Exploring optimization strategies with the prisoner’s dilemma

Jay Chandrasekaran, Jaxon Hornsey, Ali Naqvi, Daniela Espinosa-Perdomo

Dept. of Computer of Science

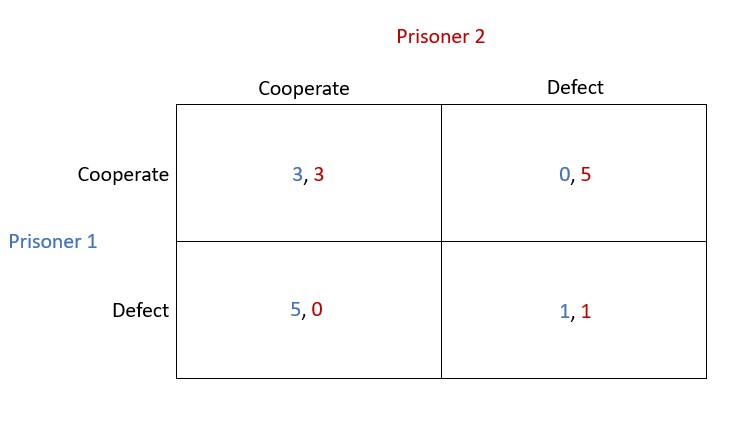
University of Windsor, Windsor, Canada

{thennam, espinos, hornseyj} @uwindsor.ca

***Abstract*—** **Prisoners’ dilemma is a popular game used in game theory to show why rational individuals may not choose to cooperate. The ‘dominant strategy’ in prisoner’s dilemma is to always ‘defect’ rather than ‘cooperate’. In iterated prisoner’s dilemma (IPD), the players play the game repeatedly, and base their moves on prior moves of the opponent. Unlike prisoner’s dilemma, in iterated prisoner’s dilemma, the desire for long-term reward provides good incentive for players to cooperate.  The players can adopt several strategies to maximize their gain based on past moves of their opponent. For the current project, the existing online code for Axelrod’s iterated prisoner’s dilemma was used to run simulations and generate sample data. Different optimization methods like brute-force search, hill climbing search and tabu search were applied to find excellent strategies for iterated prisoner’s dilemma (IPD). Our research found the most successful strategies to be ‘Tit-for-tat’ (TFT) and its variant ‘Tit-for-2-tat’ (TF2T). Following these two, ‘Grudger’ emerged as a good strategy.**

# Introduction

A game-theoretic paradox, “Prisoners’ Dilemma”, is a situation modelled by cooperation and conflict. In this paradox, two individuals who are unable to communicate with each other must choose to either cooperate with each other or not. The reward system for everyone varies based on their decisions and the highest reward is given if both choose to co-operate.

*Sample model of the “Prisoner’s Dilemma”*

The purpose of this report is to analyze and compare a variety of optimization methods for the “Iterated

Prisoners’ Dilemma” in order to find the best possible strategy. The performance of each method will be monitored and tweaked with factors such as population size and mutation probability. A variety of optimization methods will be used such as Genetic Algorithms, Hill Climbing, Tabu Search, and Exhaustive search. These strategies have been tested in a simulated environment and modelled for proper analysis.

# RELEVANT LITERATURE REVIEW

sd

# experimental setup and methodology

 The environment that was used for the experimentation of this paper used Python as its programming language. For all experiments, the popular Axelrod library was utilized []. Each experiment uses an optimization method to find out the best strategy for different scenarios. The Axelrod library contains strategies that will be implemented in the simulations []. As mentioned before, this paper is to analyze and compare the results of winning strategies from different optimization methods and the effect of their parameters.

The methodology for this research was comparing different types of optimization strategies: the Tabu search, Brute force as well as the hill-climbing strategy. The hill-climbing method is a search strategy that continuously moves in the direction of the value that is increasing within a study in order to find the top of the hill (the very best solution to the prisoner dilemma). The hill climber method ends whenever it hits the topmost value where there is no counterpart with a higher value. The main features of the hill-climbing method are things like: the greedy approach; the hill-climbing strategy goes to the direction that optimizes on any cost that might be involved; generate the variant and test it. It will generate the best possible result, this strategy does not take into account/ remember the previous states. The tabu search method of optimization is based on the memory structures, where the information of the recently evaluated players is stored. Brute force generates all possibilities and tests them to find the optimal choice; this option however takes the most amount of time and computer power. The efficiency is O(n\*m). The tabu search is mostly used in the optimization of the problem with the best possible strategy. This algorithm presents a rapid convergence for searching to come up with the best solution for a prisoner dilemma situation.

## Exhaustive Search Setup

7 different strategies will be used throughout this experiment:

**Defector, Tit-for-Tat, Tit-for-2-Tats, Random, Suspicious-Tit-For-Tat, Cooperator, Grudger**

The first experiment was conducted using brute force optimization in mind. The library contains a tournament function that sets up a round-robin tournament with experimental variables such as noise (the probability that a player’s action will be swapped with another strategy), turns to play against each other, and repetitions of the tournament to smoothen out stochastic effects [ ]. A round-robin tournament with 7 players each using one of the seven strategies mentioned before was created. The round-robin allows us to match each strategy against another strategy while keeping track of points. After all, players have played their rounds, the player with the most points wins. For some strategies, the number of turns will inherently give it either an advantage or disadvantage. This experiment will explore the best strategy for our given scenarios:

Round Robin with 5 turns and no noise

Round Robin with 15 turns and no noise

Round Robin with 250 turns and no noise

These scenarios have been specifically picked to demonstrate the difference in best strategies due to changing scenarios. The tournaments will use varying turns and no noise to solely compare the strategies against one another. To ensure accurate results the tournament was run 10 times with a different seed in each case. The purpose of changing the seed allows stochastic strategies to change their moves [ ].

## Hill Climbing and Tabu Search Setup

1. *Hill Climbing Search:* Hill climbing is a type of local search for optimization where we start with an arbitrary solution and make small incremental changes repeatedly until we reach our goal state. We place all the strategies we have from our complete set of strategies(S) into an array. We can pick any random strategy and start the tournament. Here we start by picking the first pair of strategies from the strategies array and play a tournament with the third strategy. Out of this trio, the strategy that accumulates the least points (based on pay-off matrix) is the loser and the other two winning strategies advance to play next tournament with the next strategy. At any point in the game, the best two strategies from the preceding list of strategies would be playing the current strategy. When the last trio of strategies complete the tournament, the strategy (or strategies in case of tie) with the most points is deemed to be the best strategy from the set of strategies we started with.
2. *Hill Climbing Experiment Results Strategy Set (S):* {Defector, Random, Cooperator, Grudger, TitForTat(TFT), Alternator, Bully, TitFor2Tats}

Round-1

Participating Trio: [Defector, Random, Cooperator]

Simulation output:

(Defector, Random: 0.5) 5 [(D, C), (D, C), (D, D), (D, C), (D, D)]

(Random: 0.5, Defector) 5 [(D, D), (C, D), (D, D), (D, D), (C, D)]

(Defector, Cooperator) 5 [(D, C), (D, C), (D, C), (D, C), (D, C)]

(Cooperator, Defector) 5 [(C, D), (C, D), (C, D), (C, D), (C, D)]

(Random: 0.5, Cooperator) 5 [(D, C), (C, C), (C, C), (C, C), (D, C)]

(Cooperator, Random: 0.5) 5 [(C, C), (C, D), (C, D), (C, C), (C, D)]

Losing strategy: Cooperator

Strategies advancing to next round: Defector, Random

Round-2

Participating Trio: [Defector, Random, Grudger]

Simulation output:

(Defector, Random: 0.5) 5 [(D, C), (D, C), (D, C), (D, C), (D, D)]

(Random: 0.5, Defector) 5 [(C, D), (D, D), (D, D), (C, D), (D, D)]

(Defector, Grudger) 5 [(D, C), (D, D), (D, D), (D, D), (D, D)]

(Grudger, Defector) 5 [(C, D), (D, D), (D, D), (D, D), (D, D)]

(Random: 0.5, Grudger) 5 [(C, C), (D, C), (D, D), (C, D), (C, D)]

(Grudger, Random: 0.5) 5 [(C, C), (C, D), (D, D), (D, D), (D, C)]

Losing strategy: Random

Strategies advancing to next round: Defector, Grudger

Round-3

Participating Trio: [Defector, Grudger, TitForTat]

Simulation output:

(Defector, Tit For Tat) 5 [(D, C), (D, D), (D, D), (D, D), (D, D)]

(Tit For Tat, Defector) 5 [(C, D), (D, D), (D, D), (D, D), (D, D)]

(Defector, Grudger) 5 [(D, C), (D, D), (D, D), (D, D), (D, D)]

(Grudger, Defector) 5 [(C, D), (D, D), (D, D), (D, D), (D, D)]

(Tit For Tat, Grudger) 5 [(C, C), (C, C), (C, C), (C, C), (C, C)]

Losing strategy: Defector

Strategies advancing to next round: TitForTat, Grudger

Round-4

Participating Trio: [Defector, Grudger, TitForTat]

Simulation output:

(Defector, Tit For Tat) 5 [(D, C), (D, D), (D, D), (D, D), (D, D)]

(Tit For Tat, Defector) 5 [(C, D), (D, D), (D, D), (D, D), (D, D)]

(Defector, Grudger) 5 [(D, C), (D, D), (D, D), (D, D), (D, D)]

(Grudger, Defector) 5 [(C, D), (D, D), (D, D), (D, D), (D, D)]

(Tit For Tat, Grudger) 5 [(C, C), (C, C), (C, C), (C, C), (C, C)]

Losing strategy: Defector

Strategies advancing to next round: TitForTat, Grudger

Round-5

Participating Trio: [Bully, Grudger, TitForTat]

Simulation output:

(Bully, Tit For Tat) 5 [(D, C), (D, D), (C, D), (C, C), (D, C)]

(Tit For Tat, Bully) 5 [(C, D), (D, D), (D, C), (C, C), (C, D)]

(Bully, Grudger) 5 [(D, C), (D, D), (C, D), (C, D), (C, D)]

(Grudger, Bully) 5 [(C, D), (D, D), (D, C), (D, C), (D, C)]

(Tit For Tat, Grudger) 5 [(C, C), (C, C), (C, C), (C, C), (C, C)]

(Grudger, Tit For Tat) 5 [(C, C), (C, C), (C, C), (C, C), (C, C)]

Losing strategy: Bully

Strategies advancing to next round: TitForTat, Grudger

Final Round

Participating Trio: [Tit For 2 Tats, Grudger, TitForTat]

Simulation output:

(Tit For 2 Tats, Tit For Tat) 5 [(C, C), (C, C), (C, C), (C, C), (C, C)]

(Tit For Tat, Tit For 2 Tats) 5 [(C, C), (C, C), (C, C), (C, C), (C, C)]

(Tit For 2 Tats, Grudger) 5 [(C, C), (C, C), (C, C), (C, C), (C, C)]

(Grudger, Tit For 2 Tats) 5 [(C, C), (C, C), (C, C), (C, C), (C, C)]

(Tit For Tat, Grudger) 5 [(C, C), (C, C), (C, C), (C, C), (C, C)]

(Grudger, Tit For Tat) 5 [(C, C), (C, C), (C, C), (C, C), (C, C)]

Winning strategies: TitForTat, Tit For 2 Tats, Grudger

*Please note that the score for strategies were calculated based on ‘IPD Payoff Matrix’ presented in APPENDIX.*

1. *Tabu Search:* Tabu search is a metaheuristic search employing a local search method used for optimization. The inferior solutions can be placed in a forbidden list, known as tabu list so that revisiting those solutions can be avoided. In our experiments we apply tabu search along with the hill climbing search. We allocate a random list of strategies (S1, S2, ...Sn) to our subsets (SS1, SS2) from the complete set of strategies(S) we have. Then, we apply the Hill Climbing search (as detailed previously) to the first subset of strategies (SS1) in first level(L1). The highest scoring strategy from S1 is selected and promoted to play in the next level(L2). The worst performing strategy from SS1 is placed in a tabu list (TL). After this, we pick the next subset of strategies (SS2) and proceed to perform Hill Climbing search which would be the current level(L1). When we pick a current strategy, we go through the tabu list (TL) to know if the strategy is inferior based on prior validation done during Hill Climbing search with prior subset (SS1). If the current strategy is in tabu list (TL), the strategy is skipped. If not, the strategy is picked to participate in the tournament. At the end of Hill Climbing search, we will have the best strategy from the current subset (SS2). This strategy is promoted to play in the next level (L2). We can apply Hill Climbing search to select the best from the list of strategies promoted from prior levels. The winning strategies proceed to higher levels to compete and at the end result in a set of highly successful strategies. zzz When the last trio of strategies complete the tournament, the strategy (or strategies in case of tie) with the most points is deemed to be the best strategy from the set of strategies we started with.
2. *Tabu Search Experiment Results Strategy Set (S):* {Defector, Random, Cooperator, Grudger, TitForTat(TFT), Alternator, Bully, TitFor2Tats}

**Strategy Subset1 (SS1):** {Defector, Random, Cooperator, Grudger}

**Strategy Subset2 (SS2):** {Cooperator, Grudger, TitForTat(TFT), Alternator, Bully}

Level-1(SS1) Round-1

Participating Trio: [Defector, Random, Cooperator]

Tabu List (TL): empty

Simulation output:

(Defector, Random: 0.5) 5 [(D, C), (D, D), (D, D), (D, D), (D, C)]

(Random: 0.5, Defector) 5 [(C, D), (D, D), (C, D), (C, D), (C, D)]

(Defector, Cooperator) 5 [(D, C), (D, C), (D, C), (D, C), (D, C)]

(Cooperator, Defector) 5 [(C, D), (C, D), (C, D), (C, D), (C, D)]

(Random: 0.5, Cooperator) 5 [(D, C), (C, C), (D, C), (C, C), (C, C)]

(Cooperator, Random: 0.5) 5 [(C, C), (C, D), (C, C), (C, C), (C, D)]

Losing strategy: Cooperator

Strategy added to Tabu List (TL): Cooperator

Strategies advancing to next round: Defector, Random

Level-1(SS1) Round-2

Participating Trio: [Defector, Random, Grudger]

Tabu List (TL): Cooperator

Simulation output:

(Defector, Random: 0.5) 5 [(D, C), (D, C), (D, C), (D, C), (D, D)]

(Random: 0.5, Defector) 5 [(C, D), (D, D), (C, D), (C, D), (D, D)]

(Defector, Grudger) 5 [(D, C), (D, D), (D, D), (D, D), (D, D)]

(Grudger, Defector) 5 [(C, D), (D, D), (D, D), (D, D), (D, D)]

(Random: 0.5, Grudger) 5 [(D, C), (C, D), (D, D), (D, D), (D, D)]

(Grudger, Random: 0.5) 5 [(C, D), (D, D), (D, C), (D, D), (D, C)]

Losing strategy: Random

Strategy added to Tabu List (TL): none

Winning strategies of Level-1: Defector, Grudger

Level-1(SS2) Round-1

Participating Trio: [Cooperator, Grudger, TitForTat]

Tabu List (TL): Cooperator

Since Cooperator is in Tabu List, the strategy is skipped.

Losing strategy: Cooperator

Strategy added to Tabu List (TL): none

Strategies advancing to next round: Grudger, TitForTat

Level-1(SS2) Round-2

Participating Trio: [Grudger, TitForTat, Alternator]

Tabu List (TL): Cooperator

Simulation output:

(Alternator, Tit For Tat) 5 [(C, C), (D, C), (C, D), (D, C), (C, D)]

(Tit For Tat, Alternator) 5 [(C, C), (C, D), (D, C), (C, D), (D, C)]

(Alternator, Grudger) 5 [(C, C), (D, C), (C, D), (D, D), (C, D)]

(Grudger, Alternator) 5 [(C, C), (C, D), (D, C), (D, D), (D, C)]

(Tit For Tat, Grudger) 5 [(C, C), (C, C), (C, C), (C, C), (C, C)]

(Grudger, Tit For Tat) 5 [(C, C), (C, C), (C, C), (C, C), (C, C)]

Losing strategy: Alternator

Strategy added to Tabu List (TL): none

Strategies advancing to next round: Grudger, TitForTat

Level-1(SS2) Round-3

Participating Trio: [Bully, Grudger, TitForTat]

Tabu List (TL): Cooperator

Simulation output:

(Bully, Tit For Tat) 5 [(D, C), (D, D), (C, D), (C, C), (D, C)]

(Tit For Tat, Bully) 5 [(C, D), (D, D), (D, C), (C, C), (C, D)]

(Bully, Grudger) 5 [(D, C), (D, D), (C, D), (C, D), (C, D)]

(Grudger, Bully) 5 [(C, D), (D, D), (D, C), (D, C), (D, C)]

(Tit For Tat, Grudger) 5 [(C, C), (C, C), (C, C), (C, C), (C, C)]

(Grudger, Tit For Tat) 5 [(C, C), (C, C), (C, C), (C, C), (C, C)]

Losing strategy: Bully

Strategy added to Tabu List (TL): none

Winning strategies of Level-1: Defector, Grudger

Level-2(SS1 vs SS2 from Level-1) Final Round

Participating Trio: [Defector, Grudger, TitForTat]

Tabu List (TL): Cooperator

Simulation output:

(Defector, Tit For Tat) 5 [(D, C), (D, D), (D, D), (D, D), (D, D)]

(Tit For Tat, Defector) 5 [(C, D), (D, D), (D, D), (D, D), (D, D)]

(Defector, Grudger) 5 [(D, C), (D, D), (D, D), (D, D), (D, D)]

(Grudger, Defector) 5 [(C, D), (D, D), (D, D), (D, D), (D, D)]

(Tit For Tat, Grudger) 5 [(C, C), (C, C), (C, C), (C, C), (C, C)]

(Grudger, Tit For Tat) 5 [(C, C), (C, C), (C, C), (C, C), (C, C)]

Losing strategy: Defector

Strategy added to Tabu List (TL): none

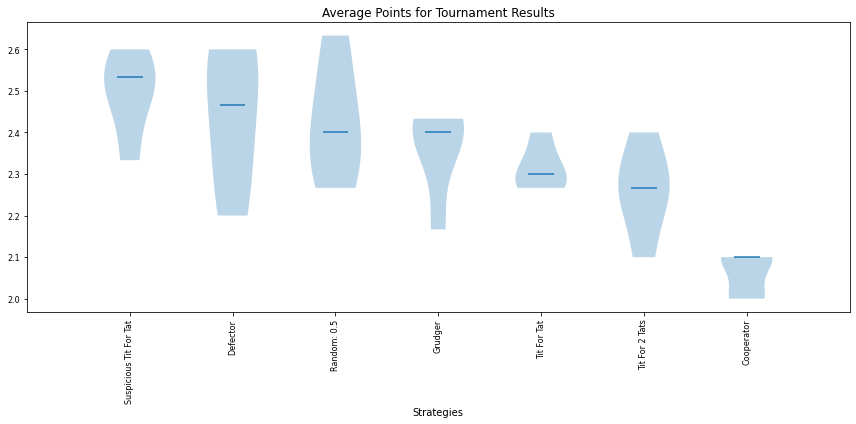
Winning strategies: Grudger, TitForTat

*Please note that the score for strategies were calculated based on ‘IPD Payoff Matrix’ presented in APPENDIX.*

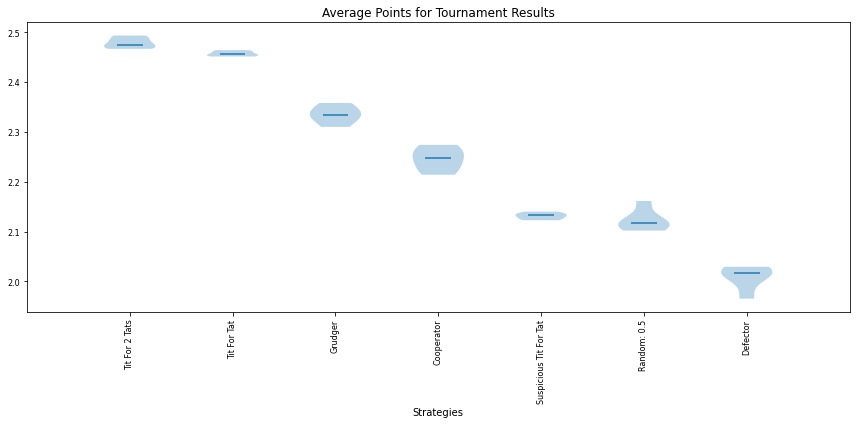
# DISCUSSION

Results for Exhaustive Search

For the round-robin with 5 turns and no noise, there are 2 main winners. The defector strategy was the tournament winner 50% of the time and suspicious-tit-for-tat (STFT) won 40%. This is logical because a defector player would be able to take advantage of all cooperative strategies as there aren’t enough rounds for the defector strategy to be punished. Interestingly the average score of STFT was higher than the defector. Since few rounds are played there is a high variance in average scores.

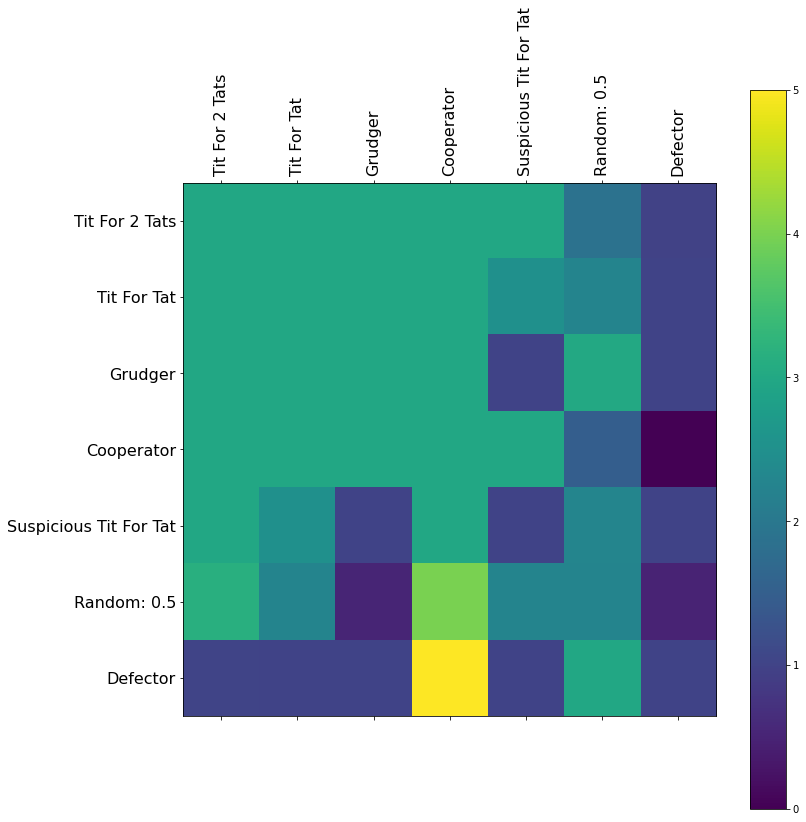
*(figure 1: average points for strategies with a tournament of 5 turns)*

For the round-robin with 15 turns and no noise, there are again 2 main winners. The Tit for tat strategy won the tournament 60 % of the time and Tit for 2 tats won 40 % of the time with less variance than tit for tat. As more rounds are added, the cooperative strategies are starting to punish defector strategies. With Defector having the worst average score overall. For the round-robin with 250 turns and no noise after 10 simulations, there is 1 clear winner. Tit for 2 tats seems to be the most dominant with a tournament win rate of 100%.



*(Figure 2: average points for strategies with the tournament of 250 turns and no noise)*

  Comparing figure 1 to figure 2 we can see as rounds are increased the variation decreases and we are receiving a true representation of what strategies are good. A pattern is starting to emerge, as more rounds are added the most effective strategies have the same property of being cooperative. This pattern can be seen visualized by figure 3. The cooperating strategies such as TFT, TF2T, and Cooperator are shown to have better scores with each player on average (the lighter the box the higher the score) when they are playing against each other. In summary when rounds are unknown it seems likely that the best strategies are: TFT, TF2T and Grudger.



*Figure 3: Average score between other strategies*

Results for Hill Climbing and Tabu Search

.

# CONCLUSION

##### References

1. S. Jurisiˇc, Marko, et al. “A Review of Iterated Prisoner’s Dilemma Strategies.” *CiteSeerX*, 16 July 2012. https://www.egr.msu.edu/~kdeb/papers/k2006002.pdf. accessed March. 05, 2022).
2. Placement
3. K Placement
4. J. Placement
5. Placement
6. Placement
7. .
8. D. Placement