Trust Simulation and Modeling Using Iterated Prisoner’s Dilemma

Arief Rahman
Informatics Department
Institut Teknologi Bandung
Bandung, Indonesia
ariefrahman95@gmail.com

***Abstract*—The changing trust level in society is affected by several factors, like other people’s past actions, misunderstanding, the dominant behavior in the environment, and extortion tactics. The ever-changing level of trust can be modeled using game theory, particularly with iterated prisoner’s dilemma game. In this paper, we propose a model and simulation scheme to simulate trust evolution in a population of people (agents) with several trust strategies and experimented using many types of agents.**

Keywords—model, iterated prisoner's dilemma, simulation, trust, zero-determinant

I. Introduction

In society, we as humans choose to cooperate or not depending on how much faith we put in other people. The level of trust we place to others depends on their past actions. The more a person cooperates with others, the more other people will trust them. However, this also makes some other people want to exploit the cooperativeness of this person. The dynamics of trust based on past actions are fascinating to model and simulate.

In some cases, some of these actions can result from misunderstanding. There are cases when someone wants to help others, but inadvertently ended up betraying or cheating the people they want to help. In this case, the recipient of the action may end up not trusting the other person, even if it is a mistake. The introduction of misunderstanding in the environment can affect trust to other people radically.

In addition to past actions and misunderstanding, people can also adapt to how they handle trust to other people based on others. This phenomenon is inherently present in humans, especially on those who regularly interact with many kinds of people. In a society with adaptable people, the dynamics of trust consisting of many types of people can result in an interesting analysis regarding human psyche itself.

Another major factor in changing trust environment is the presence of extortion tactics. This phenomenon occurs when a person forces another person to work for them, in exchange for a certain benefit to the other person. Extortion causes the extorter to receive a significant benefit, while the extorted person only receives a small benefit that is much smaller in comparison. The presence of extortion phenomenon surprisingly affects the trust level in an environment significantly, especially when it is widespread.

The way trust evolves based on people's interactions with others and the environment itself can be modeled using game theory. Game theory allows us to determine how we make a decision. By analyzing what choices would result in the best outcome possible for them, a definite answer can be determined, even if it at first appears complicated.

One of the most often-used game theory to analyze trust and interpersonal relations is the iterated prisoner’s dilemma game. The original game (prisoner’s dilemma) is often used for making a case of why two rational people might not cooperate even in their best interest to do so. However, the expansion of this game, called the iterated prisoner's dilemma, is used more for simulating evolution of trust, because it shows the nature of cooperation and trust better than the original game.

In this paper, we propose a plan to model and simulate the evolution of trust in a population consisting of people with many kinds of trust principles (called strategies) using iterated prisoner’s dilemma. Section II will discuss the prisoner's dilemma and its variant, the iterated prisoner's dilemma. Section III will present our proposed model and simulation for trust evolution. Finally, Section IV will discuss the implementation plan of the proposed model in this paper.

II. Prisoner’s Dilemma

In this section, we discuss several types of prisoner’s dilemma and their properties.

A. Prisoner's Dilemma

Prisoner's dilemma is one of the most standard games analyzed in game theory [1]. In this game, two players are given a choice of whether to cooperate with (a.k.a. ally) or defect against (a.k.a. cheat, betray) the other player. Each player will receive or lose points depending on both player's choice. Below are three possible outcomes of the game:

1. If both players cooperate, they both get *R* (reward) points.
2. If both players cheat, both players receive *P* (punishment) points.
3. If a player defects while the other player cooperates, the defecting player gets *T* (temptation) points, while the cooperating player receives *S* (sucker) points.

Table I shows the generalized outcome chart of a standard prisoner's dilemma.

Using the standard outcomes and assuming a self-contained result with rational decision making, the best decision in a round of prisoner's dilemma would be to defect, because defecting would never make a player's points lower than the other player's [1]. However, contrary to this result, humans display a systemic bias to cooperate in a similar situation [2]. To represent humans better, a prisoner's dilemma model usually uses a specific decision-making strategy for each agent. This model has been shown to make better model and simulations for similar cases [3].

TABLE I. PRISONER'S DILEMMA OUTCOME

A \ B DECISION	**Cooperate**	**Cheat**
Cooperate	$R \setminus R$	$S \setminus T$
Cheat	$T \setminus S$	$P \setminus P$

The following condition is necessary for a prisoner's dilemma game to have a strong condition: $T > R > P > S$. Depending on the value of T, R, P, and S, the dominant strategy for a prisoner's dilemma game becomes different. If $R > S$, the best approach is for both agents to cooperate. If $T > R$ and $P > S$, the best strategy for both agents is to cheat [1].

B. Iterated Prisoner's Dilemma

Iterated prisoner's dilemma is an expansion of the original prisoner's dilemma [4]. In this game, two players play the prisoner's dilemma game more than once in succession, remembering both their own and the other player's previous choices. This allows the players to plan their next decisions accordingly with a specific strategy. This variant of prisoner's dilemma is used more than other similar game theories for simulating human cooperation and trust in an environment, which is the focus of this paper.

Below are the three conditions necessary for an iterated prisoner's dilemma game to have a strong condition [1]:

1. The original prisoner's dilemma's strong condition ($T > R > P > S$).
2. $2R > T + S$. This condition is necessary to prevent both players from alternating between cooperating and cheating. Without this condition, there is a chance that this strategy results in greater reward than mutual cooperation.
3. The number of rounds is not known to both agents. If an agent knows the number of rounds, by inductive reasoning, the best decision for each round is to cheat.

III. TRUST SIMULATION

Our model and simulation for trust evolution adopts three popular iterated prisoner's dilemma models: Axelrod [4] and Case [5]. The next two sections will explain these three models in detail.

A. Axelrod Iterated Prisoner's Dilemma Model

Axelrod [4][6][7] proposed a model to simulate how agents with different personalities react to each other using iterated prisoner's dilemma. Axelrod executed two separate tournaments, with the same game setting but different amount of strategies. The simulation is done by having a round-robin style tournament of iterated prisoner's dilemma for every possible pair of agents. Every agent then plays 200 rounds of prisoner's dilemma games with other agents using different strategies (including an agent with the same strategy as their own), using a *TRPS* value of 5, 3, 1, and 0 respectively. The score of each agent is tallied. A score is defined as the payoff an agent receives after a Prisoner's Dilemma game. The agents are then sorted based on their average total score across twenty tournaments with different random seed.

Axelrod discovered that the Tit-for-Tat strategy, which has one of the simplest implementations (after cooperator and defector), also has the best average score, compared to other sophisticated decision strategies. Even in the second tournament [7], the Tit-for-

Tat strategy still managed to place first. Using this information, Axelrod concluded that agents with the best average scores tend to have these four properties:

1. Nice (the agent tries to cooperate as much as they can)
2. Retaliating (the agent will not cheat before being cheated by their opponent)
3. Forgiving (the agent will become nice again if the opponent does not continue to cheat)
4. Non-envious (the agent does not strive to score more than their opponent).

There are 15 unique agents in the first tournament and 66 unique ones in the second tournament. The list of strategies used in the first Axelrod tournament can be viewed in Appendix A, while the list for the second tournament can be seen in Appendix B.

It was later discovered that one of the strategies in the first tournament was implemented incorrectly, resulting in a worse placement during the tournament. This strategy, created by Downing, decides the next action by estimating the probability of the opponent cooperating or defecting, given their last action ($P(C|C)$ and $P(C|D)$). The strategy won with the biggest score when it is implemented correctly. This discovery leads to a conclusion that an agent with an adaptive decision strategy also tend to perform better than the static ones.

B. Case Simulation - Evolution of Trust

Case's "Evolution of Trust" simulation [5] is a web simulation published in 2017 that simulates how people's interactions evolve trust in an environment. Case's simulation adopts Axelrod's simulation model [4]. In Case's simulation, the iterated prisoner's dilemma is done in tournament style. For each match, the agents will have several rounds of prisoner's dilemma game with each other in round-robin styles. After the scores are tallied, the agents with the lowest scores will be deleted and replaced by clones of the agent with the highest score. In other words, the agents with the lowest scores will imitate the strategy of the best agent. Everyone's score is reset to 0 for the next match. The match is then done repeatedly.

Case's "Evolution of Trust" simulation also introduces the concept of choice mistake, in which the agent does the opposite of their initial decision within a certain probability. The addition of mistake mechanic in iterated prisoner's dilemma has been observed to affect trust evolution radically [8].

Below are the necessary user inputs for Case's simulation.

1. The agent population. There can be more than one agent that uses the same strategy.
2. The outcome of each decision scenario (*TRPS*). The value of *TRPS* has to obey the strong conditions defined for iterated prisoner's dilemma.
3. The number of prisoner's dilemma rounds for each pair of agents.
4. The probability of an agent making a mistake.
5. The number of agents with lowest scores that are replaced at the end of every match.

Below are the assumptions the Case simulation uses.

1. The agent does not know the number of rounds, thus relying only on the strategy they adopt for their decision-making process.
2. Every pair of agents will have the same number of prisoner's dilemma rounds.
3. Making mistake applies to both cooperate and defect choice.
4. Every agent will play an iterated prisoner's dilemma game with every other agent in the population.
5. At the end of every match, the history of an agent's past interactions with other agents will be deleted.

Case used 8 simple strategies that are present in the previous studies, although some of them have different names. The list of strategies used in the first the Case simulation can be seen in Appendix C.

Case's simulation confirmed Axelrod's result: Copycat (Tit-for-Tat) strategy has the best total score in a mistake-free environment. However, in an environment with low non-zero mistake percentage, Copykitten (Tit-for-Two-Tats) strategy overall performs better than Copycat. In short, Case concluded three things from this simulation:

1. Forgiveness allows better performances in repeated interactions.
2. Trust breaks down in high level of misunderstanding.
3. Being forgiving resulted in a much better trust environment when misunderstanding level is low.

C. *Stewart-Plotkin Tournament with Zero Determinant Strategies*

Stewart and Plotkin [9] ran an Axelrod-style tournament that involves agents with Zero-Determinant strategies [10]. These strategies, also called ZD, are strategies that can set the opponent's score deterministically regardless of their reaction to the ZD agent. The strategies can also enforce and extort the opponent (especially one with evolutionary strategy) to help the ZD agent, but not managing to catch up with the ZD agent's score.

Stewart and Plotkin used 19 unique strategies in their tournament. Two of them are ZD strategies. Appendix D shows the list of strategies used in this tournament.

Stewart and Plotkin discovered that ZD strategies do manage to fare better than any other strategy presents in the tournament. Even the best strategy in the first Axelrod tournament, Tit-for-Tat, still had lower payoff than both ZD strategies. Steward and Plotkin concluded that extortion tactics employed by ZD strategies will win most iterated prisoner's dilemma games that uses round-robin style tournament.

IV. Proposed Simulation

Axelrod's [4] and Stewart & Plotkin's [9] simulation in itself has a very basic scheme of iterated prisoner's dilemma. However, it has a diverse variety of strategies to compare with. Case's [5] simulation, on the other hand, is much more customizable than Axelrod's simulation. However, all of the strategies present in this simulation is simple. Because of these limitations, we propose a model that is a hybrid of Axelrod's, Case's, and Steward & Plotkin's simulations, which combines Case's customizable simulation and Axelrod's and Stewart and Plotkin's diverse variety of strategies.

Our simulation used both Axelrod's and Case's simulation format. Axelrod's tournament will be used to determine the average payoff of every agent during an entire round of iterated prisoner dilemma game. We use this information as a supplemental analysis for the Case's simulation. We use the same number of games per agent pair as the one Axelrod used, which is 200.

From Case's simulation, we intend to adopt the model setting, with one main difference: making mistake only applies to cooperation. We adopt the real condition where almost all mistakes in real world interactions end up in detrimental for the other receiving person. We detail the assumptions we used in our simulation below.

1. The agent knows the number of rounds, but the iterated prisoner's dilemma game can end after any game with a fixed probability. This assumption is used to allow strategies that cheats at the very end of the game (e.g. Borufsen and Stein & Rapoport) while keeping the fairness of the games. We used 0.005 probability of ending an iterated prisoner's dilemma game after each prisoner's dilemma game, with 200 rounds as the maximum number of games in one iterated prisoner's dilemma game.
2. Making mistakes can apply to both cooperate and defect choice or only to cooperate choice (mistake bias).
3. Every agent will play an iterated prisoner's dilemma game with every other agent in the population.
4. At the end of every match, the history of an agent's past interactions with other agents will be deleted.

We implemented the simulation in Python, using `axelrod` library[1] and used as many strategies as possible that are available in the implementation at the time of the writing of this paper (May 4th, 2018). The list of strategies we used in our simulation is listed in Table II.

We also implemented our own Case simulation to be used in the `axelrod` library, in the form of `CaseProcess` class. This class accepts the following parameters during the instantiation:

1. List of agents.
2. The maximum number of prisoner's dilemma games each pair of agents will play. We use a default value of 200, the same as the one Axelrod and Stewart and Platkin used.
3. The probability of ending an iterated prisoner's dilemma game after a game of prisoner's dilemma. We use a default value of 0.005 in all scenarios.
4. The maximum number of matches. This parameter is used in case the agents never converge into one type of strategy. Our implementation checks whether the population only contains homogenous cooperative agents or the maximum the number of matches is reached. If one of these conditions are fulfilled, the simulation immediately terminates. We used a default value of 50.
5. The probability of making a mistake. The default value is 0.

[1] http://axelrod.readthedocs.org

6. Whether mistake bias is activated or not. If mistake bias is activated, a mistake will only be allowed on cooperation choice. In other words, if an agent cooperates, there's a chance they will flip their choice to defect. However, if they defect, the choice will never be flipped to cooperate. The mistake bias is disabled by default.
7. The number of agents to replace at the end of each match. The default value is 1.
8. The *TRPS* value. The default values use Axelrod's *TRPS* value: 5, 3, 1, and 0 for each.

V. Experiment

In this section, we explain the experiment scenarios we used in this research. We differentiate the simulation based on these parameters:

1. The list of agents. We used five lists of agents in this research: Axelrod's first tournament's agents and second tournament's agents, Case's simulation's agents, Steward and Plotkin's tournament's agents, and the best agents from the previous four simulations. We define the best agents as the ones that: (1) consistently survived to the end of the Case's simulation, and/or (2) had more than 1/3 of the entire agent population consistently during the Case's simulation. We used 20 best strategies that are gotten from the four tournaments.
2. The tournament types. We used both Axelrod's original tournament format and Case's simulation format. Axelrod's original tournament is used to analyze each agent's average score gain throughout the iterated prisoner's dilemma games. The Case's simulation on the other hand is used to analyze the effect of the three trust aspects that we discussed in the background section: multiple interactions, mistakes, and adaptations.
3. Mistake percentage. We used three mistake percentages: 0, 0.05, and 0.2. Furthermore, we used the mistake bias to differentiate the later two's scenarios.
4. Seed. We did the experiment using twenty different seeds to ensure prevent random bias on our results and analysis.
5. For all scenarios, we used the following parameters:
    a. Number of rounds for each pair of agents: 200
    b. Maximum number of matches: 50
    c. Number of agents to replace per match: 1
    d. *TRPS*: 5, 3, 1, and 0.

Due to the enormous amount of the graph results, we will not put all the results in this paper. We will only show one of the most common results from the simulation. We will also discuss some of the cases not shown in this paper. We direct you to the author's github page to see the complete results: https://github.com/ariefrahman95/Axelrod. We also provide our implementation of `CaseProcess` in the aforementioned link.

TABLE II. List of Strategies Used

No.	Simulation	Strategies Used	Description
1	Axelrod's first tournament	Tit-for-Tat, Tideman and Chieruzzi, Nydegger, Grofman, Shubik, Stein and Rapoport, Grudger, Davis, Revised Downing, Feld, Joss, Tullock, Unnamed Strategy, Random (50-50)	All strategies except Graaskamp are accounted
2	Axelrod's second tournament	Axelrod's first tournament strategies, Black, Borufsen, Cave, Champion, Colbert, Eatherley, Getzler, Gladstein, Go by Majority, Graaskamp & Katzen, Harrington, Kluepfel, Leyvraz, Mikkelson, More Grofman, More Tideman and Chieruzzi, Richard Hufford, Tester, Tranquilizer, Weiner, White, Wm Adams, Yamachi	Less than half of the strategies present are accounted
3	Case simulation	Cooperator, Cheater (Defector), Copycat (Tit-for-Tat), Detective, Grudger, Copykitten (Tit-for-Two-Tats), Simpleton (Win-Stay-Lose-Shift), Random (50-50)	Detective strategy is not available in the library, but can be easily implemented
4	Steward and Plotkin's tournament	Case's simulation strategies (except Detective), ZD Extort-2, Hard Go by Majority, Hard Tit-for-Tat, Hard Tit-for-Two-Tats, Generous Tit-for-Tat, ZD Generous Tit-for-Tat 2, Calculator, Prober, Prober 2, Prober 3, Hard Prober, Naïve Prober, Random (50-50)	All strategies are accounted. Detective strategy is not present in the original tournament.
5	Best players from Axelrod's two tournaments, Case's simulation, and Steward and Plotkin's tournament	Stein and Rapoport, Grudger, Revised Downing, Tit-for-Tat, Davis, Nydegger, Cave, Tranquilizer, White, Eatherley, Champion, Harrington, Defector, Random, Win-Stay-Lose-Shift, Generous Tit-for-Tat, ZD Generous-Tit-for-Tat 2, Tit-for-2-Tats, Naïve Prober, Hard Prober	-

A. Axelrod Tournament Result

We show in detail the results of the Axelrod's tournament based on the scenarios we proposed in the previous section.

1) Axelrod Tournament Result with Axelrod's First Tournament Agents

Figure 1 shows the payoff result of the Axelrod tournament using all agents from the Axelrod's first tournament, except the Graaskamp agent. The revised Downing agent consistently comes in the first place, followed by Grofman, Nydegger, and Stein and Rapoport almost consistently (19 out of 20 cases). The result is radically different from the ones listed in Axelrod [4], in which Tit-for-Tat won with the best average score per round. This result may be because of the revised Downing agent and the lack of Graaskamp agent.

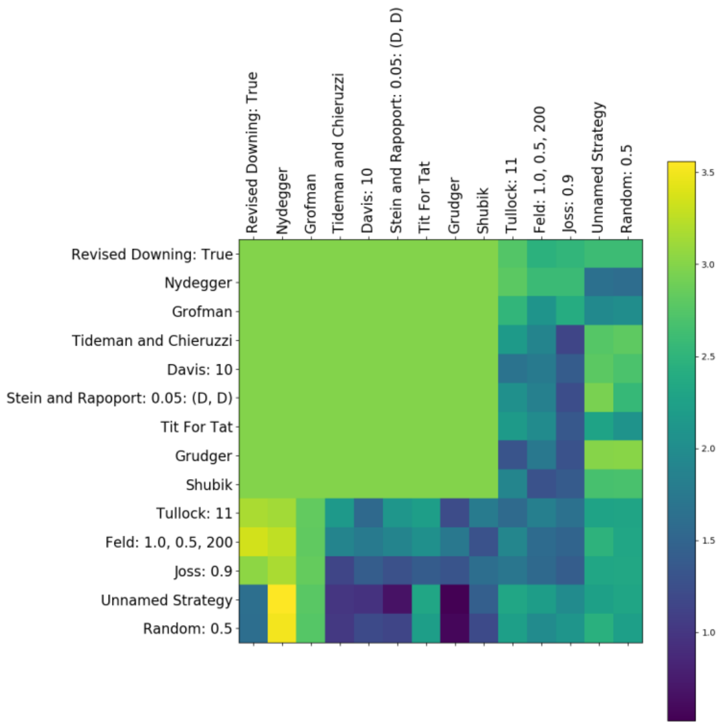

Fig. 1. Payoff matrix of Axelrod tournament with Axelrod's first tournament (except Graaskamp) agents (seed = 1143728).

The original Downing agent defects on the first two rounds, while the intended action is actually to cooperate on the first two rounds. This resulted in the Downing agent losing trust from many agents that have retaliating properties (e.g. Shubik, Grofman, and Grudger). The revised agent allows it to maintain trust with these inherently-retaliating agents, while maximizing long term play using conditional probabilities of the opponent cooperating given their action in the previous round.

The lack of Graaskamp agent also seems to shake up the result quite significantly. Graaskamp cooperates for the first fifty rounds, does defects on the 51st round, and play Tit-for-Tat for the rest of the game. However, the agent defects until the end if the opponent is determined to play randomly. In this case, strategies that play similarly to Tit-for-Tat would have the biggest advantage playing with the Graaskamp agent. The lack of Graaskamp agent in this scenario impacts the agents with nice property like Tit-for-Tat most significantly. The idea of a "kingmaker" (an agent that helps a particular agent significantly) has been alluded by Axelrod [4].

Based on Figure 1, we can also see that the average payoff from the top-performing strategies are completely homogenous. This shows that the nice agents are benefitting from each other. The only difference between the top performers lies in their interactions with the bottom scorers. The bottom performers all have the tendency to defect without any reason. We can see that the Revised Downing agent in overall managed to adapt better to these bottom performers' defection habits. We hypothesize that an adaptive agent can perform even better in this condition.

Figure 2 shows one of the common payoff results of the Axelrod tournament using all agents from the Axelrod's first tournament, except the Graaskamp agent, with a 0.05 noise. The results show a drastically different results from the first scenario, where no mistakes occur. This shows that an environment containing nice and retaliating agents can still devolve into an environment full of distrust even when only little mistakes occurs.

On 20 cases, Revised Downing and Davis won 7 times, followed by Grudger 4 times and Nydegger 2 times. We find that Revised Downing and agents that tend to retaliate heavily (like Grudger and Davis) dominated in an environment with small mistakes. Another observation that happens consistently is the fact that agents' interactions with Nydegger strategy mostly ends up in bright spots, implying that almost everyone benefits from betraying Nydegger. Grofman also consistently shows a brighter payoff column, implying benefits from betrayal like Nydegger's, though not as much as Nydegger's. We conclude that in this scenario, Nydegger and Grofman end up being the kingmaker of the top performers.

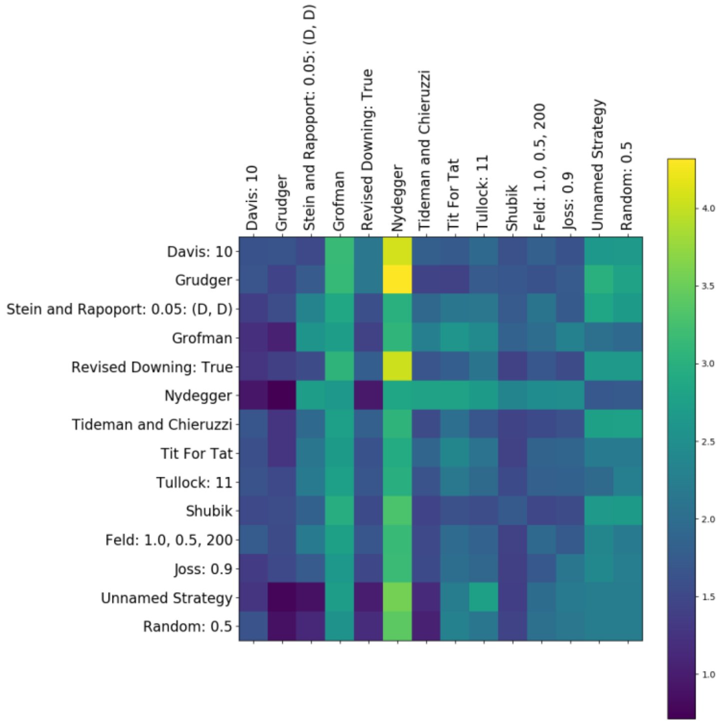

Fig. 2. Payoff matrix of Axelrod tournament with Axelrod's first tournament agents (except Graaskamp) with 0.05 mistake probability (seed = 4750670).

Figure 3 shows one of the common payoff results of the Axelrod tournament using all agents from the Axelrod's first tournament, except the Graaskamp agent, with a 0.05 noise and mistake bias enabled. Based on 20 cases, Nydegger won 9 times; Revised Downing 5 times; Grudger 4 times; Davis 2 times. Surprisingly, despite being exploited the most often by the other agents, Nydegger ends up being the most consistent winner in this scenario. Based on the payoff matrix, we can see that Nydegger mostly ends up having better payoff with the rest of the agents, while the other agents have lower payoff with each other. Nydegger's "cooperative but not gullible" methods stand out when mistake bias is enabled in this scenario.

Figure 4 shows one of the common payoff results of the Axelrod tournament using all agents from the Axelrod's first tournament, except the Graaskamp agent, with a 0.2 noise. In this scenario, out of 20 cases, Grudger won 11 times, followed by Revised Downing 7 times. There two outlier cases with one win from Davis and Tideman & Chieruzzi each. We can see that the payoff matrix has a consistent color changes now, from dark (top performer) to bright (bottom performer). We can also see that the bottom performers are consistently exploited when the noise is high. The four bottom performers, Grofman, Unnamed Strategy, Nydgger, and Random, consistently end up with the lowest scores in all cases. We can see that these four agents are the kingmaker in this scenario. Adding a mistake bias doesn't change the result significantly, which is shown in Figure 5. In 20 cases of 0.2 mistake with mistake bias, Revised Downing won 8 times; Grudger 6 times; Davis 5 times; Feld 1 time.

Based on the five scenario results, we found that Revised Downing agent is the agent that performs the most consistent across all scenarios, acquiring many wins consistently. Grudger and Davis are the next top performers, especially in an environment with big chance of mistakes happening. Surprisingly, Tit-for-Tat almost never managed to place on the top half, signifying that there is a kingmaker in Axelrod's original tournament, which turned out to be Graaskamp. The lack of a kingmaker for Tit-for-Tat drastically reduced its performance. This is similar to Rapoport's critique of the original tournament [11].

2) Axelrod Tournament Result with Axelrod's First Tournament Agents

For Axelrod's second tournament results, we will not analyze the underlying pattern due to the sheer number of agents and the time limit of the research. However, we will show the obvious trends that happen in the results. Further analysis of these results can be done in the next studies.

Figure 6 shows one of the most common the payoff results for the Axelrod tournament using agents from the Axelrod's second tournament. Out of 20 cases, Borufsen wins 16 times, followed by More Tideman & Chieruzzi and Harrington 2 times each. As with the previous list of agents, we can see that the first 32 agents have an almost consistent mutual cooperation color. We can infer that almost all of the agents are nice by default and only defects in retaliation.

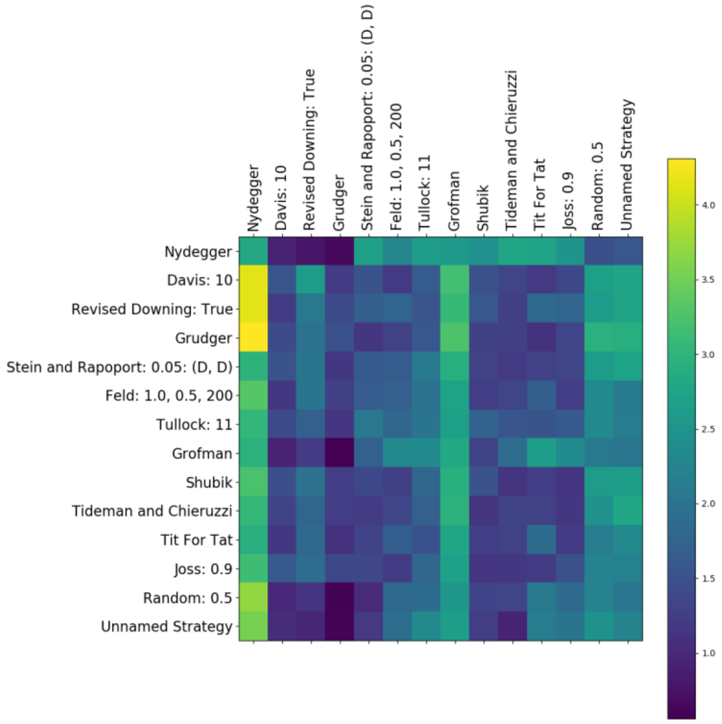

Fig. 3. Payoff matrix of Axelrod tournament with Axelrod's first tournament agents (except Graaskamp) with 0.05 mistake probability and mistake bias (seed = 901236547).

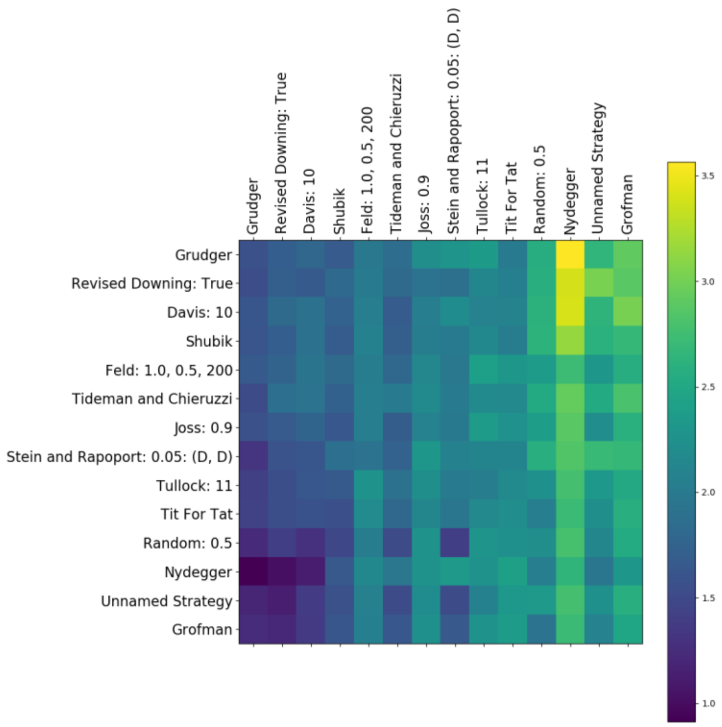

Fig. 4. Payoff matrix of Axelrod tournament with Axelrod's first tournament agents (except Graaskamp) with 0.2 mistake probability (seed = 291358764).

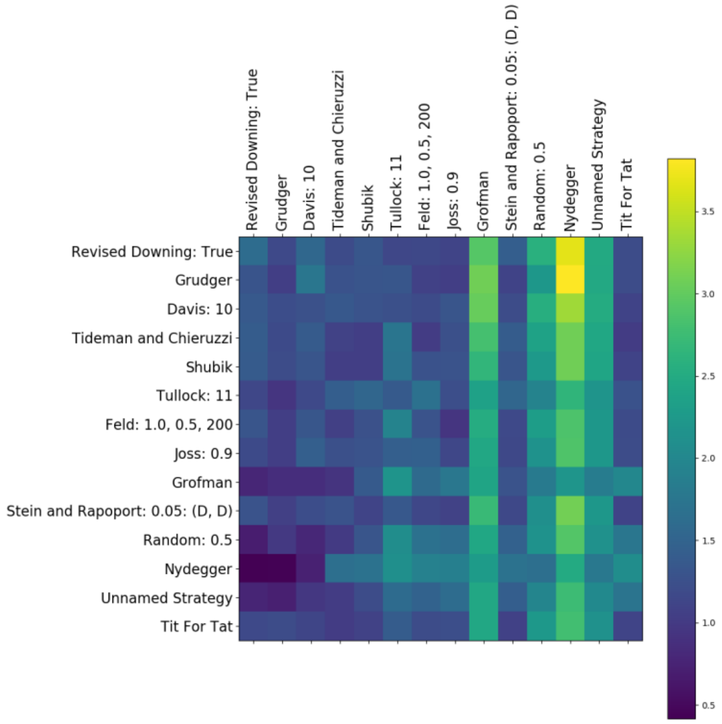

Fig. 5. Payoff matrix of Axelrod tournament with Axelrod's first tournament agents (except Graaskamp) with 0.2 mistake probability and mistake bias (seed = 771923).

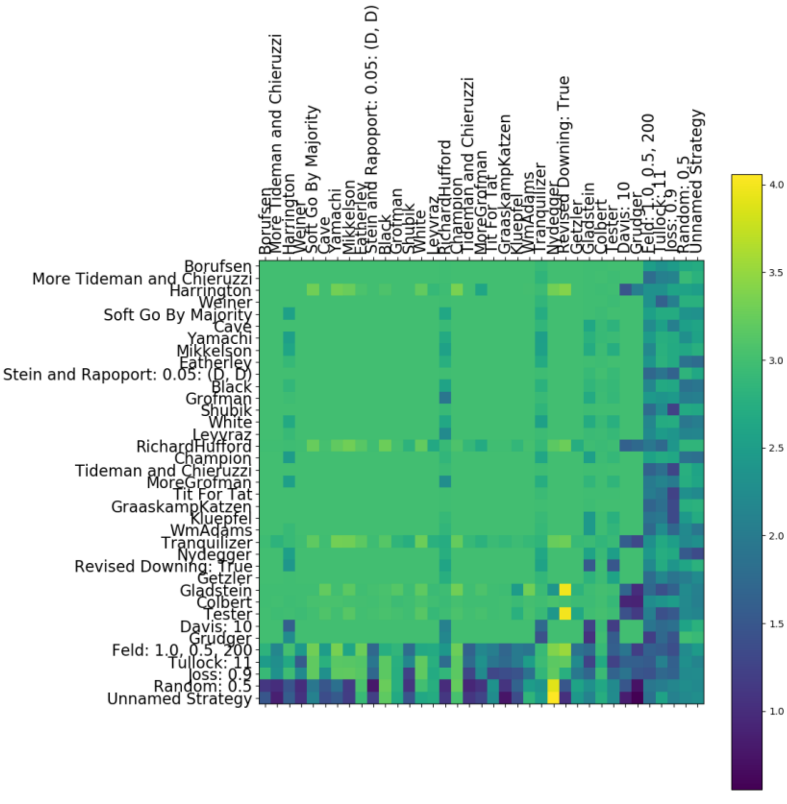

Fig. 6. Payoff matrix of Axelrod tournament with Axelrod's second tournament agents (seed = 5665363).

Figure 7 shows one of the most common the payoff results for the Axelrod tournament using agents from the Axelrod's second tournament with 0.05 mistake probability. Mikkelson won 7 times out of 20 times, followed by Yamachi 5 times, Soft Go by Majority and Tranquilizer 3 times, and finally White and Eatherley 1 time each. We can still see some kind of color degradation like the previous scenario, which means that the top performing agents still managed to cooperate with each other, despite the small mistakes that may happen. All the top five strategies (Mikkelson, Yamachi, White, Eatherley, and Soft Go by Majority) share the same property of forgiveness.

Likewise, the next scenario approximately shows the same result as the previous one. Figure 8 shows one of the most common the payoff results for the Axelrod tournament using agents from the Axelrod's second tournament with 0.05 mistake probability and mistake bias enabled. Yamachi won 10 times out of 20 cases, followed by Eatherley 5 times, Soft Go-by-Majority 4 times, and Tranquilizer 2 times.

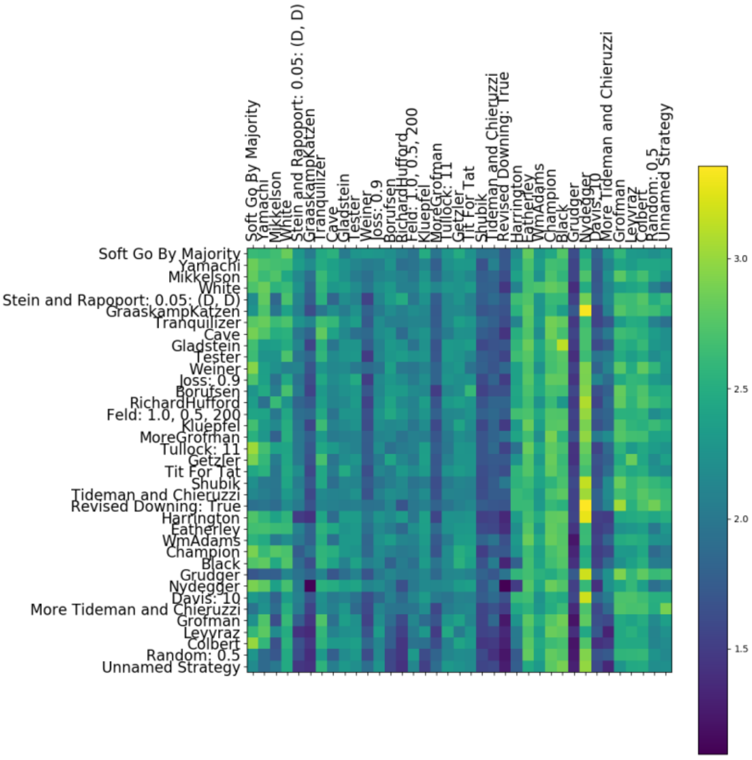

Fig. 7. Payoff matrix of Axelrod tournament with Axelrod's second tournament agents with 0.05 mistake probability (seed = 135792468).

Figure 9 shows one of the most common the payoff results for the Axelrod tournament using agents from the Axelrod's second tournament with 0.2 mistake probability. Out of 20 cases, Soft Go-by-Majority wins 8 times, followed by Yamachi 4 times and Mikkelson 3 times. There are also four outlier cases with one win each from White, Richard Hufford, Graaskamp Katzen, and Champion. We can see that the payoff with same colors are often clustered together in one column. This means that the interactions that happened between a pair of agents mostly resolve with one of the two outcomes: mutual cooperation or mutual defection.

Figure 10 shows one of the most common the payoff results for the Axelrod tournament using agents from the Axelrod's second tournament with 0.2 mistake probability and mistake bias enabled. Soft Go-by-Majority wins 16 times out 20 times, followed by Eatherley 3 times and Champion 1 time. We can see that the one column payoff color clustering is even more apparent in this scenario. There are only several agents that still had bright color, meaning either they have mutual cooperation or the agent got exploited. Some particularly noticeable agents are Nydegger, Colbert, Unnamed Strategy, and Grofman. Nydegger, in particular, consistently managed to score in the top half of the population.

Based on the five scenarios' results, we find that Borufsen performs the best in a no-mistake environment but performs much more poorly on noisy environment. We also find that Soft Go-by-Majority performs the best when the noise is high, especially with mistake bias activated. The reasoning behind this can be examined in future studies. One other thing we noticed is that Tit-for-Tat strategy never managed to get into the top half of the ranking, which reinforces our conclusion regarding the kingmaker in the previous section.

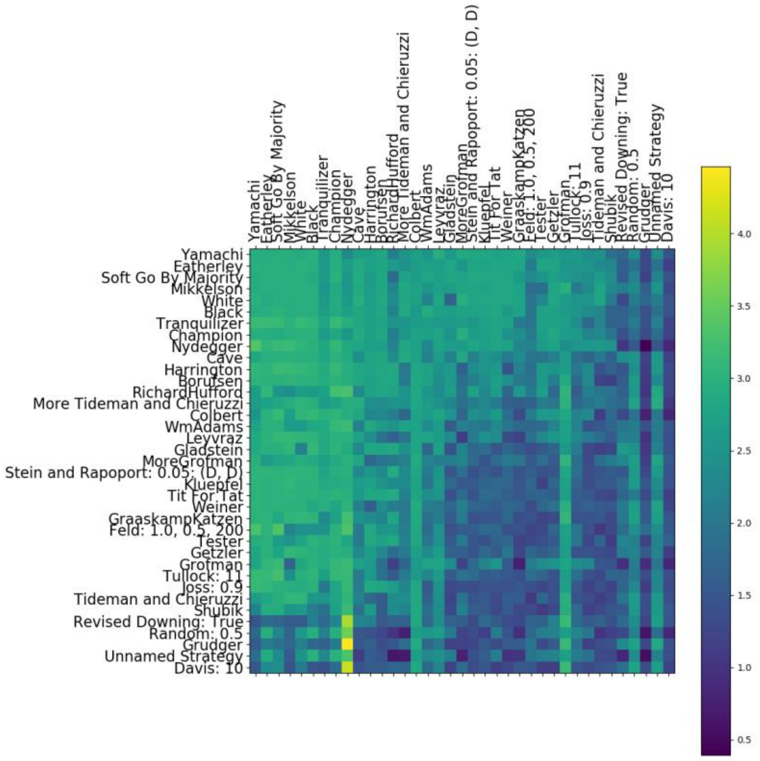

Fig. 8. Payoff matrix of Axelrod tournament with Axelrod's second tournament agents with 0.05 mistake probability and mistake bias (seed = 2231145).

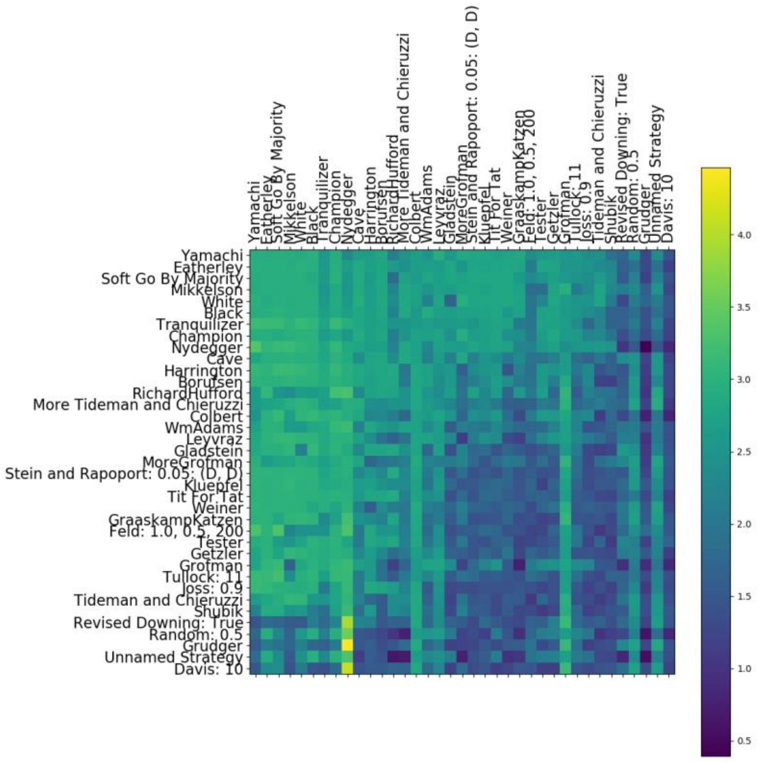

Fig. 9. Payoff matrix of Axelrod tournament with Axelrod's second tournament agents with 0.2 mistake probability (seed = 2231145).

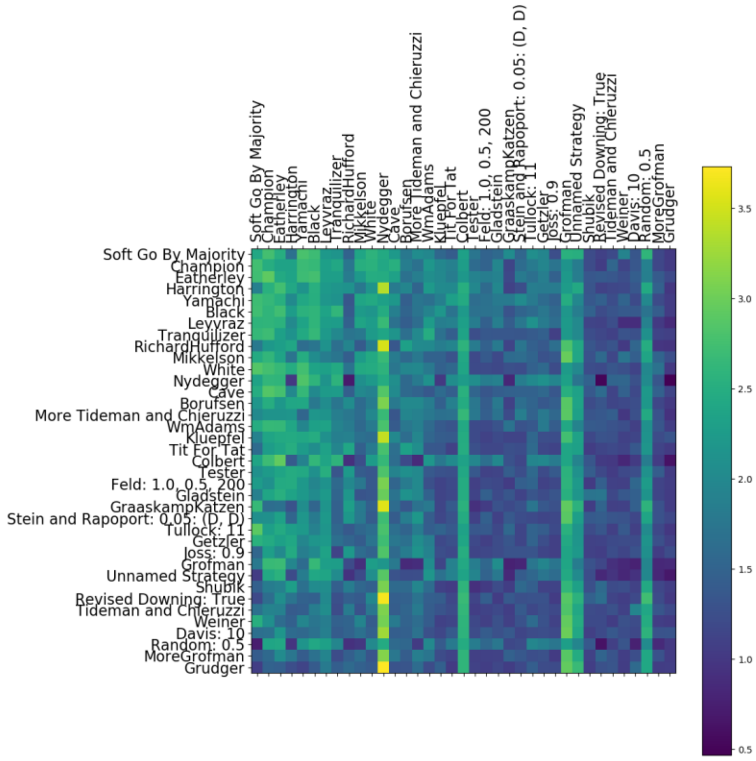

Fig. 10. Payoff matrix of Axelrod tournament with Axelrod's second tournament agents with 0.2 mistake probability and mistake bias (seed = 1).

3) Axelrod Tournament Result with Case's Simulation Agents

Figure 11 shows the payoff matrix of the Axelrod tournament using Case's simulation agents. The simulation almost consistently shows the following placement in all seeds, starting from the first place: Copycat > Grudger > Simpleton > Copykitten > Detective > Cheater > Random 0.5 > Cooperator. Like in the first Axelrod tournament, we can see that the top four agents constantly cooperate with each other, resulting in a higher payoff compared to the low performers. Tit-for-Tat in particular is able to deflect against the Defector, Detective's initial betrayal, and Random better than the other agents. Based on Axelrod's original conclusion, we can easily predict Tit-for-Tat's constant win in this scenario.

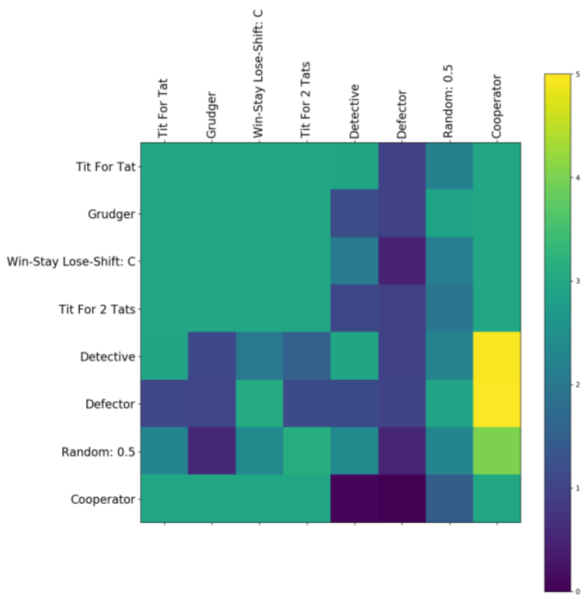

Fig. 11. Payoff matrix of Axelrod tournament with Axelrod's second tournament agents with 0.2 mistake probability and mistake bias (seed = 1).

When mistake chance is included, the results immediately converge into one of the two results, which ends with either Defector or Grudger winning. Both agents always dominate the ranking in all cases. Out of 80 cases, 62 of them were Defector's win. Even the smallest mistake chance causes the Defector to reign over others. Figure 12 a representative of these results.

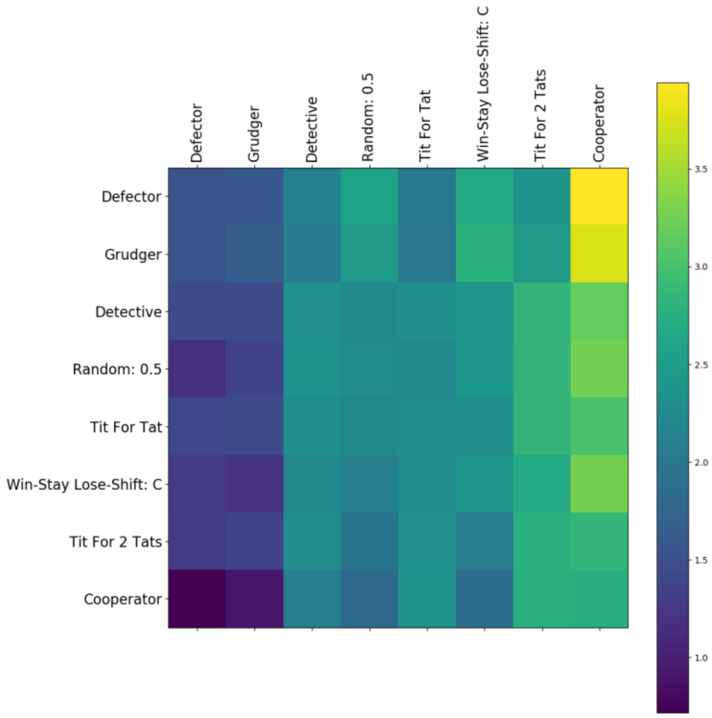

Fig. 12. Payoff matrix of Axelrod tournament with Case's simulation agents with 0.05 or 0.2 mistake probability and/or mistake bias (seed = 223145).

4) Axelrod Tournament Result with Steward and Plotkin's Tournament Agents

Figure 13 shows one of the most common results of the Axelrod tournament using Stewart and Plotkin's simulation agents. The wins are dominated by both ZD-GTFT-2 (18 wins) and GTFT (2 wins). The result is mostly the same with the original result from Stewart and Plotkin's simulation [9]. The Detective agent, despite having a tendency to exploit the other agents, managed to place in the middle of the ranking. This may be because of the fact that the entire population forces the Detective to stay in its Tit-for-Tat mode. The fact that Extort-2 and Defector ended up in the bottom also reinforces the conclusion that nice Zero-Determinant strategies helped foster cooperation. This result reinforces everything that has been concluded in Axelrod's, Case's and Stewart & Plotkin's simulation.

Figure 14 shows one of the most common results of the Axelrod tournament using Stewart and Plotkin's simulation agents with 0.05 mistake probability. ZD-GTFT-2 still manages to win the most cases (12 out of 20), followed again by GTFT 5 times. The two outlier cases are by Hard Prober (2 times) and Detective (1 time). The results in this scenario reinforces what we have learned about zero-determinant strategies. Figure 15 shows similar results, using 0.05 mistake probability and mistake bias. However, the number of wins is surprising than the previous two scenarios. GTFT wins 15 times out of 20 cases, while ZD-GTFT-2 only wins 5. We haven't been able to find a reason on why this phenomenon happens, but we hypothesize that it is caused by the fact that GTFT and ZD-GTFT-2 are always neck-to-neck in their scores.

Figure 16 shows one of the most common results of the Axelrod tournament using Stewart and Plotkin's simulation agents with 0.2 mistake probability. Hard Prober wins the most in 8 cases, followed by Defector 5 times, and Detective 3 times. Three outlier cases happen with one win each from Hard Tit-for-Tat, Tit-for-Tat, Calculator, and Prober 3. The results shown in this scenario is radically different from the previous three scenarios. GTFT and ZD-GTFT-2 strategies are nowhere to be seen in the top four performers.

Figure 17 shows one of the most common results of the Axelrod tournament using Stewart and Plotkin's simulation agents with 0.2 mistake probability and mistake bias. The result is dominated by GTFT with 18 wins out of 20 cases, followed by both ZD-GTFT-2 and Random with 1 win each. Surprisingly, the GTFT managed to have a big win margin against ZD-GTFT-2 in this scenario. The contrast difference with the previous scenario comes from the fact that adding mistake bias adds more "certainty" that agents are likely to defect. GTFT in this case is superior compared to Tit-for-Tat or Tit-for-Two-Tats because it's stochastically forgiving. The competition between GTFT and ZD-GTFT-2 can be used as an analysis material for further studies.

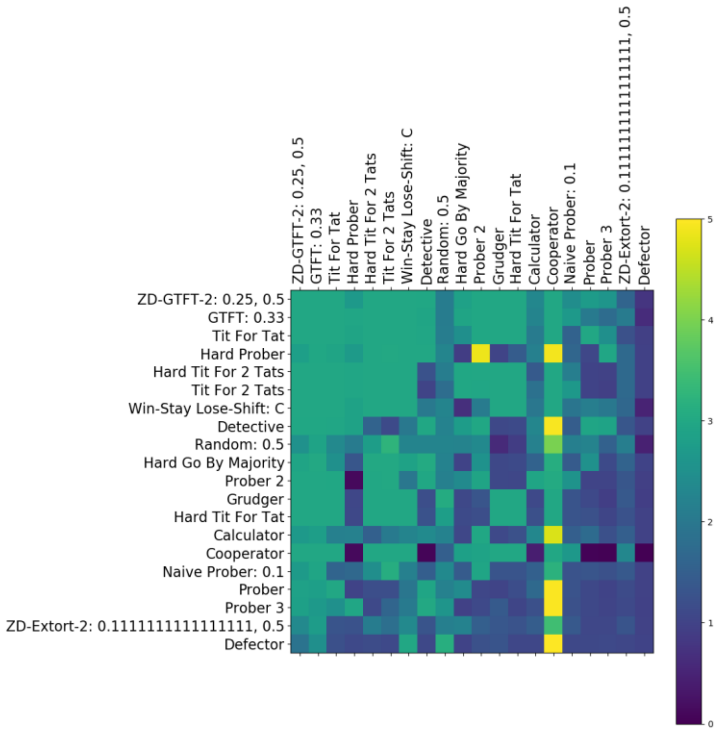

Fig. 13. Payoff matrix of Axelrod tournament with Steward-Plotkin's tournament agents (seed = 771923).

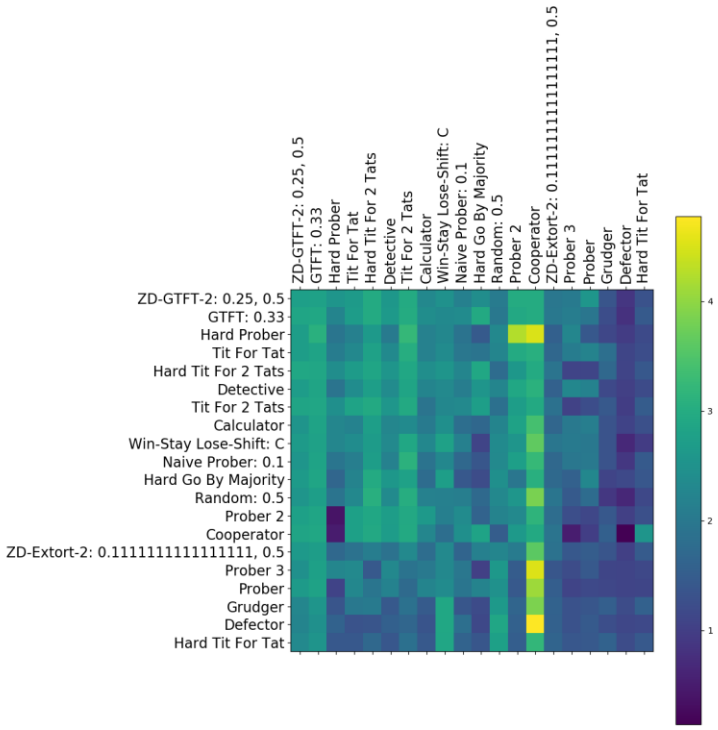

Fig. 14. Payoff matrix of Axelrod tournament with Steward-Plotkin's tournament agents with 0.05 mistake probability (seed = 1).

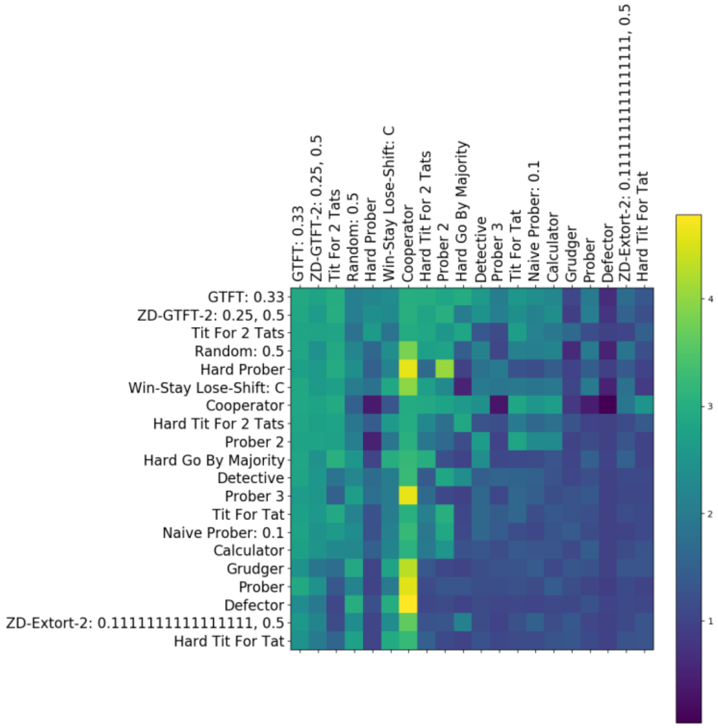

Fig. 15. Payoff matrix of Axelrod tournament with Steward-Plotkin's tournament agents with 0.05 mistake probability and mistake bias (seed = 9100021).

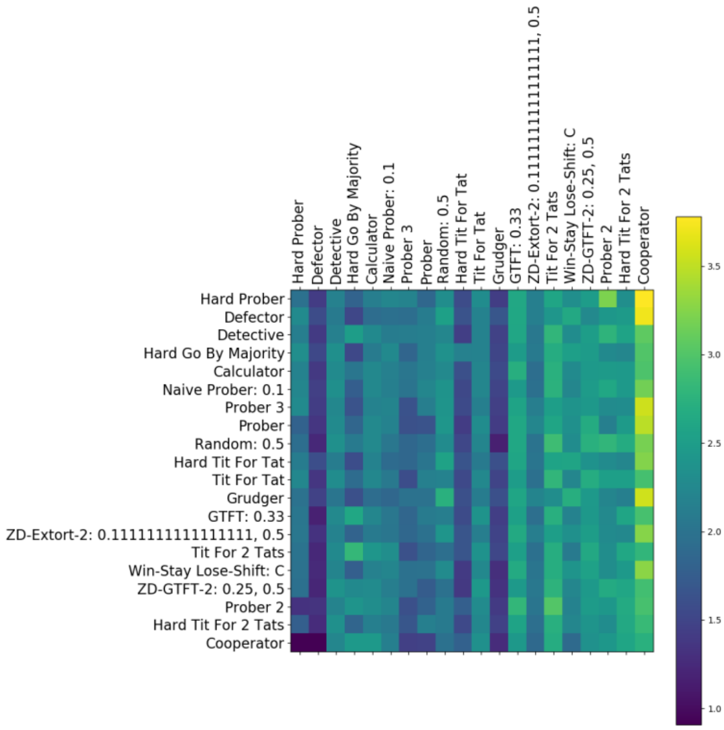

Fig. 16. Payoff matrix of Axelrod tournament with Steward-Plotkin's tournament agents with 0.2 mistake probability (seed = 19192831).

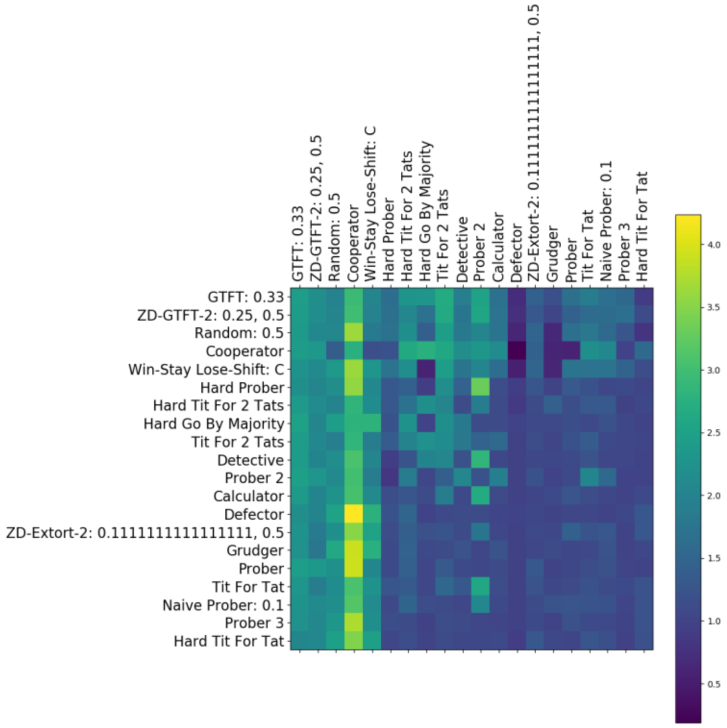

Fig. 17. Payoff matrix of Axelrod tournament with Steward-Plotkin's tournament agents with 0.2 mistake probability and mistake bias (seed = 771923).

5) Axelrod Tournament Result with Best Agents

Figure 18 shows the payoff matrix of the Axelrod tournament using the best agents based on the criteria we discussed in the previous section. ZD-GTFT-2 managed to win the most out of 20 cases with 8 wins, followed by Cave and Harrington with 4 wins each, and finally by Tit-for-Tat and Eatherley with 2 wins each. As expected, in a deterministic environment, zero-determinant strategies managed to dominate better than other types of strategies.

Figure 19 shows the payoff matrix of the Axelrod tournament using the best agents with 0.05 mistake probability. The result is surprisingly different than expected, with 17 wins by Tranquilizer, 2 wins from Cave, and 1 win from Harrington. The sudden domination by Tranquilizer is unprecedented. The same case surprisingly also happens when mistake bias is introduced, shown in Figure 20, with Tranquilizer taking over 18 wins and Harrington taking 2 wins. We haven't managed to find a concrete reason for these phenomena at the time of the writing of this paper.

Figure 21 shows the payoff matrix of the Axelrod tournament using the best agents with 0.2 mistake probability. The results show an even more different pattern than the previous three scenarios. Defector takes over with 12 wins, followed by Revised Downing with 5 wins and three outlier cases with one win from Hard Prober, Naïve Prober, and Tranquilizer each. Adding mistake bias also changes the result drastically, which is shown in Figure 22. Harrington wins 13 times, followed by Eatherley with 5 wins and Champion with 2 wins.

We suspect that the presence of zero-determinant strategies is able to change every other agent's scores drastically. Almost all of the results in the previous sections show a consistent result, even with the addition of mistake chance and mistake bias. However, we haven't been able to find a concrete reason for the sudden changes of these results only based on the mistake probability and the bias.

B. Case's Simulation Results

We now show in detail the results of the Case's simulation based on the scenarios we proposed in the previous section.

1) Case's Simulation Result with Axelrod's First Tournament Agents

We show the results of the Case's simulations using Axelrod's first tournament agents in this section. Figure 23 shows the population plot when no mistake happened. We can see that every strategy that defects without any reason (Random, Unnamed Strategy, Tullock, Joss, and Feld) got replaced before the end of the simulation. The Revised Downing strategy also has the most number of agents at the end. Both of these phenomena happen consistently across all seeds.

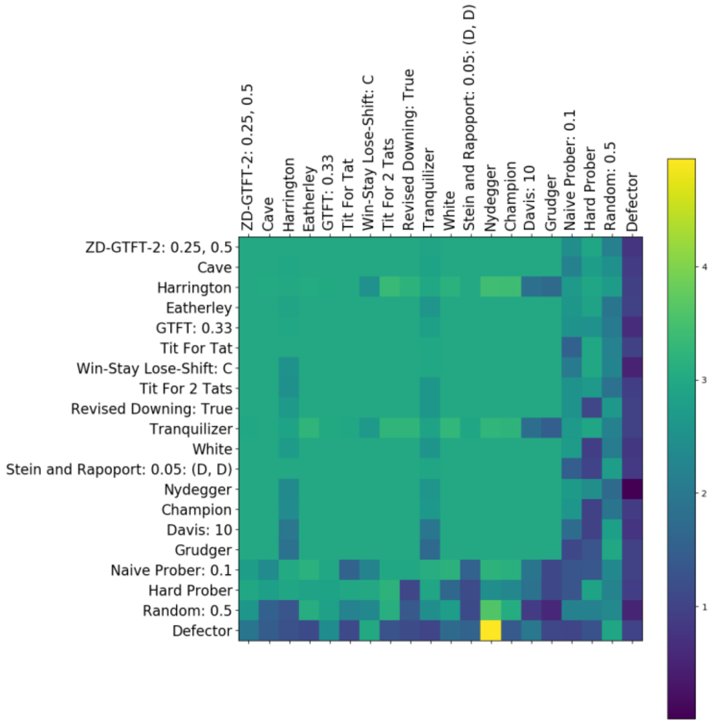

Fig. 18. Payoff matrix of Axelrod tournament with best agents (seed = 1).

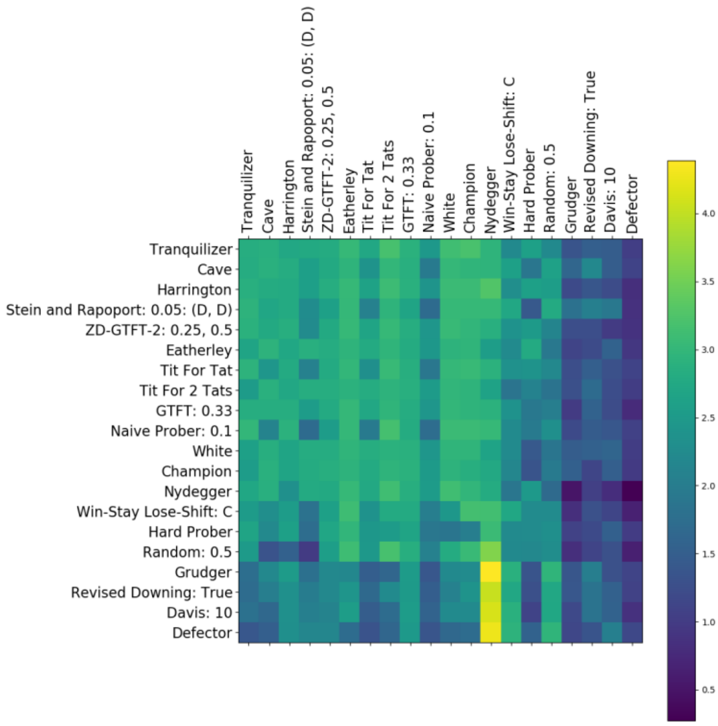

Fig. 19. Payoff matrix of Axelrod tournament with best agents with 0.05 mistake probability (seed = 918273645).

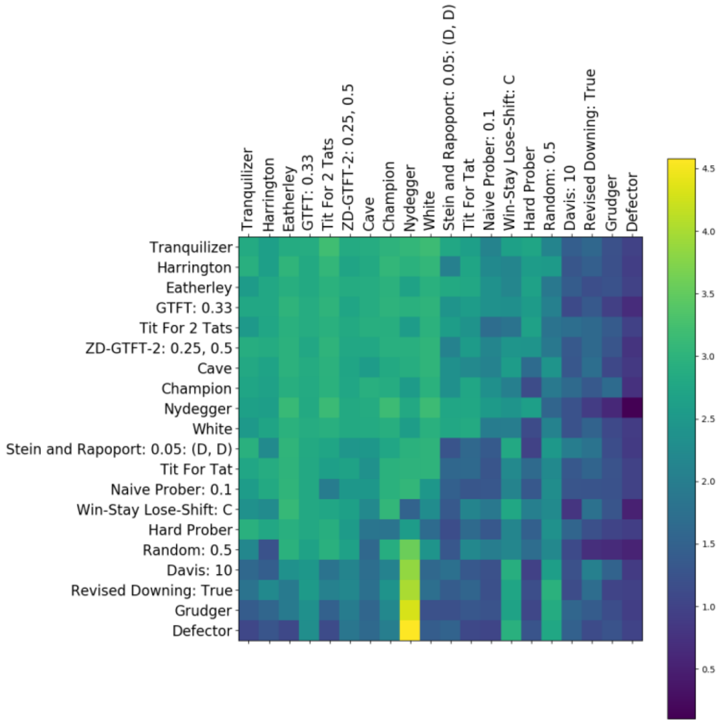

Fig. 20. Payoff matrix of Axelrod tournament with best agents with 0.05 mistake probability and mistake bias (seed = 135792468).

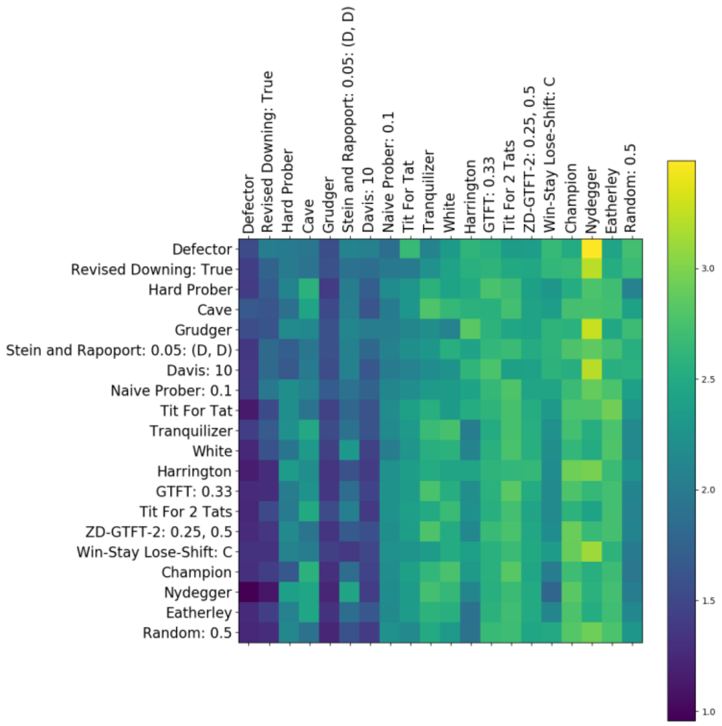

Fig. 21. Payoff matrix of Axelrod tournament with best agents with 0.2 mistake probability (seed = 2231145).

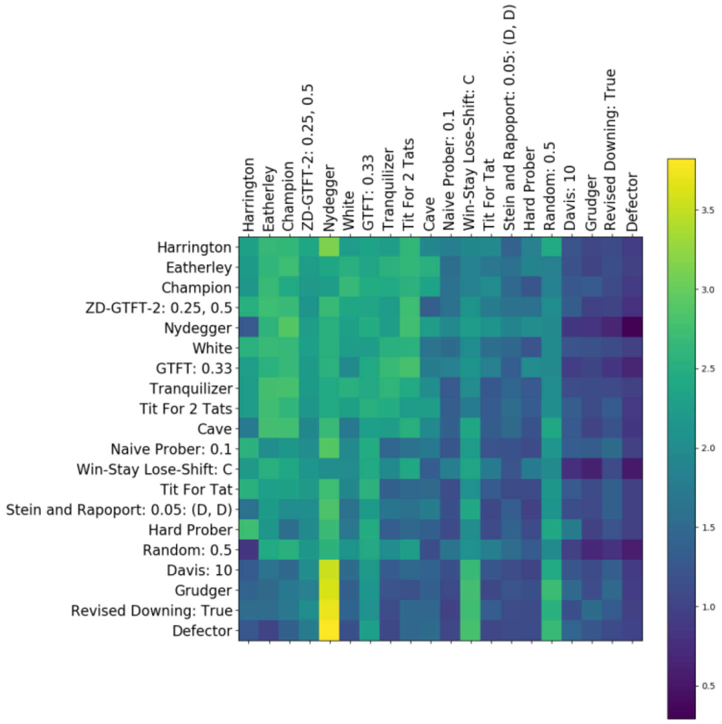

Fig. 22. Payoff matrix of Axelrod tournament with best agents with 0.2 mistake probability and mistake bias (seed = 83281327).

Based on what we know from the Axelrod tournament results, it is obvious that agents that do non-retaliating defections always got heavy retaliations from the other agents (especially those with Grudger-like tendency). This causes them to lose major benefits from any future cooperation. The results also reinforce the Revised Downing strategy's effectiveness for being adaptive in this particular scenario. One thing to note is that the environments always managed to become homogenously cooperative by the sixth match. We can easily predict this particular outcome by using the payoff chart we provided from the previous section.

The results become more mixed when mistakes are involved. Figure 24 shows the most common population plot results from the simulation with a mistake chance of 0.05. Based on our experiment results, Stein and Rapoport won 9 times out of 20; Tit-for-Tat and Grudger 5 times out of 20; Tideman and Chieruzzi 1 out of 20. Random strategy also managed to have a significant number of agents in 2 out of 20 cases. An interesting observation to note is that the Revised Downing strategy almost never managed to survive to the end of the simulation (only 1 out of 20 cases). We also note that strategies that perform deterministically tend to perform better than the stochastic ones (all top four strategies are deterministic by nature).

Figure 25 shows the most common population plot results from the simulation with a mistake chance of 0.05 and mistake bias added. The Grudger strategy performs the best, winning 8 out of 20 cases, followed by Davis and Stein & Rapoport (3 out 20 cases). We can see that strategies that heavily retaliates win most of the games. However, an interesting thing to consider is that Stein & Rapoport is basically a Tit-for-Tat strategy that retaliates when it determines its opponent to be playing randomly, which, despite not being a retaliator, ended up winning 3 times out of 20 cases. We have not managed to find a reason on how this result occurs.

Figure 26 shows the most common population plot results from the simulation with a mistake chance of 0.2. As expected, Grudger managed to survive with the biggest number of agents almost consistently (17 out of 20 cases). Surprisingly, the Revised Downing agent, which performed abysmally when the mistake chance is low (0.05), performed much better in this setting. The strategy even managed to win 4 out of 20 times and trailed behind the Grudger strategy in terms of agent number 10 out of 20 times. From these results, we hypothesize that when mistake percentage is involved, adaptive strategies perform better the higher the chance of mistake. In other words, bigger mistake causes more certainty, which makes a better "learning data" for adaptive strategies.

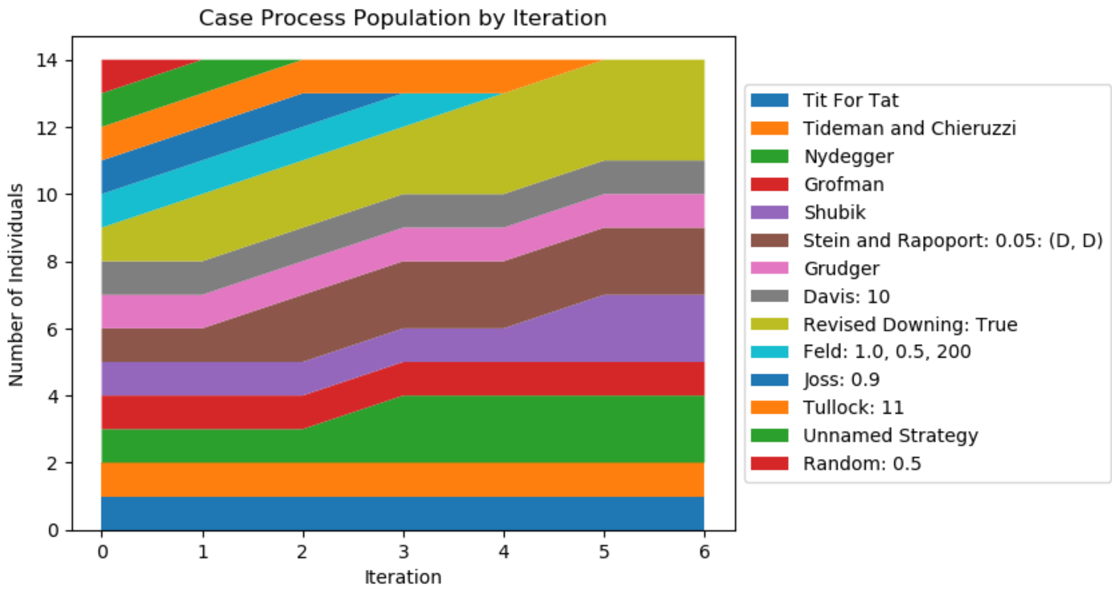

Fig. 23. Population plot of Case's simulation with the agent population from the Axelrod's first tournament (seed = 83281327).

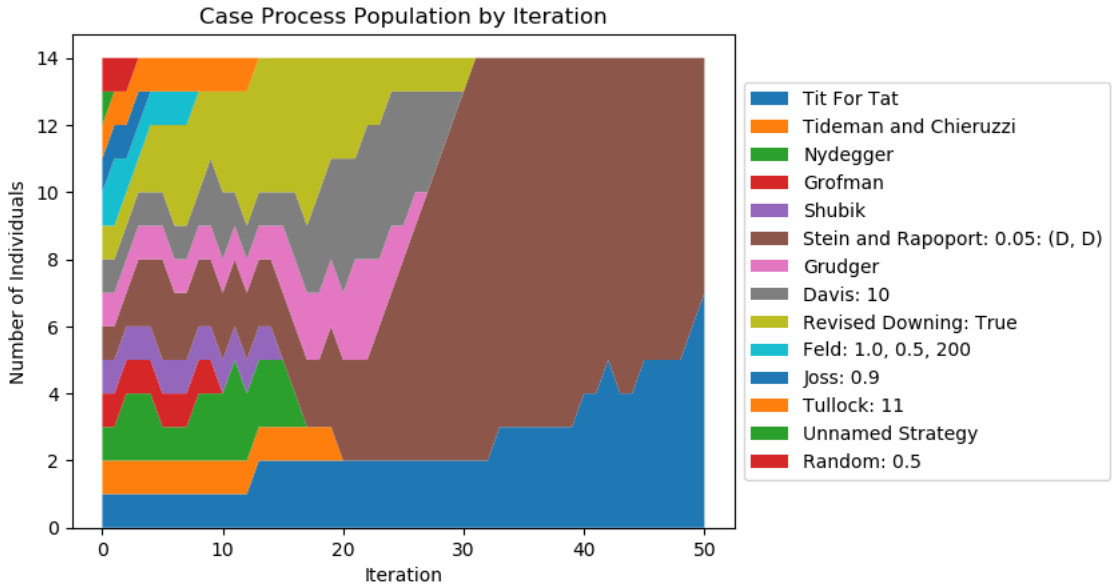

Fig. 24. Population plot of Case's simulation with the agent population from the Axelrod's first tournament with 0.05 mistake chance (seed = 918273645).

The addition of mistake bias seems to reinforce the hypothesis from the previous paragraph. Figure 27 shows the most common population plot results from the simulation with a mistake chance of 0.2 and mistake bias enabled. The Revised Downing strategy performs much better than the other strategies, winning 9 times out of 20 cases. Grudger and Tideman & Chieruzzi trailed behind with 4 wins out of 20 cases. We conclude this section by stating these hypotheses:

1. Strategies with an adaptive property perform better when there is no noise (mistake) or when the chance to do mistake is high. Small uncertainty significantly lowers this type of strategy's performance.
2. In an environment with high chance of mistake, agents that heavily retaliate to defections and perform deterministically have better performances.

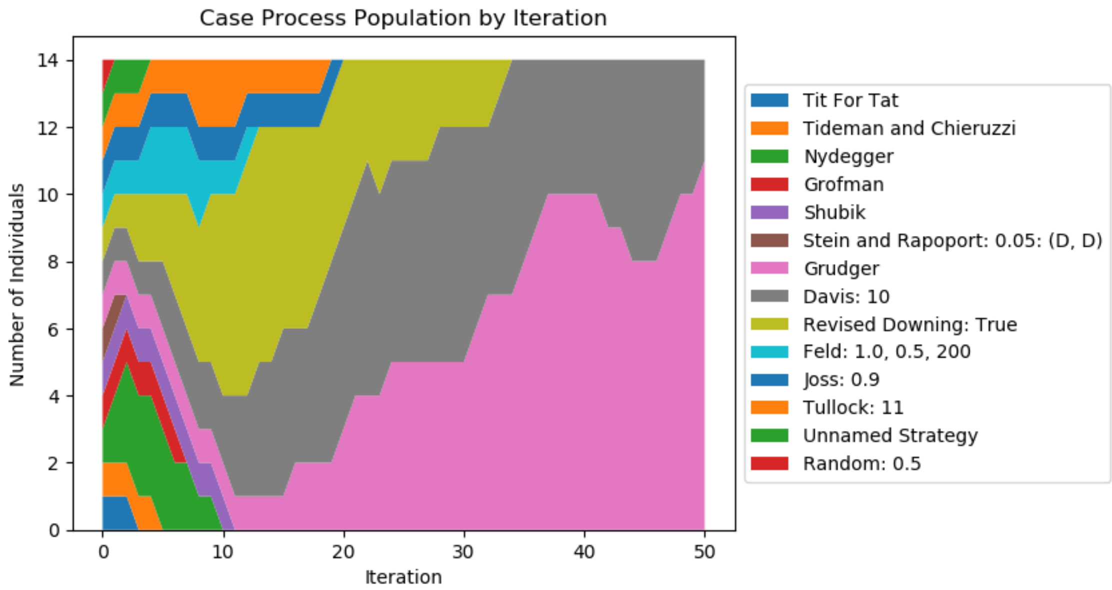

Fig. 25. Population plot of Case's simulation with the agent population from the Axelrod's first tournament with 0.05 mistake chance and mistake bias (seed = 2231145).

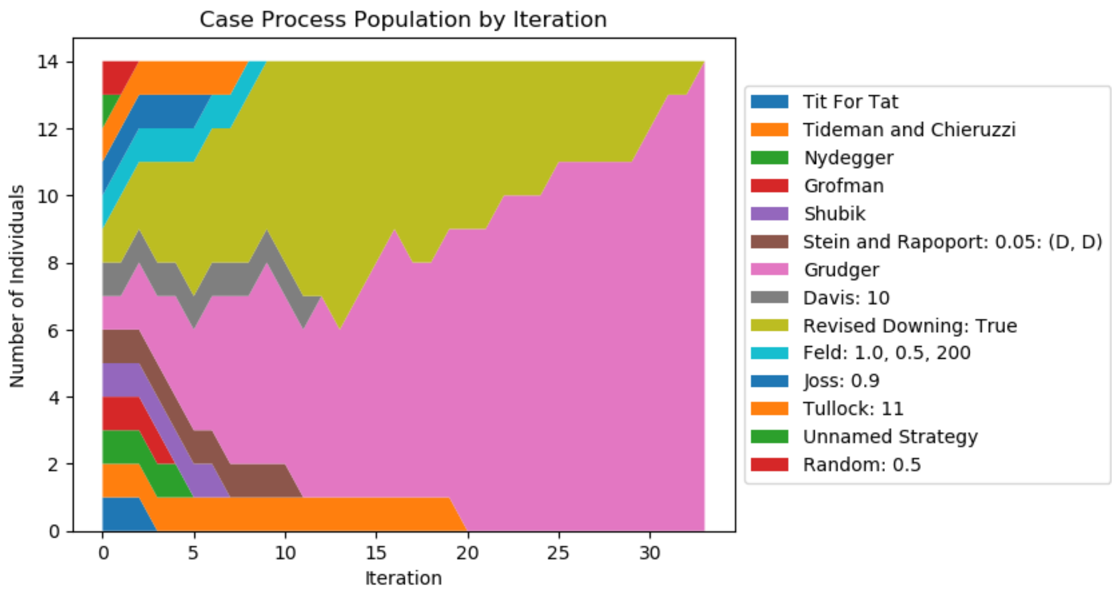

Fig. 26. Population plot of Case's simulation with the agent population from the Axelrod's first tournament with 0.2 mistake chance (seed = 34293471).

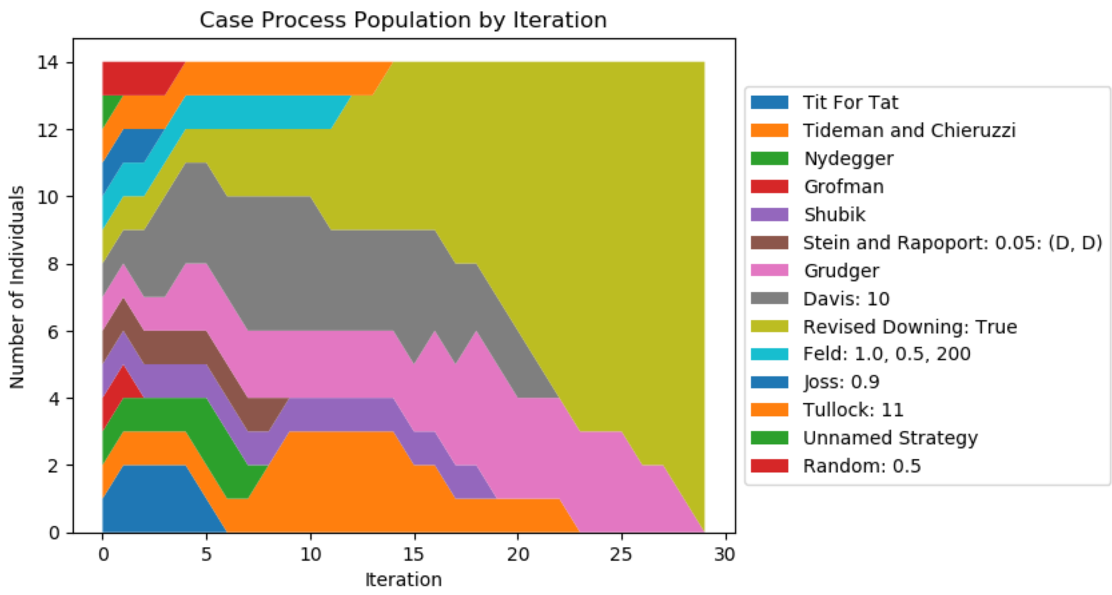

Fig. 27. Population plot of Case's simulation with the agent population from the Axelrod's first tournament with 0.2 mistake chance and mistake bias (seed = 1143728).

2) Case's Simulation Result with Axelrod's Second Tournament Agents

In this section, we show the results of the Case's simulations using Axelrod's second tournament agents. Like in the Axelrod tournament section, we will not analyze the underlying pattern due to the sheer number of agents and the time limit of the research. However, we will show the obvious trends that happen in the results. Further analysis of these results can be done in the next studies.

Figure 28 shows the population plot of Case's simulation using Axelrod's second tournament agents with no mistakes happening. The results ended up being radically different from the ones shown in the first Axelrod tournament. The simulation consistently ended up being dominated either by Tester or Gladstein strategy with equal percentage (10 wins each). Harrington also consistently held out behind the two top strategies, dominating half of the simulation before ending up in the minority. Meanwhile, the Revised Downing strategy, which we touted as the best strategy in the previous section, shows bad performances, never reaching the end of the simulation.

The most surprising thing from the results is the fact that Gladstein and Tester escaped the consequences of defecting in the beginning. Both Tester and Gladstein always defect at the beginning. If the opponent ever defects, Tester will apologize by cooperating and then plays Tit-for-Tat for the rest of the game. Otherwise, Tester will alternate between cooperation and defection. Gladstein also apologizes after the first defection. However, Gladstein will keep defecting unless its own cooperation rate so far is below 0.5, in which case it will always cooperate.

Figure 29 shows one of the most common population plot results of Case's simulation with 0.05 mistake chance. The Harrington strategy almost consistently dominates the population (19 out of 20 cases), with Tranquilizer strategy constantly trailing behind or winning in 1 case. If you look at the payoff chart for this particular scenario, you can see that Harrington consistently placed on the bottom half of the ranking. We can see that there are some agents that reduce Harrington's scores significantly: Graaskamp Katzen and Revised Downing. We can see that when both of these agents are replaced by any other agent, the Harrington strategy managed to keep winning and replace the lowest scorers. However, Harrington can still manage to survive even when both agents survive long enough (both of them almost never reach the end).

Based on its description, we can describe the Harrington strategy as a Tit-for-Tat that transform into adaptive GTFT after 37 rounds. We hypothesize that being stochastically forgiving is a good trait to have in an evolutionary trust environment. We will show a further evidence of this hypothesis in the next section with the Stewart-Plotkin results.

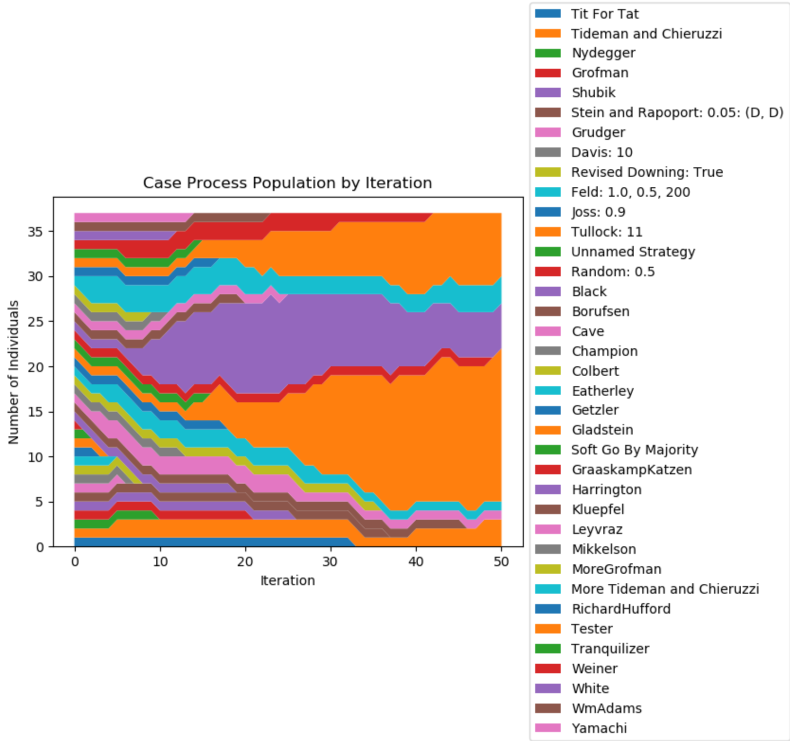

Fig. 28. Population plot of Case's simulation with the agent population from the Axelrod's second tournament (seed = 1143728).

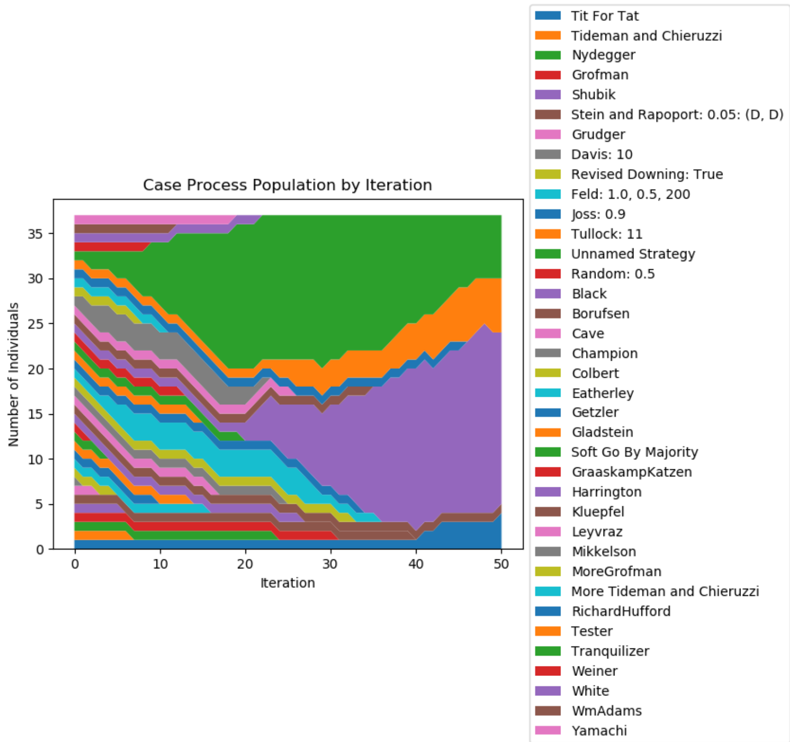

Fig. 29. Population plot of Case's simulation with the agent population from the Axelrod's second tournament with 0.05 mistake probability (seed = 2231145).

Figure 30 shows one of the most common population plot results of Case's simulation with 0.05 mistake chance and mistake bias. The Harrington strategy, again, also almost consistently dominates the end of the simulation (19 out of 20 cases), with Richard Hufford trailing behind with 1 win. We find that adding mistake bias doesn't really affect Harrington's performance in a low mistake environment.

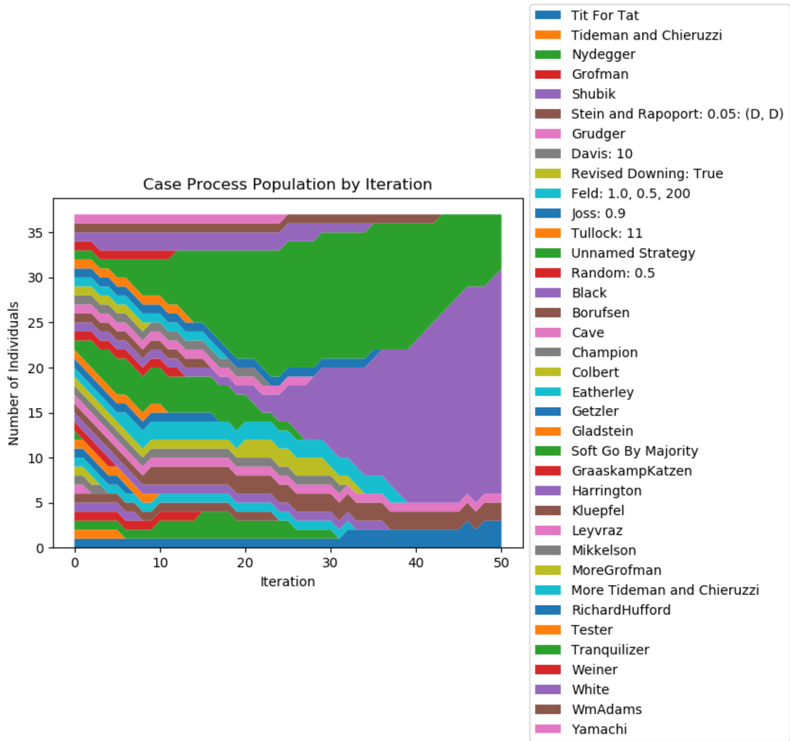

Fig. 30. Population plot of Case's simulation with the agent population from the Axelrod's second tournament with 0.05 mistake probability and mistake bias (seed = 43238133).

We did not show the result for the scenario with 0.2 mistake chance, because we found no obvious patterns from the 20 cases. The results are erratic, to say the least, with no obvious agent majorly winning through the simulation. We suspect that high mistake rate for this particular population causes them to act similarly with Random agent.

Figure 31 shows one of the most common population plot results of Case's simulation with 0.2 mistake chance and mistake bias enabled. We can see that the addition of mistake bias reduced the erratic behavior of the agents significantly. White wins the most (13 cases out of 20), with Harrington trailing behind with 4 wins, and with outlier cases from Mikkelson, Black, and Eatherley with 1 win each. White is a strategy that by default cooperates with others, but defects whenever the defection rate of the other agent is larger than the current turn. In an environment with a middle rate of mistakes, White would win most points in each match by cooperating until they deem that their opponent is not trustable anymore.

3) Case's Simulation Result with Case's Simulation Agents

In this section, we show the results of the Case's simulations using Case's simulation agents. Most of the results here are used to ensure the correctness of our simulation, since the results can be easily analyzed, unlike the previous two sections. Figure 32 shows the result of Case's simulation using Case's simulation agents with no mistakes happening. The result consistently shows Cooperator replaced at the beginning, followed by Defector, Detective, and Random getting replaced, before ending in equilibrium with Copycat, Copykitten, Simpleton, and Grudger. Tit-for-Tat also consistently has the most number of agents by the end of the simulation. The results shown here confirms that our implementation of Case's simulation is correct.

Figure 33 shows the result of Case's simulation using Case's simulation agents with 0.05 mistake chance. The result often shows Tit-for-Two-Tats dominating the population (14 out of 20 cases), followed closely by Tit-for-Tat consistently. The 6 other cases show some outlier cases where either one of Detective, Grudger, or Cheater wins. Despite the outlier cases, the main takeaway in this scenario is still the same as the original Case's simulation conclusions.

Figure 34 shows the result of Case's simulation using Case's simulation agents with 0.05 mistake chance and mistake bias. The Random strategy surprisingly managed to win in most cases (9 out of 20 cases), followed by Copykitten with 5 out of 20 cases. Some of the outlier cases are the Copycat and Simpleton strategy winning 2 out of 20 cases, and Detective 1 out of 20 cases. The mistake bias helps Random strategy significantly because it will trigger Grudger's retaliation to every other agent much earlier. This reduces the score gap that would usually happen between Random and the best-performing agents in the regular setting. The Random agent's wins also depend largely on Grudger's survival. If Grudger managed to survive long enough (because the other agents defected too late), Random is guaranteed to be replaced. However, if Grudger is eliminated, it will be able to edge out the other agent's scores.

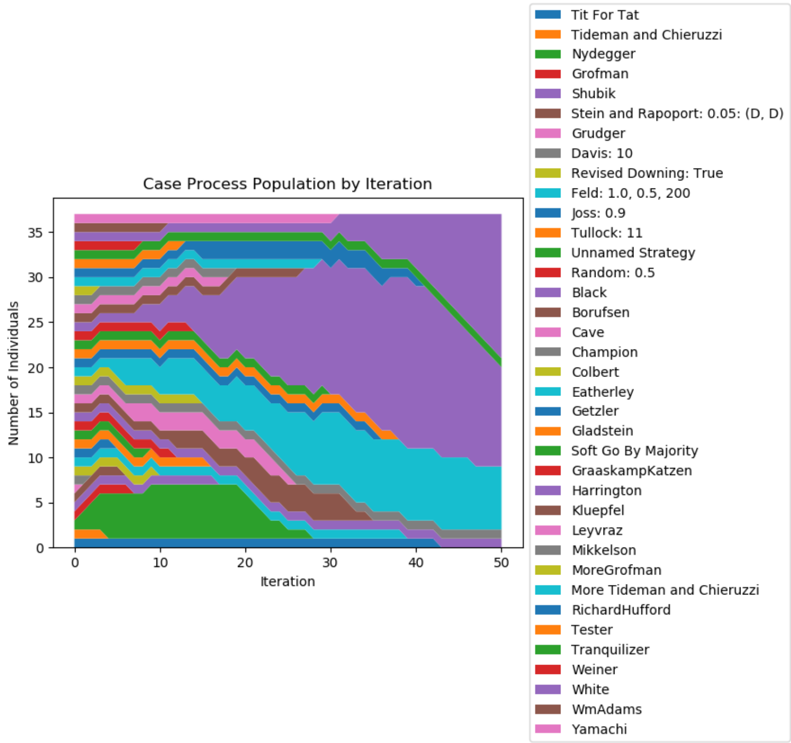

Fig. 31. Population plot of Case's simulation with the agent population from the Axelrod's second tournament with 0.2 mistake probability and mistake bias (seed = 43238133).

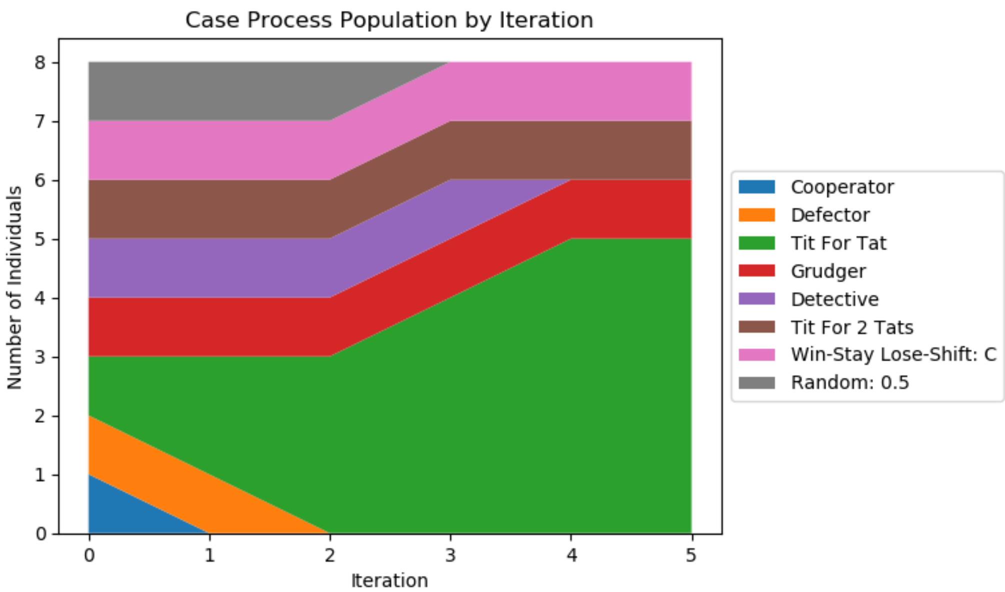

Fig. 32. Population plot of Case's simulation with the agent population from the Case's simulation (seed = 19192831).

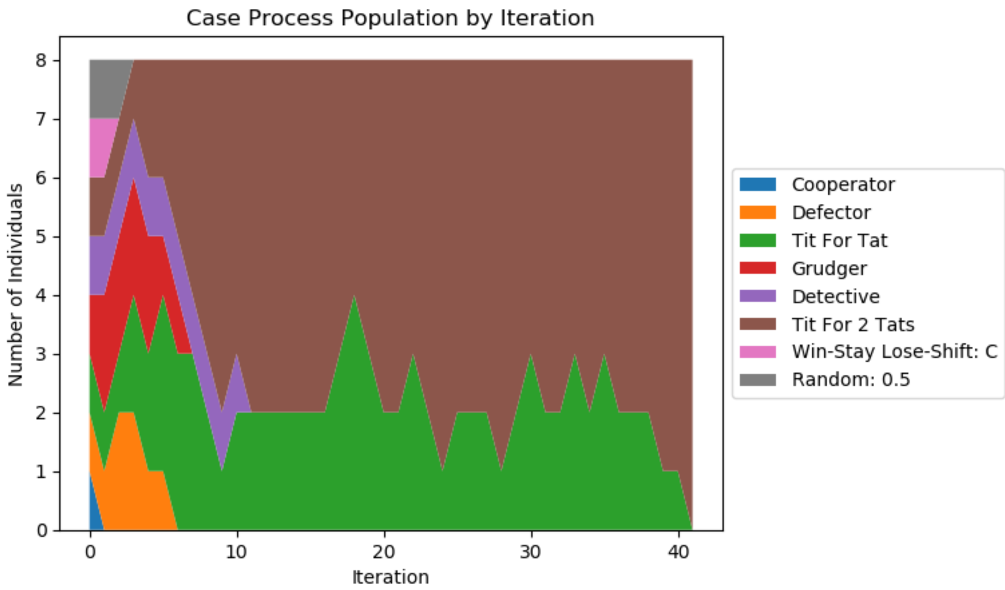

Fig. 33. Population plot of Case's simulation with the agent population from the Case's simulation with 0.05 mistake probability (seed = 43238133).

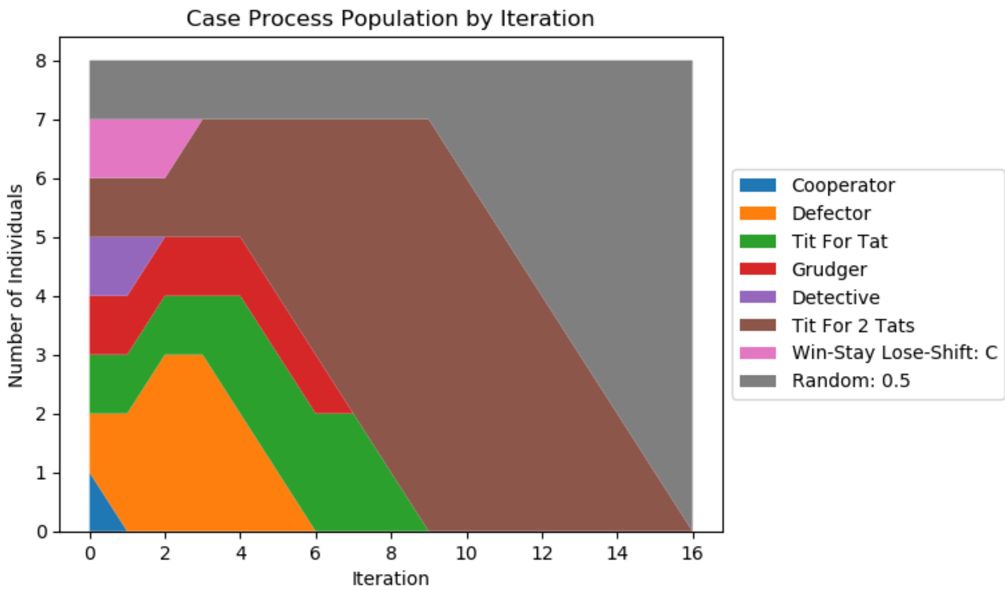

Fig. 34. Population plot of Case's simulation with the agent population from the Case's simulation with 0.05 mistake probability and mistake bias (seed = 34293471).

Figure 35 shows the one of the most common results of the simulation with 0.2 mistake chance. As expected, strategies that tends to defect often wins, which was shown by the victories of Cheater in 11 out of 20 cases and Grudger in 9 out of 20 cases. Using the setting of 0.2 mistake chance with mistake bias also shows similar results as the setting with the 0.2 mistake chance with mistake bias (12 out of 20 wins for Defector and 4 out of 20 wins for Grudger). There are also some outlier results in here, with 3 wins by Copykitten and 1 win by Detective. The trigger for these outlier results are still the same as the previous setting with mistake bias (0.05 mistake chance with mistake bias), which is the Grudger's retaliation timing. The earlier the other agents defect to the Grudger, the bigger the chance either the Cheater or the Grudger will win.

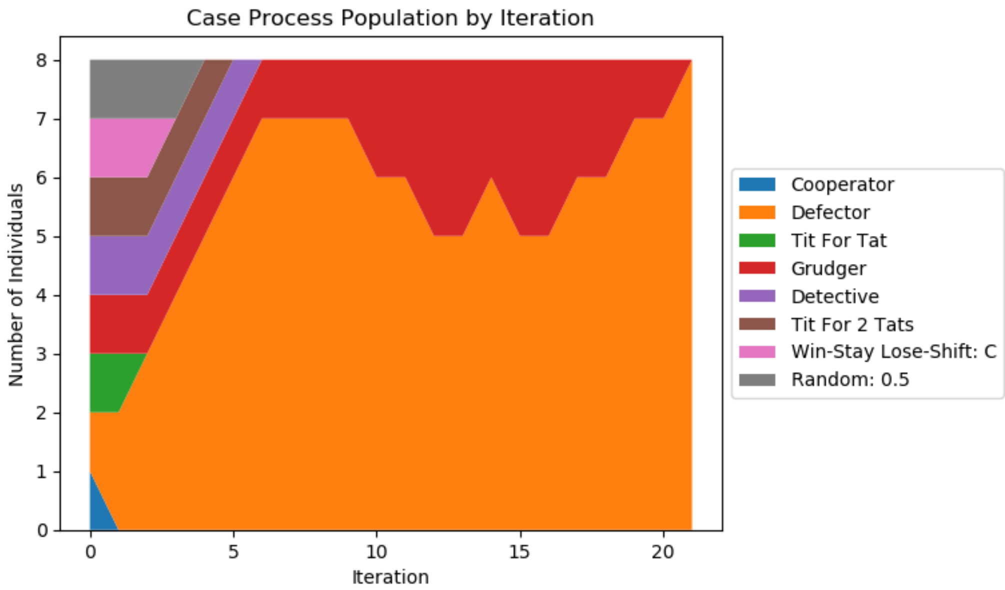

Fig. 35. Population plot of Case's simulation with the agent population from the Case's simulation with 0.2 mistake probability with/without mistake bias (seed = 123456).

4) Case's Simulation Result with Stewart-Plotkin's Tournament Agents

Figure 36 shows the most common result of the simulation with Stewart-Plotkin tournament agents. Our experiment results match Stewart-Plotkin's results, with ZD-GTFT-2 and GTFT taking over most of the wins (16 ZD-GTFT-2; 3 GTFT). One outlier case happens with Hard Prober winning 1 time out of 20 cases. 18 out of 20 cases, the population stabilized with cooperate-by-default strategies, consisting of Hard Tit-for-Tat, Hard Tit-for-Two-Tats, GTFT, ZD-GTFT-2, Tit-for-Tat, Grudger, Tit-for-Two-Tats, and Win-Stay-Lose-Shift strategy, which is similar to the payoff matrix result from the Axelrod tournament's results.

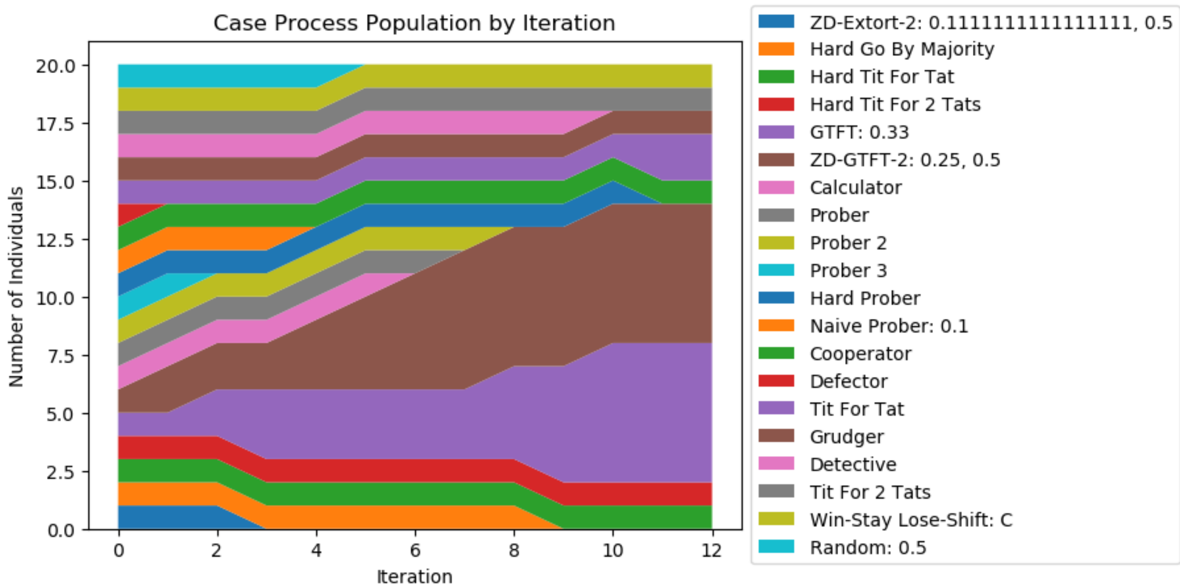

Fig. 36. Population plot of Case's simulation with the agent population from the Stewart-Plotkin's simulation (seed = 901236547).

Figure 37 shows the most common result of the simulation with Stewart-Plotkin tournament agents with 0.05 mistake chance. The result is mostly the same with the previous setting, with GTFT and ZD-GTFT-2 taking over the number of wins (14 GTFT; 6 ZD-GTFT-2). The GTFT strategy managed to win much more than a Zero-Determinant strategy like ZD-GTFT-2. Using a mistake bias doesn't change this trend. In fact, GTFT almost dominated the entire simulation constantly (19 GTFT; 1 Tit-for-Two-Tats). Rather than saying zero-determinant strategies do not work well in a low mistake environment, we are more convinced to hypothesize that strategies with stochastic forgiveness like GTFT (and Harrington in Axelrod's second tournament) works well in a similar environment.

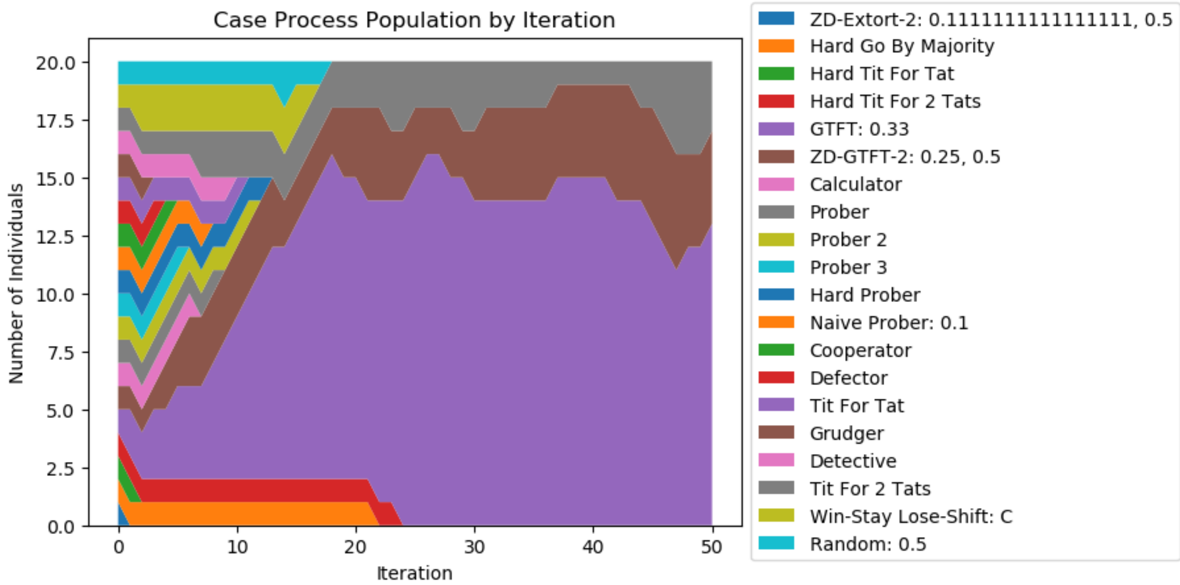

Fig. 37. Population plot of Case's simulation with the agent population from the Stewart-Plotkin's simulation with 0.05 mistake probability with/without mistake bias (seed = 5665363).

We did not show the result of the simulation with Stewart-Plotkin tournament agents with 0.2 mistake chance because most of the graphs are different. We found no underlying pattern in the results. However, we discovered the top performers. Out of 20 cases, ZD-GTFT-2 won 6 times; GTFT and Detective 4 times; Naïve Prober 2 times; Calculator, Hard Prober, Defector, and Detective 1 time; and 1 joint win between Naïve Prober and GTFT.

We prove our previous hypothesis the results from the scenario with 0.2 mistake chance and mistake bias. We see that GTFT wins 14 times out of 20 cases, followed by both Win-Stay-Lose-Shift and ZD-GTFT-2, both winning 3 times out of 20 cases. The sudden increase of Win-Stay-Lose-Shift strategy may be because of the fact that a big mistake chance and mistake bias together caused the strategy to keep staying in the defect mode.

5) Case's Simulation Result with Best Agents

Figure 38 shows the most common result of the simulation with what we considered the best agents, based on their performances on their respective tournaments. We see that most of the time, the population consistently managed to stabilize before it reached the 50 rounds threshold. The Tit-for-Tat strategy wins 10 out of 20 times; Stein & Rapoport and ZD-GTFT-2 3 times each; Win-Stay-Lose-Shift 2 times; GTFT 1 time; joint win between Stein & Rapoport and Tit-for-Tat 1 time.

Figure 39 shows the most common result of the simulation with what we considered the best agents, with a mistake chance of 0.05. The Harrington strategy wins 12 times out of 20, followed by ZD-GTFT-2 with 6 wins and an outlier win by GTFT in 1 case. There is also a case of joint-win between ZD-GTFT-2 and Harrington strategy in one case. The result of the same setting added with mistake bias doesn't show much changes, as shown in Figure 40. In this setting, ZD-GTFT-2 wins in 11 cases, followed by Harrington 7 out of 20 times, and GTFT only in 1 case out of 20.

We did not show the results of the simulation with what we considered the best agents, with a mistake chance of 0.2, because there are no underlying pattern in the results. Defector surprisingly edges out the others by winning in 8 out of 20 times, followed by Tranquilizer 5 out of 20 times. The rest of the wins are one-off occurrences, with Grudger winning 2 out of 20 cases and ZD-GTFT-2, White, Cave, TFT, and Eatherley winning once each.

Figure 42 shows the most common result of the simulation with what we considered the best agents, with a mistake chance of 0.2 and mistake bias enabled. The White strategy wins in 10 cases out of 20, followed by GTFT in 6 cases and ZD-GTFT-2 in 4 cases. The result is consistent with what we got from the Stewart-Plotkin simulation.

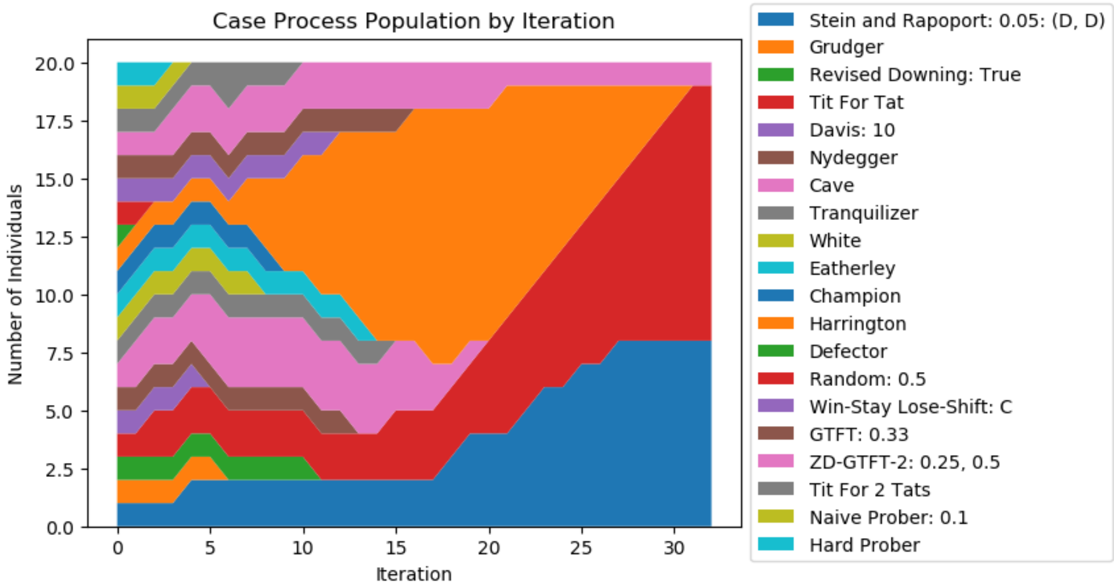

Fig. 38. Population plot of Case's simulation with the best agents (seed = 1143728).

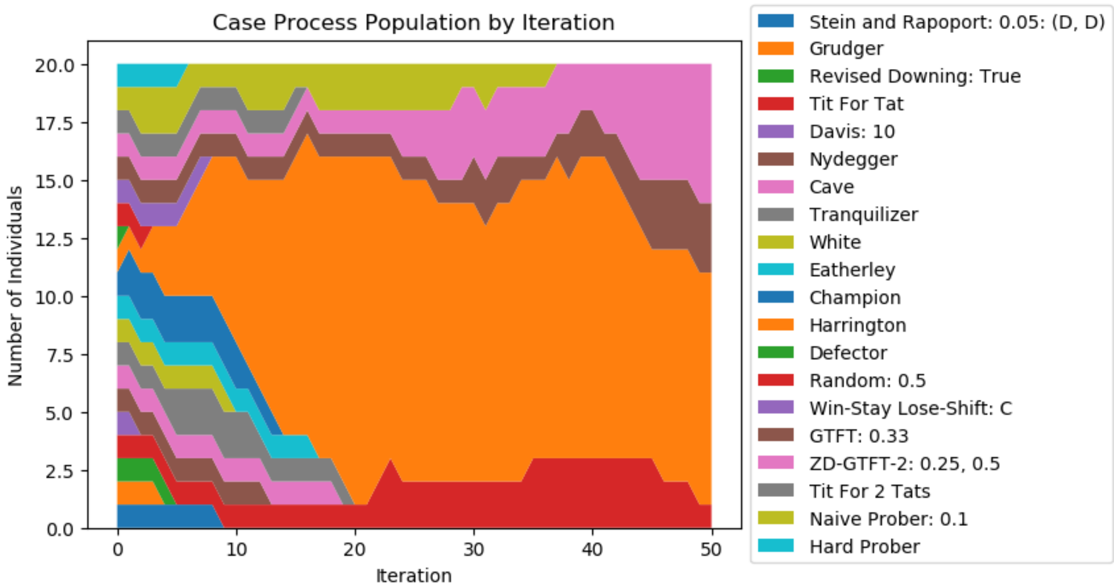

Fig. 39. Population plot of Case's simulation with the best agents with 0.05 mistake probability (seed = 83281327).

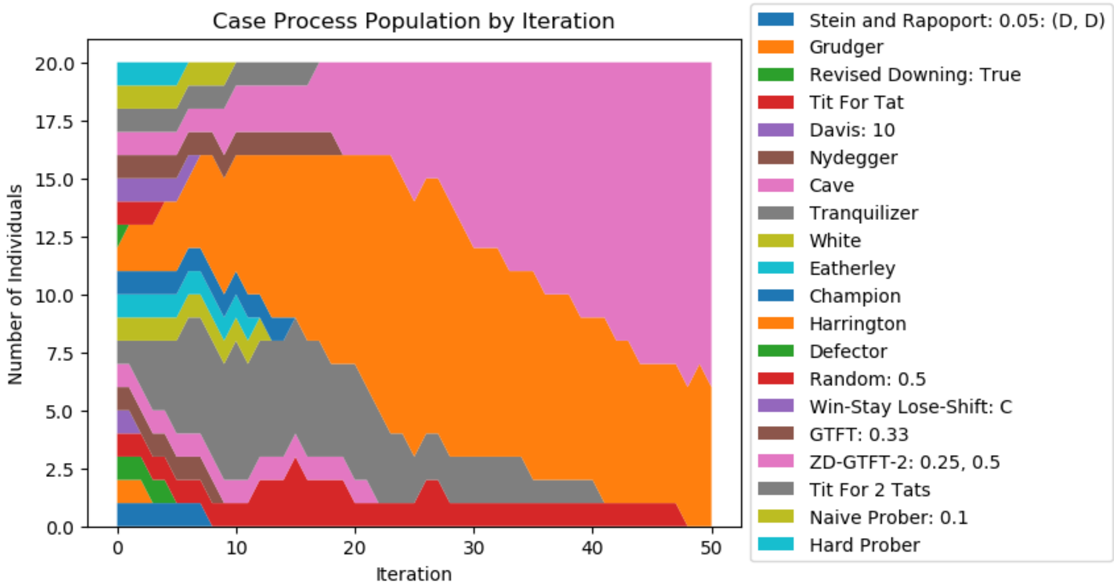

Fig. 40. Population plot of Case's simulation with the best agents with 0.05 mistake probability and mistake bias (seed = 123456).

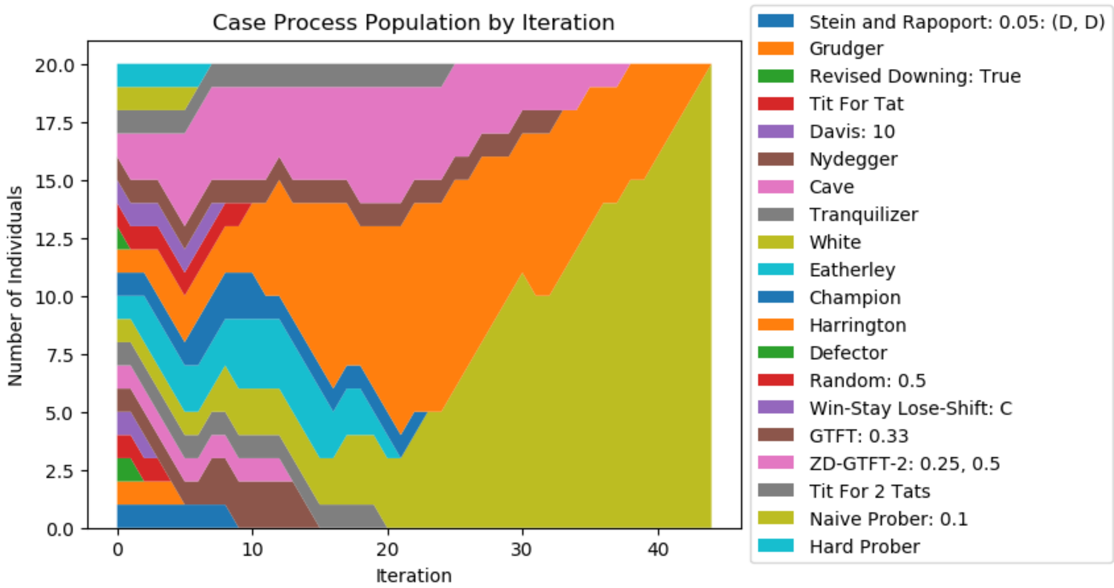

Fig. 41. Population plot of Case's simulation with the best agents with 0.2 mistake probability and mistake bias (seed = 123456).

Conclusion & Further Studies

In this paper, we proposed a model to simulate how trust between people change in response to other people and the environment using iterated prisoner's dilemma. We proposed a model and simulation that adapts Case's simulation, with additional strategies taken from Axelrod's iterated prisoner's dilemma tournament.

Based on the simulations we have done in this research, in addition to strengthening the conclusions from the previous iterated prisoner's dilemma studies, we also inferred some new observations that can be studied further.

1. Agents that adapt based on the other agent's choices can score higher and survive better in a social environment.
2. Agents with stochastic forgiveness can perform much better than agents deterministic forgiveness (e.g. Tit-for-Tat and Tit-for-Two-Tats).
3. Zero-determinant strategies perform better against agents with deterministic decision-making processes.

4. Higher level of mistakes allows agents with tendencies to defect or retaliate heavily to perform better.
5. Mistake bias increases the "certainty" of one's actions, which can be used effectively by agents that rely on stochastic processes.
6. Agent composition drastically changes the overall trust environment evolution, particularly influential agents like kingmaker agents and zero-determinant agents.

We also propose several things that can be looked upon for the next studies, based on our experiments:

1. Experimenting with how the "score" is calculated. Our research used the total payoff from iterated prisoner's dilemma games with other agents as the score. However, another possible scoring scheme can be done by counting the number of "wins" from the other agents. A win is counted when your total payoff for the current iterated prisoner's dilemma game is larger than the other agent's. Changing how the score is calculated will drastically affect the trust evolution. Based on Rapoport's analysis [11], if we count the score by the number of wins, Tit-for-Tat would always be one of the first agents to be replaced, since they will never have a total payoff score larger than their opponent's. On the other hand, a zero-determinant agent that extorts for number of wins like ZD-EXTORT-2 will surely become one of the most prominent winners in the entire simulation.
2. Experimenting with how agents are paired for an iterated prisoner's dilemma game. In reality, not every one interacts with other people. Allowing dynamic possible interactions in trust evolution simulation is one of the most important thing to be done for a better simulation results for use in the real world. There are some possible interactions that can be used here. For example, when an agent reaches a mutual cooperation with another agent, both of them would be able to connect with each other's neighboring agents, which is similar to expanding connections. Another possible way to simulate real world relationship is by cutting a tie between two agents when they reach a mutual defection.

REFERENCES

[1] B. von Stengel, "Game Theory Basics," 2007.

[2] E. Fehr dan U. Fischbacher, "The nature of human altruism -- Proximate and evolutionary origins.," *Nat. Mater.*, vol. 425, no. October, hal. 785–791, 2003.

[3] V. Capraro, "A Model of Human Cooperation in Social Dilemmas," *PLoS One*, vol. 8, no. 8, 2013.

[4] R. Axelrod dan W. D. Hamilton, "The Evolution of Cooperation.," *Science*, vol. 211, no. 4489. hal. 1390–6, 1981.

[5] N. Case, "The Evolution of Trust," 2017. [Daring]. Tersedia pada: http://ncase.me/trust/. [Diakses: 07-Feb-2018].

[6] R. Axelrod, "Effective choice in the prisoner's dilemma," *J. Conflict Resolut.*, vol. 24, no. 1, hal. 3–25, 1980.

[7] R. Axelrod, "More effective choice in the prisoner's dilemma," *J. Conflict Resolut.*, vol. 24, no. 3, hal. 379–403, 1980.

[8] R. Wu, Jianzhong; Axelrod, "How to cope with noise in the iterated prisoner's dilemma," *J. Conflict Resolut.*, vol. 39, no. 1, hal. 183–189, 1995.

[9] A. J. Stewart dan J. B. Plotkin, "Extortion and cooperation in the Prisoner's Dilemma," *Proc. Natl. Acad. Sci.*, vol. 109, no. 26, hal. 10134–10135, 2012.

[10] W. H. Press dan F. J. Dyson, "Iterated Prisoner's Dilemma contains strategies that dominate any evolutionary opponent," *Proc. Natl. Acad. Sci.*, vol. 109, no. 26, hal. 10409–10413, 2012.

[11] A. Rapoport, D. A. Seale, dan A. M. Colman, "Is tit-for-tat the answer? on the conclusions drawn from axelrod's tournaments," *PLoS One*, vol. 10, no. 7, hal. 1–11, 2015.

APPENDIX A. LIST OF STRATEGIES USED IN THE FIRST AXELROD TOURNAMENT

Most of the descriptions were taken from http://axelrod.readthedocs.io/en/stable/reference/all_strategies.html, with several corrections to make it easier to read.
For brevity, the action will be denoted by *C* and *D*, with *C* being cooperation, and *D* being defection.

1. Tit-for-Tat (by Anatol Rapoport)
   Tit-for-Tat starts the round by cooperating. Afterwards, it plays the opponent's action last round. Also called "Copycat" in Case's simulation.
2. Tideman and Chieruzzi (by T. Nicolaus Tideman and Paula Chieruzzi)
   Tideman and Chieruzzi begins by playing Tit-for-Tat and then follows the following rules:
   a. Every run of defections played by the opponent increases the number of defections this strategy plays for each retaliation by 1.
   b. The opponent is given a 'fresh start' if all these conditions are fulfilled:
      i. it is 10 points behind this strategy,
      ii. it has not just started a run of defections,
      iii. it has been at least 20 rounds since the last 'fresh start',
      iv. there are more than 10 rounds remaining in the match, and
      v. the total number of defections differs from a 50-50 random sample by at least 3.0 standard deviations.

   A 'fresh start' is a sequence of *CC* followed by an assumption that the game has just started (everything is forgotten).
3. Nydegger (by Rudy Nydegger)
   Nydegger begins by playing Tit-for-Tat for the first three rounds, except that if it was the only one to cooperate at the first round and the only one to defect at the second round, it defects at the third move.
   After the third round, Nydegger determines its choice based on the 3 preceding outcomes in the following manner.

$$A = 16a_1 + 4a_2 + a_3$$

   Where a_i is dependent on the outcome of the previous i-th round.
   - If both strategies defects, $a_i = 3$.
   - If only the opponent defects, ai = 2.
   - If only the agent defects, then $a_i = 1$.
   - If both strategies cooperate, $a_i = 0$

   Finally, this strategy defects if and only if:

$$A \in \{1,6,7,17,22,23,26,29,30,31,33,38,39,45,49,54,55,58,61\}$$

   Thus, if all three preceding rounds are mutual defections, A = 63 and the strategy would cooperate.
4. Grofman (by Bernard Grofman)
   This strategy has 3 phases:
   a. First it cooperates on the first two rounds.
   b. For rounds 3-7 inclusive, it plays Tit-for-Tat.
   c. Afterwards, it applies the following logic, looking at its memory of the last 8* rounds (ignoring the most recent round).
      i. If its own previous move was C and the opponent has done *D* less than 3 times in the last 8* rounds, do *C*.
      ii. If its own previous move was C and the opponent has done *D* 3 or more times in the last 8* rounds, do *D*.
      iii. If its own previous move was D and the opponent has done *D* only once or not at all in the last 8* rounds, do *C*.
      iv. If its own previous move was D and the opponent has done *D* more than once in the last 8* rounds, do *D*.
5. Shubik (by Martin Shubik)
   Shubik plays like Tit-For-Tat with the following modification. After each retaliation, the number of rounds that Shubik retaliates with *D* increases by 1.
6. Stein and Rapoport (by Stein and Anatol Rapoport)
   This strategy plays a modification of Tit-for-Tat.
   a. It does *C* for the first 4 moves.

   b. It does D on the last 2 moves.
   c. Every 15 moves it makes use of a chi-squared test to check if the opponent is playing randomly. If the opponent is determined to play randomly, use D. Else, play like Tit-for-Tat.
7. Grudger (by James W. Friedman)
   Grudger starts the round by doing C. It keeps doing C until the opponent does D once, after which it will always do D.
8. Davis (by Morton Davis)
   Davis starts by doing C for 10 rounds then plays Grudger for the rest of the game.
9. Graaskamp (by Jim Graaskamp)
   Graaskamp follows the following rules:
   a. Play Tit-for-Tat for the first 50 rounds;
   b. Do D in round 51;
   c. Play 5 further rounds of Tit-for-Tat;
   d. A check is then made to see if the opponent is playing randomly, in which case it uses D for the rest of the game;
   e. The strategy also checks to see if the opponent is playing Tit-for-Tat or another strategy from a preliminary tournament called 'Analogy'. If so, it plays Tit-for-Tat. If not, it uses C and randomly uses D every 5 to 15 moves.
10. Downing (by Leslie Downing)
   Downing plays the first two rounds by doing CC. After that, it attempts to estimate the next move of the opponent by estimating the probability of cooperating given that they defected ($P(C|D)$) or cooperated on the previous round ($P(C|C)$). These probabilities are continuously updated during play and the strategy attempts to maximize the long-term play.

   Downing will then calculate the following:

$$c = 6 * P(C|C) - 8 * P(C|D) - 2$$
$$alt = 4 * P(C|C) - 5 * P(C|D) - 1$$

   Using the value of c and alt:
   a. If $c >= 0$ and $c >= alt$, do C.
   b. If $c >= 0$ and $c < alt$, or alt >= 0, do the opposite of the last round.
   c. In other cases, do D.

   Note that the initial values of $P(C|C)$ and $P(C|D)$ are 0.5.
11. Feld (by Scott Feld)
   Feld plays Tit-for-Tat, always doing D if the opponent does D in the last round. However, the strategy does C when the opponent does C during the last round with a gradually decreasing probability (starting from 1) until it is only 0.5.
12. Joss (by Johann Joss)
   Joss does C with a probability of 0.9 when the opponent does C. Otherwise, it plays Tit-For-Tat.
13. Tullock (by Gordon Tullock)
   Tullock does C for the first 11 rounds then randomly does C 10% less often than the opponent has in previous rounds.
14. Unnamed Strategy
   This strategy does C with a given probability P. This probability (which has initial value 0.3) is updated every 10 rounds based on whether the opponent seems to be random, very cooperative or very uncooperative. Furthermore, if after round 130 the strategy is losing, P is also adjusted (around 0.3 to 0.7).
15. Random
   Random randomly does C and D with 50-50 probability.

APPENDIX B. LIST OF STRATEGIES USED IN THE SECOND AXELROD TOURNAMENT

Most of the descriptions were taken from http://axelrod.readthedocs.io/en/stable/reference/all_strategies.html, with several corrections to make it easier to read. Due to the lack of complete documentation, not all of the strategies' descriptions are listed. For brevity, the action will be denoted by *C* and *D*, with *C* being cooperation, and *D* being defection.

1. Two Unknown Strategies (different from the one listed in the first Axelrod tournament)
2. Go by Majority (by Gail Grisell)
   Go by Majority examines the history of the opponent: if the opponent uses *D* more than *C*, the strategy does *D*; else *C*. In case of equal number of *D* and *C*, the strategy will do *C*.
3. Kluepfel (by Charles Kruepfel)
   This player keeps track of the the opponent's responses to own behavior:
   a. `cd_count` counts: Opponent does *C* as response to the agent doing *D*.
   b. `dd_count` counts: Opponent does *D* as response to the agent doing *D*.
   c. `cc_count` counts: Opponent does *C* as response to the agent doing *C*.
   d. `dc_count` counts: Opponent does *D* as response to the agent doing *C*.

   After 26 rounds, the agent then tries to detect a random agent. The agent decides this by using the condition:
   ```
   cd_counts >= (cd_counts+dd_counts) / 2 - 0.75 * sqrt(cd_counts+dd_counts) AND
   cc_counts >= (dc_counts+cc_counts) / 2 - 0.75 * sqrt(dc_counts+cc_counts)
   ```
   If the agent decides that they are playing against a random agent, always do *D*.
   Otherwise respond to recent history using the following set of rules:
   a. If the opponent's last three choices are the same, do the same.
   b. If the opponent's last two choices are the same, do the same with a probability of 90%.
   c. Otherwise, if opponent's last action was *C*, do *C* with 70% probability.
   d. Otherwise, if opponent's last action was *D*, do *D* with 60% probability.
4. Strategy by Harold Rabbie
5. Grudger (by James W. Friedman)
   Grudger starts the round by doing *C*. It keeps doing *C* until the opponent does *D* once, after which it will always do *D*.
6. Strategy by Abraham Getzler
7. Strategy by Roger Hotz
8. Strategy by George Lefevre
9. Strategy by Nelson Weiderman
10. Strategy by Tom Almy
11. Strategy by Robert Adams
12. Weiner (by Herb Weiner)
    Weiner plays Tit-for-Tat with a chance for forgiveness and a defective override.
    The chance for forgiveness happens only if `forgive_flag` is raised. If this flag is raised and `turn` is greater than `grudge`, then override Tit-for-Tat with Cooperator (always do *C*). `grudge` is a variable that starts at 0 and increases by 20 with each forgiven *D* action (overridden through the forgiveness logic). `forgive_flag` is lowered regardless of whether the logic is overridden or not.
    The variable `defect_padding` increases each time the opponent does *D*, but resets to zero when the opponent does *C* (or `forgive_flag` is lowered) so that it roughly counts the *D* actions between *C* actions. Whenever the opponent does *C*, if `defect_padding` (before resetting) is odd, raise `forgive_flag` for the next round.
    Finally, a defective override is assessed after forgiveness. If five or more of the opponent's last twelve actions are *D*, do *D*. This will overrule a forgiveness but doesn't undo the lowering of `forgive_flag`. Note that "last twelve actions" doesn't count the most recent action.
13. Borufsen (by Otto Borufsen)
    Borufsen keeps track of the opponent's responses to the agent's response:
    - `cd_count` counts: Opponent does *C* as response to player doing *D*.
    - `cc_count` counts: Opponent does *C* as response to player doing *C*.

    The player has a defect mode and a normal mode. In defect mode, the player will always do *D*. In normal mode, the player obeys the following ranked rules:
    a. If in the last three turns, both the player/opponent do *D*, do *C* for a single turn.
    b. If in the last three turns, the player/opponent acted differently from each other and they're alternating, change next *D* action to *C*. (Doesn't block third rule.)
    c. Otherwise, play Tit-for-Tat.

The agent starts in normal mode, but every 25 turns, starting with the 27th turn, it reevaluates the mode. Enter defect mode if any of the following conditions hold:

- Detected random: Opponent does *C* 7-18 times since last mode evaluation (or start) AND less than 70% of opponent's *C* action was in response to player's *C* action, i.e. `cc_count / (cc_count+cd_count) < 0.7`
- Detect defective: Opponent did *C* fewer than 3 times since the last mode evaluation.

When switching to defect mode, immediately do *D*. The first two rules for normal mode require that last three turns were in normal mode. When starting normal mode from defect mode, do *D* on first move.

14. Strategy by R. D. Anderson
15. WmAdams (by William Adams)
    Count the number of times the opponent does *D* after their first move (`c_defect`). Do *D* if `c_defect` equals to 4, 7, or 9. If `c_defect` is larger than 9, immediately do *D* after the opponent does *D* with probability = `0.5^(c_defect-1)`. Otherwise, do *C*.
16. Strategy by Michael F. McGurrin
17. Eatherley (by Graham J. Eatherley)
    Eatherley keeps track of how many times in the game the other player did *D*. After the other player does *D*, do *D* with a probability equal to the ratio of the opponent's number of *D* action to the total moves up to that point. Otherwise, do *C*.
18. Richard Hufford (by Richard Hufford)
    Richard Hufford tracks opponent's "agreements", that is whenever the opponent's previous move is the same as the agent's move two rounds ago. If the opponent's first move is *D*, this is counted as a disagreement, otherwise it's an agreement. From the agreement counts, two measures are calculated:
    - `proportion_agree` = number of agreements (up to the opponent's last turn) + 2 divided by the current turn number.
    - `last_four_num`: The number of agreements in the last four turns. If there have been fewer than four previous turns, then is equal to the number of agreements + (4 - number of past turns).

    The agent then decides which action to take using these rules:
    a. If `proportion_agree` > 0.9 and `last_four_num` >= 4, do *C*.
    b. Otherwise, if `proportion_agree` >= 0.625 and `last_four_num` >= 2, then play Tit-for-Tat.
    c. Otherwise, do *D*.

    However, if the opponent has done *C* for the last `streak_needed` turns, the strategy deviates from the usual strategy, and instead does *D*. This deviation is called an "aberration". In the turn immediately after an aberration, the strategy doesn't override, even if there's a streak of *C* actions. Two turns after an aberration, the agent restarts the cooperation streak (never looking before this turn), do *C*, and changes `streak_needed` to:

    ```
    floor (20.0 * num_abb_def / num_abb_coop) + 1
    ```

    `num_abb_def` is 2 + the number of times that the opponent did *D* in the turn after an aberration, and `num_abb_coop` is 2 + the number of times that the opponent did *C* in response to an aberration.
19. Strategy by George Hufford
20. Cave (by Rob Cave)
    Cave first cooperates in the first round. After that, it looks for overly-defective or apparently random opponents and defects forever if the opponent is determined to be one of the two. Below are the conditions for Cave to defect.
    - round > 39 and percent of opponent's defects > 0.39
    - round > 29 and percent of opponent's defects > 0.65
    - round > 19 and percent of opponent's defects > 0.79

    Otherwise:
    - If the opponent cooperates, Cave also cooperates.
    - If the opponent has defected at least 18 times and the opponent defected in the previous round, Cave also defects.
    - If the opponent has defected less than 18 times and the opponent defected, Cave plays with a random (50-50) choice.
21. Strategy by Rik Smoody
22. Strategy by John William Colbert
23. Strategy by David A. Smith
24. Strategy by Henry Nussbacher
25. Strategy by William H. Robertson

26. Strategy by Steve Newman
27. Strategy by Stanley F. Quayle
28. Strategy by Rudy Nydegger
29. Strategy by Glen Rowsam
30. Strategy by Leslie Downing
31. Graaskamp & Katzen (by Jim Graaskamp and Ken Katzen)
    Grasskamp & Katzen plays Tit-for-Tat at first and tracks `score`. At select checkpoints, check for a high `score`. Switch to Defect Mode if:
    - On move 11, `score` < 23
    - On move 21, `score` < 53
    - On move 31, `score` < 83
    - On move 41, `score` < 113
    - On move 51, `score` < 143
    - On move 101, `score` < 293

    Once in Defect Mode, do *D* until the end.
32. Champion (by Danny C. Champion)
    Champion cooperates on the first 10 moves and plays Tit-for-Tat for the next 15 more moves. After 25 moves, the agent does *C* unless all the following conditions are true:
    a. The opponent did *D* on the previous move
    b. The other player did *C* less than 60%
    c. A random number between 0 and 1 is greater than the opponent's cooperation rate.
33. Strategy by Howard R. Hollander
34. Strategy by George Duisman
35. Yamachi (by Brian Yamachi)
    Yamachi keeps track of the play history with a dictionary (X, Y, Z) that contains the opponent's actions and the following actions by the agent and the opponent respectively. Each turn, the agent looks at the opponent's action two rounds ago and the agent's action last round. If (X, Y, C) has occurred more often (or as often) as (X, Y, D), do *C*. Otherwise, do *D*.
    Starting with the 41st turn, there's a possibility to override this behavior. If `portion_defect` is between 45% and 55% (exclusive), do *D*. `portion_defect` equals to the number of opponent's *D* actions plus 0.5 divided by the turn number (indexed by 1). When overriding this way, the agent still records (X, Y, Z) as though the strategy didn't override.
36. Strategy by Mark F. Batell
37. Strategy by Ray Mikkelson
38. Tranquilizer (by Craig Feathers)
    Tranquilizer starts by doing *C* and has 3 states.
    At the start of the strategy the agent updates its states:
    - Count the number of consecutive *D* actions by the opponent.
    - If the agent was in state 2, it moves to state 0 and calculates `two_turns_after_good_defection_ratio` and `two_turns_after_good_defection_ratio_count`.

      ```
      two_turns_after_good_defection_ratio = (((two_turns_after_good_defection_ratio *
      two_turns_after_good_defection_ratio_count) + (3 - (3 * dict[opponent.history[-1]])) +
      (2 * dict[self.history[-1]]) - ((dict[opponent.history[-1]] * dict[self.history[-
      1]]))) / (two_turns_after_good_defection_ratio_count + 1) )

      two_turns_after_good_defection_ratio_count = two_turns_after_good_defection_ratio + 1
      ```

    - If the agent was in state 1, it moves to state 2 and calculates `one_turn_after_good_defection_ratio` and `one_turn_after_good_defection_ratio_count`.

      ```
      one_turn_after_good_defection_ratio = (((one_turn_after_good_defection_ratio
      * one_turn_after_good_defection_ratio_count) + (3 - (3 *
      dict[opponent.history[-1]])) + (2 * dict[self.history[-1]]) -
      (dict[opponent.history[-1]] * dict[self.history[-1]])) /
      (one_turn_after_good_defection_ratio_count + 1) )
      ```

```
one_turn_after_good_defection_ratio_count =
one_turn_after_good_defection_ratio + 1
```

If after this the agent is in state 1 or 2, do *C*.
If the agent is in state 0 it will potentially perform 1 of the 2 following stochastic tests:

- If average score per round is greater than 2.25, the agent will do *C* with a probability of:

```
probability = ((.95 - (((one_turn_after_good_defection_ratio) +
(two_turns_after_good_defection_ratio) - 5) / 15)) + (1 / (((len(self.history))+1) ** 2))
- (dict[opponent.history[-1]] / 4) )
```

If the agent does not do *C* then the strategy moves to state 1 and does *C*.

- If average score per round is greater than 1.75 but less than 2.25, the agent will do *C* with a probability of:

```
probability = ((.25 + ((opponent.cooperations + 1) / ((len(self.history)) + 1))) -
(opponent_consecutive_defections * .25) + ((current_score[0] - current_score[1]) / 100) +
(4 / ((len(self.history)) + 1)) )
```

- If none of the above holds, the agent simply plays Tit-for-Tat.

39. Leyvraz (by Fransois Leyvraz)
    Leyvraz uses the opponent's last three moves to decide on an action based on the following ordered rules.
    a. If opponent did *D* last two turns, do *D* with a probability of 75%.
    b. If opponent did *D* three turns ago, do *C*.
    c. If opponent did *D* two turns ago, do *D*.
    d. If opponent did *D* last turn, do *D* with a probability of 50%.
    e. Otherwise, do *C*.
40. Strategy by Johann Joss
41. Strategy by Robert Pebly
42. Strategy by James E. Hall
43. White (by Edward C. White Jr.)
    White follows the following two rules:
    - If the opponent did *C* last round or in the first ten rounds, do *C*.
    - Otherwise, do *D* if and only if `floor(log(turn)) * opponent_num_defections >= turn`
44. Strategy by George Zimmerman
45. Two strategies by Edward Friedland
46. Harrington (by Paul D. Harrington)
    Harrington has three modes: Normal, Fair-weather, and Defect.

    In Normal and Fair-weather modes, the strategy begins by:
    - Updating history
    - Trying to detect random opponent if turn is multiple of 15 and >=30.
    - Checking if burned flag should be raised.
    - Checking for Fair-weather opponent if turn is 38.

    Updating history means to increment the correct cell of the `move_history`, which is a matrix where the columns are the opponent's previous move and the rows are indexed by the combo of the agent's and the opponent's moves two rounds ago. The upper-left cell must be all *C*, but otherwise order doesn't matter. After entering Defect mode, `move_history` won't be used anymore.
    If the turn is a multiple of 15 and >=30, attempt to detect random. If random is detected, enter Defect mode and do *D* immediately. If the player was previously in Defect mode, then do not reenter. The random detection logic is a modified Pearson's Chi Squared test, with some additional checks.
    Some of the agent's moves are marked as "generous." If the agent made a generous move two turns ago and the opponent replied with a *D* action, raise the burned flag. This will stop certain generous moves later.
    The agent mostly plays Tit-for-Tat for the first 36 moves, then does *D* on the 37th move. If the opponent does *C* on the first 36 moves, and does *D* on the 37th move also, then enter Fair-weather mode and do *C* this turn.
    Next in Normal Mode:
    - Check for defect and parity streaks.

- Check if *C* actions are scheduled.
- Otherwise,
    - If turn < 37, play Tit-for-Tat.
    - If turn = 37, do *D*, mark this move as generous, and schedule two more cooperations**.
    - If turn > 37, if burned flag is raised, play Tit-for-Tat. Otherwise, play Tit-for-Tat with probability 1 - `prob`. And with probability `prob`, do *D*, schedule *CC* actions for the next two rounds, mark this move as generous, and increase `prob` by 5%.

** Scheduling *CC* means to set `more_coop` flag to two. If the agent is in Normal mode and no streaks are detected, then the agent will do *C* and lower this flag, until hitting zero. It's possible that the flag can be overwritten. Notable on the 37th turn *D* action, this is set to two, but the 38th turn Fair-weather check will set this.

If the opponent's last twenty moves were *D*, do *D* this round. Then check for a parity streak, by flipping the parity bit (there are two streaks that get tracked which are something like odd and even turns, but this flip bit logic doesn't get run every turn), then incrementing the parity streak that the agent's pointing to. If the parity streak that the agent's pointing to is then greater than `parity_limit`, reset the streak and do *C* immediately. `parity_limit` is initially set to five, but after it has been hit eight times, it decreases to three. The parity streak that the agent's pointing to also gets incremented if in normal mode and the agent does *D* but not on turn 38, unless it's doing *D* as the result of a defect streak. Note that the parity streaks resets but the defect streak doesn't.

If `more_coop` >= 1, then the agent will do *C* and lower that flag here, in Normal mode after checking streaks. The agent also lowers this flag if it does *C* as the result of a parity streak or in Fair-weather mode.

After that, use the logic based on round from above.

In Fair-Weather mode after running the above logic, check if opponent did *D* last round. If so, exit Fair-Weather mode, and proceed THIS TURN with Normal mode. Otherwise do *C*.

In Defect mode, update the `exit_defect_meter` (originally zero) by incrementing if opponent did *D* last round and decreasing by three otherwise. If `exit_defect_meter` is then 11, then set mode to Normal (for future turns), do *C* and schedule two more *C* actions. Note that this move is not marked generous.

47. Gladstein (by David Gladstein)
    This strategy is also known as Tester.
    Tester is a Tit-for-Tat variant that does *D* on the first round in order to test the opponent's response. If the opponent ever does *D*, the strategy 'apologizes' by doing *C* and then plays Tit-for-Tat for the rest of the game. Otherwise, it does *D* as much as possible subject to the constraint that the ratio of its *D* actions to moves remains under 0.5, not counting the first *D* action.
48. Strategy by Scott Feld
49. Strategy by Fred Mauk
50. Strategy by Dennis Ambuehl and Kevin Hickey
51. Strategy by Robyn M. Dawes and Mark Batell
52. Strategy by Martyn Jones
53. Strategy by Robert A. Leyland
54. Black (by Paul E. Black)
    Black does *C* for the first five turns. Then it calculates the number of opponent's *D* action in the last five moves and does *C* with probability `prob_coop[number_defects]`, where:

```
prob_coop[number_defects] = 1 - (number_defects^ 2 - 1) / 25
```

55. More Tideman & Chieruzzi (by T. Nicolaus Tideman and Paula Chieruzzi)
    More Tideman & Chieruzzi does *C* if their score exceeds the opponent's score by at least `score_to_beat`. `score_to_beat` starts at zero and increases by `score_to_beat_inc` every time the opponent's last two actions are *CD*. `score_to_beat_inc` itself increase by 5 every time the opponent's last two actions are *CD*.

    Additionally, the strategy executes a "fresh start" if the following hold:
    - The strategy would do *D* by `score` (difference less than `score_to_beat`).
    - The opponent did not do *CD* in the last two turns.

- It’s been at least 10 turns since the last fresh start. Or since the match started if there hasn’t been a fresh start yet.

A “fresh start” entails *CC* actions and resetting `score`, `scores_to_beat` and `scores_to_beat_inc`.

56. Strategy by Robert B. Falk and James M. Langsted
57. Strategy by Bernard Grofman
58. Strategy by E. E. H. Schurmann
59. Strategy by Scott Appold
60. Strategy by Gene Snodgrass
61. Strategy by John Maynard Smith
62. Strategy by Jonathan Pinkley
63. Tit-for-Tat (by Anatol Rapoport)
    Tit-for-Tat starts the round by doing *C*. Afterwards, it plays the opponent’s action last round. Also called “Copycat” in Case’s simulation.
64. Win-Stay-Lose-Shift
    WSLS will do *C* in the first round. For the subsequent rounds, if the other agent does *C* in the previous round, do the same action as the last round. If the opponent does *D* in the previous round, do the opposite of the last round. Also known as “Simpleton” in Case’s simulation.
65. Random
    Random randomly does *C* and *D* with 50-50 probability.
66. Tit-for-Two-Tats
    Tit-for-Two-Tats will always do *C*, and only retaliates with *D* after the opponent does *D* two rounds in a row. Also called “Copykitten” in Case’s simulation.

APPENDIX C. LIST OF STRATEGIES USED IN THE CASE SIMULATION

Most of the descriptions were taken from http://axelrod.readthedocs.io/en/stable/reference/all_strategies.html, with several corrections to make it easier to read.
For brevity, the action will be denoted by *C* and *D*, with *C* being cooperation, and *D* being defection.

1. Always Cooperate (Cooperator)
   Cooperator always does *C*, no matter what the opponent chooses.
2. Always Defect (Defector)
   Defector always does *D*, no matter what the opponent chooses.
3. Copycat (Tit-for-Tat)
   Copycat starts the round by doing *C*. Afterwards, it plays the opponent's action last round. Also called "Tit-for-Tat" in Axelrod's and Stewart-Plotkin's tournament.
4. Detective
   Detective starts the first four rounds by doing *CDCC*. If the opponent has always done *C* in the previous rounds, always do *D*. At any point the opponent uses *D* in the previous rounds, play Copycat for the rest of the game.
5. Grudger
   Grudger starts the round by doing *C*. It keeps doing *C* until the opponent does *D* once, after which it will always do *D*.
6. Copykitten (Tit-for-Two-Tats)
   Copykitten will always do *C*, and only retaliates with *D* after the opponent does *D* two rounds in a row. Also called Tit-for-Two-Tats in Stewart-Plotkin's tournament.
7. Simpleton (Win-Stay-Lose-Shift)
   Simpleton will do *C* in the first round. For the subsequent rounds, if the other agent does *C* in the previous round, do the same action as the last round. If the opponent does *D* in the previous round, do the opposite of the last round. Also called "Win-Stay-Lose-Shift" in Axelrod's and Stewart-Plotkin's tournament.
8. Random
   Random randomly does *C* and *D* with 50-50 probability.

APPENDIX D. LIST OF STRATEGIES USED IN THE STEWART-PLOTKIN TOURNAMENT

Most of the descriptions were taken from http://axelrod.readthedocs.io/en/stable/reference/all_strategies.html, with several corrections to make it easier to read.
For brevity, the action will be denoted by *C* and *D*, with *C* being cooperation, and *D* being defection.
For memory-one strategies, P($C|XY$) denotes the probability of doing *C* given that the agent does *X* and the opponent does *Y* in the previous round.

1. Always Cooperate (Cooperator)
   Cooperator always does *C*, no matter what the opponent chooses.
2. Always Defect (Defector)
   Defector always does *D*, no matter what the opponent chooses.
3. Extort-2
   Extort-2 plays using memory-one strategy following these four conditional probabilities based on the last round of play:
   $P(C|CC) = 8/9$
   $P(C|CD) = 1/2$
   $P(C|DC) = 1/3$
   $P(C|DD) = 0$
4. Hard Majority
   Hard Majority does *D* on the first move, does *D* if the number opponent's *D* action is greater than or equal to the number of times it has done *C*, and otherwise does *C*.
5. Hard Joss (Joss)
   Joss does C with a probability of 0.9 when the opponent does C. Otherwise, it plays Tit-For-Tat.
6. Hard Tit-for-Tat
   Hard Tit-for-Tat cooperates on the first move, defects if the opponent has defected on any of the previous three rounds, and otherwise cooperates.
7. Hard Tit-for-Two-Tats
   Hard Tit-for-Two-Tats cooperates on the first move, defects if the opponent has defected twice (successively) of the previous three rounds, and otherwise cooperates.
8. Tit-for-Tat
   Tit-for-Tat starts the round by doing *C*. Afterwards, it plays the opponent's action last round.
9. Grim (Grudger)
   Grudger starts the round by doing *C*. It keeps doing *C* until the opponent does *D* once, after which it will always do *D*.
10. Generous Tit-for-Tat (GTFT)
    GTFT plays Tit-For-Tat with occasional forgiveness, which prevents cycling defections against itself.
    The agent uses the following memory-one probabilities:
    $P(C|CC) = 1$
    $P(C|CD) = p$
    $P(C|DC) = 1$
    $P(C|DD) = p$

    $p = \min(1 – (\mathrm{T} – \mathrm{R})/(\mathrm{R} – \mathrm{S}), (\mathrm{R} – \mathrm{P})/(\mathrm{T} – \mathrm{P}))$

11. Tit-for-Two-Tats
    Tit-for-Two-Tats will always do *C*, and only retaliates with *D* after the opponent does *D* two rounds in a row.
12. Win-Stay-Lose-Shift (WinStayLoseShift)
    Win-Stay-Lose-Shift will do *C* in the first round. For the subsequent rounds, if the other agent does *C* in the previous round, do the same action as the last round. If the opponent does *D* in the previous round, do the opposite of the last round.
13. Random
    Random randomly does *C* and *D* with 50-50 probability.
14. ZDGTFT-2 (ZDGTFT2)
    Extort-2 plays using memory-one strategy following these four conditional probabilities based on the last round of play:
    $P(C|CC) = 8/9$
    $P(C|CD) = 1/2$
    $P(C|DC) = 1/3$
    $P(C|DD) = 0$

15. Calculator
    Calculator plays like Joss for 20 rounds. On the 21st round, Calculator attempts to detect a cycle in the opponents history, and defects unconditionally thereafter if a cycle is found. Otherwise Calculator plays like Tit-for-Tat for the remaining rounds.
16. Prober
    Prober starts by playing *DCC* on the first three rounds and then defects forever if the opponent cooperates on rounds two and three. Otherwise Prober plays as Tit-for-Tat would.
17. Prober2
    Prober2 starts by playing *DCC* on the first three rounds and then cooperates forever if the opponent played *D* then *C* on rounds two and three. Otherwise Prober2 plays as Tit-for-Tat would.
18. Prober3
    Prober starts by playing *DC* on the first two rounds and then defects forever if the opponent cooperated on round two. Otherwise Prober3 plays as Tit-for-Tat would.
19. Hard Prober
    The strategy starts by playing *DDCC* on the first four rounds and then defects forever if the opponent cooperates on rounds two and three. Otherwise Prober plays as Tit-for-Tat would.
20. Naïve Prober
    Naïve Prober is a modification of Tit-for-Tat strategy which with a small probability of randomly defecting. Default value for a probability of defection is 0.1.