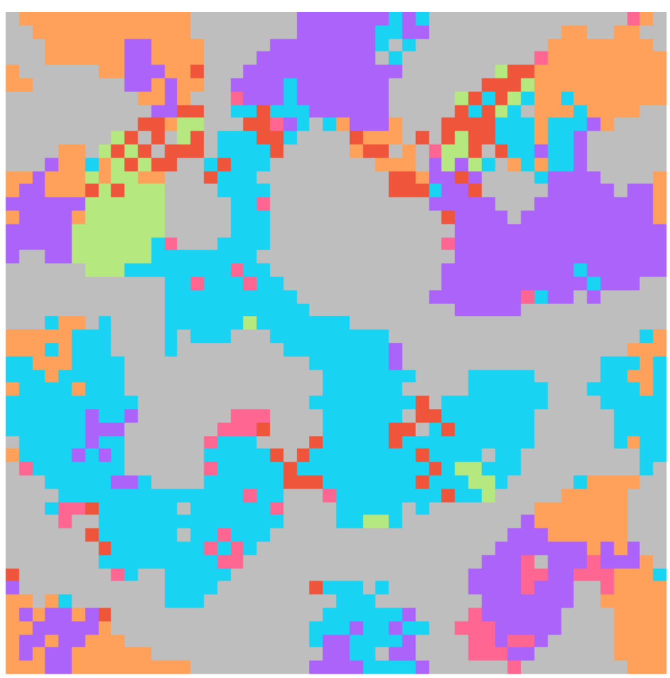
Investigations of distributional shifts and trends in the repeated and infinite spatial prisoner's dilemma

Robin Staab[1], Kim Nik Baumgartner[2], Fabrice Egger[3], Jan Urech[4], and Nando Käslin[5]

[1]staabr@student.ethz.ch, ETH Zurich
[2]kimbau@student.ethz.ch, ETH Zurich
[3]fegger@student.ethz.ch, ETH Zurich
[4]jurech@student.ethz.ch, ETH Zurich
[5]kaeslinn@student.ethz.ch, ETH Zurich

ABSTRACT

In this report, we conduct several experiments using multiple strategies in both the repeated and infinite spatial prisoners dilemma game. We investigate and identify underlying trends in the repeated spatial prisoner's dilemma when playing under various conditions. Afterward, we turn to the Infinity spatial prisoner's dilemma in which players play an infinite amount of games in each round of the simulation. We observe three major trends depending on the choice of ω and derive an upper bound on ω such that cooperating strategies can dominate. All our code, as well as experiments, is available under https://github.com/RobinStaab/CGT21_pd_simulation.



1 Introduction and Related Work

In this chapter, we first introduce the standard prisoner's dilemma. Then, we discuss the repeated prisoner's dilemma, present common strategies, and some related work on the topic. In the last part, we introduce the spatial prisoner's dilemma and highlight relevant findings in the literature.

Prisoner's Dilemma

The traditional prisoner's dilemma (PD) is a game played by two self-interested players. Each player has the option to either cooperate (C) or to defect (D). If both players cooperate, they both get a payoff of R each. If only one player cooperates and the other defects, the defecting player gets rewarded with the biggest payoff of T while the cooperating player gets punished with the lowest payoff the game has to offer: S. If both players defect, they both get a payoff of P. As implicitly assumed above, the payoffs are generally required in the following fashion: $T > R > P > S$.

	C	D
C	R,R	S, T
D	T, S	P,P

Figure 1. General Payout Matrix for the *prisoner's dilemma*. Important criteria are that $T > R > P > S$ as well as $2 \cdot R > T + S$.

Looking at the payout matrix of the game, one can easily derive that the dominant strategy for both players is to always play D. Regardless of what the opponent plays, defecting always leads to a better outcome compared to the payoff for playing C ($P > S$ and $T > R$). Hence, the actions DD (defection by both players) constitute the game's only Nash equilibrium. The dilemma lies in the fact that this Nash equilibrium is not Pareto optimal, meaning that both players could earn a higher payoff by playing CC instead of DD ($R > P$).

The PD is of particular interest as it abstracts situations that are commonly found in human civilizations, e.g., climate change[1] or war.[2] In general, the prisoner's dilemma and its variations have been a well-studied topic in the literature.[3–6] A topic that is often of central interest is the emergence and stability of cooperation in a variety of different settings.[7]

Repeated (Iterated) Prisoner's Dilemma

The repeated (iterated) prisoner's dilemma (RPD) is a repeated game with the individual stage games being classical PD's as outlined above. Players aim to maximize utility over a (unknown) amount of repetitions of the game. The players have access to the past actions of the opponents and can form strategies based on this information. Pioneering work on this game was done by Robert Axelrod in.[8] He asked scientists from various fields to submit strategies for this iterated game. The strategies were then tested against each other in a tournament format to see which strategies performed the best in a pool of different strategies. As opposed to the one-stage PD, in the iterated game, the strategy space is much more complex, and multiple

Nash equilibria strategies exist (as outlined below). Several interesting and well-known strategies will be presented here, including some additional information on them.

- **RANDOM**: In each iteration, either C or D is chosen at random with a fixed probability. It is **not** a Nash equilibrium strategy of the RPD.
- **COOPERATE**: This strategy plays C in each iteration unconditionally. It is **not** a Nash equilibrium strategy of the RPD.
- **DEFECT**: This strategy plays D in each iteration unconditionally. It is a Nash equilibrium strategy of the RPD.
- **TFT** (tit for tat): This strategy starts with C and, in the subsequent rounds, copies the opponent's last move. It is a Nash equilibrium strategy of the RPD.
- **GT** (grim trigger): This strategy starts with C and continues to cooperate as long as the opponent is cooperating. If the opponent defects in any round, **GT** will never forgive and always play D from this point on. It is a Nash equilibrium strategy of the RPD.
- **TFTD** (tit for tat defect): This strategy starts with D and, in the subsequent rounds, copies the opponent's last move. It is **not** a Nash equilibrium strategy of the RPD.
- **TF2T** (tit for two tats): This strategy starts with C in rounds 1 and 2 and plays D only if the opponent played D the last two moves. TF2T is **not** a Nash equilibrium strategy of the RPD.

One meaningful way to categorize strategies in the RPD is by the two categories of *friendly* and *unfriendly*. Friendly strategies are characterized by never defecting first. In particular, this has the effect that two friendly strategies playing against each other will always cooperate throughout the entire game. **COOPERATE**, **TFT**, **GT** and **TF2T** are friendly strategies, whereas **RANDOM**, **DEFECT** and **TFTD** are unfriendly strategies, i.e. they can defect first.

The results from [8] showed that friendly, provocable (meaning that D is returned for D and C is returned for C), and non-envious (meaning it focuses on maximizing its own score and not on scoring higher than its opponent) strategies perform the best. **TFT** is the simplest example for a strategy that has all of these characteristics, which also serves as an explanation why it yielded the best results in the tournament.

Additionally, to evaluating strategies based on performance,[9] also evaluates the strategies based on four different, clearly defined morality measures: the overall cooperation rate, the good-partner rating, the eigenjesus rating, and the eigenmoses rating. The overall cooperation rate is given by the fraction of the number of times the player played C divided by the total games. The good-partner rating is the fraction of encounters in which the player cooperated at least as much as its opponent. Finally, the eigenjesus and eigenmoses ratings are recursively defined and give higher weight to cooperation with other players that themselves cooperate more.

Infinite Prisoner's Dilemma

A natural extension to the repeated prisoner's dilemma is the infinite prisoner's dilemma (IPD). Unlike in the RPD, the IPD lets players play for an infinite amount of rounds. As the sum of payoffs in such a setting does generally not converge, one introduces a "discounted future reward function," often referred to as "shadow of the future." It generally takes the form of a parameter $\omega \in (0,1)$ such that the reward k steps into the future is discounted by ω^k. We will give a more thorough introduction to the explicit calculations in section 4.

In [10], the effects of the shadow of the future on the IPD are investigated. The author shows that cooperation increases with a greater probability of further interaction in the RPD, and players cooperate more in the infinite IPD.

Spatial Prisoner's Dilemma

A further extension to the RPD is the spatial prisoner's dilemma (SPD). It is commonly played on a two-dimensional, discrete playing field that hosts players with different strategies (however, extensions to general graph structures have been proposed, e.g., in[11,12]. In each iteration (epoch), the players then play one round of the prisoner's dilemma against their neighbors.

In [13] a simplified setup is considered where the players either always defect or always cooperate and play against their immediate neighbors, i.e., those with Manhattan distance 1. In the next round, a field is occupied by the player who scored the highest in the neighborhood of the field. This simple, deterministic version is sufficient to generate beautifully complex and chaotic patterns that bear some resemblance with Conway's game of life.[14]

Today, most works in this area either look at evolutionary strategies or the influence of imitation and migration, with our work being part of the latter. Evolutionary strategies generally evolve via the exchange of "genes" or through their mutation. Strategies are usually ranked according to some fitness metric, and offspring is produced through a survival-of-the-fittest selection procedure.[15] Depending on the gene-to-strategy mapping, this can lead to the natural evolution of well-performing strategies. Migration and imitation, on the other hand, allow players to adapt directly by copying the behavior of more successful players around them. Additionally, as shown in [16], success-driven migration, i.e., migration to a free location that promises better reward, can lead to cooperation in scenarios in which pure imitation would not.

Instead of an SPD in which all players play against their neighbors pair-wise, the concept can be extended to an n-player PD played by all players in a certain region.[17–19] In [17], the effect of player movement is investigated. They found that allowing for relocation increases the level of cooperation, with adjacent movement being the best condition for establishing regions of cooperation. The effects that the scale of interaction and reproduction have on cooperative behavior in the SPD is examined in[18]. Results show that an increase in the scale of reproduction leads to a higher level of cooperation and that there exists a value for the scale of interaction for which a peak in cooperation is achieved.

The strategy space for the SPD is extended in [20] from temporal strategies to spatial strategies that make decisions based on the configuration of actions of all neighbors. The spatial strategies were designed based on generosity and contrariness, and some of the proposed spatial strategies outperformed many previous strategies. Boundary formation and rejection of alien strategies could also be observed.

The ability of players to change their strategies throughout the game based on the neighboring strategies as well as the length of the player's memories about past actions can also have a great effect on the emergence of cooperative regions.[21,22]

Nomenclature

In the literature, one often finds IPD as the abbreviation of "Iterated Prisoner's Dilemma," which in our case refers to the "Repeated Prisoner's Dilemma." As we only deal with the spatial version of the game in this work, we will, going forward, refer to the spatial repeated (iterated) prisoner's dilemma as RPD and to the spatial infinite prisoner's dilemma as IPD.

2 Methods & Implementation

We built a general implementation of a simulation of both the RPD (repeated spatial prisoner's dilemma) and the IPD (infinite spatial prisoner's dilemma) in Python. We refer to table 2 for a complete overview of all hyperparameters used to define a specific run of the simulation. Each instance of the simulation uses a fixed rectangular grid with a fixed number of players. In all experiments, the grid wraps around the borders, allowing players to have the same maximum amount of neighbors independent of their location. The simulation runs in epochs in which each player plays a game against all his neighbors.

Parameter	Description
T, R, P, S	Payoff matrix
runs	Number of runs t
epochs	Epochs that each experiment run will simulate
snapshots	List of epochs at which to save an image of the grid
grid_x, grid_y	Dimensions of the grid
infinite	Switches experiment from RPD to IPD
<player classes>	Player classes in the experiment

Figure 2. Experiment parameters

In the RPD, a player's decision to defect or cooperate in one epoch is determined by his current strategy. These strategies can use the history of all previous games the player had against the current opponent to influence their decisions. Histories persist on a player-by-player basis, i.e., even when the player moves on the grid or, more importantly, changes his strategy. The player thus remembers if another player defected in a previous encounter even if they used other strategies and they met in another part of the grid. Note that the concept of a history becomes unclear in the case of the IPD. As players already play an infinite amount of games at every encounter, there is no inherent notion of temporal order, and players (in our simulation) do not keep histories between encounters. After every player made a decision for the current epoch, the players calculate their reward based on the payoff matrix in the RPD case or based on explicit expected reward formulas in the IPD case. This payoff is visible to other players. Thus, in each epoch, the players have a fixed (fixed per player, but in most experiments, we take a fixed value overall players) chance to imitate the strategy of their most successful neighbor. Afterward, players also have a fixed probability of migrating to a free spot within their migration window. Players may only migrate to a currently free spot and only do so if it provides a higher expected payoff. In case multiple spots lead to the same payoff, they choose the first one they found. A full overview of the algorithm is given in 1

Algorithm 1 SPD Simulation

```
procedure SIMULATE(epochs, playerParams, X, Y)
    grid ← initializeGrid(X, Y, players)
    players, matchups ← initializePlayersAndGrid(playerParams, grid)
    history ← ∅
    for i ← 1, epochs do                                          ▷ Simulate a single epoch
        history ← updateHistory(history, players, grid, i)
        for (p1, p2) ∈ matchups do                                ▷ Simulate all matchups
            rewards, decisions ← play(p_1, p_2)
            p_1, p_2 ← updatePlayers(p_1, p_2, rewards, decisions)
        for p ∈ players do                                        ▷ Imitation
            if rand(0,1) < p.imitationProb then
                bestStrategy, bestUtil ← p.strategy, p.latestPayoff
                for p' ∈ p.playNeighborhood do                    ▷ Local play neighborhood
                    if p'.latestPayoff > bestUtil then
                        bestUtil ← p'.latestPayoff
                        bestStrategy ← p'.strategy
                p.strategy ← bestStrategy
        for p ∈ players do                                        ▷ Migration
            if rand(0,1) < p.migrationProb then
                bestLoc, bestUtil ← p.loc, p.latestPayoff
                for loc ∈ p.migrationNeighborhood do              ▷ Local migration neighborhood
                    // loc is a free location
                    if expectedPayoff(p, loc) > bestUtil then
                        bestUtil ← expectedPayoff(p, loc)
                        bestLoc ← loc
                if bestLoc ≠ p.loc then
                    migrate(p, bestLoc, grid, player)             ▷ Updates matchups and neighborhoods
```

Each experiment is defined by a set of classes of players. Each class hereby uses the same starting parameters: strategy, number of players, migration, imitation probability, and ω for IPD. The results of an experiment are generally analyzed independently over the defined classes. The simulator gathers all relevant data during its execution to provide real-time access to statistics.

Parameter	Description
nr	Nr of players in this class
strat	Strategy used for games
imit_prob	Imitation probability
migrate_prob	Migration probability
omega	Shadow of the future used in the IPD

Figure 3. Player class parameters

The simulator can be used in one of two ways:

- In a web dashboard that allows quick adjustments of strategies, hyperparameters, and live updates on all statistics and the board state.
- A console-based application that enables a user to run multiple thousands of experiments without supervision and receive aggregated results.

This dashboard allowed for fast prototyping and enabled a reactive design process. It also allows users to get up to speed with the simulator quickly and get an intuition for its behavior and results.

<table>
<tr><th>T</th><th>R</th><th>P</th><th>S</th><th>runs</th><th>epochs</th><th>snapshots</th><th>grid_x</th><th>grid_y</th><th>infinite</th></tr>
<tr><td>3</td><td>2</td><td>1</td><td>.5</td><td>5</td><td>100</td><td>[25,50,75,100]</td><td>10</td><td>10</td><td>False</td></tr>
<tr><td>nr</td><td colspan="2">strat</td><td colspan="2">init_prob</td><td colspan="2">migrate_prob</td><td>omega</td><td colspan="2"></td></tr>
<tr><td>10:20</td><td colspan="2">RANDOM</td><td colspan="2">.1</td><td colspan="2">.1;.5;.1</td><td>.5</td><td colspan="2"></td></tr>
<tr><td>10</td><td colspan="2">DEFECT</td><td colspan="2">.1</td><td colspan="2">.1</td><td>.5</td><td colspan="2"></td></tr>
<tr><td>10</td><td colspan="2">TF2T</td><td colspan="2">.1</td><td colspan="2">.1</td><td>.5</td><td colspan="2"></td></tr>
<tr><td>10</td><td colspan="2">GT</td><td colspan="2">.1</td><td colspan="2">.1</td><td>.5</td><td colspan="2"></td></tr>
</table>

Figure 4. Example experiment configuration that contains both a set of values (10:20) and a range definition (.1;.5;.1)

In the console-based application, one can define a set of experiments in a configuration file. This file supports both value ranges as well as sets to allow the creation of many similar experiments without the need of copying parameters multiple times. In addition, the results of each experiment are saved to files after it has finished. Using these configurations can be helpful when running multiple and extensive experiments since after their launch, the user does not need to interact with the app during or in between experiments.

Testing the simulation

To ensure that our simulation works as expected we ran multiple test experiments. These experiments contain simple parameters that result in predictable results. After running these experiments we compared the results with our expectations. Since our expectations were met we conclude that our simulation works as intended.

3 Results and Discussion

Using this SPD simulation implementation, two groups of experiments were performed. We define a strategy to be "spatially dominating" when the highest percentage of players plays this strategy at the end of the simulation (e.g., after 400 epochs). We extend this definition to a set of strategies S with the additional condition that the strategies are mutually in a stable equilibrium (e.g., $\{TFT, TF2T, GT\}$ which are mutually cooperative amongst each other). First, we investigate the emergence of a dominating strategy or dominating set of strategies in the setting where an equal number of players starts for each strategy. Afterward, we reduce the starting player count of said dominating strategies to investigate possible follow-up dominating strategies, which are kept back in the base case. During this experiment, the dominance of *Tit-for-Tat* was established, and when the amount of TFT-starting players was reduced, surprisingly *Grimm Trigger* emerged as a potentially dominating strategy. Even more surprising was an emerging Nash-equilibrium which consisted of GT and TFT or a TFT-variant ($TF2T$, $TFTD$) and equalized at around a **20%** defection-rate and at around **90%** of the optimal payout (social optimum). Furthermore, the impact of imitation probabilities in highly $DEFECT$-centered environments was examined. We found that at imitation rates of up to 70% for strategies other than TFT, TFT remains dominant even with a very small starting population. We observed maximum TFT dominance at 50% with it lowering afterward. At even higher imitation rates of 80-90%, $TFTD$ became the most successful strategy.

Dominating Strategy and emerging Equilibrium

The base case

For easy comparison of the experiments, we first constructed a base case. It consists of a constant amount of players (350), which are evenly distributed into the seven different strategies playing on a 20×20 grid. Each player is placed at a random location on the grid to start. The base case experiment is repeated 10 times and averages, median, and deviation are reported. The payout matrix was chosen to be:

	C	D
C	20,20	1, 30
D	30, 1	10,10

The migration as well as the imitation probability are set to 10%.

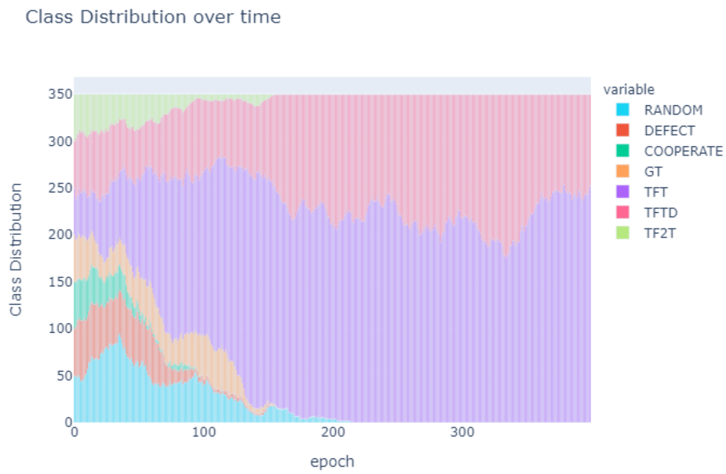


Figure 5. Sample base case. Strategy distribution over time. Total number of players playing each strategy in the epoch (round).

This distribution shows that significantly more players choose to play *Tit-for-Tat* over the duration of the games (starting at 50 and going up to 252, which constitutes $\mathbf{72\%}$ of the player-base. Over the 10 replications of the experiment, the median value for TFT-playercount was 219 ($\mathbf{62\%}$). With a 95% confidence interval mean of 232±42 ($\mathbf{66\%\pm12\%}$) *Suspicious Tit-For-Tat* ($TFTD$) was also fairly successful, which is not surprising as after the first round against new opponent they play just like TFT. (95% confidence interval mean of 117±29 ($\mathbf{33\%\pm8\%}$)) (The interaction of a $TFTD$ player and a TFT player can lead to bad cycles where they exploit one another repeatedly). Additionally, we tracked the defection rate of the different strategies and the percentage of optimum (social optimum).

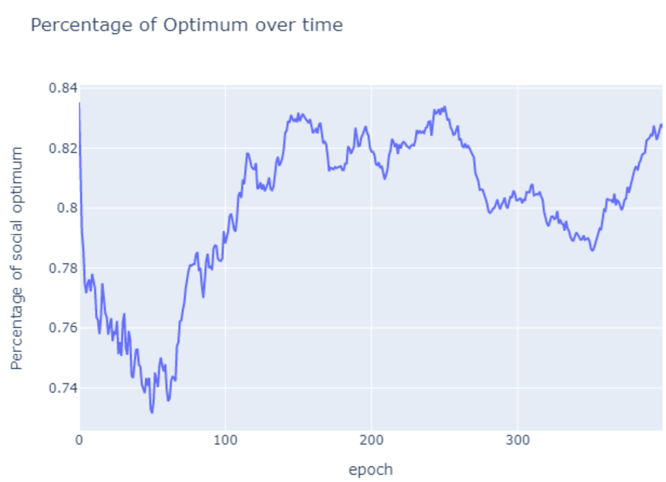


Figure 7. Sample base case. Percentage of optimum over time. Shows how close the total payoff achieved in a given round is to the social optimum (every player plays cooperate against everybody in his vicinity).

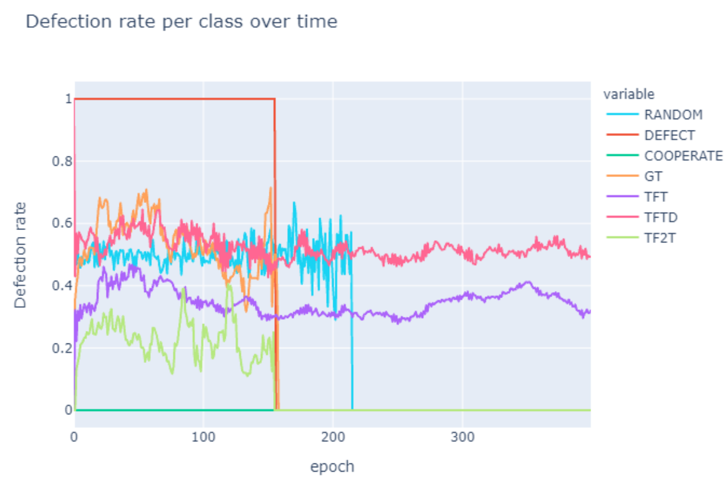


Figure 6. Sample base case. We show the defection rate per strategy over time. The x-axis is over the simulated epochs. The y-axis shows the percentage of players of each strategy that defect in a given epoch (with 1 being total defection). A sudden drop to 0 indicates that the strategy died out in that round.

6 shows where the different strategies die out (defection suddenly drops to 0 except for $COOPERATE$, which already starts at 0 defection rate), but it also shows the critical flaw of TFT. Even when all players play TFT at high epochs, the defection rate will never be 0 as TFT is a *non-forgiving* strategy. Even if a higher payout could be achieved by only playing C, TFT-players who were once "betrayed" by a neighboring TFT or TFT-variant player will forever alternately play D as retribution for the round before. The near-linear rise of percentage in 7 results from the imitation of more successful strategies at a constant rate of 10%. As the dominant strategy gains more players playing it, the payout is closer to the optimum payout.

Reducing TFT starting player-count to offset domination

In this experiment, the only change to the base case was that the number of players starting with TFT as their strategy was reduced incrementally. All the other hyperparameters, such as the payout matrix 3 were left the same. Reducing the starting player-count of the TFT-strategy, a lower limit of starting players necessary for TFT to become the dominating strategy was determined. Running the experiments with 20 (6.25%) players starting with TFT, the strategy was still able to dominate 50% of the time. Reducing this even further to 10 players, TFT only dominated 25% of simulations run. At 8 players (2.3%), the strategy died out as often as it dominates (20% of cases). Dying out could hypothetically be attributed to "bad luck" when spawning in the grid, e.g., next to a great number of $RANDOM$- or $DEFECT$-players, where switching to a different strategy would increase the short-term payout, with a small starting-player base the probability of this would logically increase as there are not enough players to balance out "unlucky" spawns. We determined 7.14% (25 starting players) as the threshold for TFT to dominate reliably (80%) in an otherwise equal game.

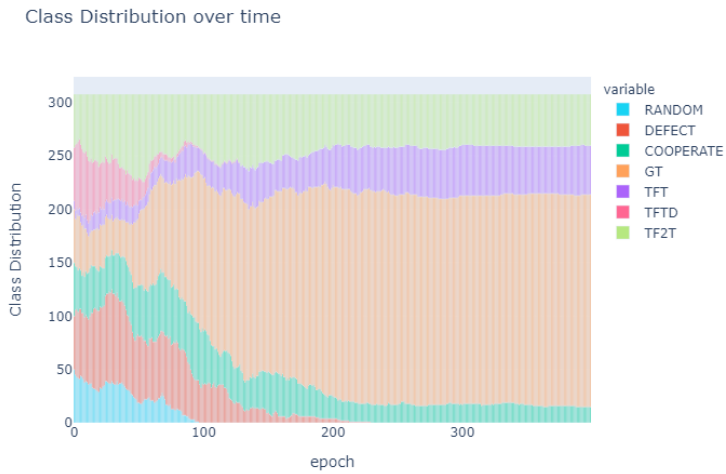


Figure 8. Strategy distribution over time, with a starting population of 10 TFT-players and 50 of each other strategy.

One of the most exciting developments found in these experiments, however, was the emergence of *Grimm Trigger* as a follow-up dominating strategy instead of the (expected) other TFT-Variants like $TF2T$ and $TFTD$.

Equilibria, with *Grim Trigger* overtaking the other strategies, such as the one seen in 12 frequently appeared with these parameters. Additionally, in 80% of cases, players playing *Grim Trigger* and one of the TFT-Variants fell into a Nash equilibrium where they have nothing to gain by imitating other players or migrating to another spot on the grid.

Figure 9 shows that this equilibrium not only occurs sooner than in the base case, the percentage of optimum is also on average 7% higher! This means the variant with fewer TFT-players is closer to the social optimum than the base case, even when no non-friendly strategy was adjusted.

Impact of imitation-probability in a *DEFECT*-dense playspace

The impact of the migration-probability on the base case had expected results: the game was faster in a equilibrium (see Appendix B). For investigating the impact of a higher imitation probability a $DEFECT$-dense playspace was chosen because $DEFECT$ is the weakest non-friendly strategy which means it will create more "chaos" in early rounds.

In this experiment a 20×20 grid was chosen and filled with the following amounts of players starting with each strategy: $RANDOM$: 20 (8%), $DEFECT$: 100 (40%), $COOPERATE$: 70 (28%), GT: 20 (8%), TFT: 5 (2%), $TF2T$: 20 (8%), $TFTD$: 20 (8%). For a total of 255 players. With this a highly $DEFECT$-dense grid was created which should theoretically lower the percentage of optimum based on the lower amount of cooperation. The migration probability was set to 10%, as was the imitation probability at the beginning. The payout matrix was as follows:

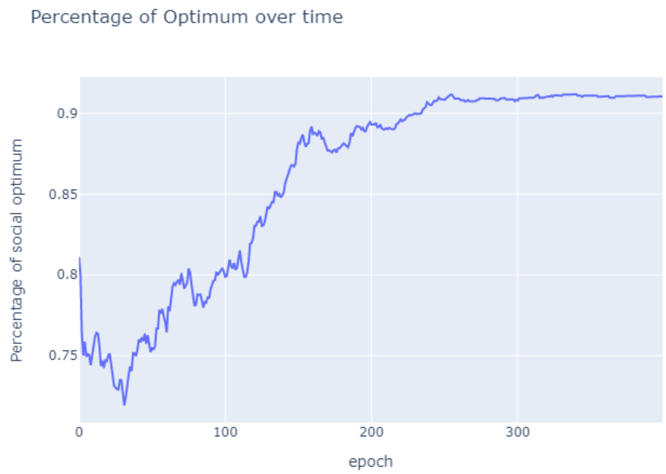
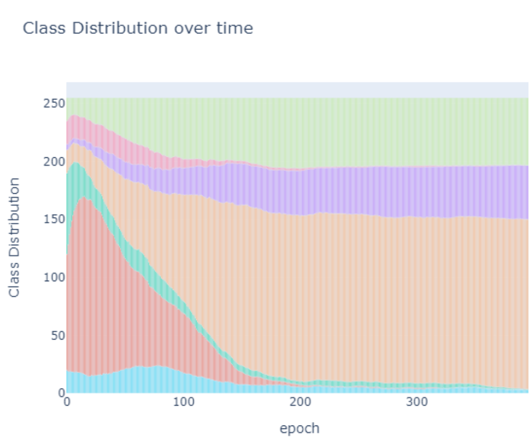


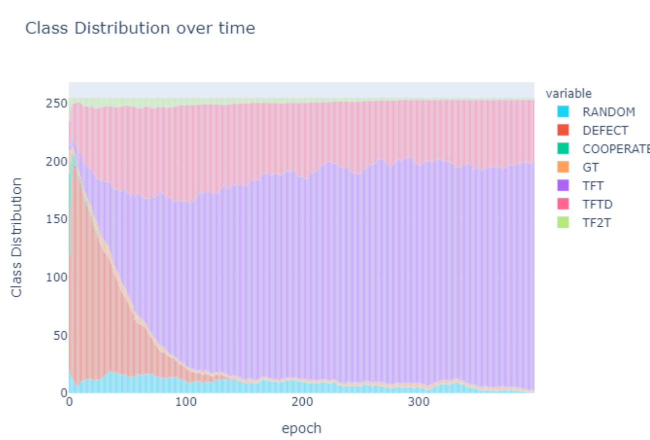
Figure 9. Percentage of Optimum over time, with a starting population of 10 TFT-players and 50 of each other strategy.

	C	D
C	20,20	1, 30
D	30, 1	10,10

Then the imitation probability for every strategy except TFT (as the most successful strategy in earlier experiments) was continuously raised from the starting 10% up to 90%. The hypothesis for this experiment was, that with higher imitation probabilities TFT would, despite its 2% starting player base, come to be the most played strategy again, as a switch to TFT is the most optimal move for each player looking to imitate a more successful one. In the range from 10% to 70% our hypothesis turned out to be correct. TFT became the dominating strategy at 30% imitation and its player number grew until a maximum was achieved at 50% (see 10a, 10b).



(a) Imitation-probability 10%.



(b) Imitation-probability 50%.

At very high imitation rates of over 70%, $TFTD$ became the most successful strategy. This can be attributed to the high early game payoff $TFTD$ has in our (predominantly friendly) grid. At very high imitation-rates the early rounds decide which strategy becomes the most prominent as almost all players, change strategy in the first 50 epochs. (Except the ones playing optimal strategies from the beginning). In later epochs there is no advantage to switching from $TFTD$ to TFT as it plays the same after the first game against each player in its vicinity. This effect can be seen in a base-case with high imitation but the transition from low to medium to high rates is illustrated better in a "unfriendly" $DEFECT$-dense grid (see Appendix B).

(a) Imitation-probability 80%.

(b) Imitation-probability 90%.

4 Infinite Spatial Prisoners Dilemma with success-driven migration and imitation (IPD)

In the real world, there is no certainty in the number of encounters between two actors. While for some constellations, the number of meetings follows more predictable patterns, it is never deterministic. Hence the assumption of a fixed and finite number of meetings can be seen as a simplification that trades realism with computational feasibility. As shown in Section, this simplification also holds for the Repeated Prisoners Dilemma. Experiments in [23] show that real players start to change their behavior as soon as they know that they will have a limited number of encounters – generally favoring defection (a dominant strategy) in the last round. The strategies discussed in this paper are not aware of the number of rounds that they play with each other (both in total and per epoch). More sophisticated strategies, however, could exploit such repeated patterns to their advantage.

An additional disadvantage of a finite number of repetitions is that they struggle to capture long-term dependencies between actions and rewards. In reality, people tend to be more sceptical when they encounter someone new but will start to develop trust after multiple interactions with one another. Strategies that build up trust initially are at a disadvantage until their reputation[24],[25] has built up, and they can be successful. If a simulation does not take this into account, it may lead to a misevaluation of the corresponding strategy.

Arguments in Favor of the IPD

This opens up whether the Infinite Prisoner's Dilemma can address the problem of long-term interactions and dependencies. Naturally, parties will not interact an infinite number of times with each other, but using the concept of "the shadow of the future," i.e., a discounted future reward function we can achieve an explicit trade-off between the concept of infinite iterations and finite payoff. After each round of playing the Prisoners Dilemma, there is a chance $0 < \omega < 1$ to go on playing another round of the game. This ω corresponds to the probability that two people who interacted in the real world would meet each other again at a later point in time. While it is therefore technically possible to play infinitely many times against the same opponent, such an event happens with exponentially vanishing probability. This is again an abstraction since such an ω would be specific to each person as well as depend on the shared history of interactions. Our model simplifies this assumption by fixing the omega on a player basis (i.e., not for each ordered pair of players) independent of previous outcomes. However, we argue that this can already accurately capture different long-term planning behaviors associated with a heterogeneous set of (real-world) players.

Calculation of success

We calculate the success of the strategy as the expected reward against another strategy. Let S be the set of all possible strategies in the IPD. We define $u : S \times S \to \mathbb{R}$ as the reward function which maps a pair of strategies $(s_1, s_2) \in S^2$ to a real number corresponding to the expected reward of s_1 against the strategy s_2. In particular, one can write u as follows:

$$u(s_0, s_1) = \sum_{i=0}^{\infty} \omega^i \cdot r(s_{0_i}, s_{1_i})$$

Where s_{k_i} represents the decision between defection and cooperation of strategy s_k at the $i-th$ round. The function $r : \{D,C\}^2 \to \{T,R,S,P\}$ where $r(s_{0_i}, s_{1_i})$ stands for the reward of s_0 after its decision s_{0_i} against s_{1_i}. Naturally r and therefore also u depends on the values of T, R, P, and S. In the special case of the Prisoners Dilemma we assume that: $T > R > P > S$ and $2 \cdot R > T + S$. As simulating infinitely many repetitions for each encounter in the simulation is impossible, we explicitly calculated the expected reward of each strategy against each other strategy in advance (refer to Appendix A). For the treatment of spatial IPD we only focused on the following seven strategies: $RANDOM$, AD, AC, GT, TFT, $TFTD$,$TF2T$.

Experimental set-up

We choose 50 players of each strategy in our simulation on a 20×20 grid. Each strategy is placed at a random location on the grid to start. Choosing T = 30, R = 20, P = 10, and S = 1 we get the payoff matrix:

	C	D
C	20,20	1, 30
D	30, 1	10,10

In particular, there is always an incentive to play against other players due to the positive reward in all outcomes of the game.

This experiment simulates players who can migrate to the best free location up to three fields away with a migration probability of 10%. With a probability of 10% a player adopts the strategy of the most successful player directly around him. The focus of this experiment is how different values of $\omega \in \{0.2, 0.4, 0.6, 0.8\}$ influence the outcome of the simulation. Each of the seven strategies starts with the same imitation rate, migration rate, ω value, and player count. All simulation results are averaged over ten random runs for 400 epochs each.

Results

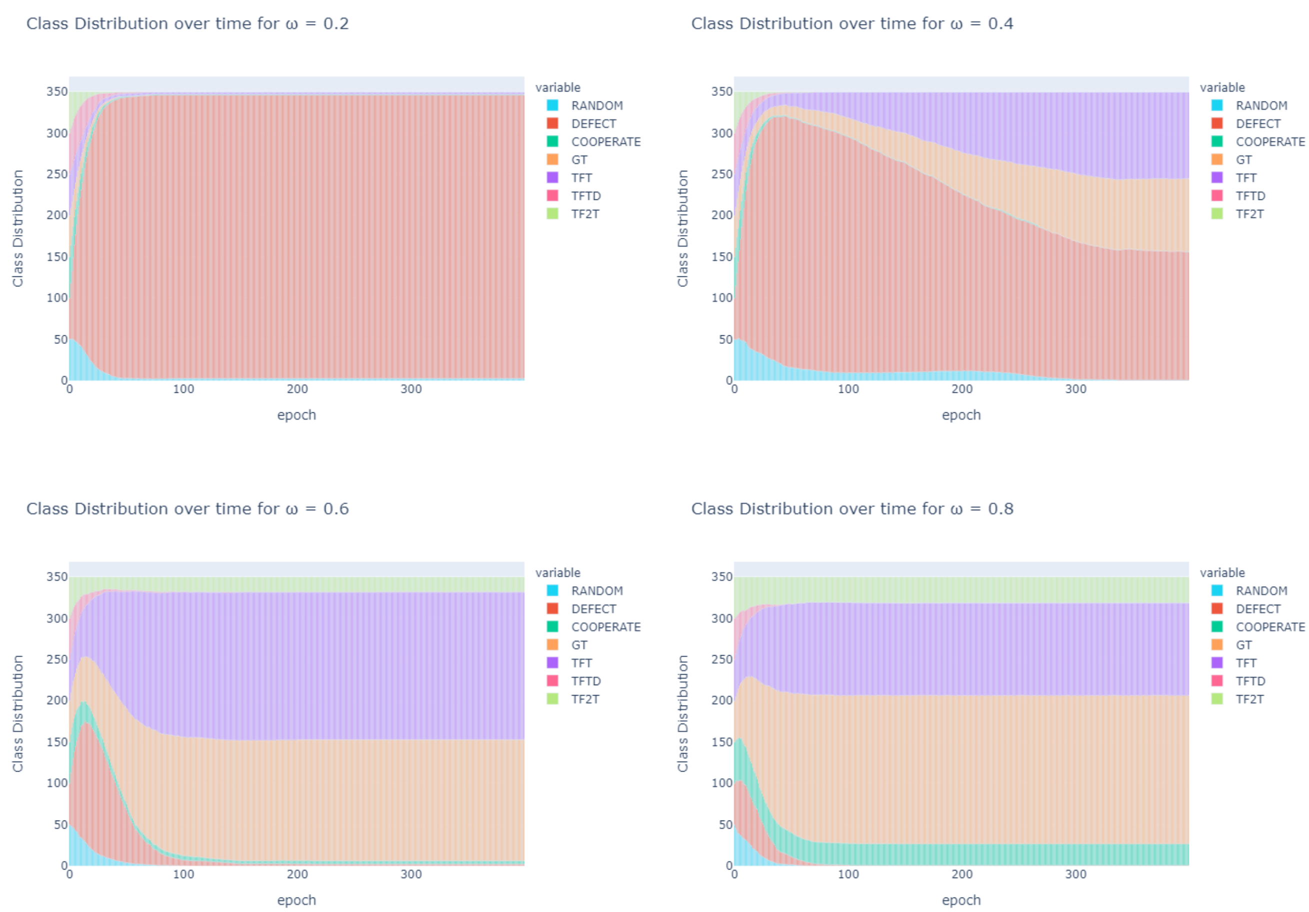


Figure 12. Strategy distribution over time for the infinitely repeated prisoners dilemma with different ω values of 0.2(top-left), 0.4(top-right), 0.6(bottom-left), 0.8(bottom-right)

The graphs in figure 12 clearly show how the different ω values lead to entirely different strategy distributions. One can see how smaller values of ω lead to DEFECT becoming an increasingly dominant strategy. Vice-versa, a higher ω favors friendlier strategies.

Both GT and TFT are favored if $\omega > 0.4$ while DEFECT is dominant for $\omega < 0.4$. Other unfriendly strategies such as RANDOM and TFTD are going extinct independent of the value of ω. Both TF2T and COOPERATE have a small chance of survival for a big ω.

Additionally, as highlighted in figure 13, we can observe a significant change in the "percentage of optimum" for the different ω values.

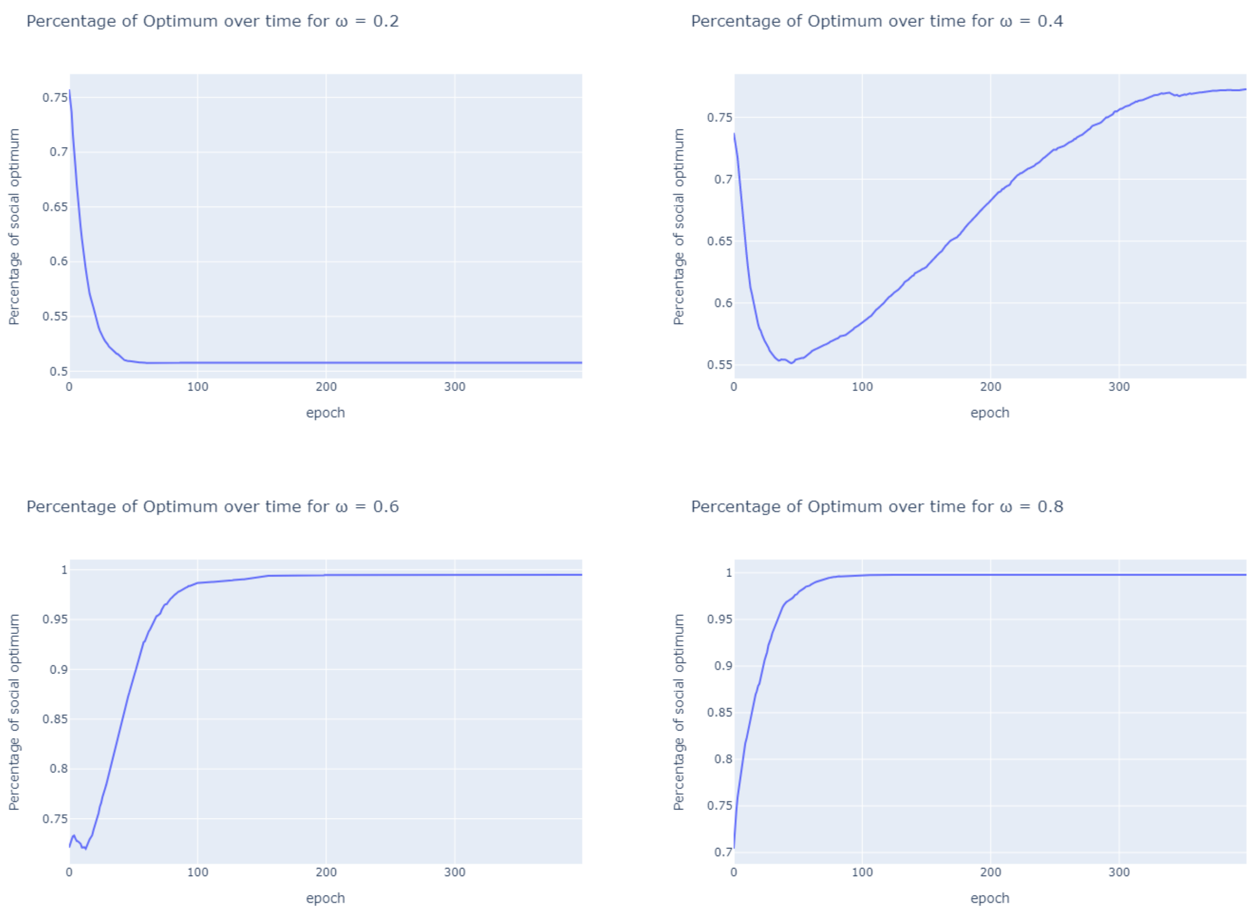


Figure 13. Percentage of optimum over time for the infinitely repeated prisoners dilemma with different ω values of 0.2(top-left), 0.4(top-right), 0.6(bottom-left), 0.8(bottom-right)

As expected, the optimum increases whenever the relative percentage of friendly strategies increases. Thus, as DEFECT emerges as the leading unfriendly strategy, the POO curve roughly matches the inverse of the DEFECT curve in figure 12. This confirms the initial assumption of an inverse correlation of the POO and the relative fraction of DEFECT players.

Interpretation

The result of the experiment showed three different types of behaviors for different ω values.

The first and most expected result is achieved whenever $\omega > 0.4$. Intuitively, the higher ω, the higher the probability for players to meet each other again in the future. Naturally, this is a strong incentive to adopt a friendly strategy for infinite cooperation. Furthermore, this leads to faster convergence to friendly strategies for $\omega \gg 0.4$.

If $\omega \approx 0.4$, we get a mixture of DEFECTION accounting for about 50% of the total players and TFT, and GT accounting for the other half. Let us call this point ω^*. We find that ω^* is not fixed to 0.4 but rather depends on the form of the payout matrix. However we can derive an upperbound on ω^* as follows: We know that DEFECT must have a better payout against the friendly strategies than the friendly strategies between each other. In particular the following relation must hold:

$$u(AD,TFT) = u(AD,GT) = T + \frac{\omega}{1-\omega} \cdot P > R \cdot \frac{1}{1-\omega} = u(TFT,GT) = u(GT,TFT) \tag{1}$$

$$\omega^* < \frac{T-R}{T-P} = \frac{1}{2} \tag{2}$$

A conflicting force is that friendly strategies always receive more reward against each other than DEFECT receives against DEFECT (which follows from our assumption of $R > P$), completely independent of the ω value. Therefore, the more DEFECT players there are, the higher the probability that they meet each other, and the lower the expected reward of the average DEFECT player. This is in line with our observation of ω^* being close to 0.5 (2), in our case around 0.4.

For $\omega < 0.4 \approx \omega^*$ DEFECT is the dominant strategy. A ω value that is low means that the players do not expect to play against each other again in future rounds. The lower ω gets, the more a single match up of the IPD behaves like an isolated instantiation of the prisoner's dilemma game. As seen in other papers, playing D in the never repeated prisoner's dilemma is a dominant strategy.[26]

Another prominent trend is that the strategies form clusters instead of spreading out. This follows from the fact that all rewards at the end of the game are positive, and hence a player's overall reward scales linearly with the number of its neighbors. An interesting future direction here could be investigating the clustering behavior under a payout matrix that contains negative entries.

Discussion

We find that the Infinite prisoner's dilemma does not automatically lead to cooperation. In particular, we have highlighted a strong dependence between the emergence of cooperation and the games payout matrix and ω value. A society that favors cooperation should therefore try to maximize the ω value. By generally expecting future encounters, cooperation can become a dominant strategy over defection. However, as soon as the ω falls below some critical value ω^* (2) where:

$$\omega \leq \omega^* < \frac{T-R}{T-P}$$

there will always be some defectors in society. Even in the idealized, infinitely repeated version of the prisoner's dilemma.

Future Work

During our experiments, we found several interesting future directions that one could try out with none or only slight modifications to our current simulator:

- **K-Step RPD**: Let players play k rounds of the PD before we allow imitation and migration (bridges to IPD but allows to keep history).
- **Threshold Imitation Migration**: Players only imitate or migrate when their new strategy / location is better than the current one by some threshold.
- **Switch Imitation and Migration Order**
- **Allow abstention**: Building on the work of Perc et al. in our setting.
- **Harsh environments**: Constantly decrease utility and introduce the concept of survival.
- **Common good**: Simulate a shared common good against one can always defect.
- **Extended strategy set**: Use additional strategies (e.g., use the history to a varying extent[27,28])

Appendices

A Explicit IPD Formulas

RANDOM with parameter p

vs. RANDOM with param p_2

$$
\begin{aligned}
& u(RAND_{p_1}, RAND_{p_2}) \\
& = p_1 \cdot p_2 \cdot R + p_1 \cdot (1-p_2) \cdot S + (1-p_1) \cdot p_2 \cdot T + (1-p_1)(1-p_2) \cdot P + \omega \cdot u(RAND_{p_1}, RAND_{p_2}) \\
& = \frac{p_1 \cdot p_2 \cdot R + p_1 \cdot (1-p_2) \cdot S + (1-p_1) \cdot p_2 \cdot T + (1-p_1)(1-p_2) \cdot P}{1-\omega}
\end{aligned}
$$

vs. AD

$$
\begin{aligned}
& u(RAND_p, AD) \\
& = p \cdot S + (1-p) \cdot P + \omega \cdot u(RAND_p, AD) \\
& = \frac{p \cdot S + (1-p) \cdot P}{1-\omega}
\end{aligned}
$$

vs. AC

$$
\begin{aligned}
& u(RAND_p, AC) \\
& = p \cdot R + (1-p) \cdot T + \omega \cdot u(RAND_p, AC) \\
& = \frac{p \cdot R + (1-p) \cdot T}{1-\omega}
\end{aligned}
$$

vs. GT

$$
\begin{aligned}
u(RAND_p, GT) &= p \cdot (R + \omega \cdot u(RAND_p, GT)) + (1-p) \cdot (T + \omega \cdot u(RAND_p, AD)) \\
&= p \cdot \omega \cdot u(RAND_p, GT) + p \cdot R + (1-p) \cdot (T + \omega \cdot \frac{p \cdot S + (1-p) \cdot P}{1-\omega}) \\
& + p \cdot \omega \cdot u(RAND_p, GT) + p \cdot R + (1-p) \cdot (T + \omega \cdot \frac{p \cdot S + (1-p) \cdot P}{1-\omega}) \\
&= \frac{p \cdot R + (1-p) \cdot (T + \omega \cdot \frac{p \cdot S + (1-p) \cdot P}{1-\omega})}{1 - p \cdot \omega}
\end{aligned}
$$

vs. TFT

Results from: $u(TFT, RAND_p)$ by swapping S and T

$$
\begin{aligned}
&u(RAND_p, TFT)\\
&= \frac{p^2 \cdot \omega \cdot (T+S-R-P) + p \cdot (2\omega P + R(\omega-1) - 2\omega T + T - \omega S) - \omega P + T(\omega-1)}{\omega - 1}
\end{aligned}
$$

vs. TFTD

Results from: $u(TFTD, RAND_p)$ by swapping S and T

Note: $u(TFTD, RAND_p)$ is part of the calculation of $u(TFT, RAND_p)$

$$
u(RAND_p, TFTD) = \frac{\omega p^3 + \omega p^2(R-S-T-1) + \omega pT + p(S+1-p)}{1-\omega}
$$

vs. TF2T

Let

$$
\begin{aligned}
u_1 &:= u(RAND_p, TF2T) = p(R + \omega \cdot u(RAND_p, TF2T)) + (1-p)(T + \omega \cdot u(RAND_p, TF2T_1))\\
u_2 &:= u(RAND_p, TF2T_1) = p(R + \omega \cdot u(RAND_p, TF2T)) + (1-p)(T + \omega \cdot u(RAND_p, TF2T_2))\\
u_3 &:= u(RAND_p, TF2T_2) = p(S + \omega \cdot u(RAND_p, TF2T)) + (1-p)(P + \omega \cdot u(RAND_p, TF2T_2))
\end{aligned}
$$

then

$$
\begin{aligned}
&u_3 = \frac{pS + \omega p \cdot u_1 + p - p^2}{1 - \omega + \omega p}\\
&\implies u_2 = p(R + \omega \cdot u_1) + (1-p)(T + \omega \cdot \frac{pS + \omega p \cdot u_1 + p - p^2}{1 - \omega + \omega p})\\
&= p\,(R +_1) + (1-p)\left(T + \frac{\omega\left(pS + \omega p u_1 + p - p^2\right)}{1 - \omega + \omega p}\right)\\
&\implies u_1 = p(R + \omega \cdot u_1) + (1-p)(T + \omega \cdot (p\,(R +_1) + (1-p)\left(T + \frac{\omega\left(pS + \omega p u_1 + p - p^2\right)}{1 - \omega + \omega p}\right)))\\
&= \frac{-\omega^2 p^4 + 3\omega^2 p^3 + \omega^2 p^3 T - \omega^2 p^3 R + \omega^2 p^3 S - 3\omega^2 p^2 - 3\omega^2 p^2 T + 2\omega^2 p^2 R - 2\omega^2 p^2 S + \omega^2 p}{-\omega + 1}\\
&+ \frac{3\omega^2 pT - pT + pR - \omega^2 pR + \omega^2 pS + T - \omega^2 T}{-\omega + 1}\\
&= \frac{-\omega^2 p^4 + \omega^2 p^3 \cdot (3 + T - R + S) + \omega^2 p^2 \cdot (-3 - 3T + 2R - 2S) + \omega^2 p(1 + 3T)}{1 - \omega}\\
&+ \frac{p(-T+R) + \omega^2 p \cdot (-R+S) + T - \omega^2 T}{1 - \omega}
\end{aligned}
$$

Always Defect (AD)

vs. RANDOM with param p

Results from: $u(RAND_p, AD)$ by swapping S and T

$$u(AD, RAND_p) = p \cdot T + (1-p) \cdot P + \omega \cdot u(AD, RAND_p) = \frac{p \cdot T + (1-p) \cdot P}{1-\omega}$$

vs. AD

$$u(AD, AD) = P + \omega P + \omega^2 P + \ldots = \frac{1}{1-\omega} \cdot P$$

vs. AC

$$u(AD, AC) = T + \omega T + \omega^2 T + \ldots = \frac{1}{1-\omega} \cdot T$$

vs. GT

$$u(AD, GT) = T + \omega \cdot u(AD, AD) = T + \frac{\omega}{1-\omega} P$$

vs. TFT

Same behaviour as GT

$$u(AD, TFT) = T + \omega \cdot u(AD, AD) = T + \frac{\omega}{1-\omega} P$$

vs. TFTD

$$u(AD, TFTD) = P + \omega \cdot u(AD, AD) = \frac{1}{1-\omega} P$$

vs. TF2T

$$u(AD, TF2T) = T + \omega \cdot T + \omega^2 \cdot u(AD, AD) = T + \omega T + \frac{\omega^2}{1-\omega} P$$

Always Cooperate (AC)

vs. RANDOM with param p

Results from: $u(RAND_p, AC)$ by swapping S and T

$$
\begin{aligned}
& u(AC, RAND_p) \\
& = p \cdot R + (1-p) \cdot S + \omega \cdot u(AC, RAND_p) \\
& = \frac{p \cdot R + (1-p) \cdot S}{1-\omega}
\end{aligned}
$$

vs. AD

$$u(AC, AD) = S + \omega S + \omega^2 S + ... = \frac{1}{1-\omega} \cdot S$$

vs. AC

$$u(AC, AC) = R + \omega R + \omega^2 R + ... = \frac{1}{1-\omega} \cdot R$$

vs. GT

$$u(AC, GT) = R + \omega R + \omega^2 R + ... = \frac{1}{1-\omega} \cdot R$$

vs. TFT

$$u(AC, TFT) = R + \omega R + \omega^2 R + ... = \frac{1}{1-\omega} \cdot R$$

vs. TFTD

$$u(AC, TFTD) = S + \frac{\omega}{1-\omega} \cdot R$$

vs. TF2T

$$u(AC, TF2T) = R + \omega R + \omega^2 R + ... = \frac{1}{1-\omega} \cdot R$$

Grim Trigger (GT)

vs. RANDOM with param p

Results from: $u(RAND_p, GT)$ by swapping S and T

$$u(GT, RAND_p) = \frac{p \cdot R + (1-p) \cdot (S + \omega \cdot \frac{p \cdot T + (1-p) \cdot P}{1-\omega})}{1 - p \cdot \omega}$$

vs. AD

$$u(GT, AD) = S + \omega \cdot u(AD, AD) = S + \frac{\omega}{1-\omega} P$$

vs. AC

$$u(GT, AC) = R + \omega R + \omega^2 R + ... = \frac{1}{1-\omega} \cdot R$$

vs. GT

$$u(GT, GT) = R + \omega R + \omega^2 R + ... = \frac{1}{1-\omega} \cdot R$$

vs. TFT

$$u(GT, TFT) = R + \omega R + \omega^2 R + ... = \frac{1}{1-\omega} \cdot R$$

vs. TFTD

$$u(GT, TFTD) = S + \omega \cdot T + \frac{\omega^2}{1-\omega} \cdot P$$

vs. TF2T

$$u(GT, TF2T) = R + \omega R + \omega^2 R + ... = \frac{1}{1-\omega} \cdot R$$

Tit-for-Tat (TFT)

vs. TFT

vs. RANDOM with param p

$$u_1 := u(TFT, RAND_p) = p \cdot (R + \omega \cdot u(TFT, RAND_p)) + (1-p) \cdot (S + \omega \cdot u(TFTD, RAND_p))$$
$$u_2 := u(TFTD, RAND_p) = p \cdot (T + \omega \cdot u(TFT, RAND_p)) + (1-p) \cdot (P + \omega \cdot u(TFTD, RAND_p))$$

Then:

$$\begin{aligned}
&u_2 = p \cdot (T + \omega \cdot u_1) + (1-p) \cdot (P + \omega \cdot u_2) \\
\Leftrightarrow &u_2 = p \cdot (T + \omega \cdot u_1) + (1-p) \cdot P + (1-p) \cdot \omega \cdot u_2 \\
\Leftrightarrow &u_2 - (1-p) \cdot \omega \cdot u_2 = p \cdot (T + \omega \cdot u_1) + (1-p) \cdot P \\
\Leftrightarrow &u_2(1 - (1-p) \cdot \omega) = p_(T + \omega_u \cdot 1) + (1-p) \cdot P \\
\Leftrightarrow &u_2 = \frac{p \cdot (T + \omega \cdot u_1) + (1-p) \cdot P}{1 - (1-p) \cdot \omega} = \frac{p \cdot (T + \omega \cdot u_1) + (1-p) \cdot P}{1 - (1-p) \cdot \omega} \\
&u_1 = p \cdot (R + \omega \cdot u_1) + (1-p) \cdot (S + \omega \cdot u_2) \\
\Leftrightarrow &u_1 = p \cdot \omega \cdot u_1 + p \cdot R + (1-p) \cdot (S + \omega \cdot u_2) \\
\Leftrightarrow &u_1(1 - p \cdot \omega) = p \cdot R + (1-p) \cdot (S + \omega \cdot u_2) \\
\Leftrightarrow &u_1 = \frac{p \cdot R + (1-p) \cdot (S + \omega \cdot u_2)}{1 - p \cdot \omega} \\
&u_2 = \frac{p \cdot (T + \omega \cdot \frac{p \cdot R + (1-p) \cdot (S + \omega \cdot u_2)}{1 - p \cdot \omega}) + (1-p) \cdot P}{1 - (1-p) \cdot \omega} \\
&u(TFTD, RAND_p) = \frac{\omega p^3 + \omega p^2 (R - T - S - 1) + \omega p S + p(T + 1 - p)}{1 - \omega} \\
&u_1 = \frac{p \cdot R + (1-p) \cdot (S + \omega \cdot \frac{p \cdot (T + \omega \cdot u_1) + (1-p) \cdot P}{1 - (1-p) \cdot \omega})}{1 - p \cdot \omega} \\
&u(TFT, RAND_p) = \frac{\omega p^3 + \omega p^2 (R - T - S - 2) + \omega p (1 + T - R + 2S) + p(R - S) + S - \omega S}{1 - \omega}
\end{aligned}$$

vs. AD

$$u(TFT, AD) = S + \omega P + \omega^2 P + \ldots = S + \omega P \cdot \sum_{i=0} \omega^i = S + \frac{\omega}{1 - \omega} P$$

vs. AC

$$u(TFT, AC) = R + \omega R + \omega^2 R + \ldots = R \cdot \sum_{i=0} \omega^i = \frac{1}{1 - \omega} R$$

vs. GT

$$u(TFT,GT) = R + \omega R + \omega^2 R + ... = R \cdot \sum_{i=0} \omega^i = \frac{1}{1-\omega} R$$

vs. TFT

$$u(TFT,TFT) = R + \omega R + \omega^2 R + ... = R \cdot \sum_{i=0} \omega^i = \frac{1}{1-\omega} R$$

vs. TFTD

$$u(TFT,TFTD) = S + \omega T + \omega^2 S + \omega^3 T + ... = \sum_{i=0} \omega^{2i} S + \omega \sum_{i=0} \omega^{2i} T$$
$$u(TFT,TFTD) = \frac{1}{1-\omega^2} S + \frac{\omega}{1-\omega^2} T$$

vs. TF2T

$$u(TFT,TF2T) = R + \omega R + \omega^2 R + ... = R \cdot \sum_{i=0} \omega^i = \frac{1}{1-\omega} R$$

Suspicious Tit-for-Tat (TFTD)

vs. RANDOM with param p

Results from: $u(RAND_p, TFTD)$ by swapping S and T Note: see $u(TFT, RAND_p)$ for the calculation

$$u(TFTD, RAND_p) = \frac{\omega p^3 + \omega p^2 (R - T - S - 1) + \omega p S + p(T + 1 - p)}{1-\omega}$$

vs. AD

$u(TFTD,AD) = u(AD,AD) = \frac{1}{1-\omega} \cdot P$

vs. AC

$$u(TFTD,AC) = T + \omega \cdot u(AC,AC) = T + \frac{\omega}{1-\omega} \cdot R$$

vs. GT

$$u(TFTD,GT) = T + \omega \cdot S + \omega^2 \cdot u(AD,AD) = T + \omega S + \frac{\omega^2}{1-\omega} \cdot R$$

vs. TFT

see u(TFT, TFTD):

$$u(TFTD, TFT) = \frac{1}{1-\omega^2}T + \frac{\omega}{1-\omega^2}S$$

vs. TFTD

$$u(TFTD, TFTD) = u(AD, AD) = \frac{1}{1-\omega} \cdot P$$

vs. TF2T

$$u(TFTD, TF2T) = T + \omega \cdot u(AC, AC) = T + \frac{\omega}{1-\omega} \cdot R$$

Tit-for-2-Tat (TF2T)

vs. RANDOM with param p

Results from: $u(RAND_p, TF2T)$ by swapping S and T

$$\begin{aligned}
&u(TF2T, RAND_p) \\
&= \frac{-\omega^2 p^4 + 3\omega^2 p^3 + \omega^2 p^3 S - \omega^2 p^3 R + \omega^2 p^3 T - 3\omega^2 p^2 - 3\omega^2 p^2 S + 2\omega^2 p^2 R - 2\omega^2 p^2 T}{-\omega+1} \\
&+ \frac{\omega^2 p + 3\omega^2 pS - pS + pR - \omega^2 pR + \omega^2 pT + S - \omega^2 S}{-\omega+1} \\
&= \frac{-\omega^2 p^4 + \omega^2 p^3 \cdot (3 + S - R + T) + \omega^2 p^2 \cdot (-3 - 3S + 2R - 2T)}{1-\omega} \\
&+ \frac{\omega^2 p(1+3S) + p(-S+R) + \omega^2 p \cdot (-R+T) + S - \omega^2 S}{1-\omega}
\end{aligned}$$

vs. AD

$$u(TF2T, AD) = S + \omega \cdot S + \omega^2 \cdot u(AD, AD) = S + \omega S + \frac{\omega^2}{1-\omega}P$$

vs. AC

$$u(TF2T, AC) = S + \omega \cdot S + \omega^2 \cdot u(AD, AD) = S + \omega S + \frac{\omega^2}{1-\omega}P$$

vs. GT

$$u(TF2T, GT) = R + \omega R + \omega^2 R + \ldots = \frac{1}{1-\omega} \cdot R$$

vs. TFT

$$u(TF2T, TFT) = R + \omega R + \omega^2 R + \ldots = \frac{1}{1-\omega} \cdot R$$

vs. TFTD

$$u(TF2T, TFTD) = S + \frac{\omega}{1-\omega} \cdot R$$

vs. TF2T

$$u(TF2T, TF2T) = R + \omega R + \omega^2 R + \ldots = \frac{1}{1-\omega} \cdot R$$

B Effect of higher imitation- and migration-probabilities on the base case

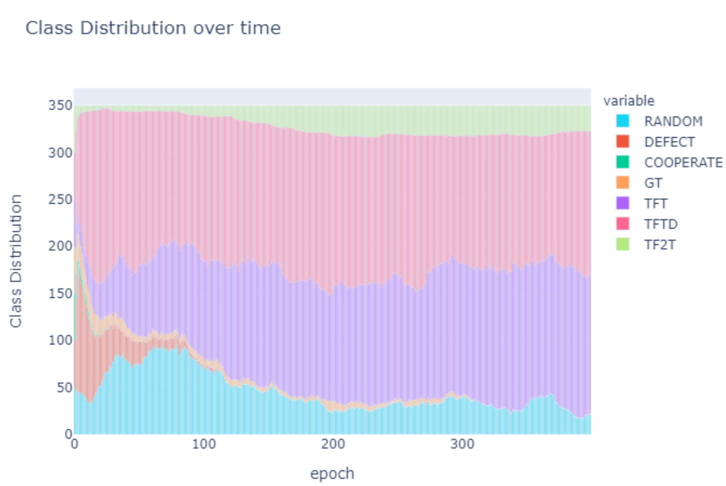


Figure 14. Strategy distribution over time with 50% migration rate

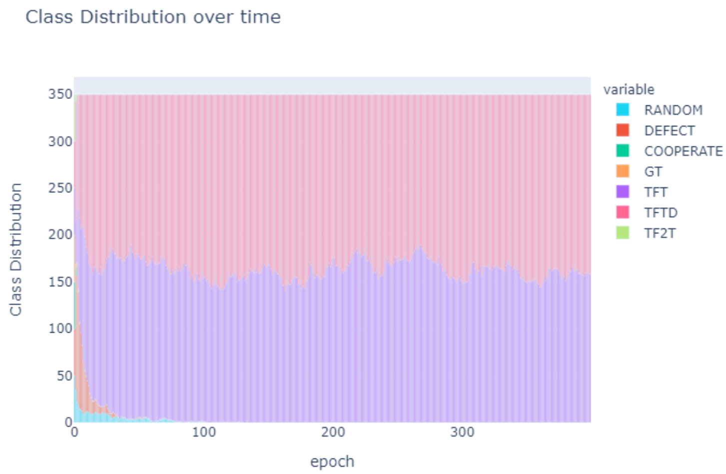


Figure 15. Strategy distribution over time with 90% migration rate

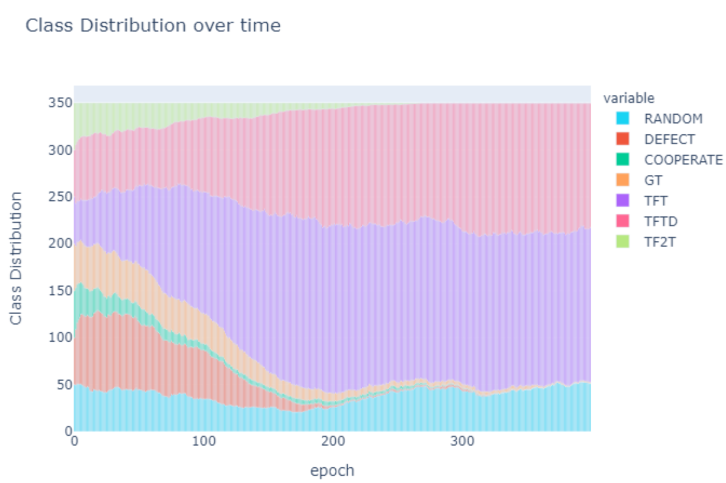


Figure 16. Strategy distribution over time with 50% migration rate

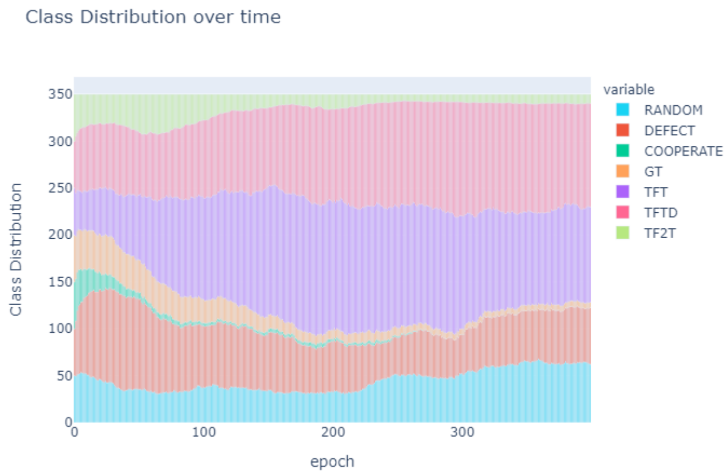


Figure 17. Strategy distribution over time with 90% migration rate

References

1. Pittel, K. & Rübbelke, D. T. Transitions in the negotiations on climate change: from prisoners dilemma to chicken and beyond. *Int. Environ. Agreements: Polit. Law Econ.* **12**, 23–39 (2012).
2. Lumsden, M. The cyprus conflict as a prisoners dilemma game. *J. Confl. Resolut.* **17**, 7–32 (1973).
3. Tullock, G. Adam smith and the prisoners dilemma. *The Q. J. Econ.* **100**, 1073–1081 (1985).
4. Snidal, D. Coordination versus prisoners dilemma: Implications for international cooperation and regimes. *The Am. Polit. Sci. Rev.* 923–942 (1985).
5. Leibenstein, H. The prisoners dilemma in the invisible hand: an analysis of intrafirm productivity. *The Am. Econ. Rev.* **72**, 92–97 (1982).
6. Lewis, D. Prisoners dilemma is a newcomb problem. *Philos. & Public Aff.* 235–240 (1979).
7. Cardinot, M., Griffith, J., O'Riordan, C. & Perc, M. Cooperation in the spatial prisoner's dilemma game with probabilistic abstention. *Sci. Reports* **8** (2018).
8. Axelrod, R. & Hamilton, W. D. The evolution of cooperation. *science* **211**, 1390–1396 (1981).
9. Singer-Clark, T. Morality metrics on iterated prisoners dilemma players. (2014).
10. Bó, P. D. Cooperation under the shadow of the future: experimental evidence from infinitely repeated games. *Am. economic review* **95**, 1591–1604 (2005).
11. Ashlock, D. Cooperation in prisoner's dilemma on graphs. 48 – 55, DOI: 10.1109/CIG.2007.368078 (2007).
12. Durán, O. & Mulet, R. Evolutionary prisoner's dilemma in random graphs. *Phys. D: Nonlinear Phenom.* **208**, 257–265 (2005).
13. Nowak, M. A. & May, R. M. Evolutionary games and spatial chaos. *Nature* **359**, 826–829 (1992).
14. Games, M. The fantastic combinations of john conways new solitaire game @articleehlert2020human, title=Human social preferences cluster and spread in the field, author=Ehlert, Alexander and Kindschi, Martin and Algesheimer, René and Rauhut, Heiko, journal=Proceedings of the National Academy of Sciences, volume=117, number=37, pages=22787–22792, year=2020, publisher=National Acad Scienceslife by martin gardner. *Sci. Am.* **223**, 120–123 (1970).
15. Kendall, G., Yao, X. & Chong, S. Y. *The Iterated Prisoners' Dilemma* (WORLD SCIENTIFIC, 2007). https://www.worldscientific.com/doi/pdf/10.1142/6461.
16. Helbing, D. & Balietti, S. *Social Self-Organization: Agent-Based Simulations and Experiments to Study Emergent Social Behavior*, 25–70 (2012).
17. Majeski, S., Linden, G., Linden, C. & Spitzer, A. A spatial iterated prisoners dilemma game simulation with movement. In *Simulating social phenomena*, 161–167 (Springer, 1997).

18. Suzuki, R. & Arita, T. Evolutionary analysis on spatial locality in n-person iterated prisoners dilemma. *Int. J. Comput. Intell. Appl.* **3**, 177–188 (2003).

19. Chiong, R. & Kirley, M. Random mobility and the evolution of cooperation in spatial n-player iterated prisoners dilemma games. *Phys. A: Stat. Mech. its Appl.* **391**, 3915–3923 (2012).

20. Ishida, Y. & Mori, T. Spatial strategies in a generalized spatial prisoners dilemma. *Artif. life robotics* **9**, 139–143 (2005).

21. Sheng, Z.-H., Hou, Y.-Z., Wang, X.-L. & Du, J.-G. The evolution of cooperation with memory, learning and dynamic preferential selection in spatial prisoners dilemma game. In *Journal of Physics: Conference Series*, vol. 96, 012107 (IOP Publishing, 2008).

22. Alonso-Sanz, R. Spatial order prevails over memory in boosting cooperation in the iterated prisoners dilemma. *Chaos: An Interdiscip. J. Nonlinear Sci.* **19**, 023102 (2009).

23. Bó, P. D. Cooperation under the shadow of the future: Experimental evidence from infinitely repeated games. *The Am. Econ. Rev.* **95**, 1591–1604 (2005).

24. Axelrod, R. The emergence of cooperation among egoists. *The Am. Polit. Sci. Rev.* **75**, 306–318 (1981).

25. Cooper, R., DeJong, D. V., Forsythe, R. & Ross, T. W. Cooperation without reputation: Experimental evidence from prisoner's dilemma games. *Games Econ. Behav.* **12**, 187–218, DOI: https://doi.org/10.1006/game.1996.0013 (1996).

26. Tullock, G. Adam smith and the prisoners' dilemma. *The Q. J. Econ.* **100**, 1073–1081 (1985).

27. Brede, M. Short versus long term benefits and the evolution of cooperation in the prisoner's dilemma game. *PLOS ONE* **8**, 1–9, DOI: 10.1371/journal.pone.0056016 (2013).

28. Rogers, N. & Ashlock, D. The Impact of Long-Term Memory in the Iterated Prisoner's Dilemma. In *Intelligent Engineering Systems through Artificial Neural Networks*, DOI: 10.1115/1.802953.paper31 (ASME Press, 2009). https://asmedigitalcollection.asme.org/book/chapter-pdf/2795594/802953_paper31.pdf.