Investigating Social Trends in the Iterated Prisoner's Dilemma

A Thesis Presented in Partial Fulfillment of
the Honors Bachelor's Degree

Jessica Finocchiaro

Abstract

In ethics, many academics make the assumption that all people want to be good. Evil comes in where there is a conflict of "good" decisions, where a decision that is good for one person contradicts the good of another. In this case, a person will make a different decision depending on their definition of the "good" they want to accomplish. In a society that starts with an equal proportion of "selfishly good" and "selflessly good" people, we aim to investigate the convergence patterns motivations through simulation of populations playing the Iterated Prisoner's Dilemma over time.

Department of Computer Science
Under the supervision of Dr. H. David Mathias
Florida Southern College
May 2017

Contents

Acknowledgments

I'd like to thank my professors, friends, and family for their support over the past four years at Florida Southern College. Specifically, I would like to thank Dr. H. David Mathias for advising my thesis and pushing me to be the best student I can and to believe in myself.

1 Introduction

Prisoner's Dilemma is a non-zero-sum game in which two players must make one of two decisions. Each player can either cooperate with the other, or they can defect from the other. Both players make their decision simultaneously without any knowledge of the other player's decision. Mutual cooperation brings the highest communal payoff, but not necessarily the highest personal payoff. Thus, cooperation comes at a personal cost. A chart of payoffs is seen in Table 1. The first number is the payoff for player 1, and the second is the payoff for player 2.

P1 \ P2	Cooperate	Defect
Cooperate	3 : 3	0 : 5
Defect	5 : 0	1 : 1

Table 1: Scoring for the version of IPD in this work.

Prisoner's Dilemma was originally framed by Merrill Flood and Melvin Dresher, working at RAND, an American think tank, in 1950. Albert Tucker later formalized the game and called it the "Prisoner's Dilemma."

Iterated Prisoner's Dilemma is arguably the most well-known problem in Evolutionary Game Theory. In our research, we investigate multi-objective optimization in the Iterated Prisoner's Dilemma amongst a population of individuals with different objectives. Special cases of the game have emerged, but the game originally was created shortly after World War II with the escalating arms race in mind as nuclear weapons had just been unleashed a few years before.

Consider the Cold War: by its end, the Soviet Union was spending over one-fourth of its Gross National Product on defense. Mutually Assured Destruction (MAD) suggests that both countries spent absurd amounts of money on weapons, and if they had been used, it would have resulted in not only the annihilation of the opponent,

but self-annihilation as well. While it's best for the United States to prepare 100 nuclear weapons and the USSR prepares none (and the converse), the best outcome for both countries is if both cooperate and neither prepares any nuclear weapons for use.

One of the main scenarios we considered designing our model was the cost of "going green." In order to slow and reduce the degradation of the Earth's resources, countries need to cooperate with each other, since emissions in one country still affect the atmosphere over other countries. Enacting environmental policies, however, is often expensive in the short-term. Many developing countries cannot afford to industrialize with the most environmentally friendly policy, and there has been an ongoing debate on whether countries should develop in a "green" manner. Our model considers the group benefit of saving the planet, while members still want to do well personally from an economic standpoint.

In our research, we have designed each member of the population to try to accomplish two tasks, chosen from the following:

1. Maximizing their own score
2. Minimizing their opponent's score
3. Maximize their opponent's score
4. Maximize mutual cooperation

Some of these objectives are conflicting. For example, minimizing your opponent's score and maximizing your opponent's score. Because of this, we have made options for members to aim to accomplish the following pairs of tasks shown in Table 2.

The problem we plan to investigate is how members of a population learn from each other. If a selfish person plays against a selfless person and defects on them

Name	Max Own	Min Opp	Max Opp	Max Co-op
Selfish	•	•		
Communal	•		•	
Cooperative	•			•
Selfless			•	•

Table 2: Objective pairs used in experiments reported.

every time, will that selfless person, after achieving no mutual cooperation, realize they didn't do well? And after not doing well, will they change their objectives and learn to play more selfishly?

2 Related Works

Robert Axelrod wrote the original work [1] that designed an Iterated Prisoner's Dilemma as a tournament-styled learning game amongst members of a population with different strategies. Many works have been done evaluating single objective evolutionary learning to maximize score, but Mittal and Deb, in [2], were the first to approach the problem using Multi-objective Optimization. Other approaches have included coevolutionary learning and spatially restricted selection of players in the Iterated Prisoner's Dilemma. Areas of future work may include temporal punishment and its effect on learning for members of the population.

In a scenario where players are (to their assumption) playing infinitely many games, Axelrod proposed the research question of what the optimal solution to Iterated Prisoner's Dilemma is. He found that often depended on history of the past few moves. However, when he originally proposed the experiment, he had members of his population play each other 150 times each and calculated the maximum score a player scored on average against an opponent. Members of his population, to whom we compare our members in our experiments, include:

- Always comply
- Always defect
- Tit-for-tat: Comply on the first move, and every move after, do what your opponent did
- Suspicious tit-for-tat: Same as above, but defect on the first move
- Pavlov: Comply on the first move, then defect if there was disagreement on the previous move
- Spiteful: Comply until the opponent defects, then always defect
- Random
- Periodic CD: Alternates playing C and D
- Periodic DDC: Plays D, D, and C in a cycle
- Periodic CCD: Plays C, C, and D in a cycle
- Tit for two tats: Cooperates on the first move then defects when the opponent has defected on the past two moves
- Soft majority: Complies as long as the number of times the opponent complies is greater than or equal to the number of times they have defected.
- Hard majority: Complies as long as the number of times the opponent complies is greater than to the number of times they have defected.
- Hard tit for tat: Cooperates on the first move, then defects if the opponent has defected on any of the three previous moves.

- Naive prober: Like tit for tat, but defects at a probability of 0.01
- Remorseful prober: Like naive prober, but tries to break the series of mutual defections after defecting.

Axelrod's tournament included every member of his population playing every other member of the population, and tit-for-tat was the most successful strategy. From then on, tit-for-tat became the standard to beat for the evolutionary algorithms that came to follow. On average, tit-for-tat scored a little less than three points per round, and so three points per round is our baseline. The only way tit-for-tat was exceeded was by using evolutionary algorithms, at first with one objective, but Mittal and Deb's experiment [2] with multi-objective optimization further beat the average with almost three points (2.987) per round.

Shashi Mittal and Kalyanmoy Deb [2] wrote a paper that applies Multi-objective Optimization to the Iterated Prisoner's Dilemma problem. For the most part, our work extends theirs. They designed members to make a decision based on the history of the past three moves between their player and their opponent, playing members of the population 150 times each, round robin style. In this paper, Mittal and Deb tried to find the optimal solution when every member of the population was trying to maximize their score and minimize their opponent's score. They conducted another experiment with different, but very similar objectives, but found better results and a stronger pareto-optimal solution with their first experiment. In our research, we depart from Mittal and Deb by initializing a population of members with different objectives. Depending on interactions with other members, the population evolves the numbers of members with the different objective pairs.

We conduct a handful of different experiments in this research. In one experiment, we allow members to change their objectives in order to be more successful. In

the second experiment, we disallow switching objectives, and we are evaluate if one specific subpopulation is more successful than the others.

Many of the related works from the literature apply coevolution of two different populations in Iterated Prisoner's Dilemma, like [3], [4], [5], [6], [7], [8], [9], [10], [11], which would be another interesting experiment we ran. Coevolution is where subpopulations are evolved alongside each other, not in competition, but in a symbiotic relationship. This is typically done by training one population on the best member of another population from the previous generation. Applications using coevolution include "solutions" for war theory of missile defense, while another compares different types of coevolution on success at the game Othello. Additionally, Tanimoto [4] expanded research in the field to use coevolution to find a solution for a family of 2x2 games, including Iterated Prisoner's Dilemma, as well as Leader and Hero.

Jaskowski, Liskowski, Szubert, and Krawiec [8] present work that could be used in an expansion of our research interests by using non-uniform coevolution to evolve strategies for different games. Using the game Othello, Jaskowski *et. al.* compares 5 different types of evolutionary algorithms. The first algorithm learns to play against a population of randomly generated individuals, called the RSEL population. The 1SEL population is trained on members of its same population, which is what we use in the majority of our experiments. Their 2SEL population is what is commonly known as coevolution: one population is trained against the best member of another population and the other population is trained against the best member of the first, back and forth. 1SEL-RS and 2SEL-RS are the same as their respectively named populations, but half of the 1SEL-RS is trained within the population and half is trained on random members. The same follows for 2SEL-RS.

John J. Nay and Yevgeniy Vorobeychik wrote the paper "Predicting Human Cooperation" [12], presenting a computational model for predicting human behavior in the

Iterated Prisoner's Dilemma. Their model is constructed in two pieces, the first being on the first-period action based solely on the game's parameters, while the second is constructed to predict dynamic actions using history in addition to game parameters. Their dataset includes samples from over 160,000 individual games played by human participants. They calibrate a model for individual behavior prediction and fine-tune it for aggregate behavior prediction. They found that after defecting, a person tends to cooperate, almost as a way of trying to "make up" for their defection. Their paper also investigates the "inertia" of decision making by people and how patterns of decision making are developed. This paper develops and actualizes some of the applications used in our model.

A handful of other papers have Electrical Engineering applications, rather than Game Theory applications. Wu, for example in [13], writes on making dynamic fitness adjustments for processor management, and other applications include program scheduling.

3 Variations in the Prisoner's Dilemma Game

As stated earlier, the Prisoner's Dilemma game is a non-zero-sum simultaneous game, meaning that the payoffs received by each player are not necessarily reciprocal of the other player's payoff. Therefore, it is possible for both players to do well, as well as possible for both to do poorly. Zero-sum games occur when one player's outcome is the opposite of their opponent's. This is exemplified in Poker, when the money that one player wins is taken directly from another player. Being simultaneous implies that both players make their decision at the same time without knowledge of the decision their opponent is making.

Earlier, we presented the payoff table we use in our experiments given concrete

numbers. However, the Prisoner's Dilemma game can be played with a more general payoff table, shown in Table 3.

P1 \ P2	Cooperate	Defect
Cooperate	R : R	S : T
Defect	T : S	P : P

Table 3: Generic payoff table.

Note that $T > R > P > S$. It is noted in the paper by Deb and Mittal that T stands for Temptation to defect again, R for Reward for complying, P for Penalty faced for mutual defection, and S for Sucker. Because $T > R$ and $P > S$, we can say that defection is the dominant strategy, taking just the player's outcome into consideration. However, defecting every time will most likely not yield a result as high-scoring as a combination of Complying and Defecting, since compliance by a player gives their opponent an incentive to comply at times instead of always defecting.

In most simulated variations of Iterated Prisoner's Dilemma, all of the members of the population have the same objective and Axelrod's population is used as the canonical training set. Our model adds the variation of member objectives, and while we train our members against Axelrod's population, we also conduct experiments training members against each other as well as a combination of the two populations.

4 Genetic Algorithms

In our research we utilize *Genetic Algorithms*, modeled after Darwinian evolution. Genetic algorithms are approximation algorithms to calculate either an approximation of the best solution, or a pareto-optimal solution to a problem. In our research, we are interested in finding a pareto-optimal best "solution" to succeeding in the Iterated

Prisoner's Dilemma in a mixed population, measured by their success at the objectives they are trying to achieve. Our experiment, because we use members with different objectives, is unique because it has four different pareto-optimal fronts, one for each set of objectives. Though objective pairs differ, scores are scaled to allow comparison of individuals without regard for their objectives.

The reason is use genetic algorithms in different problems varies. In our problem, there is no definitive correct next move since there is no way of predicting human behavior with 100% certainty. However, these approximation algorithms are also often applied to problems that are considered NP-Hard. NP-Hard problems mean they cannot be solved in polynomial time, so computing an exact answer would take longer than a lifetime. In most of these cases, we can approximate solutions using different strategies, one of which includes genetic algorithms.

In a genetic algorithm, a population containing members is initialized. Through the evolutionary steps a new, more successful population is generated, and the steps are repeated for many (usually 5,000 or 10,000 in our experiments) generations. The essential steps of a genetic algorithm include reproduction, mutation, fitness evaluation, and selection over multiple generations, each of which will be explained in detail below. No changes to the genome of parents are made in the reproduction or mutation stages, but they create and alter children. Fitness evaluation and selection do not change the genome of any members.

4.1 Crossover

Reproduction, or *crossover*, generally occurs with a relatively high probability (greater than or equal to 0.5) in most applications of genetic algorithms. There are three commonly used methods of crossover: one point, two point, and uniform crossover. In

our experiments, we utilize one point crossover. In one point crossover, depicted in Figure 1, with the predetermined probability, we select a number at random in the range of the genome length and switch the genome of two members after this point. In two point crossover, two numbers are generated and the genome between those two points is switched between the two parent members in the children. In uniform crossover, reproduction happens bit by bit within the child genome. The child inherits from parent A at probability p and from parent B at probability (1-p) in order to construct the child's genome. p is usually set to 0.5, so the child is expected to be composed of half of each parent's genome. Figure 1 is an example of one-point crossover, where the point in the genome to switch is indicated by the color change of the text.

While we don't take advantage of this, if the situation permits, crossover may consider spatial information. In our experiments, crossover occurs between the next two players in the population list. However, niching techniques like those found in [14] simulate reproduction of nearby members of the population.

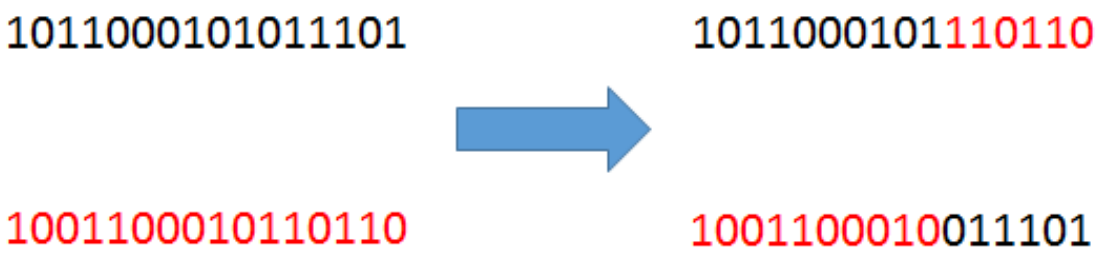

Figure 1: Example of one-point crossover (left: parents, right: children)

4.2 Mutation

Mutation can occur given a certain mutation probability. However, different methods of mutation can be used to mutate real valued genes by different amounts. In our experiment, we mutate a gene by flipping the bit value, or switching it from 0 to 1 or 1 to 0 with a mutation probability of 1/70, so there is 1 bit expected to be flipped in every child. Figure 2 is an example of a bit-flip mutation, as used in our experiments.

Figure 2: Example of bit-flip mutation

4.3 Fitness

Fitness evaluation makes a model unique with each application. Like Darwinian evolution, the term "survival of the fittest" is taken literally in computation. Fitness measures an algorithm's success at optimizing the objectives given. In single-objective optimization, this looks like a maximization or minimization problem, as demonstrated in Figure 3, learned over time measured in generations. Figure 3 shows the fitness of an algorithm on the y-axis over time, shown on the x-axis in generations. The growth of the fitness function is marked with small spikes because genetic algorithms are approximation algorithms, and fitness may vary from trial to trial. The overarching trend of growth, however, is what is essential to learning.

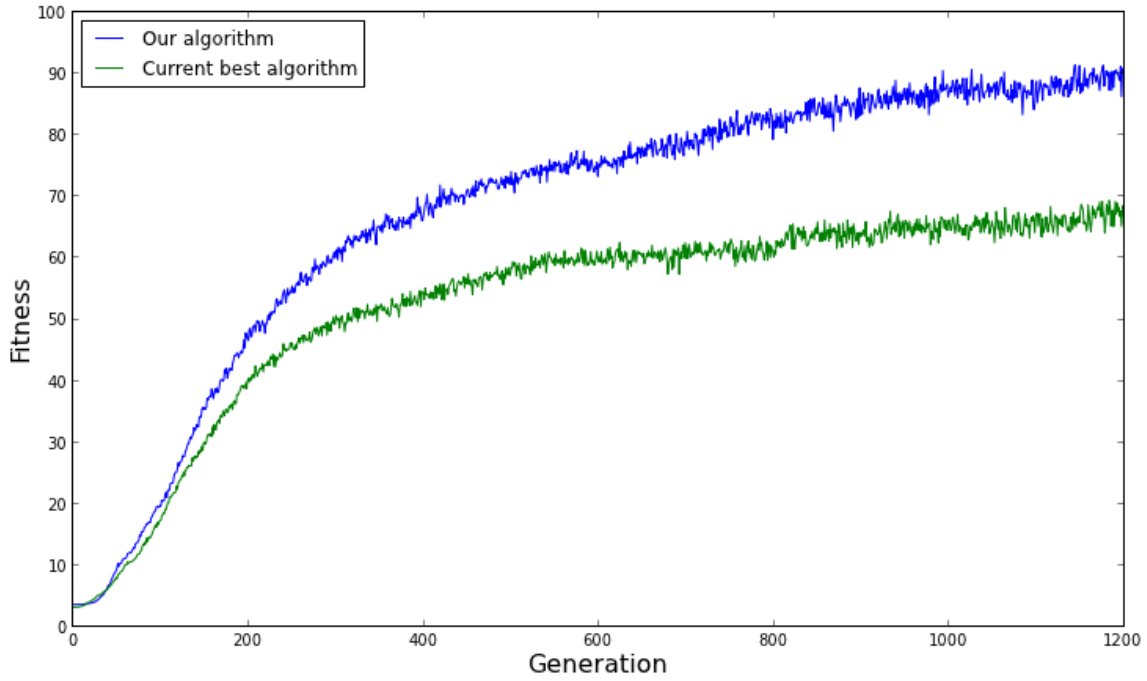

Figure 3: Example of single objective fitness

However, in a multi-objective problem, like ours, we aim to maximize (or minimize) two or more objectives. Measuring fitness in a multi-objective problem yields an n-tuplet, where n is the number of objectives in the problem. Trade-offs are often required in multi-objective optimization if objectives conflict in some way with each other. Figure 4 shows fitness as an ordered pair in two dimensions and shows how there are multiple ways to achieve an optimal solution, here called pareto-optimal. A pareto-optimal solution allocates resources so that it is impossible to reallocate in order to make any one individual or preference criterion better off without making at least one individual or preference criterion worse off.

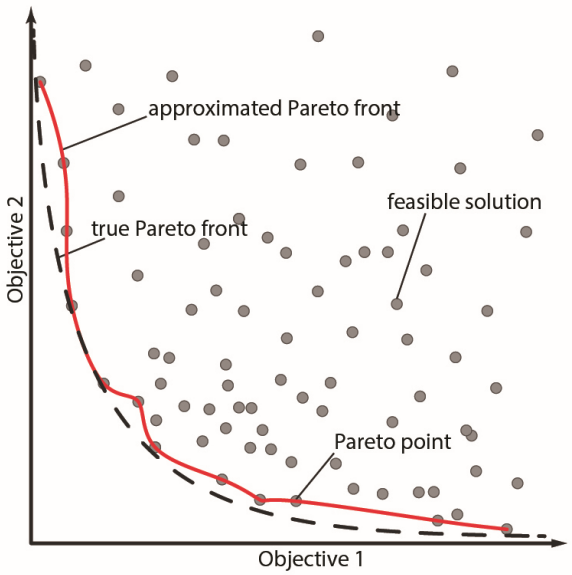

Figure 4: Example of a pareto front

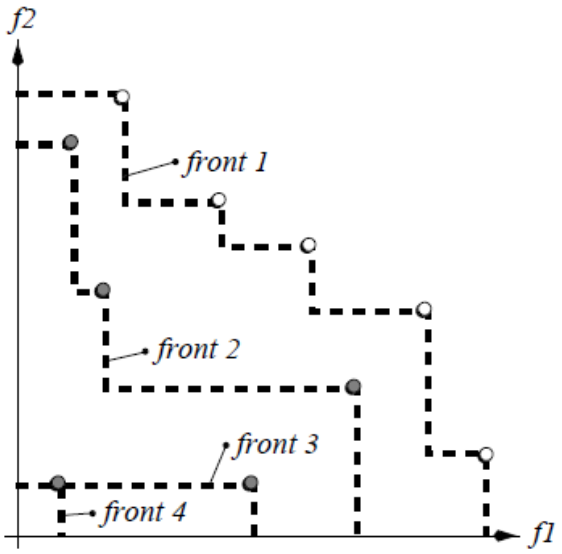

Figure 5: Labelled fronts in a maximization problem

4.4 Selection

Selection is the process of choosing which members of the population survive to the next generation. In a single-objective problem, this is as simple as choosing the top x percent of the population. In a multi-objective optimization problem, selection becomes more difficult. If a member of the population is excellent at one objective, but bad at the other objective, how is one to say it is any better or worse than a member of the population who does well, but not excellent, at both of its objectives? To solve this, we use Deb's NSGA-II algorithm presented in [15]. NSGA, or Nondominated Sorting Genetic Algorithm creates "fronts" that measure success. A box is constructed from each point to the origin, and any point in that box is considered dominated by that point. If the selection takes the top x members of the population, all the members from the highest performing fronts will be selected until a front cannot be entirely fit into the next generation, and at that point the selection of members from that front to go to the next generation is arbitrary within the front. An example of front construction is shown below. The points in front 1 in Figure 5 are not dominated by any other points, while the points in front 2 are only dominated by points in front 1, and so on. Of the ten points in this sample, if we were to select the top half, or 5, to survive to the next population, all of the points in front 1 would be selected, but since there are 5 points in front 1, no other points would be selected. In 5, we can see four different fronts, where front 1 dominates front 2, front 2 dominates front 3, and so on.

5 Our Model

Our model is based on Mittal's [2], using many of the same parameters for the sake of comparing results. We initialize a population of 60 members, each with a 70-bit string

```
while(inNextGen >= popSize)
    if(sizeOfFront <= popSize - inNextGen)
         nextGen.add(front)
    else
     while (inNextGen <= popSize)
         nextGen.add(memberInNextFront)
```

Figure 6: Pseudocode of NSGA-II

that is their genome. Additionally, we track each member's cumulative score in the games they play, how many points they allow their opponents to score, the number of times both players cooperate in a round, and the total number of rounds played. These scores are used later to calculate fitness of the member. We refer to Mittal's model for the types of crossover, crossover probability, and mutation probability. Mittal and Deb let their experiments run for 20,000 generations, although fitness converged at its maximum after around 2,500 generations. In our experiments, we run some trials at 5,000 and some at 10,000 to ensure that no further improvement is likely to be made.

In his canonical paper on Evolutionary Algorithms and Iterated Prisoner's Dilemma, Robert Axelrod proposed a round robin tournament against 16 different pre-determined strategies, where each member plays each strategy 150 times, and tried to evolve an optimal strategy that would score the most against his members. In fact, his work became so influential that an annual tournament to determine the best solution, that is, one that could beat his tit-for-tat strategy, emerged. Deb's paper is inspired by Axelrod's, and he runs experiments that play against Axelrod's 16 strategies.

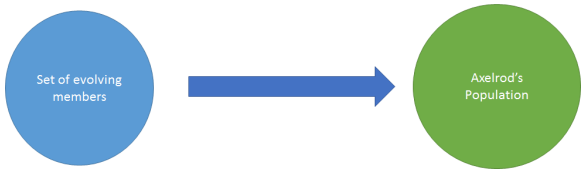

Figure 7: Model of Axelrod's evolution

In Mittal's experiment, players are trying to maximize their score and minimize their opponent's score. However, in our experiment, we uniformly initialize all of the members in our population with one of four different pairs of objectives. Some of them, like Mittal's members, are trying to maximize their score and minimize their opponent's scores. However, those members are competing against other members with different objectives, including those who are trying to maximize their own score and their opponent's score. Additionally, some members want to maximize their score and the number of times mutual cooperation occurs, and the last subcategory of members want to maximize their opponent's score while maximizing cooperation.

In life, people are naturally motivated to do the same things by different factors. Taking the Prisoner's Dilemma literally (although that is not its intention) for the sake of an example, people have different motives for committing a crime. In the same sense, people have different motives to playing the game. For example, someone may play selfishly if they are committing crimes solely in order to feed their family and they don't like the people they were working with. Another person may play cooperatively if they are close to the people they committed the crime with, maybe a fellow gang member. Continuous mutual cooperation is known as *Nash Equilibrium*, where neither player has incentive to change their decision pattern and both are doing well. In this case, a player may strive for mutual cooperation for the betterment of the large-scale crime organization, but still may want to do well as an individual. However, a player may be selfless if the person they got arrested with and are playing

against is the leader of the gang and they are low-ranking within their group. While this is an extreme example, it can easily be applied to other situations, and the reasons for mixing objectives becomes obvious.

With environmental tradeoffs, each country forms policy with different backgrounds. Countries that are economically sound and can afford to take the economic cost of implementing environmentally friendly policy will have very different things in mind when making a decision than a developing country that is trying to industrialize.

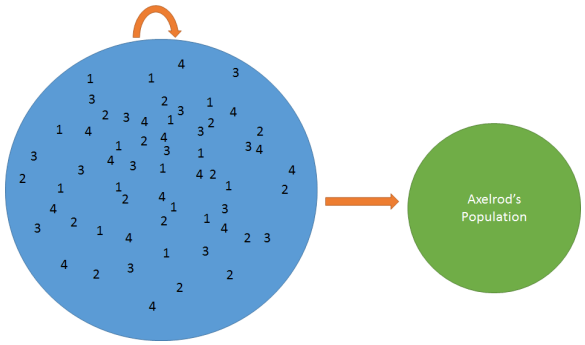

Figure 8: Internal evolution with testing on Axelrod's population

In our model, we are less interested in which members succeed, but how objective pairs converge when given the opportunity for people to learn. We also fix objectives later and evaluate which members of the population succeed when there is a uniform distribution of objectives within the population. The problem, however, originated with the investigation of convergence. We also are interested in extracting "key" decision patterns. For example, if opponents, in the past three moves, all complied with each other, will members learn to keep cooperating, doing well for themselves and perpetuating the state of Nash Equilibrium, or will the member take advantage of this opportunity and defect? This may change depending on the member's objectives, and so success, as well as objectives, must be taken into consideration when evaluating members of the population.

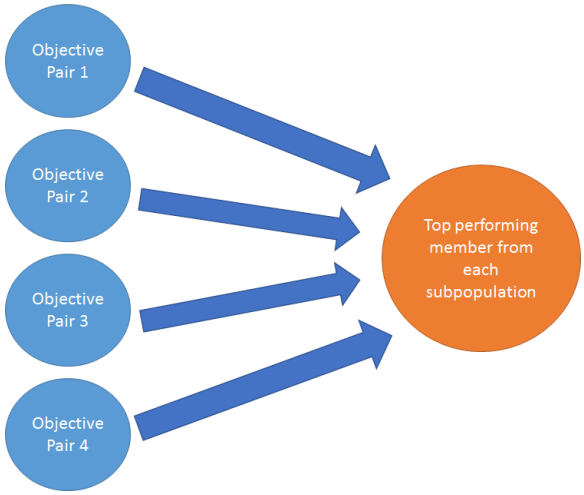

Figure 9: Our model of coevolution

Previous work on this subject has revealed that an average score of 3 is the benchmark score reaching Nash equilibrium, where players are optimally cooperating. Since mutual cooperation, one of our possible objectives, is dependent upon not only personal decision but also the opponent's decision, the benchmark is 0.5 if the current player always cooperates, since the other player's expected cooperation is 0.5. Using this logic, we multiplied the cooperation score by 6 so the benchmark would be 3 and would be on the same scale as the personal and opponent score benchmarks.

In our Leave One Out testing, we do, however, change the multiplication constant to see if results change. Using a constant of 6 gives a maximum score (all cooperation) of 6, which, in a cooperative situation, poses an unfair advantage, since the highest personal score possible is 5 per round. We found, however, that this did not change the general outcome of which members were the most successful.

6 Experimental Results

We ran a series of experiments, at first for 5,000 generations, then for 10,000 generations. In some experiments we fixed the objectives to see, when players play a uniform sampling from each population the entire time, which objective pair performs the best. Our original experiment was to allow freedom to change objectives during the evolution process. We sought to determine if there was a trend in the convergence of objectives. We found there was generally a convergence every experiment, but the most popular objective pair changed from experiment to experiment. Thinking this was at part due to chance, we adjusted our model to reset a player's score when they were a new member of the population so old scores didn't affect future success. Then, we added a freeze value n of 20% of the number of generations, and fixed the objectives for the first n generations to see if this made results more consistent in what was succeeding.

We also ran the tournament, fixing the objectives for the first 20% of generations, against Axelrod's 16 original members, for 5,000 and 10,000 generations. Playing against Axelrod's members, we observed a much more significant convergence of final objectives, since members were no longer playing against themselves. Selfish objectives weren't as popular against Axelrod's members as they had been within the population. Against Axelrod, selfless and communally good objectives tended to be most successful. This makes sense, since many of Axelrod's members will cooperate if their opponent cooperates, but will defect if their opponent does. The selfish members are more likely to defect, in an effort to minimize their opponent's score, leading to mutual defection. However, when members play within the population, selfish and selfless members feed off of each other. Axelrod's players end up in a series of mutual defections against selfish players, where selfless players are able to be taken advantage

of, because with their objectives, they are succeeding as well.

Some work has gone into extracting key decisions that are uniform across members. We have only done this within experiments, seeing which decision points are common among the top five performing members, regardless of objectives.

In Mittal and Deb's highest-scoring experiment, their multi-objective strategy scored an average of 2.98 points per game. Our results, in the same experiment, usually yielded a personal score of 2.30 points per game. However, with different objectives, our model yields a better communal result and our highest achieving members achieve better (scaled) results on some of their objectives. Our highest achieving members of the population tend to aim to maximize cooperation, whether aiming for communal good or selfless good. Our members have been trained in different experiments against Axelrod's population, other members of this population, and a combination of the two. We also coevolved members of subpopulations (each objective pair being one subpopulation) against the highest performing members of other subpopulations, just like Jaskowski *et. al.* Our results were better without coevolving members, insted evolving them against Axelrod's members.

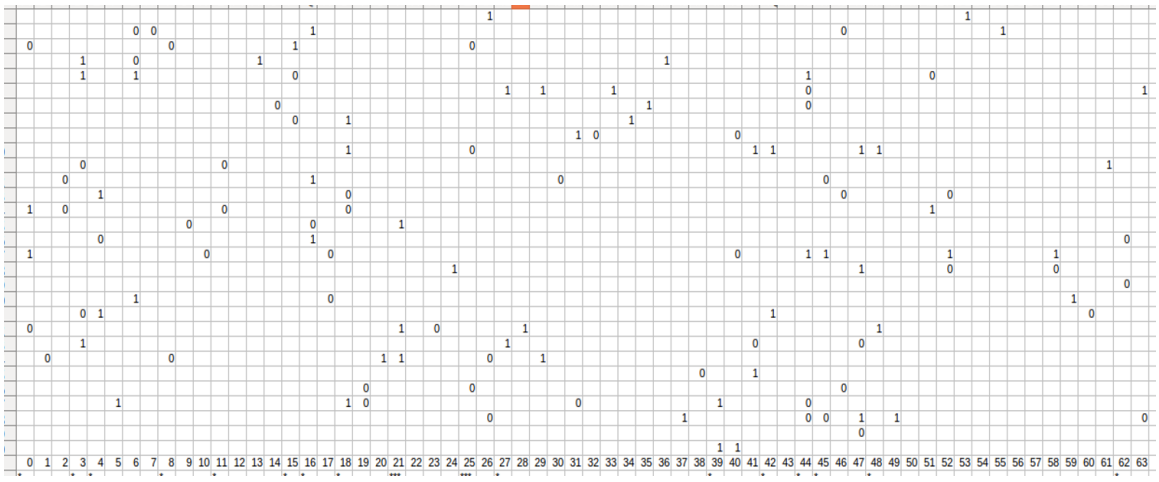

Figure 10: Common bit positions over 30 trials

From thirty trials per experiment, we extracted common decisions from the top 5 performing members of the population. There is one column per experiment, and if at least 4 of the top 5 performing members would make the same decision, we recorded it in this graph by either a 0 (cooperation) or 1 (defection). We then compared the decisions made by the top 5 members from each of the trials and mapped them as in Figure 10.

In this example, where members have free objectives and are playing against other members of the population, we extract a handful of common decisions. Some history patterns have conflicting decisions, which looking at the trials, depended on what objectives thrived most in that simulation. In this set of trials, we draw conclusions at 8, 11, 16, 21, 25, 27, 39, 42, 44, 48, and 62. For example, we notice cooperation at spot 8, which is a history of 000100 in binary. By this we know the players cooperated, our player got screwed, but then they cooperated again. Our players tend to cooperate here, which they are already in a habit of doing. On the other hand, 21 received an implied decision to defect. This history is 010101, so the person has been screwed over the past three rounds and is learning to defect since they keep getting the short end of the stick. However, a history of 25, or 011001 suggests complying after alternating players have been screwed over the past three rounds.

While the average of all the members doesn't vary extremely from one objective pair to the next, the top 5 members of each objective pair performs significantly better at their objectives than other members of the population. Similarly, we present these example tables for the other main trials (free/fixed objectives, against the population/against Axelrod.) These are just the result of one trial from each, provided as an example of trends from each set of trials.

Next, we conducted a set of experiments on training against only one of Axelrod's members at a time. We trained a population against the one member, then selected

Objective Pair	Avg./Top 5	Self Score	Opp Score	Co-op
0	Avg	**2.27934**	**2.2128**	2.97535
0	Top 5	**2.36111**	**2.0988**	2.8857
1	Avg	**2.24259**	**2.28793**	3.0965
1	Top 5	**2.32642**	**2.16432**	2.92497
2	Avg	**2.24719**	2.26753	**3.06432**
2	Top 5	**2.31981**	2.20539	**3.11723**
3	Avg	2.2562	**2.24173**	**3.01335**
3	Top 5	2.22415	**2.35713**	**3.28277**

Table 4: Playing against other members of the population with objective pairs free to change.

Objective Pair	Avg./Top 5	Self Score	Opp Score	Co-op
0	Avg	**2.23501**	**2.22356**	2.87043
0	Top 5	**2.27992**	**2.14351**	2.76386
1	Avg	**2.23491**	**2.22828**	2.90563
1	Top 5	**2.28068**	**2.17099**	2.84097
2	Avg	**2.23976**	2.24739	**2.93457**
2	Top 5	**2.28188**	2.18633	**2.86745**
3	Avg	2.24062	**2.25106**	**2.95219**
3	Top 5	2.20946	**2.33366**	**3.14768**

Table 5: Playing against other members of the population with objective pairs that are fixed.

Objective Pair	Avg./Top 5	Self Score	Opp Score	Co-op
0	Avg	**2.22718**	**2.43067**	3.62658
0	Top 5	**2.27611**	**2.14351**	4.39061
1	Avg	**2.22008**	**2.30601**	3.47205
1	Top 5	**2.26787**	**2.33426**	3.89833
2	Avg	**2.21002**	2.33083	**3.48085**
2	Top 5	**2.24083**	2.38243	**3.77887**
3	Avg	2.21197	**2.31391**	**3.49426**
3	Top 5	2.22246	**2.38778**	**3.74632**

Table 6: Playing against Axelrod's population with objective pairs that are free to change.

Objective Pair	Avg./Top 5	Self Score	Opp Score	Co-op
0	Avg	**2.21422**	**2.30917**	3.55639
0	Top 5	**2.25451**	**2.33711**	3.74954
1	Avg	**2.22085**	**2.31562**	3.62603
1	Top 5	**2.26213**	**2.33615**	3.93126
2	Avg	**2.22119**	2.2927	**3.5398**
2	Top 5	**2.26849**	2.31657	**3.82275**
3	Avg	2.21035	**2.27242**	**3.35759**
3	Top 5	2.2143	**2.37353**	**3.62149**

Table 7: Playing against Axelrod's population with objective pairs that are fixed.

the best player and had that player- without training- play 150 rounds against the member of Axelrod's population. This produced artificially high results, since the player is over-specialized against that player, but from this, we were able to conclude that players were in fact learning the optimized strategy against their opponent.

Our last set of experiments included Leave One Out testing, where members of the population trained against 15 of Axelrod's members and their score was extracted from playing against the 16th member based on training from the other 15. During testing, objective pairs are not adjusted, but scores are kept here to track results.

This is where Mittal and Deb claim their strongest results in [2]. We conducted Leave One Out testing, both using the best member of a selfish population and using the best member of our mixed population. We do this to generalize the results of our training to see how it scales against different players. In our population with multiple objectives, we also note which member of the population is most successful and how their results do against the tested member. Note that here, a "win" is not necessarily better than a "loss," depending on the objectives of the player being tested.

Personal Score	Opponent Score	Cooperation Score	Objective Pair	Tested On
5	0	0	3	Cooperate
0.5	3	0	1	Defect
2.6133	2.58	1.28	3	TFT
2.667	2.667	2	3	Soft TFT
3	3	6	3	Pavlov
0.56	2.9933	0.08	3	Spiteful
2.64	1.64	1.32	3	Random
2.0067	2.00667	2.96	3	CD
1.54667	2.68	1.36	3	DDC
3.33	1.667	0	3	CCD
3	3	6	3	Tit-for-two-tats
2.5	2.5	0	3	Soft majority
0.5	3	0	3	Hard majority
0.55	2.98667	0	3	Hard TFT
2.9533	2.98667	5.76	3	Naive Prober
3	1.33	2	3	Remorseful

Table 8: Leave One Out Results- Best player of entire population.

7 Analysis of Leave One Out Results

The main comparison between our results and those of Mittal and Deb is between the Leave One Out testing rounds. We compared Leave One Out testing amongst a population of all selfish members, testing on the best member of that population, in order to verify the strength of our code compared to that of Mittal and Deb. Additionally, we compare our Leave One Out results with all selfish members to a population with mixed objectives, and then selected the most successful member of that population for comparison. We conduct two rounds of Leave-One-Out testing with multiple objectives, adjusting the multiplication constant for mutual cooperation from 6 to 5 to see if this significantly alters results. Using a constant of 6 generally yields the most successful member of a population, so we ran another set of trials to see if members seeking mutual cooperation were still the most successful.

From our results, it becomes apparent that cooperation can be learned if one

Personal Score	Opponent Score	Cooperation Score	Objective Pair	Tested On
3	3	6	0	Cooperate
0.00667	4.97333	0	0	Defect
1.04667	1.0133	0	0	TFT
2.42	2.42	1.68	0	Soft TFT
0.1133	4.88	0.08	0	Pavlov
0.04	4.94	0	0	Spiteful
2.08	2.64667	1.84	0	Random
2	2	3	0	CD
1.667	3.33	0	0	DDC
2.5	2.5	3	0	CCD
3	3	6	0	Tit-for-two-tats
4	1.5	3	0	Soft majority
3.58	2.04667	3.88	0	Hard majority
0.533	2.9667	0	0	Hard TFT
2.5133	2.58	1.76	0	Naive Prober
2.766667	2.1	1.48	0	Remorseful

Table 9: Leave One Out Results- Best selfish player.

member of the population, is willing to take the first step in cooperation. However, on the opposite side, member of the population may become jaded and respond defensively against aggressive opponents. Learning becomes apparent in Leave-One-Out testing, and there is an evident robustness to specifically learning how to combat one specific strategy.

8 Optimal Strategies by Objective

In this section, we go through each of Axelrod's players in details and outline what the ideal decision each of our players would be if they were conscious of their opponent's decision. This gives us a baseline for the top score against each opponent. A generalization of our results is also presented for each opponent.

Strategy Name	**Summary**
Cooperate only	Cooperates every turn
Defect only	Defects every turn
Tit-for-tat	Cooperates, then does what opponent did on previous turn
Suspicious tit-for-tat	Tit-for-tat that defects on the first move
Pavlov	Cooperates, then defects if there was disagreement on previous move
Spiteful	Cooperates until opponent defects. Defects the rest of the game after
Random	Plays a random move
CD	Rotates between cooperating and defecting
DDC	Defects twice, then complies. Repeat.
CCD	Complies twice, then defects. Repeat.
Tit-for-two-tats	Tit-for-tat, but only defects if opponent has defected the past two moves
Soft majority	Begins by cooperating, and cooperates as long as the number of times the opponent has cooperated >= defected
Hard majority	Like soft majority, but cooperation must be strictly greater than defection
Hard tit-for-tat	Tit-for-tat, but defects if opponent has defected in any of past three moves
Naïve prober	Tit-for-tat, but randomly defects at 0.01 probability
Remorseful	Like Naive Prober, but it tries to break the series of mutual defections after defecting by cooperating.

Table 10: Summary of Axelrod's Players.

8.1 Cooperate Only

For a selfish member playing against a member that only cooperates, the optimal strategy will be to always defect, as that both maximizes personal score (5) and minimizes opponent score (0). For a member that wants to maximize personal and opponent score, the optimal strategy will be to always cooperate, as the communal score (3+3) is greater than the score would be if the main player defected. Similarly, cooperation would be optimal for the last two players as cooperating would yield a score of 3 for personal/opponent and 6 for cooperation, giving a total of 9. Even in the case of the person trying to maximize personal score and cooperation, defecting would add less than would be lost in the cooperation score. In LOO testing, our best selfless players generally learned to cooperate once they were placed in a series of cooperating. Our best selfish players actually generally learned cooperation in the LOO setting, being influenced by the selfless nature of the opponent.

8.2 Defect Only

For the selfish player, it is optimal to defect against an opponent that only defects. In this situation, the maximum score the player can achieve is 1, and this decision also minimizes the opponent's score (from 5 to 1). For the communally good player, cooperating is optimal, although only in one sense. It minimizes personal score, although the difference is less than the increase in the opponent's score should they have defected (from 1 to 5). The cooperatively good player is best off defecting. Regardless of their action, their opponent will not cooperate with them on the next move, and so they should make the move that maximizes their personal score, which is defecting. The selflessly good player, following that logic, should do what will maximize their opponent's score, which is cooperating. Our top players against defecting players

varied in objective from round to round, but generally followed the manner of what is written above. With the exception of one trial, almost all of our top selfish players learned to mutually defect.

8.3 Tit-for-Tat

A selfish player will have no way to beat a tit-for-tat opponent (without knowledge of which round is the last). Alternating cooperation and defection will yield 2.5 points per round for each player. Cooperating will yield 3 points per player, and defection will yield 1 point per player. Communally good players will want to cooperate, since their opponent will cooperate if they do, which does well both for personal and opponent scores. Cooperatively good players will want to communicate since the trade off of alternating cooperation and defection decreases a mutual cooperation from 5 to 0 but only decreases personal score by 2. A similar argument follows for the selflessly good player. Our top selfish players had nearly identical scores to tit-for-tat scores in every trial, ranging from 2.2 to 3 points per player per round. The same followed for selfless players, which were all the top members in these trials, but had more pairs of opponents at 3 points per round.

8.4 Suspicious Tit-for-Tat

The same follows as tit-for-tat, as cooperation is required to get the opponent on the track of cooperation. With the best of all objectives, once again all selfless members, we noticed patterns of alternating defection, but had four of ten trials that were at almost complete cooperation. Looking at selfish players, there was only one round that almost cooperated fully, but there was also a trial of all defection, while the rest were in the middle, but generally leaned towards forced cooperation.

8.5 Pavlov

Our selfish player will want to start with a defection. One the next move, the opponent will defect, so we will want to defect in order to get the Pavlov opponent to cooperate. This will result in an alternating pattern where personal score averages to be 3 and opponent score averages to 0.5. Every other player will want to always cooperate since the player average will be 3 and opponent average will also be 3, which is the communally optimal solution. The best players of all objectives were either being taken advantage of in our trials or learned mutual cooperation. Many of our top selfish players, however, alternated decisions every other round and had scores close to 3 and 0.5.

8.6 Spiteful

Cooperation will be ideal for the selfish player here, since the moment the controlled player defects, there is no way of getting the opponent to get back to cooperating, so our player is forced into mutual defection, which just results in a low scoring tie. Technically the player can win if they screw their opponent over and then know to defect the rest of the game and not try to make it up to their opponent. The rest of the players will all want to (always) cooperate, for more obvious reasons, as mutual cooperation will be maximized. Playing against the spiteful player, our best overall players almost all were being taken advantage of, maximizing opponent score. Our best selfish players either had patterns of full cooperation or generally knew to defect after they had defected first time.

8.7 Random

There isn't a particularly defined strategy playing against a random opponent. The selfish player will defect, since their decision will not affect their opponent's next decision, and this is the safest decision to make for their sake. The rest of the players, however, will tend to cooperate, as that is the strategy to maximize communal score as well as cooperation. For the cooperatively good player, there is a conflict between if a person should cooperate, sacrificing their score but making mutual cooperation possible, as playing against a random player will, if our player always cooperates, yield a personal score of approximately 1.5 and cooperative score of 3, while defection will yield a personal score of 3 and cooperative score of 0, so it still seems better to cooperate. All of our members had average and personal scores of approximately 2.5

8.8 CD

For the selfish player, it will always be best to defect, as their opponent's pattern of behavior will not change based on their behavior. If they always defect, they will average 4 points per round and their opponent will average 0.5. A communally good player will always cooperate, since their average decrease from defecting won't increase as much as their opponent's score will increase if they cooperate even on the defecting turns. Their score will average 1.5 and opponent's will average 4 if they cooperate. The cooperatively good player will want to follow the same pattern as the opponent, since their score will be 2 on average (average mutual cooperation and defection) and cooperative score will be 3. Some of our selfless players learned to almost always cooperate, while others got screwed over, mutually cooperated, and then defected alongside the opponent on the most recent turn.

8.9 DDC

Similarly, our selfish player will always want to defect against the periodic DDC player since our decision does not affect our opponent's next decision. The Communally good player will want to always cooperate, yielding an average personal score of 1, but opponent's average score of 4.33, which is greater than following their pattern (average of 1.66 and 1.66), DCC for an average of 1.33 and opponent's average of 3, or always defecting (average of 2.33 personally and 0.66 for the opponent.) A cooperatively good player will want to follow their opponent's pattern of play, as that maximizes cooperation given their opponents decisions and maximizes personal score on the defecting moves. A selflessly good player will always cooperate, allowing their opponent to flourish on their defecting moves. Our selfless players generally cooperated, even along defection, but some knew to follow the same pattern as the opponent. Many selfish players followed the same pattern as the opponent as well, and a few just decided to always defect.

8.10 CCD

Similar logic follows for all four players as above. For our players, some selfless players still took advantage of their opponents on cooperating turns, yet others cooperating on every turn. Most of our selfish players almost always defected in their past three moves.

8.11 Tit-for-Two-Tats

The selfish player in this situation will want to alternate between defection and cooperation in order to keep the opponent cooperating. This will yield an average personal score of 4 and opponent score of 2.5. The cooperative player will act in

the same manner, as this yields a higher total score than the sum of a cooperative score of 3 and 3. The communal players, however, will always cooperate in order to maximize cooperative score. Our top players generally either always cooperated or alternated between defection and cooperation between all objectives. Our selfish players, however, were in defecting bouts with the opponent more often.

8.12 Soft Majority

Our selfish player will want to rotate between complying and defecting, starting by complying. They will obtain 4 points per round where their opponent will average 1.5. If they maintain the number of complies to be greater than or equal to their number of defects, the opponent will comply on the next turn. The communally good player will want to cooperate the whole time since they and they opponent will cooperate together and will average 3 points per round each. Half of our top players of all objectives settled into patterns of cooperation, while less (but still some) selfish players settled into cooperation. The better performing selfish players won approximately 3.4-2.0, which is not as good as the idea 4-1.5.

8.13 Hard Majority

Our selfish player will do the same as above, but must comply on the first two turns and then alternate, since the number of complies must strictly be greater than the number of defects. The rest of our players will maintain the same strategies. Our top performing members won with scores close to 4 and reached approximately half cooperation, which is the optimal solution.

8.14 Hard Tit-for-Tat

Our selfish player, if they defect every third move, will average 1.66 points per round and are almost forced into cooperation with a hard tit-for-tat player. A hard tit-for-tat player will cooperate upon cooperation, so every other player will benefit from mutual cooperation. With the exception of one fully cooperating game, most players could not achieve any cooperation.

8.15 Naive Prober

Since we cannot strategize for the probing, we should play the Naive Prober just like the tit for tat player. No selfless players were able to achieve full cooperation for every single round, but a handful of players were in a fully cooperative pattern at the end of the round. A handful of our selfish players cooperated, while one was in a series of alternating cooperation and defection with the opponent.

8.16 Remorseful

The selfish player should start by defecting. The remorseful player, in response, will defect on the next turn. If the selfish player also defects, the remorseful player will switch to cooperation, starting a cycle where the remorseful player alternates between cooperation and defection. This gives a personal score average of 3, and opponent score average of 0.5. The rest of the players will follow the same strategy as tit for tat. The top players of all objectives showed results similar to tit-for-tat, while the top selfish players had average scores close to 3 (generally 2.7-3) and opponent scores higher than optimal. There was one selfish player, however, that achieved the exact optimal score.

9 Conclusions

Our results, since we use genetic algorithms, are approximations of the optimal results discussed in the previous section. This does, however, open up an interesting question for future research. If we could train an artificial neural network to recognize opponent strategy, we could play these optimal strategies. The merit of our research is that these strategies are not limited to Axelrod's sixteen strategies like a neural network designed in that manner would be.

When considering the social implications of this work, we note that cooperation can be evolved and we learn about social trends in our model. While we cannot make a claim about the direction of convergence (*ie.* converge to cooperation or converge to defection), we found significant results showing that there is almost always a convergence of objective pairs in our experiments.

There are almost infinitely many factors to consider when deciding foreign policy, but our model is the first to consider preliminary social factors in making decisions. While we cannot track exact market types or methods of environmental policies and their impacts, our model demonstrates the effect that considering social interactions has in decision making.

The work done in this thesis lays the groundwork for future work and approaches the Iterated Prisoner's Dilemma from a cooperative perspective. This work, to our knowledge, is the beginning of cooperation research from a multi-objective perspective. We are able to juxtapose and evaluate the effect of conflicting and non-uniform objectives in the context of the Iterated Prisoner's Dilemma game.

References

[1] R. Axelrod *et al.*, "The evolution of strategies in the iterated prisoner's dilemma," *The dynamics of norms*, pp. 1–16, 1987.

[2] S. Mittal and K. Deb, "Optimal strategies of the iterated prisoner's dilemma problem for multiple conflicting objectives," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 554–565, 2009.

[3] P. J. Darwen and X. Yao, "Co-evolution in iterated prisoner's dilemma with intermediate levels of cooperation: Application to missile defense," *International Journal of Computational Intelligence and Applications*, vol. 2, no. 01, pp. 83–107, 2002.

[4] J. Tanimoto, "Co-evolution model of networks and strategy in a 2× 2 game emerges cooperation," in *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, pp. 117–122, IEEE, 2008.

[5] W. Zeng, M. Li, F. Chen, and G. Nan, "Risk consideration and cooperation in the iterated prisoner's dilemma," *Soft Computing*, vol. 20, no. 2, pp. 567–587, 2016.

[6] C.-K. Goh and K. C. Tan, "A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 103–127, 2009.

[7] S. Y. Chong and X. Yao, "Behavioral diversity, choices and noise in the iterated prisoner's dilemma," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 6, pp. 540–551, 2005.

[8] W. Jaśkowski, P. Liskowski, M. Szubert, and K. Krawiec, "Improving coevolution by random sampling," in *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pp. 1141–1148, ACM, 2013.

[9] J. H. Miller, "The coevolution of automata in the repeated prisoner's dilemma," *Journal of Economic Behavior & Organization*, vol. 29, no. 1, pp. 87–112, 1996.

[10] H. Richter, "Analyzing coevolutionary games with dynamic fitness landscapes," in *Evolutionary Computation (CEC), 2016 IEEE Congress on*, pp. 609–616, IEEE, 2016.

[11] K.-B. Sim, D.-W. Lee, and J.-Y. Kim, "Game theory based coevolutionary algorithm: a new computational coevolutionary approach," *Int J Control Autom Syst*, vol. 24, pp. 463–474, 2004.

[12] J. J. Nay and Y. Vorobeychik, "Predicting human cooperation," *PloS one*, vol. 11, no. 5, p. e0155656, 2016.

[13] A. S. Wu, H. Yu, S. Jin, K.-C. Lin, and G. Schiavone, "An incremental genetic algorithm approach to multiprocessor scheduling," *IEEE Transactions on parallel and distributed systems*, vol. 15, no. 9, pp. 824–834, 2004.

[14] S. W. Mahfoud, "Niching methods for genetic algorithms," *Urbana*, vol. 51, no. 95001, pp. 62–94, 1995.

[15] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.