From Simple Games to Complex Behaviours: Prisoner’s Dilemma

Andrew Lee

17 May 2017

Partner: Samuel Kirby

Supervisor: Dr Dave Clements

Imperial College London

Abstract

In this report, we extend the iterated prisoner’s dilemma to a two-dimensional regular network of agents on a periodic cartesian grid, and explore the propagation of 5 simple deterministic strategies for the game (game strategies) across the network under a small variety of different strategy changing methods (update strategies). We find a purely local update strategy favours ‘mean’ strategies who will defect with no provocation, whereas a less strict ‘semi-local’ update strategy that considers neighbours’ total scores favours ‘nice’ strategies who will never pre-emptively defect. The semi-local update strategy resulted in average equilibrium scores across the network 2.3 times than the average equilibrium scores for the local update strategy.

1. Introduction

1.1 The Prisoner’s Dilemma

The prisoner’s dilemma is a classic simple nonzero-sum game from game theory which has been studied since 1950. The game was invented by the RAND Corporation, a US thinktank, with the intent of understanding the incentives associated with nuclear war. The game consists of two players, each of whom have two possible moves, ‘Cooperate’ and ‘Defect’. The game can be explained as a minimisation problem, where two conspirators are being held by the authorities. They can either cooperate with each other by telling the authorities nothing, or defect by telling the authorities about the conspiracy. If both players defect, then both receive a punishment, e.g. spending two years in prison. If both players cooperate, then both receive a reward, e.g. spending

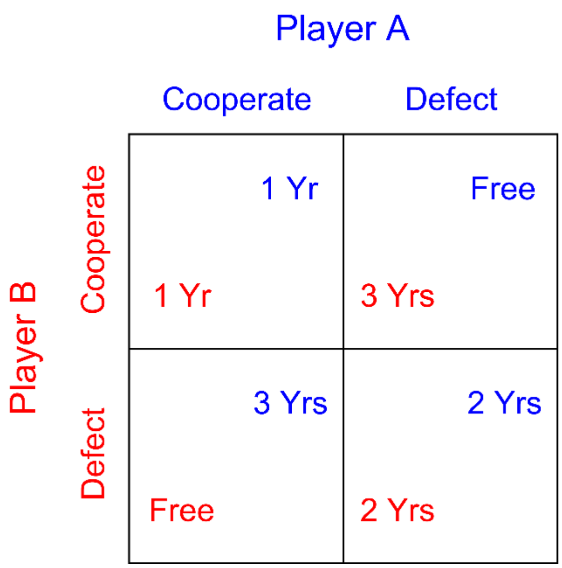


Figure 1: This example payoff matrix for the Prisoner's Dilemma shows amount of jail time served by each player for each of the four possible outcomes. The outcome for each player is written in the same colour as the player themselves, for example, if Player A cooperates and Player B defects, then Player A serves 3 years, and Player B walks free.

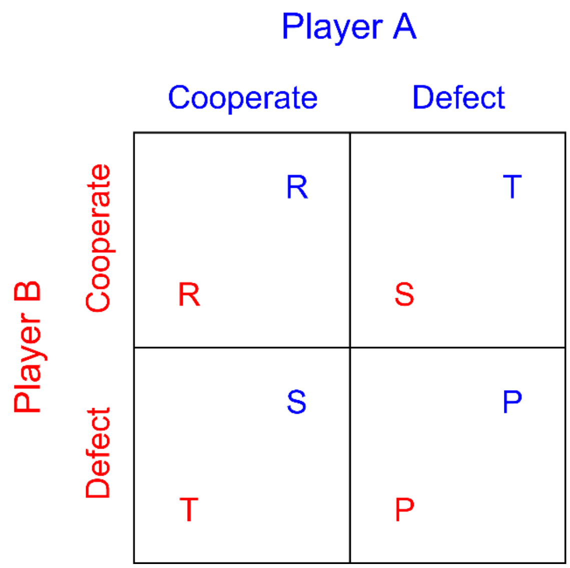


Figure 2: A generalisation of the payoff matrix in figure 1. Formulating the problem as a maximisation problem, where high scores are good, we require the values to follow the inequality T > R > P > S, so that the incentives match those of the original problem.

only one year in prison. However, if one player defects and the other cooperates, then the cooperator receives the 'sucker's payoff', e.g. three years, while the defector is allowed to walk free with zero time in jail – the temptation (see figure 1). The game is referred to as nonzero-sum because the total jail time in each outcome is not the same – both players agree that mutual cooperation leads to a better outcome than mutual defection.

More generally, if we reframe the problem as a maximisation problem, then it follows the payoff matrix in figure 2, with values such that **T** > **R** > **P** > **S**. This same game can be used as a simple model for the Cold War [1] – both sides wanted to destroy the other without being destroyed themselves, analogous to the **T S** cells of the matrix, and both sides would prefer for both to continue existing than for both to be destroyed, analogous to the **R** and **P** cells.

For a single round of the Prisoner´s Dilemma, the incentives are such that a rational player should defect. This is because whatever the opponent plays, your own score will be higher if you defect. Therefore, if both players are playing rationally, the only outcome will be mutual defection – a rather depressing outcome, especially for modelling nuclear strategy. This outcome is an example of a 'Nash equilibrium', as no party can improve their outcome by changing only their own strategy[2].

1.2 Iterated Prisoner's Dilemma

A way to break this mutual defection is to play multiple consecutive rounds of the Prisoner's dilemma, and consider a player's aggregate score across all rounds. This then means that a player can take the chance at cooperating with their opponent, and still be able to retaliate if the opponent defects. In this game, an additional constraint is placed on the score matrix so that mutual cooperation between the players results in higher scores than alternating cooperate-defect and defect-cooperate moves, i.e. 2**R** > **T** + **S**. This shifts the incentives in favour of cooperation, as demonstrated by considering a player who switches from always cooperating to always defecting after some number of rounds as shown in figure 3. However, this is conditional on the expected move by the opponent to be cooperation. If you live in a world where most players defect by default, then the correct move should also be to defect. I.e. a population of permanent defectors is stable – no matter what strategy you pick, you will always be outperformed by your opponent if they always defect, unless you also always defect. This stability of selfish behaviour is an open problem not only in game theory, but in the development of cooperative – or altruistic – behaviours in many species, including humans. If the selfish nature favoured by Darwinian natural selection is correct, then how have so many cooperative social dynamics arisen across nature?

This question has many proposed solutions, including so-called 'kinship theory', the idea that acting altruistically towards close relatives directly increases the chances of your genes surviving as many of your genes are shared with your relatives, and 'group selection', where a group of cooperating individuals outcompetes groups of selfish individuals by the more efficient utilisation of resources[3]. However, both of these explanations have issues, namely the inability of kinship theory to explain cooperation between non-relatives, and group selection's inability to explain why a group of cooperators are cooperating in the first place.

An alternative explanation for the development of cooperative behaviour was investigated by Robert Axelrod in 1980 using the iterated prisoner's dilemma as a mathematical model for cooperative and selfish behaviours[4]. He ran an experiment where 15 different strategies submitted by experts including economists, mathematicians and political scientists played each other in a round robin tournament to see how different strategies performed against each other. Effectively, 15 'agents' were created, and each programmed with a different strategy. After all the games were played, the agent with the highest average score across the tournament used the so-called 'Tit for Tat' strategy – one of the simplest submitted – where the player cooperates for the first round then repeats the opponent's previous move back at them. Thus cooperating

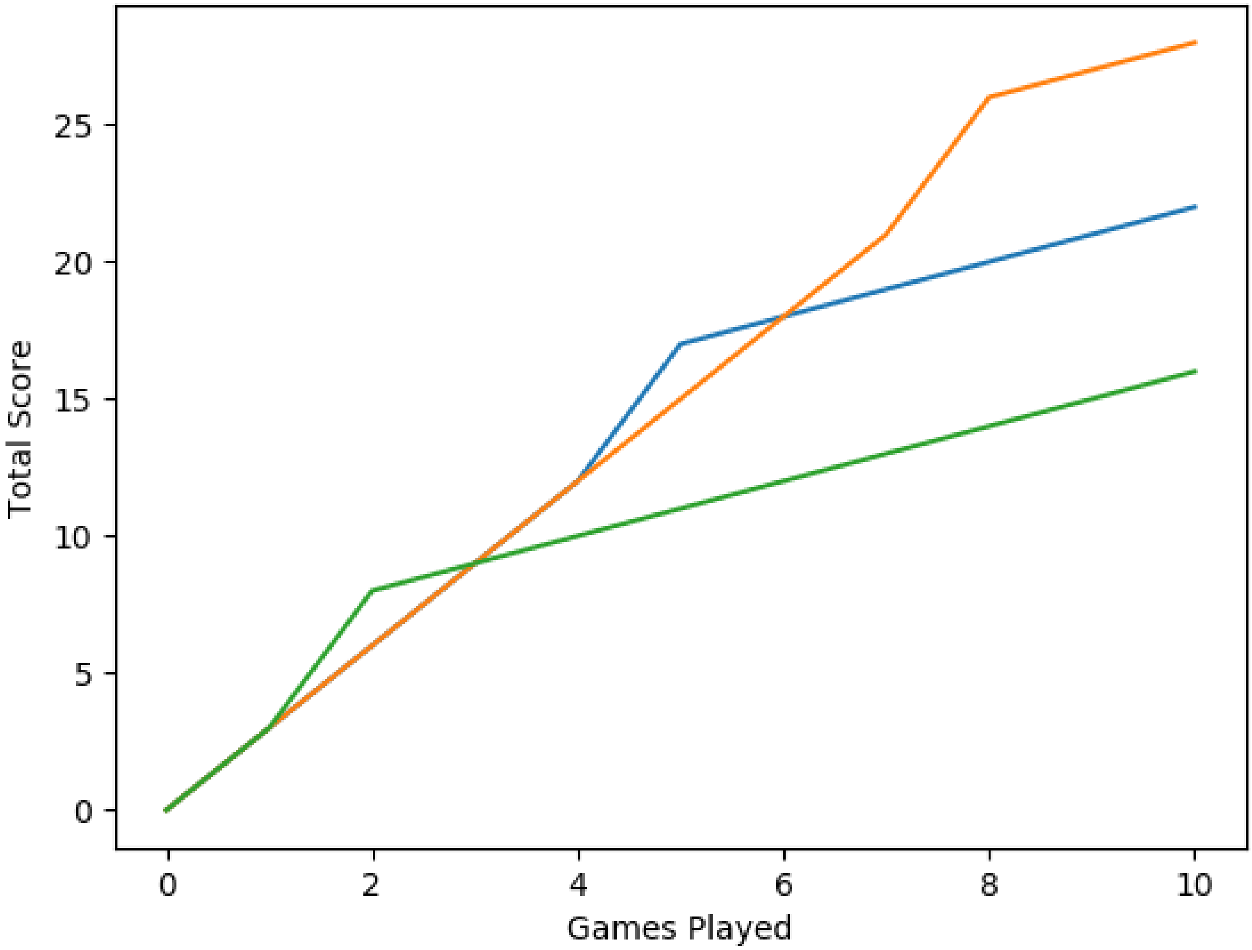


Figure 3: This graph shows the total score for one of two agents playing each other, under three scenarios. Both agents start cooperative, but after some number of rounds, d*, one agent defects against the other, followed by mutual defection henceforth. The three lines show the total scores for* d *= 1, 4 and 7 for the green, blue and orange lines respectively. These were calculated using R = 3, T = 5 and P = 1.*

with the agent is rewarded with a cooperation in the next round, and defection against the agent is punished with a retaliatory defection.

Axelrod later repeated this experiment with a larger number of strategies from a much wider range of sources, both academically and geographically, but again the Tit for Tat strategy came out on top despite strategies being submitted specifically to exploit it[5].

1.3 Networks of the Iterated Prisoner’s Dilemma

While the Axelrod tournaments are very interesting, and helped progress the understanding of the iterated prisoner’s dilemma, it effectively assumes that every agent in the tournament knows every other agent, as every possible pair of agents play the game with each other.

If we were to use the prisoner's dilemma as a model for cooperative versus selfish behaviour in the real world, then this round robin system does not make sense – most pairs of people never meet. To model this kind of incomplete information, we created networks of agents where each agent is only connected to a small subset of other agents in the network, and each agent assigned some 'game strategy' to decide whether to cooperate or defect. Then, the iterated prisoner's dilemma was played between each pair of connected agents independently of every other connected pair – effectively each edge in the network played the prisoner's dilemma simultaneously. Each agent then updated its game strategy based on its own performance according to some 'update strategy'. This cycle of playing then updating was repeated to observe how different strategies behaved and propagated across the network, and to see what emergent phenomena would arise.

2. Simulation Details

The set of possible game strategies for the Iterated Prisoner's Dilemma are infinite, as are the set of all possible update strategies. Therefore, to achieve results in finite time, we only considered few of each, as detailed in table 1 and table 2.

Game Strategy	**How it works**
Always Cooperate	Always plays Cooperate
Always Defect	Always plays Defect
Tit for Tat	Cooperates for first turn then replays the opponent's last move
Suspicious Tit for Tat	Defects for first turn then replays the opponent's last move
Tit for Two Tats	If opponent's last two moves are both Defect then Defect, else Cooperate

Table 1: This table details the five game strategies investigated in this report, and states how each game strategy decides whether to cooperate or defect.

The update strategies themselves can be broken down into three categories; Local, Semi-local, and Global update strategies. These refer to the amount of information available to the agent when it decides whether to change its game strategy, and what to change it to.

In a local update strategy, the agent only considers information it can directly see, i.e. the results of games the agent itself has played. This is the purest form of the prisoner's dilemma, as the agents are not supposed to be able to communicate with each other, except through playing the game.

With a semi-local update strategy, the agent can see how well its neighbouring agents have performed in the previous round, effectively 'interrogating' them on their performance. For example, the agent could know the total or average score of each neighbour. One can visualise

Update Strategy	Changes When	Changes To
Interrogate *Local*	Change when no neighbouring agent has lower total score than this agent	Change strategy to that of the connected agent with the highest average score across all of their games in the previous round
Threshold *Semi-Local*	Change strategy when this agent's average score is worse than the majority of average scores achieved against this agent by its neighbours	Change strategy to that of the connected agent with the highest score against this agent
Gross Domestic Produce (GDP) *Global*	Always change	Change strategy to the strategy with the highest aggregate score across the entire network in the previous round
Produce Per Person (PPP) *Global*	Always change	Change strategy to the strategy with the highest average agent score across the entire network in the previous round

Table 2: This table details the main update strategies implemented in the simulation. The two listed global update strategies, GDP and PPP, are not investigated in this report, but were implemented in code, and are listed here to provide further context for the space of possible deterministic update strategies. In the case of ties where multiple agents fulfil the criteria to be selected as the new strategy, the simulation falls back on a directional tiebreak, as detailed in section 4.1.

this scenario as either each agent meeting up with its neighbours at the end of a day's work and boasting about how well they have done, or as agents inferring their neighbours scores much like people infer their neighbours' salaries based on their homes, clothes, cars etc.

Finally, global update strategies use some information calculate from the state of the entire network of agents, for example the total number of agents using each strategy, or the average score achieved by each strategy. One can look at this as modelling people reading the news or looking at the stock market at the end of the day, and seeing how different corporations are performing.

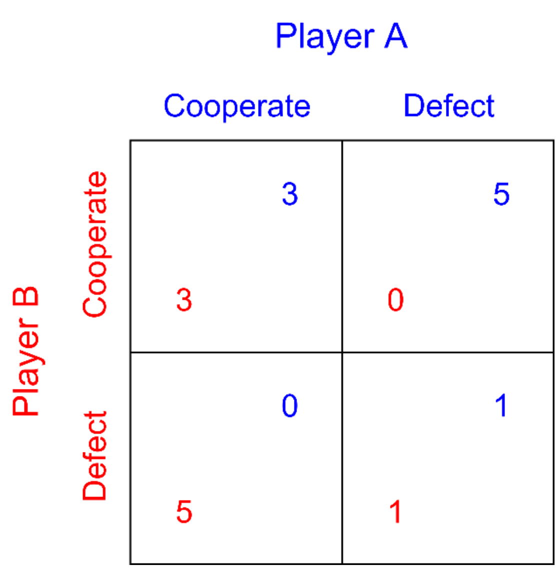


Figure 4: This shows the payoff matrix used in these simulations, i.e. R = 3, T = 5, P = 1 and S = 0. This fulfils both inequalities specified in sections 1.1 and 1.2.

For every simulation explored in this report, the payoff matrix used was that shown in figure 4, the same as that used in the Axelrod tournaments[4].

3. Method Random Square Network

For the random square network, each simulation had 4 parameters:

- l, the side length of the square
- m, the number of Games Per Round (GPR), or the number of consecutive games of the Prisoner's Dilemma in each game of the Iterated Prisoner's Dilemma
- N, the number of rounds of the Iterated Prisoner's Dilemma, and subsequent strategy updates processed in the simulation
- U, the update strategy used by the agents

An l by l array of agents was created, with each agent at a different integer 2D coordinate. Each agent was randomly assigned one of the game strategies listed in table 1, and every agent assigned the same update strategy, U, one of those listed in table 2. A list of 'edge' objects was then created by connecting each agent to the surrounding 8 agents. Each edge therefore had two component agents, and each agent was a component of 8 different edges.

These edges between the nodes then all independently played the iterated prisoner's dilemma between their two agents with m consecutive games, and the results of the Iterated Prisoner's Dilemma recorded by both agents. The agents then updated their game strategies according to their update strategy, U. This play-update process was then repeated N times to propagate changes across the network.

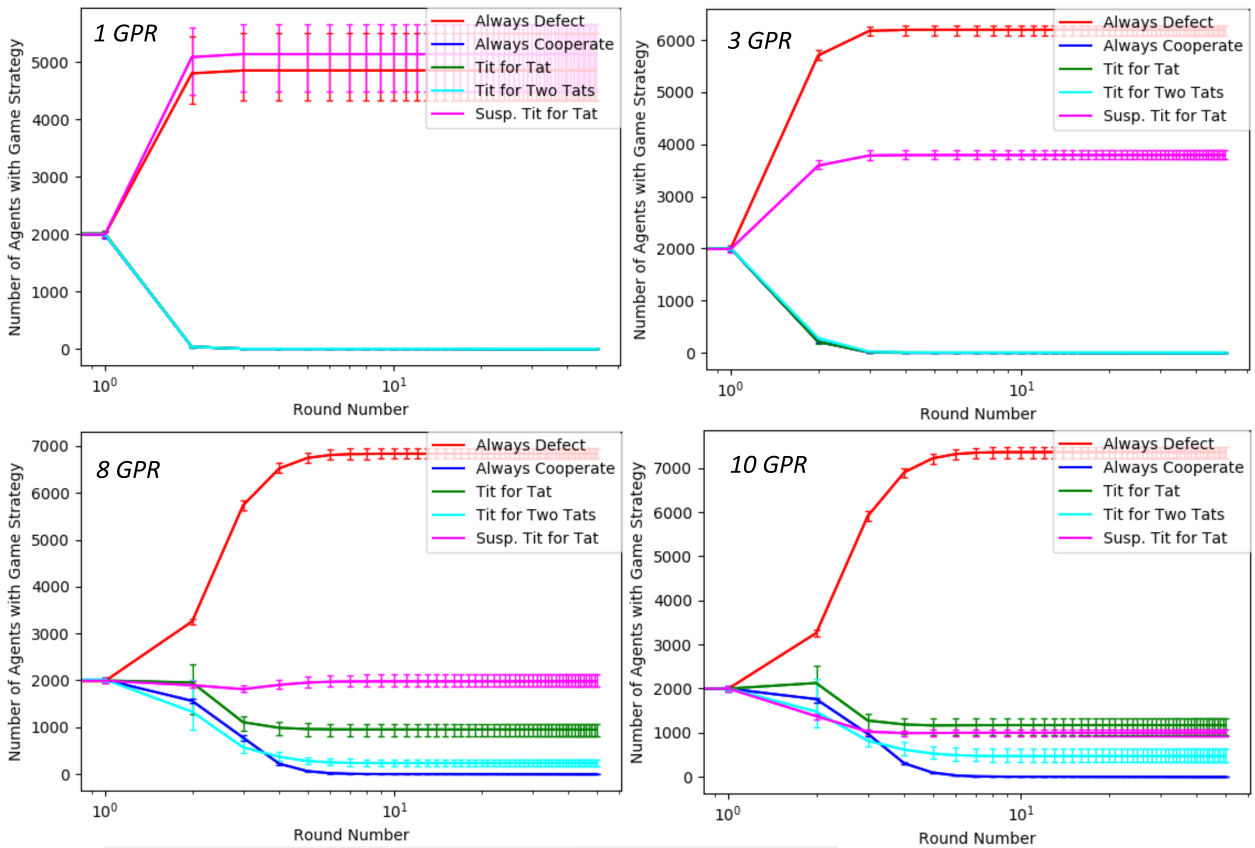
Initially this was implemented in Python, but it was then rewritten in C++ to reduce runtime.

The program was run for a range of network sizes, to see if there were any characteristic length scales of the problem. This process was repeated for a range of strategies, sizes, GPRs, and numbers of rounds, as discussed below.

4. Results and Discussion

4.1 Local Update Strategies

Initially, the program was run for a range of GPRs, on a 100x100 network of agents with all of them using the local Threshold update strategy, as this was the update strategy closest to the original form of the prisoner's dilemma, as it requires the least information for each agent to decide whether to change strategy. The simulation was run 200 times for each GPR value, and



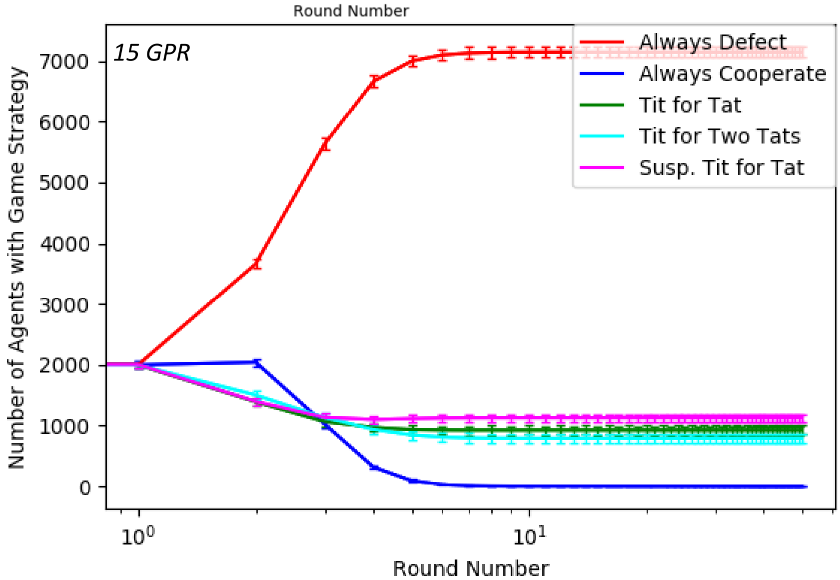


Figure 5: These graphs show the number of agents using each of the 5 game strategies in the network over the first 50 rounds for a range of GPRs from 1 to 15. The values for the line are the mean populations of agents over 200 simulations, and the error bars show the 10th and 90th percentiles of these 200 simulations, indicating the spread of results.

the mean population of each strategy over the first 50 rounds for a few of these is shown in figure 5. The upper and lower error bars on each point show the 90th and 10th percentile in the distribution for that point, showing the spread of outcomes in strategy fractions. The graphs are displayed on a log scale in round number to emphasise the first few rounds, as this is where interesting behaviour is taking place.

For the remainder of the report, we will focus on the 15GPR case, as beyond this, the equilibrium state has stabilised by this point, as shown by figure 6, but the computational load was still

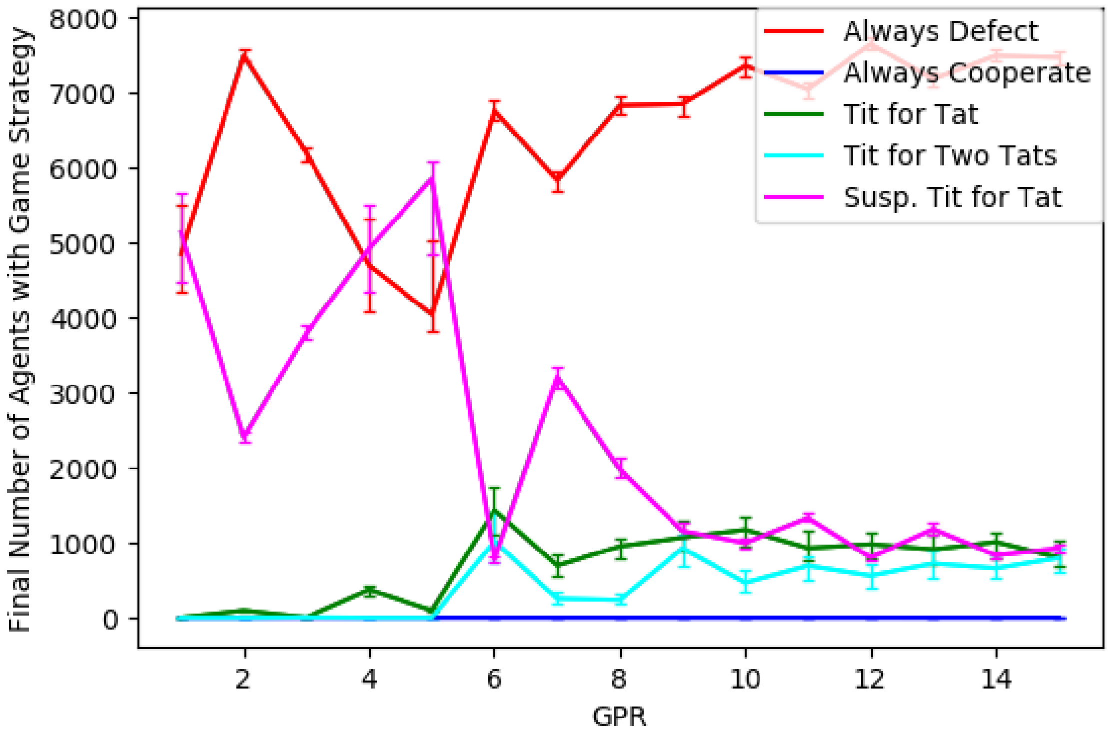


Figure 6: This graph shows the mean number of agents for each game strategy for 200 simulations after the first 50 rounds for a range of GPRs. Again, the error bars indicate the 10th and 90th percentiles of the 200 simulations. The lines stabilise above ~12 GPR, with the primary oscillation due to the advantage Suspicious Tit for Tat holds over Tit for Tat for an odd GPR, and parity for an even GPR thanks to their constant defect-cooperate, cooperate-defect back and forth.

manageable for a couple of hundred simulations. The decision to use 15 GPR was also partly due to 15 being odd. This means that the Suspicious Tit for Tat game strategy scores higher than the Tit for Tat game strategy when they play each other. It is a somewhat arbitrary decision, as for even GPRs, there is no difference in score, but it was decided that overstating the effect relative to an average of multiple GPRs was preferable to neglecting the effect entirely, and so an odd GPR of 15 was selected.

The Always Defect game strategy rapidly dominated the networks in these simulations, as it is not possible to score higher than an Always Defect agent when playing against it – the best possible result against it is to always defect yourself. This means that it is not possible to displace an Always Defect game strategy from an agent. In addition to this, it is the most favourable game strategy to displace Always Cooperate agents, as it racks up a huge score of 75 from its 15 consecutive defect-cooperate moves. This allows it to rapidly spread across the network, as shown by the images of an example network in figure7.

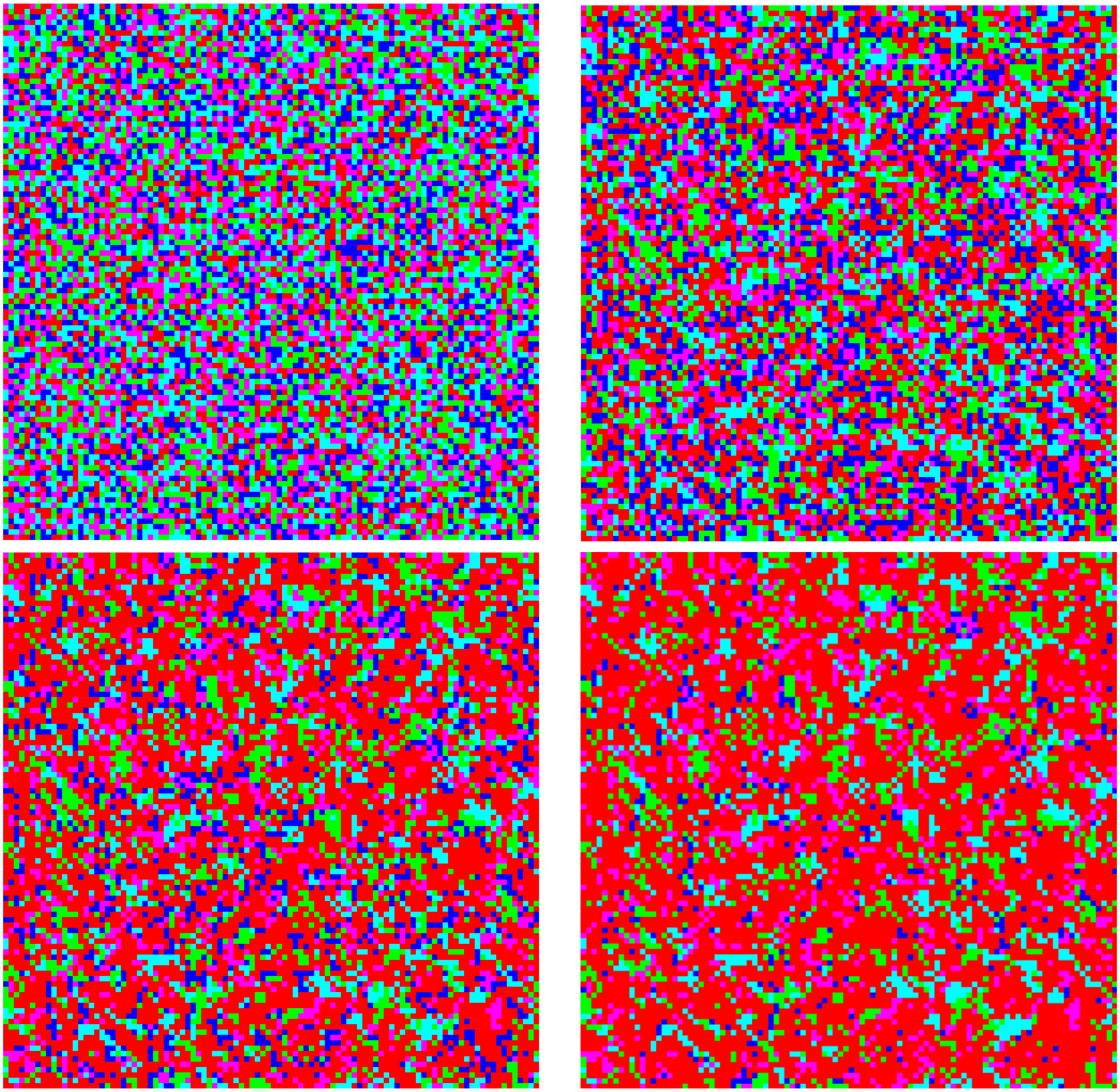


Figure 7: These images show the initial state of a network (top left), and the state of the network after rounds, 1, 2 and 3, in the top right, bottom left and bottom right respectively. The colours of each square indicate the game strategy used by the agent at that position, with the same colour scheme as previous figures, i.e. Always Defect in red, Always Cooperate in blue, Tit for Tat in green, Tit for Two Tats in cyan, and Suspicious Tit for Tat in magenta.

The graphs show that the equilibrium state of the network is reached after only the first 10 or so rounds. Beyond this there is still some variation, as there are infinite cycles that can occur in the network, for example the cycles shown in figure 8. This shows there both exist cycles that change the global populations of the strategies, and cycles that have no effect on the global populations.

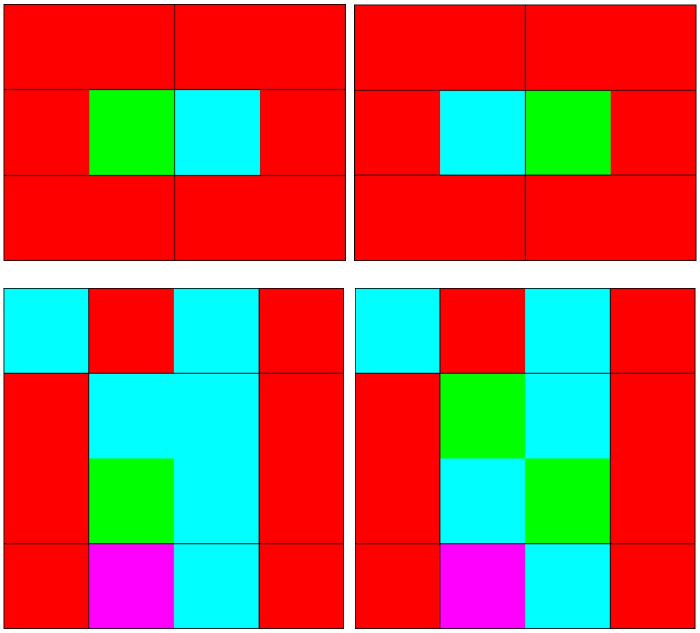


Figure 8: These images are samples from one of the 200 simulations. The top two images show the two states of a cycle that does not affect the global populations as the two game strategies simply swap places forever. However, the bottom two images show the two states of a cycle which does change the global populations – in aggregate, one agent constant flips from Tit for Tat to Tit for Two Tats and back again forever. In both of these cases, the network has reached equilibrium, as there are no further genuine changes in state, merely short cycles.

I.e. even if the strategy populations have reached equilibrium, that does not mean the network as a whole is unchanging.

For local update strategies, there is clearly a size above which a periodic network appears infinite to all agents, at least so long as the simulation only runs for a finite number of rounds. This is because information can only propagate across the network at a rate of one agent per round. For example, if we consider a 5x5 network played only for 1 round, then the network appears infinite – the information an agent receives from one side is entirely independent to the information it receives from the opposite side. Or more generally, using this definition for an infinite network we can say a square network with $l > 2N$ will appear infinite for a purely local update strategy. For a semi-local update strategy, the formula would be $l > 4N$, as information can propagate two nodes per round. Thus the 100x100 network appears infinite for at least the first 49 rounds.

However, this is merely an upper bound on the threshold value of l for which the network will appear infinite, as in most cases the information does not actually propagate at this maximum speed for the entire duration of the simulation. It is possible to engineer networks with initial conditions where this is the case (see figure 9), but for the vast majority of networks, the state of the network reaches an equilibrium value much faster than this theoretical limit. Once the network reaches this state, there is no further information propagation, so effectively the network appears infinite if $l > 2E$ for local and $l > 4E$ for semi-local update strategies, where E is the number of rounds taken to reach equilibrium.

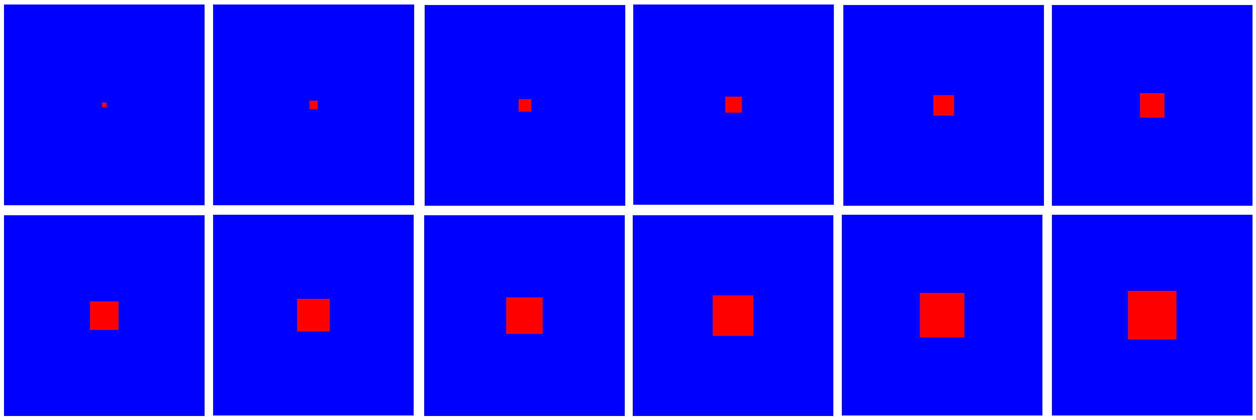


Figure 9: This figure shows images of a network where information travels at the maximum rate for the duration of the simulation. It is initially a single Always Defect agent in a 100x100 network full of Always Cooperate, and so Always Defect game strategy spreads at one agent per round. However, this is an astronomically unlikely scenario for a random network – the vast majority of networks will reach an equilibrium state much faster than this will.

Whether a network appears infinite is a separate question from whether there are characteristic length scales to the problem. For example, while a 3x3 network run for 1 round does appear infinite, it is not a representative demonstration of behaviour of all such 3x3 networks. A very small network is highly sensitive to the initial configuration of strategies.

The Threshold update strategy has a slight quirk – the agent checks its average score against the score of each of its neighbours, and if its average score is worse than a majority of its neighbours scores, it changes to the game strategy of the neighbour with the highest score against it. However, if multiple agents have the same score, which neighbour should it pick? The answer is it checks through the neighbours in order from top left (or north west) to bottom right (south east) as shown in figure 10, and picks the first one it finds with the correct score. This actually occurs quite often as many of the strategies score the same against each other, for example, Always Cooperate, Tit for Tat, and Tit for Two Tats are all always mutually cooperative with each other and themselves, and Always Defect and Suspicious Tit for Tat are always mutually defective.

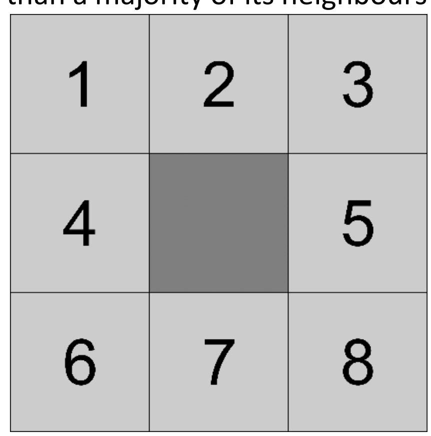


Figure 10: For an agent at the centre of the image above who has tied strategies to choose from, it will check through its neighbouring agents in ascending numerical order, and change strategy to that of the first tied agent it finds.

This preference for the top left agent's game strategy being copied both leads to diagonal lines of agents being created, and to some game strategies – primarily Always Cooperate – 'migrating' south east, as shown by the four images of a network using 14 GPR in figure 11. As the Always Cooperate

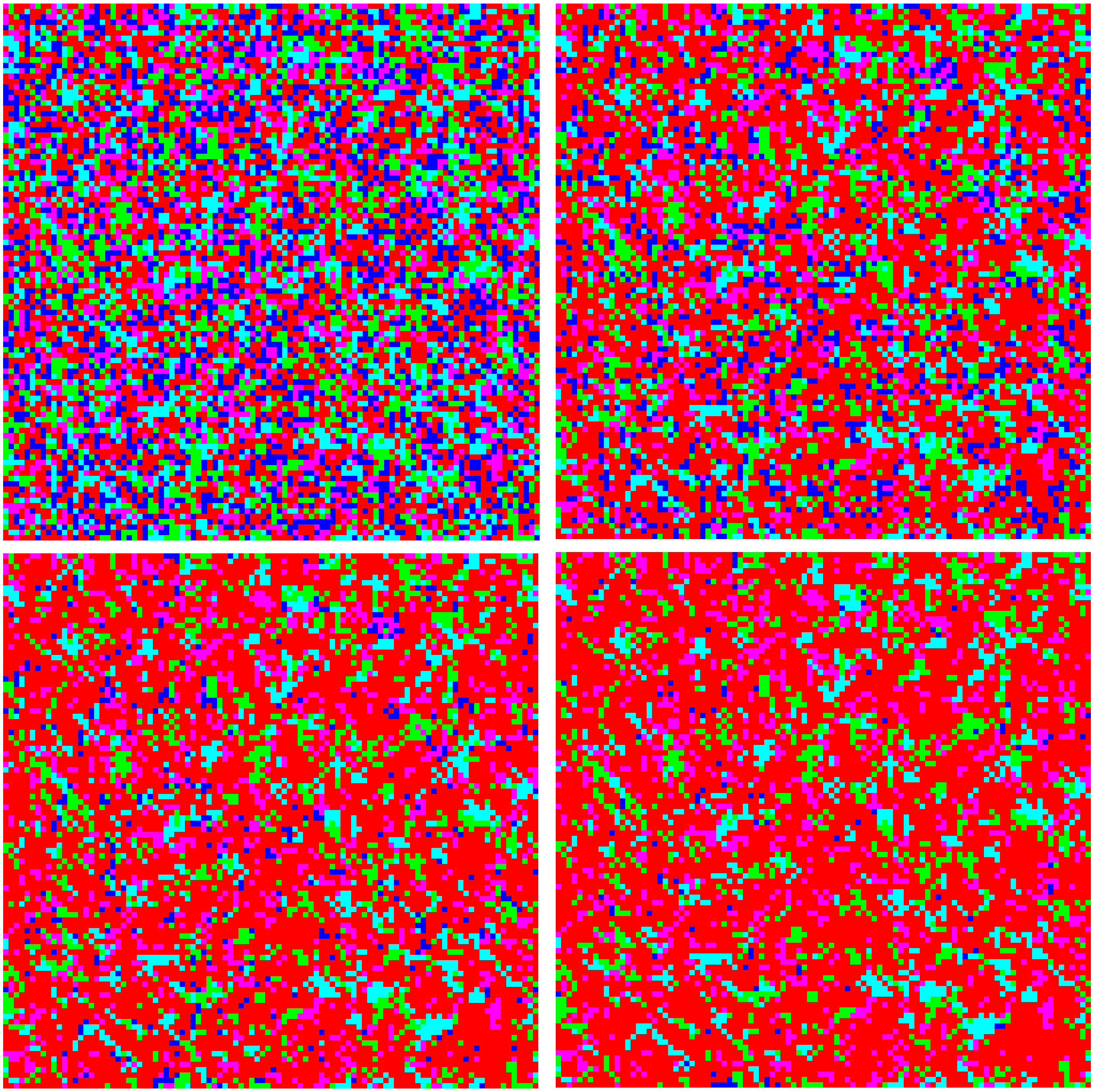


Figure 11: These images show the migration of the Always Cooperate game strategy in the 'south east' direction. This is most clear when looking at the second image, where the Always Cooperate game strategy is mostly eradicated from the south western sides of cooperative regions, but is still very much present on the north and west of the blocks. In truth it was mostly annihilated everywhere after the first round, but was chosen as the new strategy for many cooperative agents on the north west side, due to the directional tie break.

scores just as well against Tit for Tat and Tit for Two Tats as other Tit for Tat or Tit for Two Tats agents, it will often displace them if to the north west of them

One easy solution to this would be to randomly select a direction in case of a tie, however that would make the game non-deterministic – the equilibrium state of the network would not be solely dependent on the initial state of the network. We decided we preferred to have a strict logical chain in the simulation so that every decision was justifiable in isolation, However, this does leave investigation into probabilistic strategies open for future work.

Another proposed solution would be to switch to the most numerous strategy of the tied strategies that the agent can see. While this would be deterministic, there would still be many times where there are equal numbers of the tied strategies, e.g. two Tit for Tat and two Always Cooperate agents, and so the simulation would still frequently have to fall back on the directional tie break – it wouldn't actually resolve the directional favouritism in any substantial way, it would just deal with

Instead, we abandoned a purely local solution to the problem, and considered tiebreaking to the neighbour with the highest total in the round – effectively a hybrid of the Threshold and Interrogate update strategies. There is only a small chance of two agents having identical total scores, and so the simulation will only need to fall back on the directional tiebreak much more rarely, effectively removing the directional favouritism. As such was a semi-local update strategy, and is contrasted with the Interrogate update strategy below.

4.2 Semi-Local Update Strategies

The Threshold with score tiebreak has very similar population curves to the purely local version, as shown in figure 12. The main differences between the two are the slight increase in population share of the Tit for Two Tats update strategy with the new tiebreak, and a slightly smaller number of rounds to reach the equilibrium state, E. Both of these changes make sense. The improved performance of the Tit for Two Tats update strategy makes sense, as it is the only strategy able to 'convert' the Suspicious Tit for Tat strategies to mutual cooperation. As such, we would expect it to have a higher average score than other strategies, and so be favoured in the tiebreaks. The smaller E is explained by the breaking of the top left favouritism, which prevents the creation of the migratory strategies seen in figure 11. These migrations delay the equilibrium state, so removing them will reduce E.

It is worth noting that as this is now a semi-local update strategy, the network would have to be of size $l > 4N$ for it to be truly pseudo-infinite. However, we continue to use a 100x100 network and 50 rounds for these tests, violating this rule, as we found the equilibrium was reached after only around 6 rounds for the Threshold with tiebreak – well above the $l > 4E$ threshold.

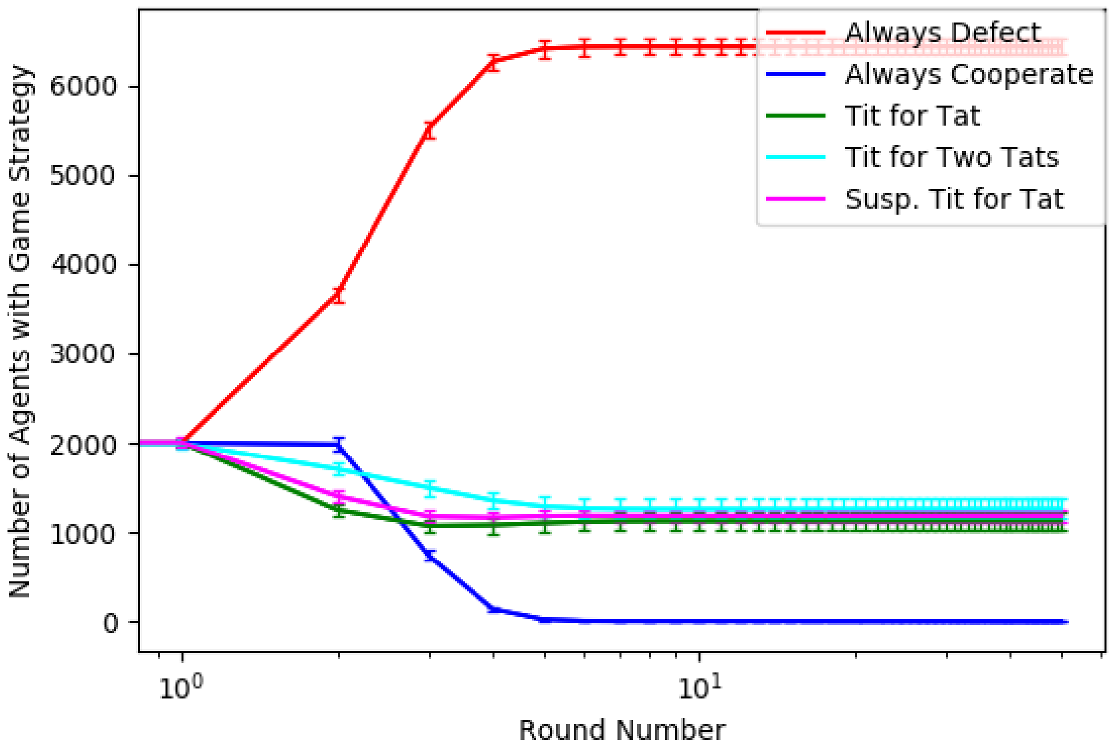


Figure 12: This graph show the number of agents using each of the 5 game strategies in the network over the first 50 rounds with 15 GPR using the semi-local Threshold with Score Tiebreak update strategy. The values for the line are the mean populations of agents over 200 simulations, and the error bars show the 10th and 90th percentiles of these 200 simulations, indicating the spread of results. The shape here is almost identical to the purely local Threshold update strategy, with the main difference being a slight shift in favour of the Tit for Two Tats game strategy.

However, we also have another semi-local update strategy, the 'Interrogate' update strategy, as stated earlier in table 2. The differences between this and the Threshold strategies are striking (see figure 13). The Always Defect game strategy – the most successful under the Threshold game strategy – is almost eradicated, along with Suspicious Tit for Tat, the other 'nasty' game strategy. This seems to be because while defection is unbeatable in a one-on-one game, a mutual defection game scores far worse than a mutual cooperation game (see table 3), and so is heavily penalised by the Interrogate update strategy. For example, an Always Defect agent playing against a Tit for Tat agent will score a single free hit where it defects and the opponent cooperates, but beyond this mutual defection takes over leading to final scores of 19 and 14 respectively. Contrasting this to a mutual cooperation game, where both agents would score 45, it is clear how 'nice' strategies outperform 'nasty'. Under the Threshold update strategy, each agent only has information of how other strategies perform against their own strategy. This means that an Always Defect agent will never be able to see the potential benefits of a 'nice'

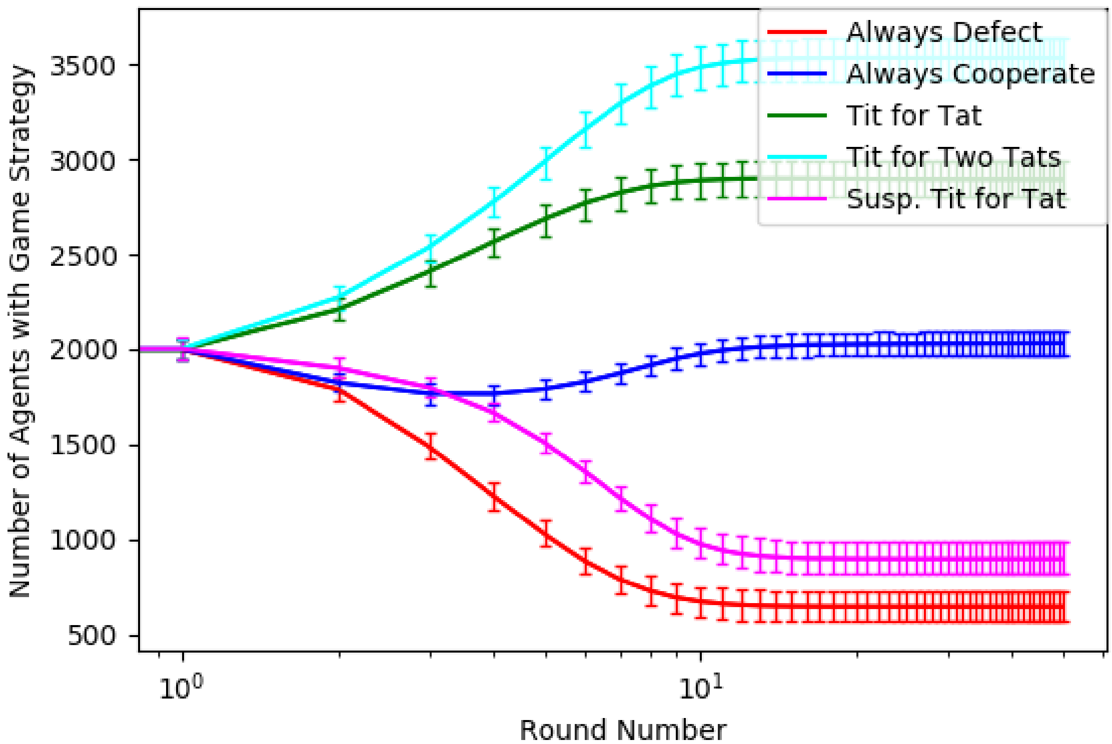


Figure 13: This graph show the number of agents using each of the 5 game strategies in the network over the first 50 rounds with 15 GPR using the semi-local Interrogate update strategy. The values for the line are the mean populations of agents over 200 simulations, and the error bars show the 10th and 90th percentiles of these 200 simulations. Of particular interest here is the massive shift in favour of the cooperative 'nice' strategies, and the interesting shape of the Always Cooperate line.

strategy, whereas under the Interrogate update strategy, the benefits of 'nice' strategies is broadcast clearly by the much higher aggregate scores from mutual cooperation games.

The only game strategy which ever significantly outperforms the mutual cooperation score of 45 is the Always Defect against Always Cooperate, which scores 75. However, every other outcome for the Always Defect game strategy is substantially worse than the mutual cooperation outcome, dragging the average score below that of cooperative strategies, as shown in table 4.

This great increase in cooperative behaviour when considering neighbours' total scores could be viewed as a crude model for empathy between agents. From this viewpoint, the observed increase in cooperative, or altruistic behaviour is consistent with results from sociology showing that increased communication between agents leads to more equal splits of some reward[6].

Agent 1 Game Strategy	Agent 2 Game Strategy	Agent 1 Score	Agent 2 Score	Total Score
Always Defect	Always Defect	15	15	30
Always Defect	Susp. Tit for Tat	15	15	30
Susp. Tit for Tat	Susp. Tit for Tat	15	15	30
Always Defect	Tit for Tat	19	14	33
Always Defect	Tit for Two Tats	23	13	36
Susp. Tit for Tat	Tit for Tat	35	30	65
Always Defect	Always Cooperate	75	0	75
Susp. Tit for Tat	Tit for Two Tats	47	42	89
Susp. Tit for Tat	Always Cooperate	47	42	89
Tit for Tat	Tit for Tat	45	45	90
Tit for Tat	Tit for Two Tats	45	45	90
Tit for Tat	Always Cooperate	45	45	90
Tit for Two Tats	Tit for Two Tats	45	45	90
Tit for Two Tats	Always Cooperate	45	45	90
Always Cooperate	Always Cooperate	45	45	90

Table 3: This table shows the results of each possible game strategy matchup in the simulation, assuming 15GPR, and using the payoff matrix in figure 4. 'Nice' game strategies are highlighted in pale green, while 'nasty' game strategies are in pale orange. Scores above the mutual cooperation result of 45 are shown in blue, scores below 22.5 (50% of the cooperative result) are shown in orange and red, between 22.5 and 45 in yellow, and exactly 45 in green. These show that the mutual cooperation result is only very occasionally outperformed.

Perhaps one of the most interesting phenomena of the population graph for the Interrogate update strategy is the decline and subsequent resurgence of the Always Cooperate game strategy. Using the above data we can attempt to explain it.

Game Strategy	Average Score
Always Cooperate	35.4
Always Defect	29.4
Tit for Tat	35.8
Susp. Tit for Tat	31.8
Tit for Two Tats	38

Table 4: This table shows the average score for each game strategy if they were to play a round robin tournament like Axelrod's over 15 rounds. Note that Tit for Two Tats performs best, even though it loses more to the Always Defect strategy than Tit for Tat, and that Always Defect has the lowest average score, despite being able to achieve the highest possible score of 75 against Always Cooperate.

Initially, there are all the game strategies randomly distributed across the network. Where an Always Cooperate agent is adjacent to an Always Defect agent, it is very strongly penalised, giving the Always Cooperate agent a high chance of switching to another strategy – probably Tit for Tat or Tit for Two Tats, as they both have a high average score. Always Defect agents not adjacent to an Always Cooperate agent are also very likely to change strategy, as their highest score against non Always Cooperate agents is 23 – barely more than half of the mutual cooperation score of 45.

Therefore after the first couple of rounds, most Always Cooperate agents will have either switched to a different strategy, or have had no Always Defect agents adjacent to them to begin with. This then allows the Always Cooperate game strategy to outperform the Tit for Tat game strategy, as it is no longer being weighed down by its poor score against Always Defect, and, unlike Tit for Tat, is able to 'convert' Suspicious Tit for Tat agents to mutual cooperation, rather than the endless eye-for-an-eye cycle of the Tit for Tat and Suspicious Tit for Tat strategies. This means low-scoring Suspicious Tit for Tat are more likely to swap to an Always Cooperate strategy than a Tit for Tat strategy, meaning the Always Cooperate strategy gains agents again.

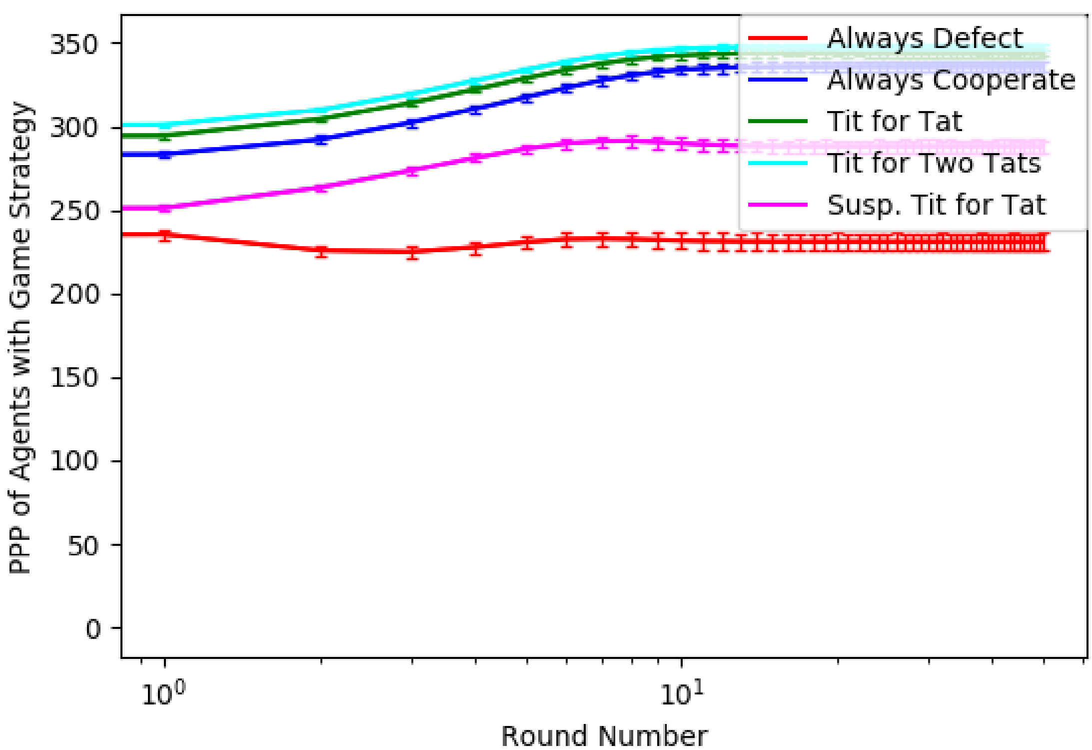


Figure 14: This graph shows the average PPP for the 200 simulations of the Interrogate update strategy. Note the order of which game strategy has a high PPP is the same as which game strategy has a high population from figure 13.

This hypothesis was tested by recording the total score across the entire network for each game strategy after each round of the simulation, then dividing the total score by the population of each game strategy to obtain the average score for each game strategy over the course of the simulation as shown in figure 14. In terms of how this quantity is analogous to real world concerns, it is similar to measuring the average salary for a set of job types, or the Produce per Person (PPP) of a state.

While we do not see the mean score for the Always Cooperate ever going above that of the Tit for Tat game strategy, the fact the average score increases is consistent with the hypothesis of Always Cooperate and Always Defect agents becoming disconnected over the course of the simulation. This is further supported by the fall in mean score for the Always Defect strategy after the first round – agents using the strategy are no longer able to pick up large scores from Always Cooperate agents. The average score per edge for an Always Cooperate agent after round 2 is 37.6 ± 0.3, or essentially that on average each Always Cooperate agent is next to one Always Defect agent, and 7 others (the other game strategies all score approximately the same against Always Cooperate – between 42 and 45, as per table 3). Therefore, the Always Cooperate agents with zero adjacent Always Defect agents will have a score of approximately 43.0 ± 1.8 – higher than the average Tit for Tat score at this point in the simulation of 38.7 ± 0.7, and therefore consistent with the hypothesis.

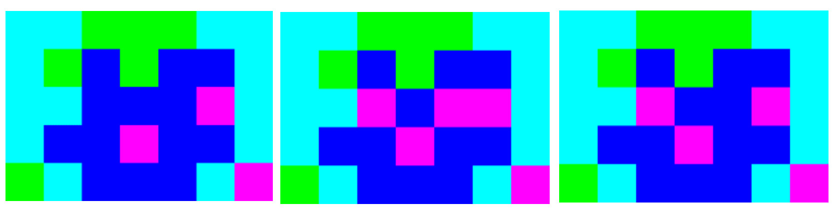


Figure 15: These images show a length three cycle that occurs under the Interrogate update strategy. Note the global populations are affected by the cycle.

Even under the Interrogate update strategy, the equilibrium state is not static, though the equilibrium cycles are of a different kind to the Threshold cycles. Under the Threshold update strategy, we found cycles occurred when two neighbouring agents would both switch to the strategy of the other as in figure 8, even though the two strategies performed equally well in both cases. In contrast, under the Interrogate update strategy we found both longer cycles, as shown in figure 15, and cycles where only one agent changed its strategy, as shown in figure 16. These cycles are not just common, but inevitable under the Interrogate update strategy. This is because the Interrogate update strategy forces an agent to update its strategy if there is no neighbour with a lower score than it. Effectively, if you consider the score of each agent in the network to form a surface over the network, then updates occur at every local minimum of the surface. As both the network and the possible scores are finite, local minima must always exist, and so in the equilibrium state at least one agent must be updating every round.

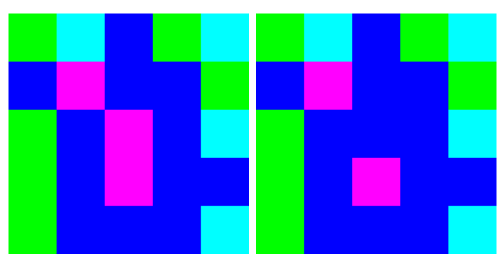


Figure 16: These images show a length two cycle that occurs under the Interrogate update strategy, where only one agent changes its strategy.

However, in many cases the agent will just update to the same strategy it had before, creating a 'cycle' of length one. This happens when an agent is at a local minima, but the highest scoring agent it can see has the same game strategy, for example in the arrangement shown in figure 17.

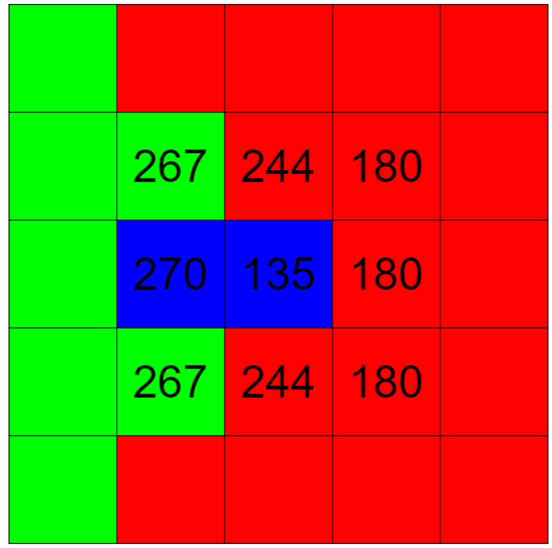


Figure 17: This image shows a hypothetical length one cycle that could occur under the Interrogate update strategy. The central Always Cooperate agent is at a local minimum in score, but its highest scoring neighbour is also an Always Cooperate agent, so no strategy changes occur.

5. Conclusions

The simulations described in this report show that the amount of information available to an agent has huge effects on the ability of the agent to make choices of strategy that are both good for the agent itself, and good for their immediate environment.

We found that under the Threshold update strategy, the purest extension of the iterated prisoner's dilemma to a network investigated, the Always Defect strategy rapidly dominated the network,

We found that allowing agents to decide their strategy based on the total score of their neighbours rather than just the result of their own games led to an increase in the average score of each agent from 142.7 ± 4.3 to 331.2 ± 11.5, an increase of factor of 2.3 in average score. This was caused by 'nice' strategies outperforming the 'nasty' strategies when agents were able to see their neighbours' total scores.

Further work is necessary to investigate how more complex strategies behave in such a network, and how other update strategies affect the equilibrium state of the network.

References

[1] "THE SECURITY DILEMMA IN ALLIANCE POLITICS", Glenn H. Snyder (July 1994). World Politics, Vol. 36, No. 4

[2] "Equilibrium points in n-person games", John Nash (1950). Proceedings of the National Academy of Sciences 36(1):48-49

[3] "Five rules for the evolution of cooperation", Martin A. Nowak (Feb 2012). https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3279745/

[4] "The Evolution of Cooperation" Robert Axelrod, William D. Hamilton (1981). Science. 1981;211:1390 http://science.sciencemag.org/content/sci/211/4489/1390.full.pdf

[5] “Short history of iterated prisoner’s dilemma tournaments” ARTEM KAZNATCHEEV (2015) https://egtheory.wordpress.com/2015/03/02/ipd/

[6] “THE POWER OF ASKING: HOW COMMUNICATION AFFECTS SELFISHNESS, EMPATHY, AND ALTRUISM” James Andreoni, Justin M. Rao (Sep 2010). NBER No. 16373 http://www.nber.org/papers/w16373.pdf