisoner A and B both betray each other they will get 2 years in jail each.
- If prisoner A betrays prisoner B and prisoner B keeps quiet, prisoner A will get off scot-free and prisoner B will get 3 years, and vice-versa.
- If prisoner A and B both keep quiet they will both get 1 year in jail.

### Iterated Prisoner's Dilemma

The iterated prisoner's dilemma is an extension of the classic prisoner's dilemma. The base rules of the prisoner's dilemma still apply, only the two actors must now consider that they will continue their interactions with the same partner. The number of times which the two actors will interact is not strictly defined.

In the regular prisoner's dilemma, there is a proof that shows that defection is best choice to take for a rational, selfish actor competing with another rational, selfish actor (Axelrod, 2006, 14a). Despite that certitude, the iterated prisoner's dilemma has no such solution. (Axelrod, 20016, 17a) In the iterated prisoner's dilemma, the optimal strategy for interacting with another is determined wholly by the competitor themselves.

In our simulation, we are applying this theory on a large scale. Every rule will be subject to the environment of their neighbour's rules in a two-dimensional world. Therefore, the optimal rule will be determined in large part by the demographics of the model's world at any time.

### Axelrod's Tournament Structure

In *The Evolution of Cooperation*, Axelrod constructed the tournaments a number of times. The first time, only 14 rules were submitted. Axelrod added the rule RANDOM (which, surprisingly, submitted a random choice for every interaction) and ran the tournament in a round-robin style. Each rule competed against every rule (including itself). Every competition between rules consisted of 200 interactions. The best performers in the tournament were those rules which performed *on average* the best.

Our primary simulation differs significantly from Axelrod's tournament structure in that rules are not submitted, but evolved. Furthermore, the spatial dimensions of our simulation means that any rule may not compete with every other - but only with a subset within its neighbourhood.

### Genetic Algorithms

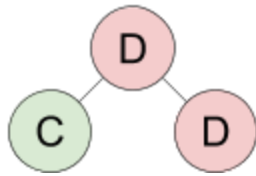
A genetic algorithm is a method for solving optimization problems. It accomplishes this by selecting subsequent solutions for performance in some criterion defined by the problem. Solutions are combined to yield new solutions which are tested for performance in the following simulations. Our approach to this meant defining a criterion for which to select by. Thankfully, the iterated prisoner's dilemma provided a straightforward measurement of performance.

Furthermore, the employment of a genetic algorithm requires that solutions can be encoded in a form which is amenable to 'mixing'. If current solutions cannot be recombined, new solutions won't approximate optimal solutions. Our approach to this required encoding rules as decision trees implemented by lists of characters.

### Decision Trees

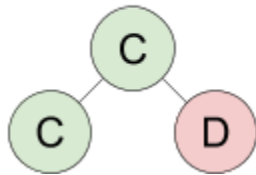
A decision tree maps previous actions to the next action to be taken. Decision trees are made up of decision nodes and paths. The gene which the genetic algorithm mutates is a slight variation on the classical decision tree. We map the actions that have previously been taken against the cell with the gene to a path along the gene's encoded tree. As there are only 2 actions that can be taken the gene's decision tree is represented as a binary tree with a default action (the root node).

### Hostile Tit-For-Tat



Given the Hostile Tit-For-Tat (TFT) tree, the first action the cell with this gene will take is to defect against other cells. If different cell cooperates with this cell, it cooperates back. If the other cell defects, then this cell defects back. This rule only examines the previous action taken against it as there is only one decision that can be made in this tree.

### Trusting Tit-For-Tat



If the hostile TFT rule competes alone with a rule that always cooperates, it will win out with a score of  $Rn+T$  vs  $Rn$ . Against a rule that always defects, it will tie. Finally, if it competed against a 'trusting' TFT rule, we would observe the chain of events:  $[(d,c),(c,d),(d,c),(c,d),(d,c),\dots]$ , where the two rules alternate receiving the  $T$  and  $S$  payoff, bringing the game to a tie.

## Goals

One of the primary goals of this simulation project was to successfully integrate a genetic algorithm with an iterated prisoner's dilemma. As there is no best move in the iterated prisoner's dilemma independent of the actor's partner, it was our hope that the dynamics of an evolving ecosystem of rules would give rise to co-operative rules in interesting ways. Considering that this has already been done successfully (Axelrod, R., 1997, p. 20b), our own results are satisfying. These results are explored later.

A second goal of this project was to endogenize spatial behaviour of rules on the Surface of our simulation. This too was successfully accomplished. Our model's assumptions about reproduction, death, and movement resulted in the emergence of definite spatial patterns. Although often simple, the spatial patterns were easily recognizable manifestations of two-dimensional interactions. For example, our model showed that when successful rules reproduce, and poorly performing cells migrate, there will almost invariably emerge colonies of cells in which Tit-For-Tat is the dominant rule.

In our appendices, we have pictures showing just that. At first, individual Tit-For-Tat rules spring up by chance. However, they do not survive because their neighbours will not cooperate with them at all. They die from attrition. These lone Tit-For-Tat rules continue to pop in and out of the existence until approximately generation 30, when a few can finally be found adjacent to each other. At this point, they are guaranteed a stable relationship of reward between themselves. By generation 44, there are almost 10 of them grouped together. By generation 53, they have grown to over a dozen. Figure 4 and 5 begin to show greater coherence within that subset of the population.

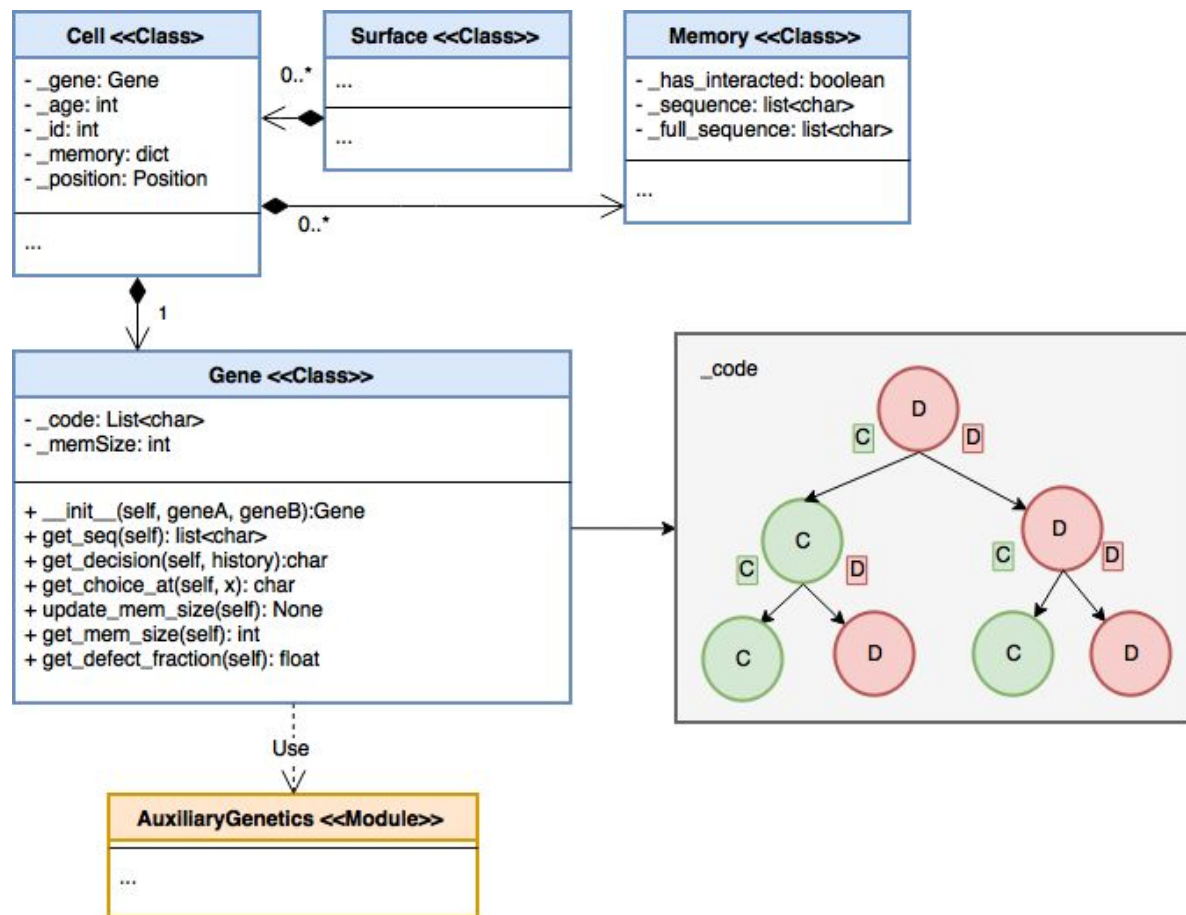
This also marks the beginning of an interesting spatial relationship between a few types of rules. Refer to Figure 4 in the appendices; a cell 'xddd' is shown highlighted in red. If you look closely, you can see it preying on two other cells, both 'occc'. These represent a rule which always defects, and a rule which always cooperates, respectively. What is occurring is that the stable Tit-For-Tat population ( otft ) is harboring the all-nice rules 'occc'. The all-nice rules, in turn, attract the all-mean rules 'xddd'. This is because the nice rules perform well on the periphery of the population of Tit-For-Tats, but cannot retaliate on their own against the all-d rules. If the all-nice rule was more ingrained in the area dominated by

the Tit-For-Tat rule, it would not be so vulnerable to exploitation. This is shown again in Figure 6, wherein Tit-For-Tat rules occupy the greater part of the surface.

## Model

### Genetic Algorithm

Our approach maintained a reasonably decoupled architecture. The gene of an actor in our simulation and the implementation of the genetic algorithm rest behind the interface to the simulated world that our actor (a 'Cell') provides.



The auxiliary genetics module contains the functionality for combination and mutation of genes. Gene combination was the tool we used to evolve new rules. To combine genes, we overlay gene A and gene B and pick randomly between the two for each position. We do this until we reach the end of the shorter gene. The length of the new gene is the average of its two parent genes. The new gene's remaining length is made up of the longer parent's choices. This process is shown in the following example:

```

          CDCDDDC
combined with DCDCCDCDDCDC
results in   CDDCCDCDDCDC
  
```

Finally, a series of mutations are applied to the gene. These are intended to roughly approximate the natural processes of gene combination. There are three defined processes:

a flip mutation, a deletion mutation, and an insertion mutation. Every mutation functions by iterating over each choice ('c' or 'd') in the gene, and for each position, performing the mutation according to some probability parameter,  $P(f)$ ,  $P(d)$ , or  $P(i)$ .

First, we apply a flip mutation to the gene, in which there is a chance per position to flip the current choice from co-operate to defect, or vice versa. Then we apply the deletion mutation, which has a probability per position to delete the choice from the gene. Finally, we apply an insertion mutation, which has a probability per position of inserting a new random choice into the gene.

### Grid Based Model

Our simulation was run on a toroidal world, where cells occupy positions in two-dimensional space. Every space on the surface has eight neighbouring connected spaces, meaning the maximum number of neighbours any cell can have is 8. From the surface of the two-dimensional world, we move in forward in distinct time-steps. Every time-step sees a number of processes occur:

```

191     self.__clean()
      ...
194     for x in range(inters):
195         self.__interaction_tick()
196         self.__death_tick()
197         self.__alt_movement_tick()
198         self.__reproduction_tick()

```

First, every cell has their memory wiped and their scores reset to a base value. Then a *for* loop begins, and runs through as many iterations as the parameter of *interactions per tick* dictates. In `__interaction_tick()`, every cell interacts with all its neighbouring cells, and its score accumulates those results. Then, if the cell's score has fallen to 0 or lower, it dies, freeing up the space it occupied. If the cell is still alive, it has the opportunity to move. The cell may move if it is within the poorly performing bottom fraction of the entire surface's population.

Finally, after *inters* interactions, the best performing cells have the opportunity to reproduce. This fraction and the chance associated with it are also parameters. The process is not entirely trivial. First, we select the best performing cells; only they will have the opportunity to mate. We then iterate over those cells; selecting one at a time and finding another with which it can reproduce. The mate for the cell to reproduce with is another cell in its neighbourhood who has not already reproduced. If it finds one, it can reproduce into an empty adjacent space. If no empty adjacent spaces exist, it cannot reproduce.

Both the moving and reproducing for these cells have used randomness wherever possible to reduce any inherent bias in the model. The model harnesses stochastics by setting a global random seed, which allows for reproducible results. Analysis of the model can be performed confidently by running simulations with identical parameters and varying only the random seed.

## Assumptions

A number of important assumptions about the function of the model were made. These were necessary to give the model structure. The assumptions range from being trivial or inconsequential to being quite central to the model's behaviour. Higher level assumptions include the model's concepts of reproduction, movement, and death - all elements of a simulation's single time-step. As well, assumptions were made about the model's structure in regards to competition and performance measurements. More minute assumptions include details of rule behaviour, such as the incorporation of cell 'Memory' and the decision to implement rules with binary decision trees. Generally, these assumptions relate to the overarching definition and application of evolution and evolutionary pressures.

A primary goal of the simulation was to analyze the results of evolutionary processes on rule sets in an iterated prisoner's dilemma. Insofar as this was a challenging goal, evolutionary processes were made as abstract as possible. The concepts of *Natural Selection* and *Survival of the Fittest* were modeled by prescribing individual cells (and therefore their rule) a *score* value. This was the measure of rule performance. Well performing cells had higher scores, and poorly performing cells had lower scores.

These scores were used as input for a number of other model processes. Reproduction was handled by selecting a fraction of the generation's top performing cells, and giving them the opportunity to reproduce with other well performing cells in their neighbourhood. If a cell performed poorly in any particular generation, it would not have the opportunity to reproduce before the next simulation time-step. Additionally, in the interest of keeping the spatial dimension of the model simple, any reproducing cells could only reproduce if there was room adjacent to them. Finally, cells could only reproduce once per generation.

These assumptions about reproduction constituted a number of important exogenous factors for the model. Although these particular properties of sexual reproduction are not common in nature (to say the least), they do have their parallels. It was important to keep the process as simple and abstract as possible to maintain the simulation's tractability. However, the issues of tractability and time constraints are in conflict with accuracy. It is possible that a style of sexual reproduction that more closely mirrored those found in nature (which would be more complicated) would yield different results.

Similar to the handling of reproduction was the handling of spatial movement over time. Poorly performing cells were given the opportunity to move several times in a single time step. Again, a poorly performing fraction of a generation's population was given the opportunity to move so long as there were empty spaces adjacent to it. Given the opportunity to move, a poorly performing cell has only a *chance* to move - it may not. This was a decision made to give the simulation more variability in its spatial behaviour, and it attempts to model the behaviour of an organism foraging for resources. What it results in is that well performing groups of cells maintain their structure, whereas poorly performing groups behave entropically. This creates a rich environment for the development by well performing cells.

Of course, this approach is not perfect. It neglects to address the obviously complex issues of an organism's movement in favour of simplicity. The parameters involved in this process such as the fraction of cells which may move and their odds of moving, however, were simple enough to allow us to frob these values fruitfully in our experimentation.

Another important assumption was the organization of the rules themselves. We opted to represent the rules as strings of 'c's and 'd's encoded as binary trees because it was extremely conducive to the genetic algorithm we used to model sexual reproduction. Furthermore, this choice was relatively simple. It was inspired by Axelrod's approach to genetic models for his rules (Axelrod, R., 1997, p. 18b), but had one key difference: our implementation allowed for Cell to track their memory of interactions beyond only three moves. Longer genetic code corresponded to a longer memory, and a more complex rule. Axelrod's implementation only kept track of a rule's previous three interactions, whereas ours had no theoretical limit.

This still lead to some restrictions; there is a certain class of rules which cannot be imitated by our approach. Take, for example, the rule which always cooperates until it has been betrayed once with a single defection, and from then on adopts the policy of permanent retaliation against the offender. This rule is not possible to replicate with our implementation because after some finite number of turns, the single defection would always be forgotten by the rule.

Finally, the meta-assumption of the entire model's structure must be addressed. The elements of our model together represent only a very crude and abstract construct. However, previous work by Axelrod suggest that the iterated prisoner's dilemma is nonetheless a valuable tool with which to imitate the natural processes of cooperation and competition in nature and in sociology. (Axelrod, R., 1997, p. 36b) This gave us confidence in the basic correctness of our model's use of the iterated prisoner's dilemma's scoring system and how we related it to a cell's performance.

## Sensitivity Analyses

### Changes in Environmental Harshness

In our model, environmental harshness was abstractly modeled as the loss of energy per interaction per cell combined with the base score. These come into effect while calculating the score of a cell. The formula to calculate a cell's score for one tick is their base score,  $B$ , plus the sum of their scores with every particular neighbour ( $pn$ ) minus the loss per interaction ( $l$ ). A cell has  $N$  neighbours, which in practice may vary between 0 and 8. The per-neighbour scores are summed over the number of interactions per tick ( $t$ ). In mathematical terms, a cell's score every tick is calculated as:

$$B + \sum_i^t \left( \sum_j^N (pn_j - l) \right)_i$$

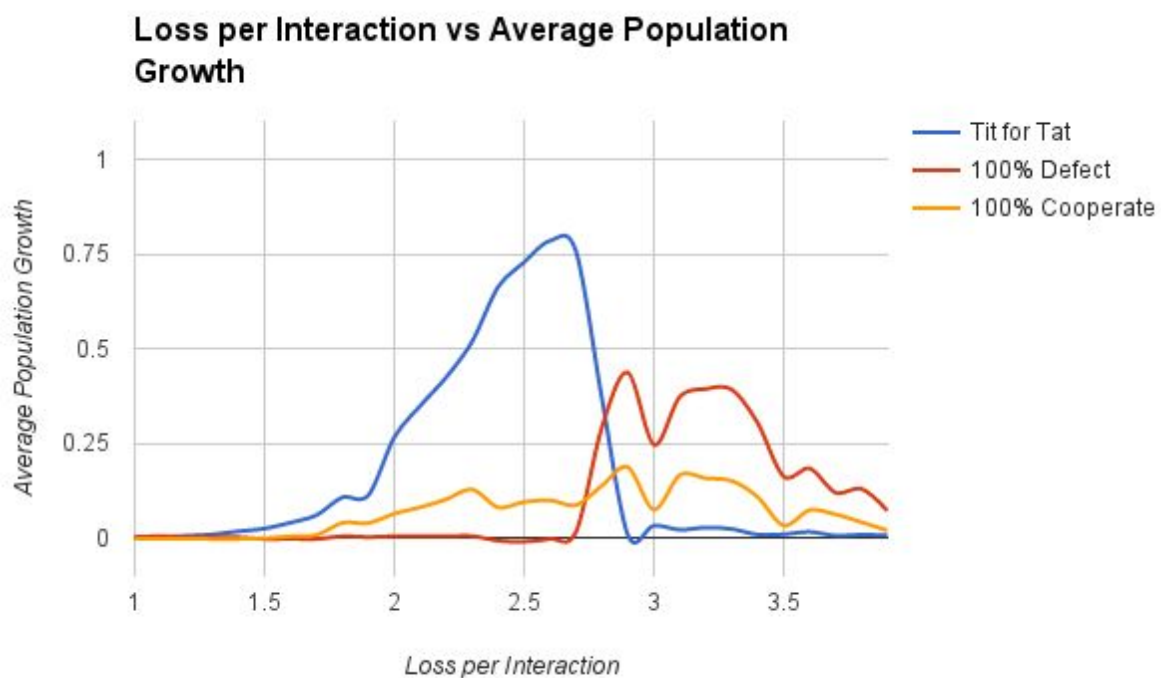


Keep in mind that a cell dies at the end of an interaction if their score has fallen to or below 0. It is important to note a few details of this formula: cells will not die if the base score,  $B$ , is large enough to offset the penalties that very poorly performing cells will incur over the course of one tick. Cells also won't die if the value  $I$  is very small, or if the values of the score matrix are all positive in such a way that even the sucker's payoff,  $S$ , is greater than  $I$ .

We decided to take a close look at the effects of varying the loss-per-tick would have on relative populations of rules. We ran simulations for every value of  $I$  from 1.0 to 3.9 in steps of .1, for 10 random seeds. This was a total of 300 simulations. Nominal parameters for these simulations were the base score,  $B = 20$ , and the score matrix, structured as:

$$T = 5, R = 3, P = 1, S = 0$$

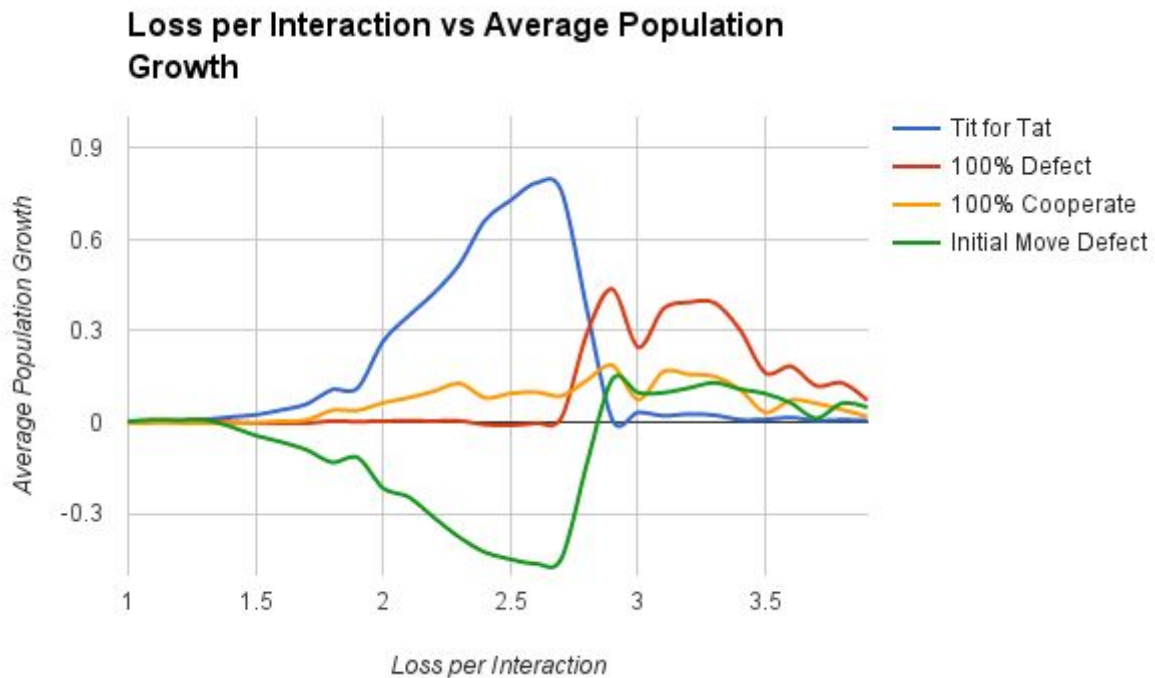
All simulations were run for 200 generations. These simulations revealed some of the figurative bounds and structures endogenous to our simulation.



This graph shows the average slope (the growth from generation 0 to 200 ) of the surface's population which are the rules *Tit-For-Tat*, *Always Defect*, and *Always Cooperate*. There are three interesting aspects of this graph. First, we see that for very low values of  $I$ , there is very little change in any population. This is because cells are not dying, and most cells in the initial random population do well enough to avoid having their score fall to 0. Without room for growth, no new strategies can be evolved.

Then, we see that for values between 1.5 and 2.8, the average slope of the surface's population of TFT rules begins to grow substantially. For simulations with relatively low selection pressures ( $I$ ), the change is small. However, for competitive environments, the relative populations of the TFT rule increases from none to as high as 78%.

Finally, we see how the advantages for TFT rules provided by the parameter  $I$  in the range 1.5 - 2.8 disappear completely when the simulation is run with a value of  $I = 2.9$ . Populations of TFT cannot become stable and flourish when the reward for mutual cooperation ( $3.0 - 2.9 = 0.1$ ) are not great enough to overcome the initial betrayal and frequent punishment ( $S = 0 - 2.9 = -2.9$ ,  $P = 1 - 2.9 = -1.9$ ) that these rules have to endure from the chaotic states that are produced at the simulation's inception. Furthermore, this tells us that the success of genetic algorithms at producing cooperative rules is influenced greatly by the harshness of the environment in which those solutions are bred.



These conclusions are further confirmed by the additional information of the relative populations of 'mean' rules on the surface. A 'mean' rule is a rule whose initial move is to defect. We see the average slope of their population for all values of  $I$  mirror the same values for which the average slope of the population of TFTs is increasing and decreasing.

The success of 'mean' rules and rules who only defect is allowed by the relatively higher values of  $I$ . As  $I$  increases past 2.8, cells which cannot prey upon other cells with success die, and cells who can prey efficiently become the majority of the cells selected for reproduction.

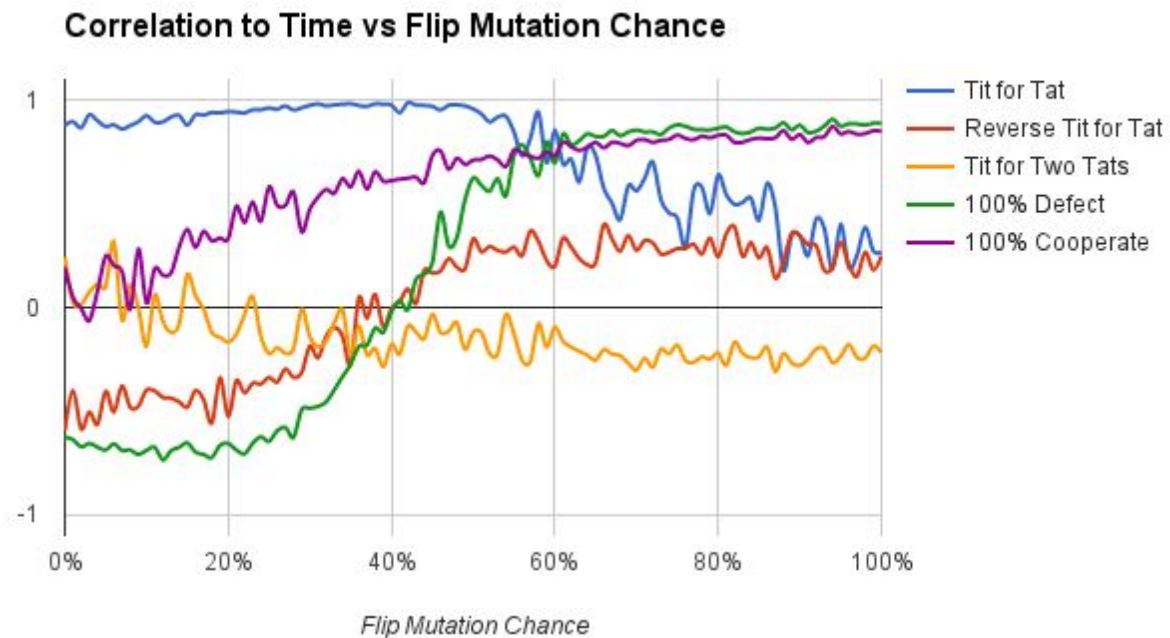
### Changes in Surface Size<sup>1</sup>

As the surface size increased we saw that the slope of all of our measured outputs became less erratic but followed the same behaviour and trends. This is an expected outcome due to the larger sample of cells that interact. We can conclude from this that the surface area has no dependant variables. We can run larger simulations to reduce uncertainty without worry of affecting the model in unintended ways.

<sup>1</sup> This took a ton of CPU time, but it really did not lead to interesting conclusions

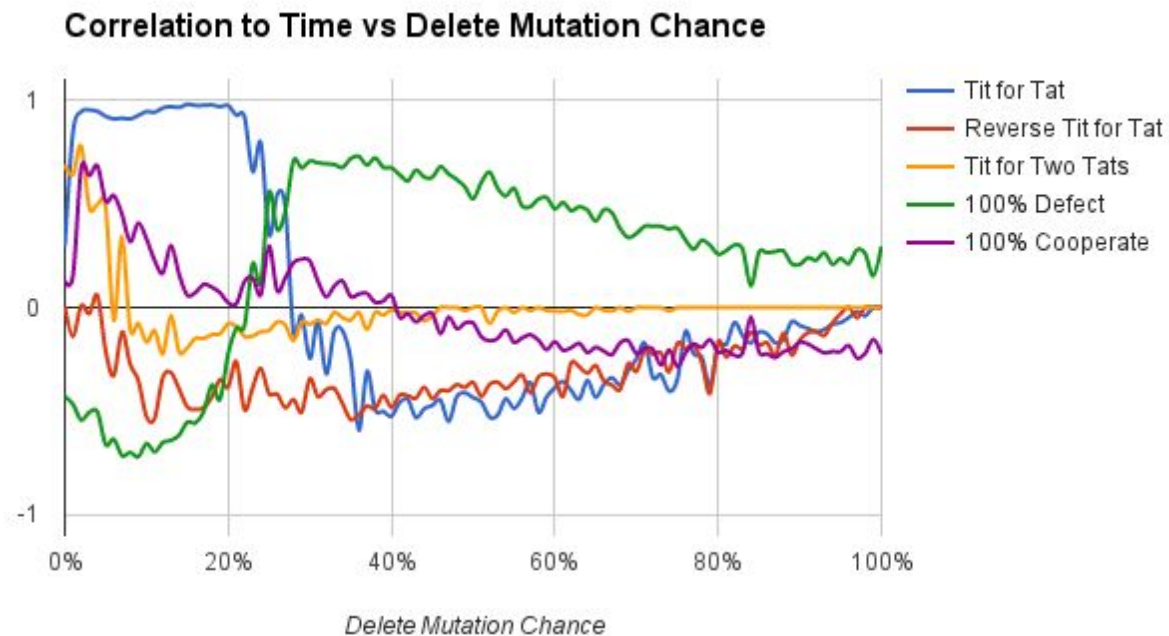
## Changes in Mutation Probabilities

To test the sensitivity for the mutation chances we varied the parameter for both flip and delete mutations between 0% and 100% at 1% intervals (0%, 1% ... 99%, 100%). For consistency, we ran each variation with 10 different seeds. We then calculated the mean correlation coefficient to time for each parameter variation. This allows us to see how the trend of our population's rule types are affected by the variation in mutation probabilities. I.e., a near 0 correlation value is equivalent to approximately random data, and a value that approaches 1 or -1 shows a significant relationship between time and our tracked population's trend.



For variation of flip chance, the development of Tit for Tat (TFT) rules is highly correlated until around 50%. This shows that for high probabilities of the flip mutation TFT rules are less likely to form in large quantities. Another interesting note is the 100% Defect rules go from a negative correlation to a positive correlation as the mutation chance increases. Similarly, we can see an increasing correlation for 100% Cooperate rules from a 0% chance of a flip to 100%. These findings show that the flip mutation rates can be important determinants of the optimal rule for a given environment, and can therefore affect the genetic demographics of the environment as a whole.

For variation of delete chance, the development of TFT rules is highly correlated with time until around 23%, at which point it then plummets. This change in correlation is mirrored by 100% Defect rules. Because genes can not be shorter than 1, we have a high prevalence of 100% Defect and 100% Cooperate rules. However, all the correlations tend towards 0, because the population variation becomes limited to only 'c' and 'd' rules. This then causes the 100% Defect rules to dominate over the 100% Cooperate rules. Once all the 100% Cooperate rules are have died, the 100% Defect rules die out shortly afterwards, because they cannot be sustained on the payoff  $P$  alone.



## Learning

### Simulation Performance

If we were to undertake this project again we would look for better ways to parallelize our simulation. We found that with larger simulations it would take exponentially more time due to our simulation being bound to a single thread/process. With the model in it's current state, we run multiple instances with different sets of parameters. However, we have multiple simulations which may not terminate at the same time. One way to accomplish a performance increase would be to pair up cells and iterate over all pairs of cells and get them to interact.

### Sensitivity Analysis

We also found that simple linear regression does not seem suited to the sensitivity analysis of models which display non-linear behaviour ( surprise! ). If the data sample for the sensitivity analysis of the input parameter *loss-per-tick* had been more condensed, it may have seemed as though simple linear regression would provide an accurate measure of the relation between *loss-per-tick* and output such as the proportion of a particular rule on the surface's population. That would have led to a false conclusion about the linear effects of that input parameter. However, our approach of using scatter plots to represent the input-output relationship still led to insight into endogenous factors of our model. Even though we had difficulty with non-linear analysis, we were able to use it for analysis of variation in mutation probabilities.

## Future Work

Many aspects of our simulation can be improved, and many more paths of exploration presented themselves throughout our research. Improvements such as computational performance stand to be vastly improved. Software engineering components of the project which (perhaps always) remain incomplete include both internal and external documentation. More importantly, basic model assumptions could be changed to focus the simulation towards other subjects. As it stands, complex or difficult to measure simulation outputs present an untapped source of results.

A good example of this is the analysis of the finer details of the simulation's two-dimensional spatial behaviour. Not only could the graphical display of the two-dimensional environment be made more colourful and detailed to better represent the environment, but we did not make any measurements of the spatial qualities of that environment. Only qualifications and hypotheses were made as to the meaning of our results. Despite our lack of focused analysis of the spatial environment, we were still able to deduce some conclusions from it.

Additionally, there is interest in identifying points of equilibrium and tipping-points of the phase space of our model. Although our simulations very obviously tend toward many particular equilibriums, they have not been well analyzed. These equilibriums include the case where all cells die before giving rise to any sustained population, as well as all the cases in which the two-dimensional space is fully saturated, but every cell is doing well enough not to die. In such cases, however, there remain a very large number of possible states.

Take the example of a simulation run in which the spatial dimension parameters are a width of 20 and a height of 20. If the surface is saturated, then there are 400 living cells. In such a case, there could be as many as  $400!$  possible states of equilibrium. Presumably, there is a lot of opportunity here to apply tools of dimensional analysis and reduction. In this case, it is likely that the intrinsic dimensionality of our model is less than the nominal dimensionality. If we restrict our scope, however, and look only at the relative proportions of our population, a restricted-dimensionality phase space could reasonably be constructed.



## Conclusions

In conclusion, our hypotheses that the simulation would see stable populations of Tit-For-Tat and cooperative rules emerge organically were validated. In fact, the success of the Tit-For-Tat rule under nominal conditions was greater than expected. The formation of stable populations of Tit-For-Tat of up to as high as 79% arose often, the difference being made up of almost exclusively cooperative rules, as in the following simulation:

```

*-----*
|otft otft otft otft otft  occc otft otft otft  occc occc otft otft otft  occc occc otft occc otft otft |
|otft otft otft otft otft  otft otft otft  occc occc otft otft  occc otft otft  otft otft otft otft |
|otft otft otft otft  occc otft otft otft  xftf otft otft  occc occc otft  occc otft otft otft otft |
|otft occc occc otft otft  otft otft  occc otft otft otft  otft  occc otft  ocde occc otft  xdcc otft otft |
|loccc otft otft otft otft  otft otft otft  otft otft otft  otft  occc occc otft  occc otft  occc otft  xftf |
|loccc otft otft otft otft  otft otft otft  otft otft otft  otft  otft otft  otft  otft  occc otft  otft |
|otft otft otft otft otft  occc otft otft  odcc occc otft otft  occc occc otft  odcc otft otft  ocde otft |
|otft otft otft otft  occc otft otft otft  otft otft otft  otft  occc otft  occc otft  otft otft  otft  xcdd |
|otft otft otft  occc otft  otft otft otft  otft otft otft  otft  otft otft  otft  otft  occc otft  otft  otft |
|otft otft otft otft otft  otft otft otft  otft otft  occc occc otft  otft otft  occc otft  otft  otft  xcdd |
|otft xftf otft otft otft  otft otft otft  otft otft  occc otft  otft otft  otft  otft  otft  occc ocde otft |
|otft otft otft otft  occc occc otft otft  otft  occc otft  otft  otft  otft  otft  otft  otft  occc ocde otft |
|otft otft otft otft otft  otft  occc otft  otft  occc otft  otft  otft  otft  otft  otft  otft  occc otft |
|otft otft  occc otft otft  otft otft otft  occc otft  occc occc otft  otft  occc otft  otft  otft  occc occc |
|otft otft  occc otft otft  otft otft otft  otft otft otft  otft  otft otft  otft  otft  otft  otft  otft  otft |
|otft otft otft  occc otft  otft  occc otft  otft otft  occc otft  otft otft  otft  otft  otft  otft  otft  otft |
|otft otft otft otft otft  otft otft otft  otft otft otft  otft  otft  occc occc otft  otft  occc otft  otft |
|otft otft otft otft otft  otft otft otft  otft otft  occc otft  otft  occc otft  occc occc occc otft  otft |
|otft otft otft otft otft  otft otft otft  otft otft  occc occc occc otft  otft  occc occc occc otft  otft |
|otft otft otft otft otft  otft otft otft  otft  xcdd otft otft  occc occc otft  otft otft  occc otft otft |
*-----*
| population: 400 | born: 3977 | died: 3577
| generation: 182
| def.frac : 27.654167 | init.move : 1.750000
| tfts : 78.750000 | ftfs : 0.750000 | t2ts : 0.000000 | all_d : 0.000000 | all_c : 18.000000

```

These results agree with those of similar simulations performed by Axelrod, wherein the results of his own experiments with evolutionarily derived solutions to the iterated prisoner's dilemma "cooperated from the beginning and employed reciprocity in a manner akin to Tit-For-Tat" (Axelrod, R., 1997, p. 26b).

As discussed previously, there remains a plethora of other avenues for exploration. Further analysis of simulation equilibria and spatial dimensionality are warranted, as well as investigation of evolutionary mechanisms for dealing with complex rules. Novel discoveries derived from our simulations include the optimal environments for fostering Tit-For-Tat rules and spatial behaviour of rules as discussed under *Goals*, as well as the effects of mutation rates on the evolution of rules and their environment.

## Citations

Axelrod, Robert., *The Evolution of Cooperation*. [Electronic edition]. Basic Books (2006)

Available from:

[https://books.google.ca/books?hl=en&lr=&id=KFf2HXzVO58C&oi=fnd&pg=PR7&ots=9iXBanYXDf&sig=71wESSCv7\\_V9sr8yPYzOGJlp0hE#v=onepage&q&f=false](https://books.google.ca/books?hl=en&lr=&id=KFf2HXzVO58C&oi=fnd&pg=PR7&ots=9iXBanYXDf&sig=71wESSCv7_V9sr8yPYzOGJlp0hE#v=onepage&q&f=false)

Axelrod, Robert., *The Complexity of Cooperation: Agent Based Models of Competition and Collaboration*. Princeton University Press (1997).





