Prisoner's Dilemma with an Adapting Population

Aaron Vanzyl

1 Introduction

The prisoner's dilemma is a competetive game played between two agents. Each agent may choose to cooperate or defect. If both cooperate, they receive a medium reward. If one defects while the other cooperates, the defector receives a large reward and the cooperate receives nothing. However, if both defect, they receive only a small reward.

The best action to take in a single round depends on what the opponent will do. Agents can choose to cooperate and share the rewards, or risk defecting for a potentially higher reward - or a much lower reward if the opponent does the same.

If only a single round is played between two agents, defecting can be considered an optimal strategy according to the Nash equilibrium. Regardless of what the opponent does, an agent that defects will always get a higher reward than an agent that cooperates in the same situation.

This extension seeks to explore two changes to the scenario:

1. Agents compete for several rounds, and can change their action based on the opponent's action in the last round.
2. Agents do not compete against a single unknown opponent, but instead exist within a population of other agents. Each agent competes against the rest of the population. Over time, an agent may adjust its strategy in response to the rest of the population.

These changes aim to reflect how people within a group may change to respond to each other over time. For example, a person within a trustworthy group may change over time to adopt to the cooperative behaviors of other members. On the other hand, a person in a group of defectors may decide that is best to defect in all situations. Or, possibly, people within a population may respond to each other in a more complex, non-linear pattern.

2 Model Description

The model consists of two different simulation components.

The first is the agent vs agent prisoner's dilemma simulation. This determines what reward is expected when agent A and agent B compete against each other in a prisoner's dilemma scenario, with each using their own strategies. This simulation is carried out

using Markov chains to determine the probability of agents being in certain states during certain rounds.

The second component is agent optimization through simulated annealing. A population of agents is maintained. During each growth iteration, agents in the population change their strategies to better perform against other agents within the population. This change is done through simulated annealing.

2.1 Agent vs Agent Prisoner's Dilemma Simulation

There are two actions that agents can take each round: cooperate (C) and defect (D).

The original prisoner's dilemma model defines the following reward matrix:

	Cooperate	Defect
Cooperate	2,2	0,3
Defect	3,0	1,1

For the extended model, we will define several variables.

2.1.1 Variables

P: 2×2 payoff matrix. $P_{i,j}$ = payoff supposing that player performs action i and opponent performs action j. In the typical prisoner's dilemma scenario, the payoff is asymmetrical. So P_A is the payoff matrix for the row agent and P_B is the matrix for the column agent.

In a typical scenario, the payoff matrices are:

$$P_A = \begin{pmatrix} 2 & 0 \\ 3 & 1 \end{pmatrix}$$

and

$$P_B = \begin{pmatrix} 2 & 3 \\ 0 & 1 \end{pmatrix}$$

M_A: 2×2 Markov transition matrix for agent A. $M_{A,i,j}$ = probability that agent A will take action j this round, given that the opponent took action i in the previous round. It is necessary that $M_{A,i,1} + M_{A,i,2} = 1$. i.e. the probabilities of choosing action 1 (cooperate) and action 2 (defect) must sum to 1. This will be adjusted over time through simulated annealing (in the next section).

For example, an agent that has a 0.9 chance to copy its opponent's previous action would have the matrix:

$$M_A = \begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix}$$

C_A: 2×1 Matrix containing the initial probability of cooperating or defecting for agent A. This will be adjusted over time through simulated annealing (in the next section).

$S_{A,n}$: 2×1 matrix containing the state (probability of cooperating or defecting) of agent A in round n. For example, suppose that during round 3, agent A has a 0.7 chance of cooperating and a 0.3 chance of defecting. Then:

$$S_{A,3} = \begin{pmatrix} 0.7 \\ 0.3 \end{pmatrix}$$

Each agent's action during the first round is determined by C, so

$$S_{A,0} = C_A$$

$R_{A,n}$: Numeric value of the expected reward obtained by agent A in round n.

2.1.2 Round Updates

State update
We apply each player's transition matrix to the opponent's previous state.

$$S_{A,n+1} = (M_A)^T \cdot S_{B,n}$$
$$S_{B,n+1} = (M_B)^T \cdot S_{A,n}$$

Reward calculation
We use the intermediary matrix X which is a 2×2 matrix such that $X_{i,j}$ is the probability that both agent A takes action i and agent B takes action j.

$$X = S_{A,n} \cdot (S_{B,n})^T$$
$$R_{A,n} = X * P_A$$
$$R_{B,n} = X * P_B$$

The expected reward R is calculated deterministically. The actual reward obtained when two agents compete would be stochastic - but the value here represents the expected reward that would be obtained over a large number of repeated competitions.

2.2 Agent Optimization Through Simulated Annealing

The population is adjusted over time through simulated annealing. Agents within the population compete against each other, and adjust their strategies to maximize their reward.

In each simulated annealing iteration, for each agent in the population:

1. The agent competes against each other agent in the population. The mechanics

of this are described in the previous section. This produces a single numeric value: the agent's average expected reward against the rest of the population. This value will be called the agent's fitness.

2. A candidate agent is generated. This agent is generated by randomly modifying the agent's starting action, and its transition matrix, C_A and M_A respectively. Each value is randomly modified by a uniform value from 0 to step size.

3. The candidate agent competes against each other agent in the population. This is the same process that the prior agent goes through in step 1.

4. If the candidate agent has a higher fitness than the prior agent, it is added to the new population. Otherwise, the metropolis criterion is used to decide if the candidate or the original agent moves forward into the new population. The probability is:

$$P = e^{\frac{\tilde{F}-F}{T}}$$

where F is the fitness of the original agent and $\tilde{F}$ is the fitness of the candidate agent. T is the temperature of the model, which is reduced each iteration by multiplication with the cooling factor.

This process is repeated for several iterations, allowing agents to gradually adjust their strategies in response to each other.

2.3 Parameter Values Chosen

These values will be held constant in all test cases in the report.

Prisoner's dilemma simulation parameters:
- rounds = 10

Simulated annealing parameters:
- population size = 10
- annealing iterations = 150
- annealing step size = 0.1
- starting temperature = 1
- cooling factor = 0.8

These values are chosen to represent a relatively small population that starts with quickly changing strategies, but gradually cools down to only making more calculated changes.

The initial agent strategies and the prisoner's dilemma reward matrix will not be kept constant. These values will be adjusted in order to answer the research questions. The specific values chosen are described in each subsection of the results.

3 Results

3.1 Effects of Starting Conditions

3.1.1 Neutral Starting Conditions

First, we look at a situation with relatively neutral starting strategies. Agents begin with a roughly 50/50 chance of cooperating or defecting - both in the first round and in response to any opponent action. Specifically, for each agent, the initial probability of cooperating in any situation is set to a random value sampled from a normal distribution with mean = 0.5 and standard deviation = 0.25.
We conduct 9 trials with randomized starting populations (as described above).

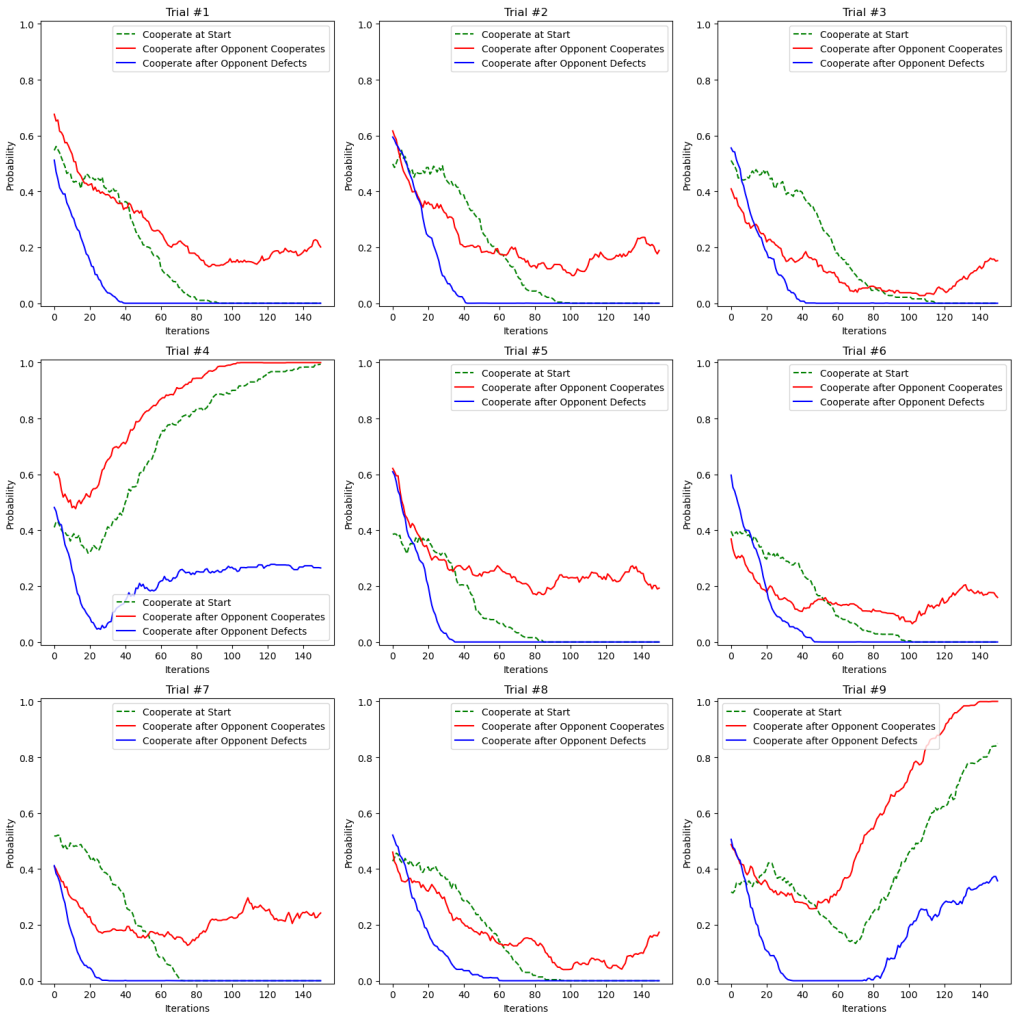

Figure 1: Agent Strategies over Time with Neutral Starting Conditions

The above figure contains 9 different subcharts, each representing a trial with slightly different starting conditions. Each subchart has 3 lines. These lines show average values of all agents within the population over time. The dashed green line shows the probability of cooperating during the first round. The solid red line shows the probability of cooperating after the opponent cooperated in the previous round. The solid blue line shows the probability of cooperating after the opponent defected in the previous round.

Two main outcomes can be seen: either the population collapses into always defecting, or - less frequently - the agents learn to cooperate in response to other agents cooperating.

In most cases, the population collapses into all defection. Agents fairly quickly learn to defect if their opponent defected in the previous round (represented by blue line dipping). This aligns with natural intuition: if the opponent defected last round, then there's no point trying to cooperate this round. This also aligns with the Nash equilibrium, which suggests that defecting is the dominant strategy. Interestingly, agents begin by learning to defect in response to the opponent cooperating (shown by the red line dipping). But, after a point, the response to cooperation either plateaus or returns to neutral. This happens because the population has collapsed into always defecting, At this point, the agents' response to an opponent cooperating no longer matters because that situation will never arise. Thus, the cooperation response (red line) has no effect on the agent's expected reward and will drift randomly.

Alternatively, in some cases, the population settles on mutual cooperation. This can be seen in trials 4 and 9. It is not immediately obvious what causes this to happen. In the following sections, the starting agent strategies will be altered to see if this outcome can be more reliably reproduced.

3.1.2 Cooperation-Biased Starting Conditions

In this section, we look at a situation with agents starting biased toward cooperation. Agents begin with a roughly 80% chance of cooperating or defecting - both in the first round and in response to any opponent action. Specifically, for each agent, the initial probability of cooperating in any situation is set to a random value sampled from a normal distribution with mean = 0.8 and standard deviation = 0.25.
As in the previous section, we conduct 9 trials with randomized starting populations.

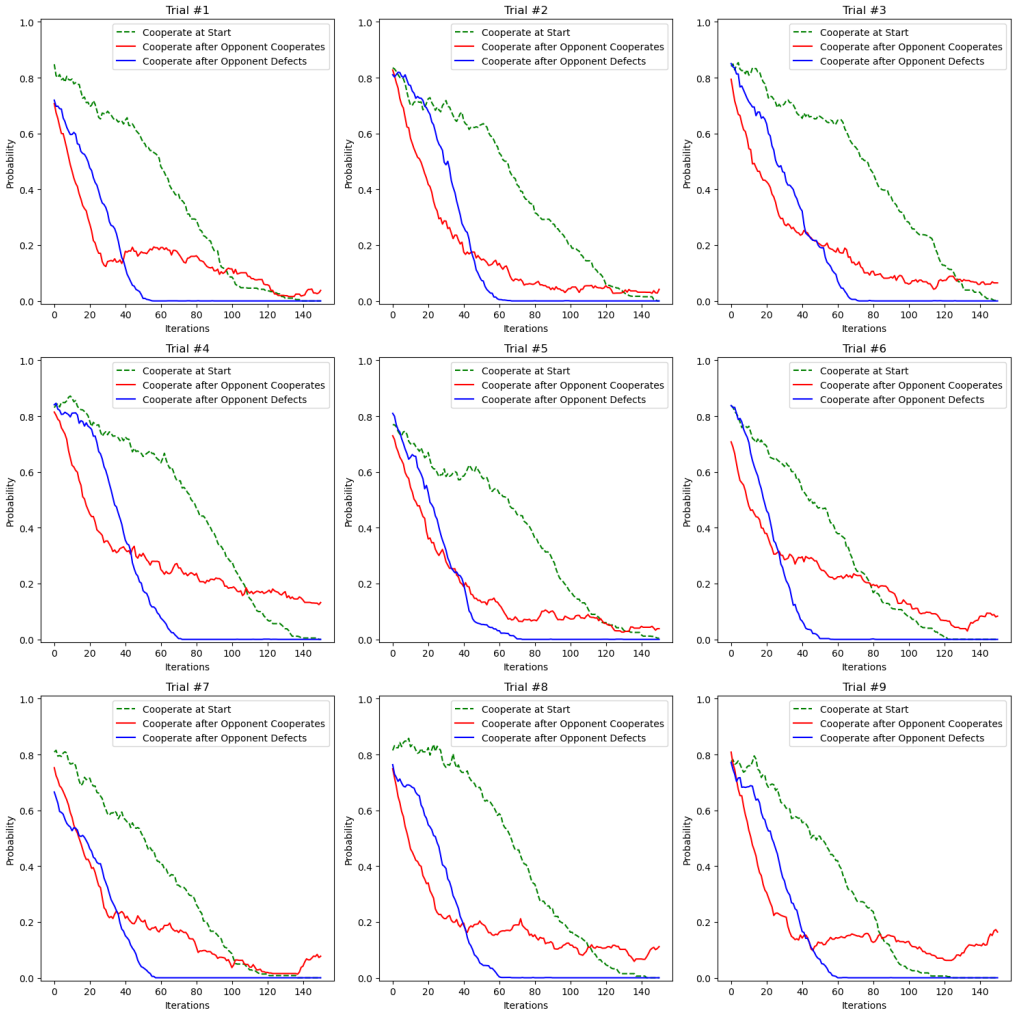

Figure 2: Agent Strategies over Time with Cooperative Starting Conditions

Biasing the population toward cooperation at the start does *not* influence the population to converge into mutual cooperation. Perhaps this is because the initial population is too trusting, even in response to defection. Each agent quickly learns that it can gain a higher reward by defecting with no consequences, and then the whole population collapses into defection.

3.2 Effects of Reward Matrix

3.2.1 Higher Mutual Cooperation Reward

In this section, we will evaluate the effects of offering a higher reward for cooperation. The following reward matrix is used:

	Cooperate	Defect
Cooperate	2.75, 2.75	0,3
Defect	3,0	1,1

The reward for mutual cooperation is raised from 2 to 2.75. This makes mutual cooperation almost as beneficial as defection into cooperation (2.75 vs 3).

We conduct 9 trials with neutral starting strategies and this new cooperation biased reward matrix.

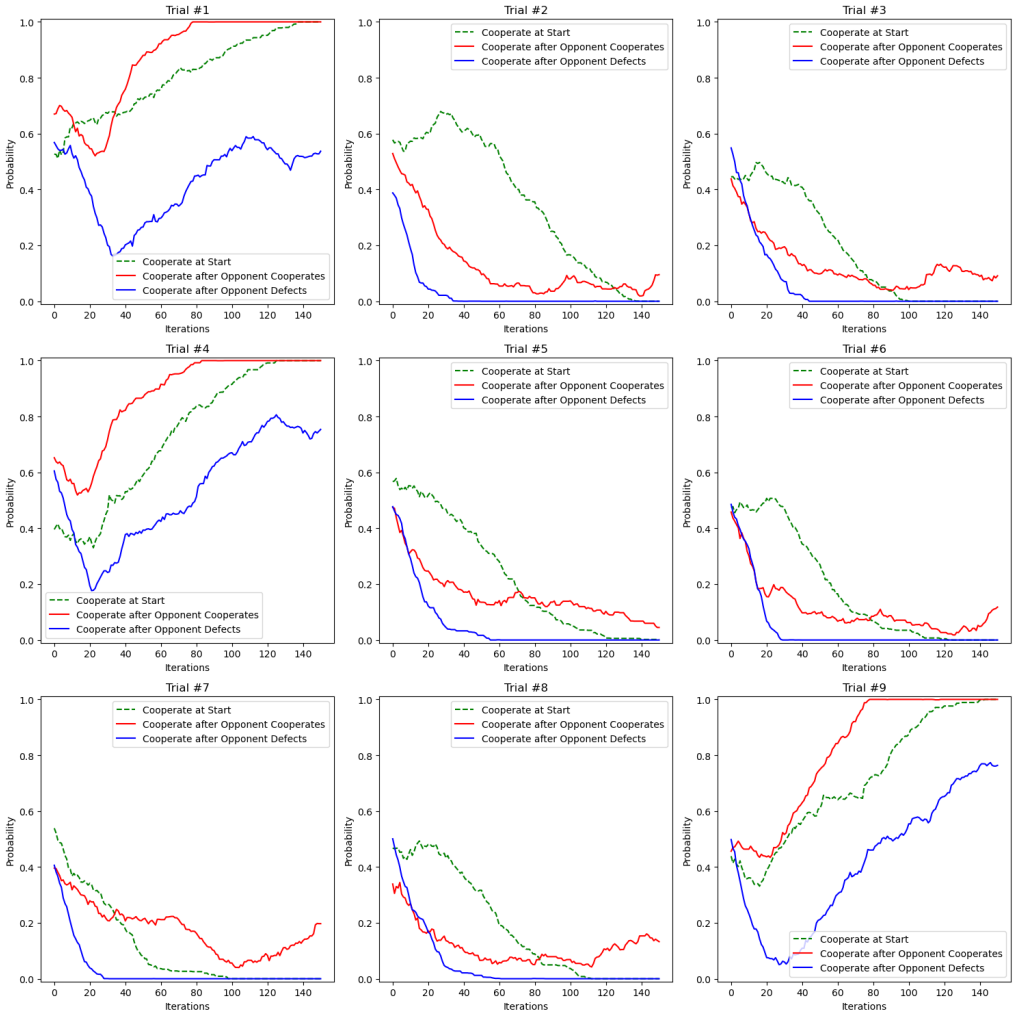

Figure 3: Agent Strategies over Time with Increased Mutual Cooperation Reward

The increase in mutual cooperation reward seems to increase the likelihood of convergence on mutual cooperation, but the outcome is not very consistent. Even though the relative incentive for defection is quite small (only 0.25 more reward than mutual

cooperation), it is still enough to cause a collapse into complete defection in many cases. Greedy little guys.

3.2.2 Mutual Defection Penalty

In this section, we alter the consequences of both agents defecting. If both agents defect, they are instantly vaporized by a gigantic hyper beam. This is represented by a -100 reward. We use the following reward matrix:

	Cooperate	Defect
Cooperate	2, 2	0, 3
Defect	3, 0	-100, -100

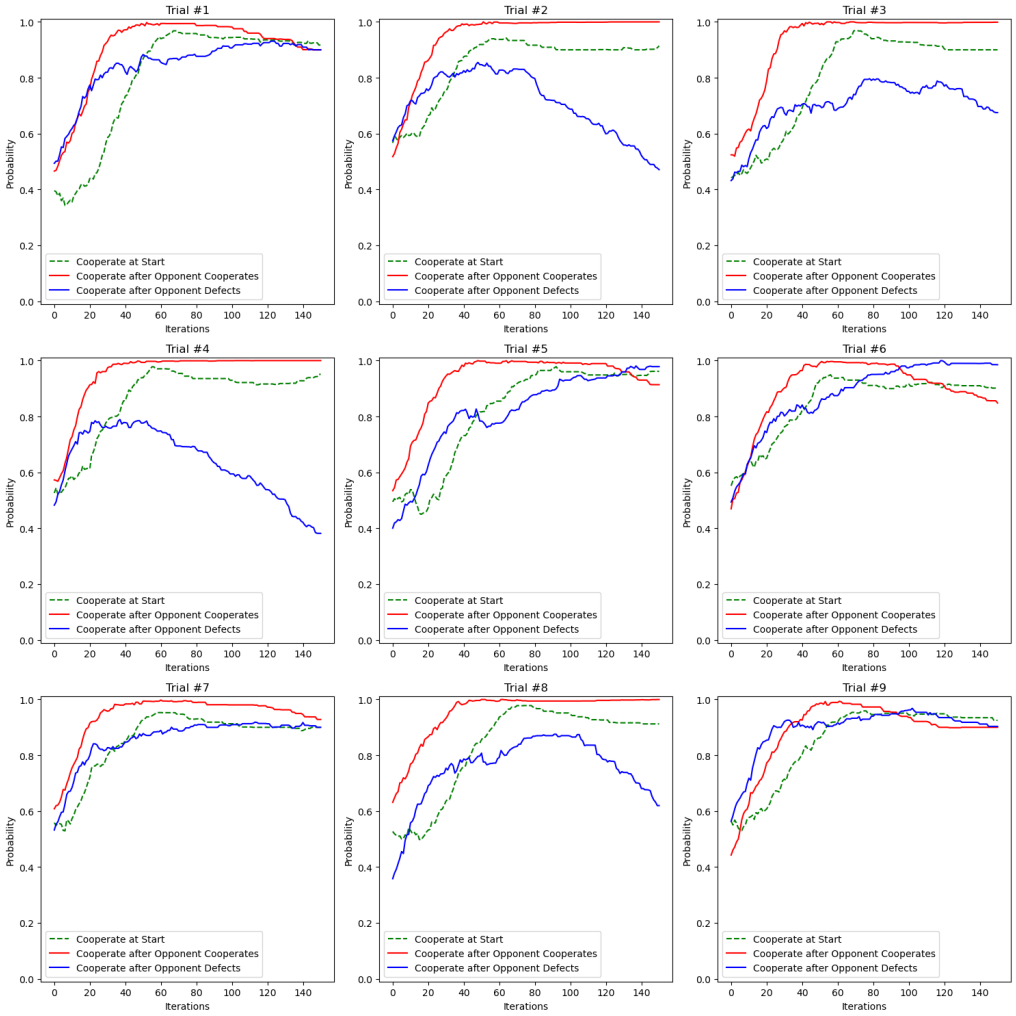

Figure 4: Cooperate or Die

Now the agents learn to cooperate (:

In a similar pattern to section 4.1.1, the response to defection (blue line) starts to drift randomly after a while since the agents are almost always cooperating.

This is more effective than adjusting the starting strategies of the agents. However, it does also change the Nash equilibrium as defecting no longer strictly dominates cooperation. Cooperation would now result in a 0 reward if the opponent defects, which beats mutual defection's -100 reward.

3.3 Conclusion

Adjusting the starting strategies of the population does not consistently cause the population to converge to a certain outcome. Moderately changing the reward matrix also does not consistently cause the population to converge to a certain outcome. But, with a sufficiently large change to the reward matrix, such as the threat of complete annihilation, the population can be effectively directed to do anything (: