This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TG.2020.3005124

Evolutionary Optimization of Cooperative Strategies for the Iterated Prisoner's Dilemma

Jessie Finocchiaro, H. David Mathias

***Abstract*—The Iterated Prisoner's Dilemma (IPD) has been studied in fields as diverse as economics, computer science, psychology, politics, and environmental studies. This is due, in part, to the intriguing property that its Nash Equilibrium is not globally optimal. Typically treated as a single-objective problem, a player's goal is to maximize their own score. In some work, minimizing the opponent's score is an additional objective. Here, we explore the role of explicitly optimizing for mutual cooperation in IPD player performance. We implement a genetic algorithm in which each member of the population evolves using one of four multi-objective fitness functions: selfish, communal, cooperative, and selfless, the last three of which use a cooperative metric as an objective. As a control, we also consider two single-objective fitness functions. We explore the role of representation in evolving cooperation by implementing four representations for evolving players. Finally, we evaluate the effect of noise on the evolution of cooperative behaviors. Testing our evolved players in tournaments in which a player's own score is the sole metic, we find that players evolved with mutual cooperation as an objective are very competitive. Thus, learning to play nicely with others is a successful strategy for maximizing personal reward.**

***Index Terms*—iterated prisoner's dilemma, cooperation, genetic algorithm, multiobjective optimization**

I. Introduction

EVERY day, people interact with others in a manner that affects both parties; one person cuts off another in traffic, causing an accident; a teenager eats the last piece of cake without asking if his brother would like some; a person volunteers their time to help those in need. In game theory, we model people or strategies as players of a game, receiving reward or punishment depending not only on their own actions but also on the actions of others. Typically, these players are selfish, and simply want to achieve the best possible outcome for themselves. They seek to optimize an objective or set of objectives and adjust their behavior in order to achieve that goal. In many games, however, the Mixed Strategy Nash Equilibrium is suboptimal in terms of social welfare. One canonical game with a globally suboptimal Nash equilibrium is the Prisoner's Dilemma (PD).

The Prisoner's Dilemma is a two-player game between criminal accomplices who have been captured by police. They are held in isolation and cannot communicate. The police incentivize each accomplice (player) to implicate the other. Game play consists of each player simultaneously and independently choosing whether to implicate (defect) or not implicate (cooperate). A penalty (symmetrically, reward) is given depending on the simultaneous choices of both players. See Table I for the matrix of rewards.

J. Finocchiaro is with the Department of Computer Science, University of Colorado, Boulder, CO, 80309 email:jessica.finocchiaro@colorado.edu

D. Mathias is with the Department of Computer Science, University of Wisconsin - La Crosse, La Crosse, WI 54601 email:dmathias@uwlax.edu

In the Iterated Prisoner's Dilemma (IPD), the participants play the game repeatedly, utilizing a history-based strategy to optimize their aggregate scores. Each has knowledge of some number of their opponent's previous decisions and uses this information to inform their next decision. IPD is especially interesting and well-studied because the Nash Equilibrium in PD is not globally optimal [1]. The globally optimal solution in PD arises from mutual cooperation while the only Nash equilibrium results from mutual defection. However, iterating the game allows players to discover a solution concept that is not necessarily equilibrium.

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

TABLE I: Scoring for the version of IPD in this work. Note that these values represent rewards rather than penalties.

Though populations achieve the best aggregate result by acting communally, previous research has focused on players that optimize their decisions in their own self-interest. Researchers have considered the role of cooperation and how to evolve cooperating behaviors [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12]. There is also work on the how social preferences can lead to cooperation [13], [14]. However, to our knowledge no previous work has incorporated a measure of cooperation as an additional optimization objective during training of strategies for the Iterated Prisoner's Dilemma.

Traditionally, IPD has been treated as a single objective problem, with players attempting only to maximize their own score. While maximizing one's score is a reasonable metric for success, we investigate the effects of using a utility function that is at least partially dependent on the opponent's score. The central question is whether training players to value their opponent's performance can encourage cooperation and improve the player's self-score. However, since we do not want to completely sacrifice the optimization of self-score, we choose to explore multi-objective optimization for training IPD players. To gauge the effectiveness of using multiple objectives, we compare players evolved in this way to a canonical single-objective player maximizing its own score, as well as a cooperative single-objective player that attempts to achieve equity with its opponent.

Mittal and Deb [15] first evolved multi-objective strategies for IPD. During training, their players attempted to maximize their own reward while minimizing that of their opponent. They showed that players trained in this way were superior

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TG.2020.3005124

to those trained using only the single objective of maximizing their own reward. In this work, we include a multi-objective player with this same objective pair, which we call Selfish, as a benchmark to compare results.

Several factors can significantly affect both the training of players for IPD and tournament outcomes. The choice of genome representation used to evolve players using a genetic algorithm one such factor. Another is the population against which players are trained. A third is noise. This takes the form of altering the decision indicated by a player's strategy. In this work, we examine the effects of each of these factors to assess the robustness of the model we develop to address the central question above.

We deviate from the typical game-theoretic assumption that to be successful, players must optimize their strategy selfishly, promoting their own self-interest to the detriment of others. Instead, we explore the hypothesis that optimizing for cooperation is not only mutually beneficial but potentially a better strategy for self-success than being selfish. To explore this, we create a population in which each player has one of four pairs of non-contradicting objectives. The four objective pairs are shown in Table IV. The population is initialized with an equal number of players with each objective pair and trained against each other or a benchmark population. We then evaluate all players in a testing phase in which they compete against a population of benchmark strategies composed of Axelrod's 16 canonical players, the Gradual strategy [16], and a sample of evolved players drawn from a candidate pool of the best players produced. In this tournament, players are scored only on their own reward, independent of the objective pair they used during training.

We find that players with cooperative objectives often outperform their selfish counterparts, even when evaluated only by self-score. That is, players trained to play to the benefit of others win tournaments in which the only metric is selfish, personal reward.

II. Related work

In 1950, Flood and Dresher, of the Rand Corporation, introduced a two player game with a matrix of payoffs determined by the binary choices made by the players [17]. Later, Tucker provided the name *Prisoner's Dilemma* and changed the payoffs to penalties in the form of prison sentences. The *Iterated Prisoner's Dilemma* is an extensive-form game in which two competing players simultaneously play the Prisoner's Dilemma game against each other for a fixed, but large, number of rounds. The number of rounds is usually unknown to the players as this knowledge fundamentally changes the game.

In 1987, Robert Axelrod [2], [18] presented 16 strategies (players) for the Iterated Prisoner's Dilemma, based on results from a tournament he created and to which he invited experts in the field to participate. The goal was to identify the best strategy for IPD. Surprisingly, the very simple Tit-for-Tat strategy, in which the player cooperates in the first round and in each subsequent round copies the opponent's previous decision, was the winner. The full list of Axelrod's players,

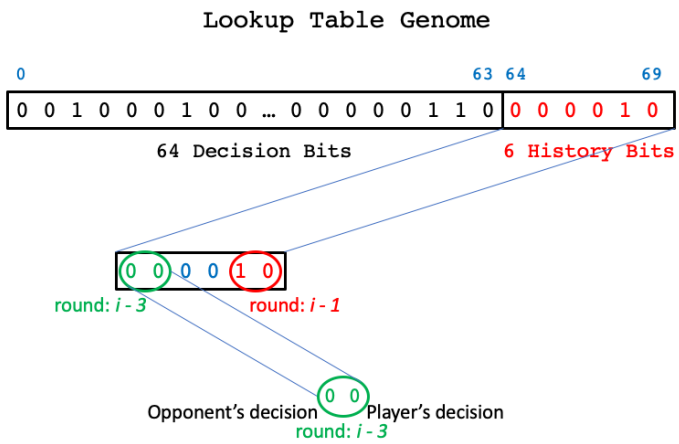

Fig. 1: Format of the history bits in the genome for our lookup representations. `L64` is shown.

and synopses of their strategies, appears in Table II. These players have formed the basis for much of the IPD research done since. Beyond demonstrating the success of Tit-for-Tat, Axelrod identified an attribute shared by most of the successful strategies in his tournaments: they are *nice*, meaning that they never defect before their opponent does so.

Since Axelrod's work, researchers have attempted, with varying degrees of success, to devise strategies that are able to outperform Tit-for-Tat [3], [16], [19].

Mittal and Deb [15] study the power of multi-objective optimization in IPD. Specifically, they implement a multi-objective genetic algorithm using a 64-bit lookup table due to Axelrod as the representation for their players. These players are evolved to maximize their own score and minimize their opponent's score. They then enter their best evolved players in a round-robin tournament against all 16 of Axelrod's players. They found that their evolved players consistently outperformed Axelrod's players, as well as a typical single-objective player trained to maximize its own score.

Subsequent to his well-known tournaments, Axelrod [2] provided 4 suggestions for successful strategies in the Iterated Prisoner's Dilemma. One of these, *"Don't be envious,"* recommends that players not attempt to outscore their opponent. They should instead try to maximize their own score without regard to their opponent's score. Selfish players, those that seek to maximize their own score and minimize their opponent's score, ignore this advice.

The emergence of cooperative behaviors in the Iterated Prisoner's Dilemma is a well-researched topic. W. Ashlock [20] analyzes the emergence of cooperative behavior under different representations, including lookup tables, finite state machines, and neural networks, evaluating how often these evolved spaces project back to each of Axelrod's original strategies. Li *et al.* [21] analyze the success of previously well-known IPD strategies in the presence of new "group" strategies that collude with other members of their group, but do not play as nicely with outside members. Ishibuchi *et al.* examine how the number of opponents affects the evolution of cooperative strategies [7] as well as how cooperation develops

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TG.2020.3005124

Name	Description
All-C	Always Cooperates
All-D	Always Defects
Spiteful	Cooperates until the first time the opponent defects, then defects henceforth
CD	Alternates between cooperating and defecting each round
CCD	Cooperates twice, then defects once
DDC	Defects twice, then cooperates once
Random	Cooperates with probability 1/2
TFT	Cooperates first round, then copies opponent's previous decision
Suspicious TFT	Tit-for-Tat that defects in first round
HTFT	Cooperates first round, the defects if opponent has defected in any of the three previous rounds
TF2T	Cooperates first round, then defects after two defections by opponent
Soft Majority	Cooperates first round, then cooperates if # opponent cooperates $\geq$ # opponent defections
Hard Majority	Defects first round, then defects if opponent has defected at least as many times as has cooperated
Naïve Prober	TFT that defects at random with low probability
Remorseful Prober	Naïve Prober, but cooperates after mutual defection
Pavlov	Cooperates first, then defects only if players did not agree on previous move
Gradual	Cooperate until opponent defection. Punish n-th defection with n defections, then cooperate twice

TABLE II: Standard players used for testing our evolved players. The first 16 are from Axelrod's 1987 tournament [18]. Gradual is due to Beaufils, *et al.* [16]. Nice strategies will not defect before their opponent does so. These strategies are shown in bold.

when IPD players are connected in a network [8]. Greenwood and Ashlock question why more progress has not been made in understanding cooperation in evolutionary games [6]. They conclude with a list of recommendations to researchers in the field for better exploring cooperation.

The effects of different representation for IPD players has also been examined. Ashlock [22] studies the importance of representation in success for the IPD, finding that the level of cooperation varies dramatically with changes in the representation used to train players. Ishibuchi *et al.* [5] consider different player representations and their effects on cooperation in the spatial IPD.

Harper *et al.* [23] evaluate various evolved IPD players in large round-robin tournaments to understand the comparative performance and robustness of evolved strategies. They run tournaments with and without noise. Ashlock, Kim, and Ashlock [24] study the effect of noise level on cooperation in the Iterated Prisoner's Dilemma, varying the probability of switching decisions from 0 to 0.05.

In order to analyze the effect of Prisoner's Dilemma payoffs on success, Ashlock *et al.* [25] use fingerprint analysis [26] to understand how scaling the payoffs affects the success of players trained using different scores in the payoff matrix. In other work, fingerprints are used to understand how several factors affect outcomes in IPD (see, for example, [27]).

III. Our model

The core of our of model is inspired by that of Mittal and Deb [15]. Thus, we implement a multi-objective genetic algorithm for the Iterated Prisoner's Dilemma using the 64 row lookup table in their work. We study a variety of player representations (see Section III-A), including the 64-bit lookup table, which we denote `L64`, and compare their performance to this original representation.

A Genetic algorithm (GA) is a population-based, stochastic approximation algorithm inspired by biological evolution. Each individual in the population represents a candidate solution to the problem and is encoded in a genome. Over a fixed number of generations, the GA refines individuals by analogs of reproduction (crossover) and random mutation (mutation). Crossover recombines the genomes of two individuals to create, most often, two new individuals. Mutation alters the genome of a single individual. From one generation to the next, a form of "survival of the fittest" (selection) uses a fitness function to determine which individuals remain in the population and which are discarded.

An important distinction between our model and that of Mittal and Deb [15] is that each individual in our population is assigned one of 6 distinct fitness functions. Motivation for this aspect of our work is given in Section I; details appear in Sections III-C and IV.

A. Representations

The representations used during evolution of strategies for the Iterated Prisoner's Dilemma is a significant factor in the outcome [5], [20], [22], [23], [28], [29]. To ensure that our results are robust against representational bias, we explore 4 representations used to evolve individuals: the 64 row lookup table mentioned previously (`L64`), an 8 row lookup table (`L8`), an 8 row Markov chain (`M8`), and a 16 state finite state machine (`FSM16`). Each appears extensively in the literature.

First used by Axelrod [18], the `L64` representation consists of a 70-bit string: a 64-bit *decision string*, followed by a 6-bit *history string*.

As illustrated in Figure 1, the history string stores the results of the last three games played. Each game is represented in the string by two bits, each representing the decision of one of the players. The history string serves as an index into the 64-bit decision string, which provides the player's decision for the next game. Thus, a player's decision in game k is determined by the outcomes of games $k-1$ to $k-3$.

The `L8` representation is similar to the `L64` representation. The difference is that only the *opponent's* last three moves are stored. Thus, there history string consists of 3 bits and the decision string of 8 bits.

`M8` consists of 8 entries, each of which represents a probability of cooperation. Like `L8`, this representation is indexed by three history bits that represent the opponent's last three moves, used to index into the table.

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TG.2020.3005124

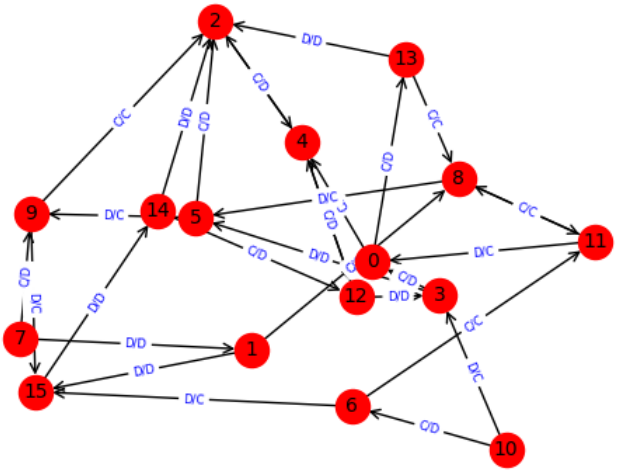

Fig. 2: Example of a trained FSM16 with TRAINAXELROD.

The FSM16 representation encodes a Mealy machine. Mealy machines are FSMs in which each transition is labeled with the character that triggers that transition and an output. For IPD, the trigger is the opponent's previous move (C or D) and the output is the encoded player's current move (C or D). An example is shown in Figure 2.

The FSM16 genome consists of 65 fields. The first represents the player's first move in a game. The remaining 64 fields form 16 4-tuples, one per state. Each tuple is comprised of: (next state if C, move if C, next state if D, move if D), where C or D indicate cooperation or defection by the opponent in their previous move.

Additionally, every representation includes an integer in $[0, ..., 3]$ indicating the objective(s) used to evolve the player, though this is not part of the genome and is not, therefore, subject to change during evolution. Section III-C details the optimization objectives used in this work. We note that objectives and representations are independent.

B. Our Genetic Algorithm

On creation of a population, the bits in each individual's decision and history strings are assigned uniformly at random. In addition, each individual is assigned one of the four objective pairs for multi-objective runs or one of the four objectives for single-objective runs. The number of members assigned each objective or objective pair is fixed at one-quarter of the population size.

In each generation, play consists of a round-robin tournament in which every pair of players play 150 consecutive games. After each game, players learn the outcome. Those players that use a history string (L64, L8, and M8) update their history strings so that they store the outcomes of the three previous games. For these players, if the decimal representation of their history string is i, their decision in that game is given by the i^{th} bit of their decision string, where 0 indicates cooperation and 1 indicates defection. FSM16 players do not use a history string. Their moves are determined by their current state and their opponent's previous move.

Crossover is one-point, meaning that the genomes of two individuals (parents), p_1 and p_2, are split at a uniformly randomly chosen position. Label the partitions p_1^1 and p_1^2 for p_1 and similarly for p_2. Two children are created by concatenating the parts of the two parents. $c_1 = p_1^1 + p_2^2$ and $c_2 = p_2^1 + p_1^2$. For L64 and L8, mutation is via random bit flips. M8 is mutated by adding a random noise $\epsilon \sim \mathcal{N}(0, 0.025)$. Mutating FSM16 is accomplished by randomly flipping the bits that represent moves and randomly reassigning states with values chosen uniformly in $[0, \ldots, 15]$. In all cases, the probability of mutating an element in genome g is $\frac{1}{|g|}$, resulting in an expected value of 1 mutation per individual.

Selection is from NSGA-II [30], a ubiquitous multi-objective genetic algorithm that is efficient, easy to implement and produces good results for small numbers of objectives. NSGA-II selection is based on *domination*, described below.

Let s_1 and s_2 be individuals and O be the set of optimization objectives being maximized (symmetrically, minimized). In multi-objective optimization, comparison of individuals is complicated when $s_1(o_i) > s_2(o_i)$ and $s_2(o_j) > s1(o_j)$, making a complete ordering on the population impossible. *Domination* is defined as follows: s_1 dominates s_2 if $\forall\ o \in O\ s_1(o) \geq s_2(o)$ **and** $\exists\ o \in O : s_1(o) > s_2(o)$. Domination is used to impose a partial order on a population via non-dominated sorting. This partitions the population into a series of domination fronts, also referred to as Pareto fronts. For fronts f_i, f_j such that $i < j$, every member of f_j is dominated by at least one member in f_i. For any two individuals in the same front, neither dominates the other.

After sorting as above, the algorithm reduces the population to its original size by selecting all members from Pareto fronts f_0 through f_k, where f_k is the last front for which all members can be added to the population for the next generation without exceeding the population size. Members from f_{k+1} are selected according to a secondary metric until the population size is reached.

Our code is implemented using the Distributed Evolutionary Algorithms in Python (DEAP) module [31].

C. Optimization Objectives

Mittal and Deb [15] demonstrated that multi-objective evolution, maximizing self-score and minimizing opponent-score, resulted in players that scored higher in a round-robin tournament than those evolved using the single objective of maximizing self-score. Further, they showed that players evolved using the single objective of maximizing the difference between self-score and opponent-score were less successful than players evolved using the objective pair above, thus justifying the use of multiple objectives.

In this work, we examine the hypothesis that players evolved to value cooperation in IPD are competitive with players evolved to maximize self-score, when self-score is the sole metric used in evaluation. To fully explore this, we evolve players using 6 measures of fitness: 2 single objectives and 4 objective pairs. All 6 measures are based on subsets of 3 quantities: self-score, opponent-score, and a measure of mutual cooperation. Table III details the values used as single objectives or as constituents in objective pairs.

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TG.2020.3005124

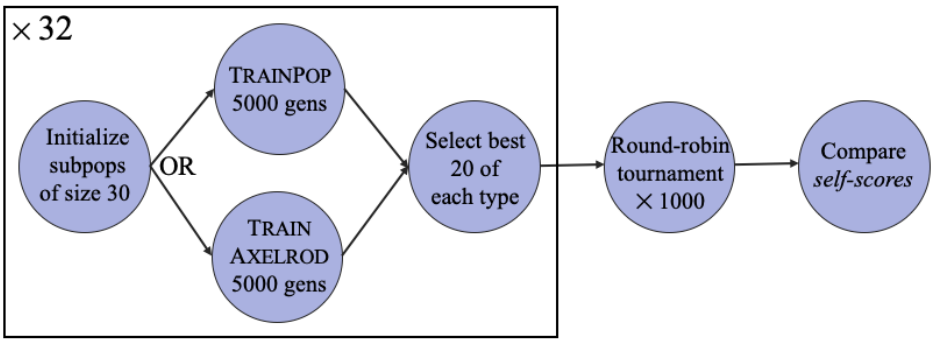

Fig. 3: Possible variations in our experiments. One training iteration shown; subroutine is repeated 32 times for training.

Table IV identifies the single objectives and objective pairs we use and the labels we use to identify them. *Selfish* players have the same objectives as those in the top-scoring algorithm implemented by Mittal and Deb [15]. Though it is non-cooperative, we include this objective pair to provide a basis for comparison with related work. The remaining three pairs reward cooperative behavior to varying degrees.

Communal players attempt to maximize their personal score and maximize their opponent's score. The *Cooperative* and *Selfless* objective pairs explicitly reward mutual cooperation: games in which both players cooperate. This is achieved via an objective that counts the number of IPD rounds in which mutual cooperation is achieved. Cooperative players are trained to maximize self-score and maximize cooperation while Selfless players are trained to maximize opponent-score and maximize cooperation.

To better evaluate the utility of multi-objective optimization for IPD, we include 2 single objectives in our experiments. *Self* individuals seek to maximize self-score. This is the default goal in most work on IPD. It is also the metric used in tournaments in the testing phase of our work. *Min-Diff* players evolve to *minimize* the difference between self-score and opponent-score. We do not use the related objective of maximizing the difference between self-score and opponent-score as Mittal and Deb [15] showed that strategy to be inferior to their selfish multi-objective player.

Single-objective individuals and multi-objective individuals are evolved in different runs of the GA. Each member of the GA population is assigned one of the 2 objectives for a single-objective run or one of the 4 objective pairs for a multi-objective run. Each objective pair is a combination of two of the objectives in Table III. Only non-contradictory pairs are allowed. In particular, simultaneously maximizing and minimizing either self-score or opponent score is not a valid combination of objectives.

Because members of our population do not share a common basis for evaluation, due to using different objectives, directly comparing them for selection during evolution is difficult. Thus, within our evolving population, we maintain four sub-populations, one for each fitness measure. In each generation of the genetic algorithm, all players participate in the same round-robin tournament without regard to sub-population. However, crossover occurs only between members in the same sub-population, as does the non-dominated sorting for survivor selection.

Objective	Value
Max Self	Maximize Personal reward
Min Opp	Maximize 3 − (opponent reward)
Max Opp	Maximize Opponent reward
Max Co-op	Maximize (# mutual cooperation)/(games played)
Min Diff	Maximize 3 − |self-score − opponent-score|

TABLE III: Fitness values are divided by number of rounds played. The scaling parameter for cooperation allows comparison with other objectives.

Single Objectives		
Name	Max Self	Min Diff
Self	•	
Min-Diff		•

Objective Pairs				
Name	Max Self	Min Opp	Max Opp	Max Co-op
Selfish	•	•		
Communal	•		•	
Cooperative	•			•
Selfless			•	•

TABLE IV: Optimization objectives, based on fitness values in Table III, used in our IPD experiments.

IV. Experimental Design

Our experiments consist of two phases: training and testing. The training phase consists of evolving IPD players using a genetic algorithm (see Section III-B). In the testing phase, we run a number of round-robin tournaments that include evolved players and fixed-strategy benchmark players.

We evolve 48 IPD player types. For each of 4 genome representations, we evolve players with 6 different optimization objectives or pairs (see Table IV). In addition, we train players in competition against either the fixed-strategy benchmark players or the other players in the evolving population.

For each triple (R, C, O), where $R \in [\texttt{L64}, \texttt{L8}, \texttt{M8}, \texttt{FSM16}]$ is a representation, $C \in$ [TrainAxelrod, TrainPop] denotes the competitors, and $O \in$ [Multi, Single] indicates the objective type, we perform 32 training runs, each of which yields an evolved population of players. A single training run consists of 5000 generations of the genetic algorithm. In each generation, members are evaluated according to their performance in a round-robin tournament. We denote a run TrainAxelrod if individuals learn against benchmark players or TrainPop if they learn amongst themselves. Each

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TG.2020.3005124

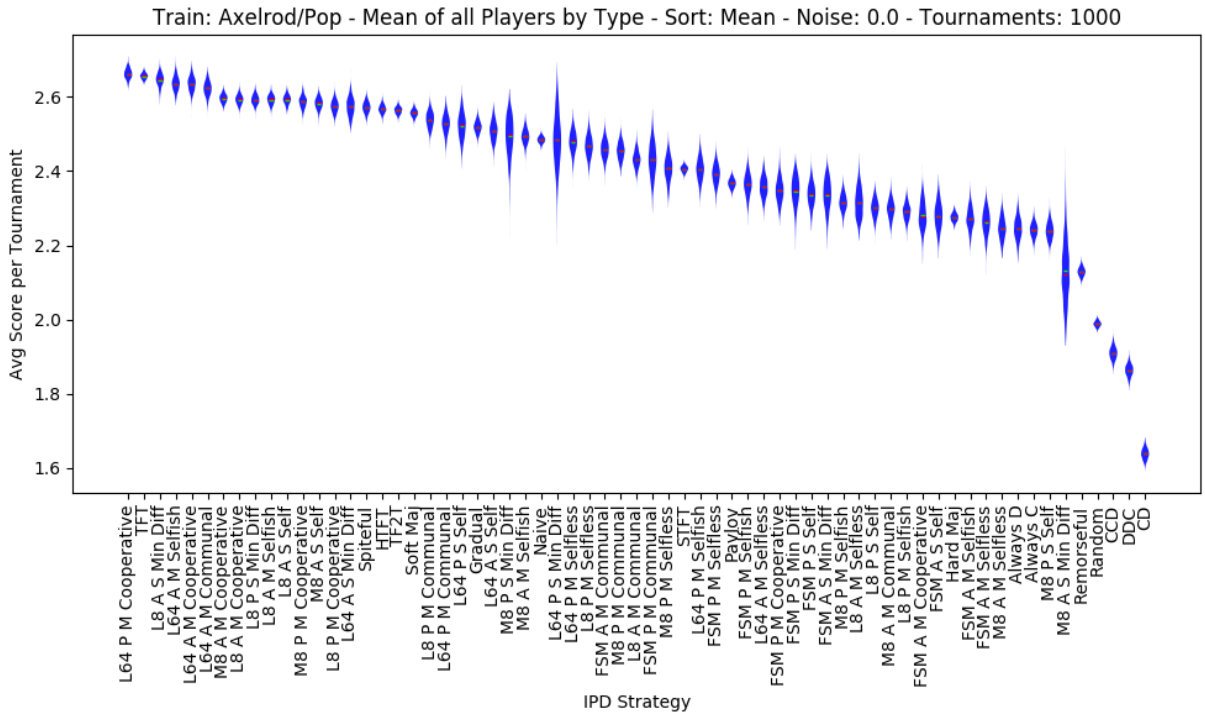

Fig. 4: Means of all players of each player type in each tournament. 1000 tournaments represented.

match in the tournament consists of 150 prisoner's dilemma rounds. The evolving population consists of 30 members with each objective pair (for multi-objective runs) or each single objective.

The benchmark population includes Axelrod's players as well as the Gradual strategy [16]. We include Gradual– a TFT-derived strategy– because it is a successful strategy that has a track record in the literature [19], [23].

For each training run, the top 20 individuals of each objective or objective pair are logged. These members are combined across the 32 trials and sorted with objective pair as the primary key and self score as the secondary key. The top 160 of these individuals for each objective or pair are stored for later use in the testing phase.

Testing takes the form of a round-robin tournament including evolved individuals and benchmark strategies (see Table II), independent of the training population. 10 individuals of each of the 4 evolved types are chosen at random, from the candidate pool of 160 formed above.

The tournament also includes 10 copies of each benchmark strategy. The total number of individuals is $10p$, where p is the number of player types participating. Each individual plays 150 rounds of prisoner's dilemma against $10(p-1)$ players, as they do not play against other individuals with the same player type. Each tournament is repeated 1000 times.

Our meta-algorithm is depicted in Figure 3. During testing, players are evaluated by only one objective: self-score. This is independent of the objective or objective pair used during evolution. Thus, success in testing demonstrates that a strategy is competitive in traditional forms of IPD rather than only according to the objectives for which they evolved their strategy.

Prior work has explored the impact of noise on results for the Iterated Prisoner's Dilemma [23], [24], [32], [33]. We repeat the training and testing described above with noise introduced into the game. This takes the form of flipping a player's decision with probability 0.05. Noise is applied to decisions by both genetic players and fixed strategies during both training and testing.

V. Results

Our experiments demonstrate the effects of several different parameters on the ability to evolve cooperative players for the Iterated Prisoner's Dilemma. These parameters include optimization objectives, representation of the genome for evolving players, the training set, and noise. In this section, we discuss the observed results.

Figure 4 shows the result of a round-robin competition that includes 65 player types. These include the 17 benchmark strategies and 48 evolved player types. For each of the six objectives/objective-pairs, there are 8 types differentiated by training set (TrainAxelrod, TrainPop) and representation (`L64`, `L8`, `M8`, `FSM16`). Competition is between all players with the restriction that two players of the same type cannot compete. An evolved player's type is comprised of the player's representation, objectives, and training set.

Each of the violin plots presented in this section represents results from 1000 tournaments. Each violin shows 1000 values, one for each tournament. Each of those values is the mean value of all players of the indicated player type in one tournament. An example to illustrate the encoding of player types on the x-axes of the plots and in the discussion in this section:

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TG.2020.3005124

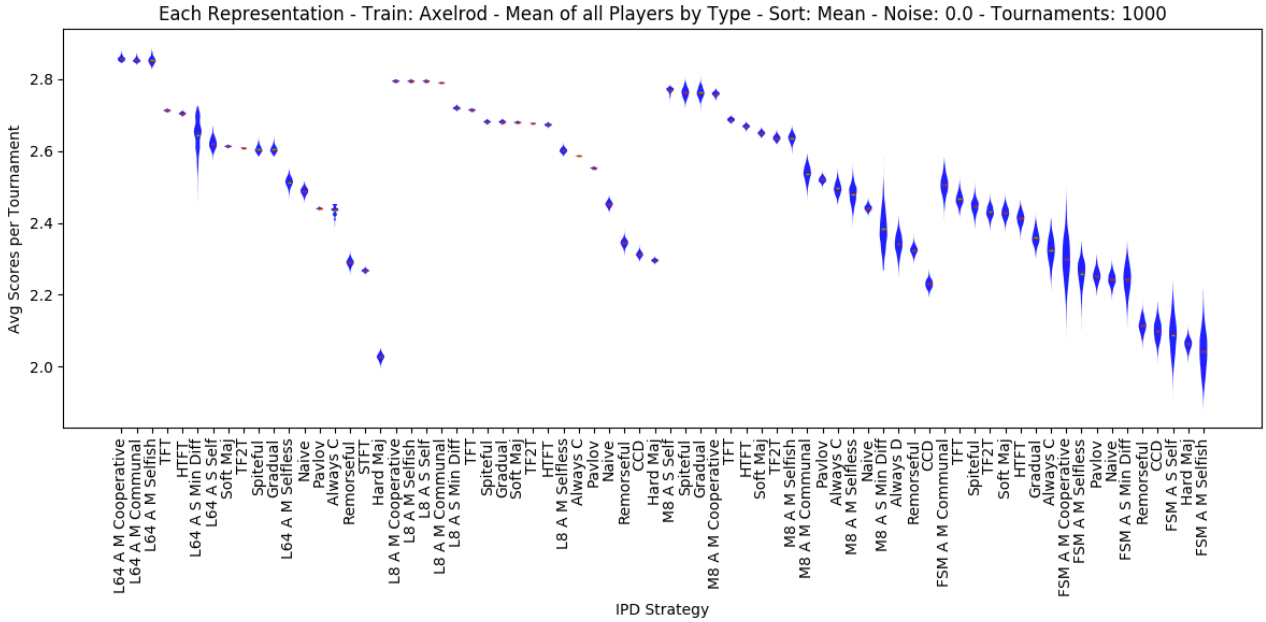

Fig. 5: Results for four round-robin competitions for players trained against the benchmark strategies. Each competition includes evolved players from only one representation. Mean of all players of each player type in each tournament. 1000 tournaments represented.

`L64 A M Cooperative` denotes the player type using the 64 row lookup table representation, TRAINAXELROD training set, and the multi-objective pair Cooperative.

In our experiments, the player type with the highest average score across 1000 tournaments is `L64 P M Cooperative`. The top 15 includes five additional Cooperative player types from three of the four representations. Thus, 40% of the top 15 player types are Cooperative. This top-line observation supports the hypothesis that players evolved to value cooperation are indeed competitive in IPD tournaments in which the players are evaluated solely by self-score.

A. The Effect of Objectives

The central question that inspired this work is this: Is it possible to evolve successful players for IPD by explicitly encouraging cooperation through the choice of optimization objectives? Traditionally, IPD players have been evolved with the single objective of maximizing their own score. The six objectives or objective pairs we explore are shown in Table IV. We do not include the single objective of maximizing the difference between self-score and opponent-score as Mittal and Deb showed this to be inferior to the Selfish objective pair [15]. We tested the single objective of maximizing mutual cooperation (one of the constituent values used in our Cooperative and Selfless objective pairs), however, it performs so poorly that we omit the results.

Figure 5 shows the results of four round-robin tournaments with populations trained against the benchmark strategies. Each tournament is for one of the four representations. Play in each tournament is restricted to the benchmark strategies and evolved players from one representation. Thus, each of the four sub-plots allows us to focus on the effect of the objectives used during evolution.

We observe that Cooperative players have the highest mean score of all player types in two of the sub-plots and fourth overall (second among evolved player types) in another. Communal players are also highly ranked in several of the competitions, finishing between first and fourth place.

B. The Effect of Representation

Ashlock *et al.* [22] describe and quantify the importance of considering different representations of evolved players when playing games, particularly the Iterated Prisoner's Dilemma. More generally, representation can significantly impact the ability of genetic algorithms to solve problems. We consider four representations for evolving players for IPD. These include two lookup tables, a Markov chain, and a finite state machine. See Section III-A for a full description.

Echoing previous work, representation is clearly very important for the results achieved by our model. Figure 4 shows that the two lookup table representations (`L64` and `L8`) tend to dominate the other representations. `FSM16` lags significantly behind the others. This is not surprising as `L64` stores the previous three moves for both itself and the opponent. Thus, it has more game state available for decision making than do `L8` and `M8`. We note that `FSM16` shows significantly greater variability than the other representations, demonstrated by the lengths of the elements in the violin plots. When judged by *best* score rather than *average* score, `FSM16` tends to looks much more competitive.

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TG.2020.3005124

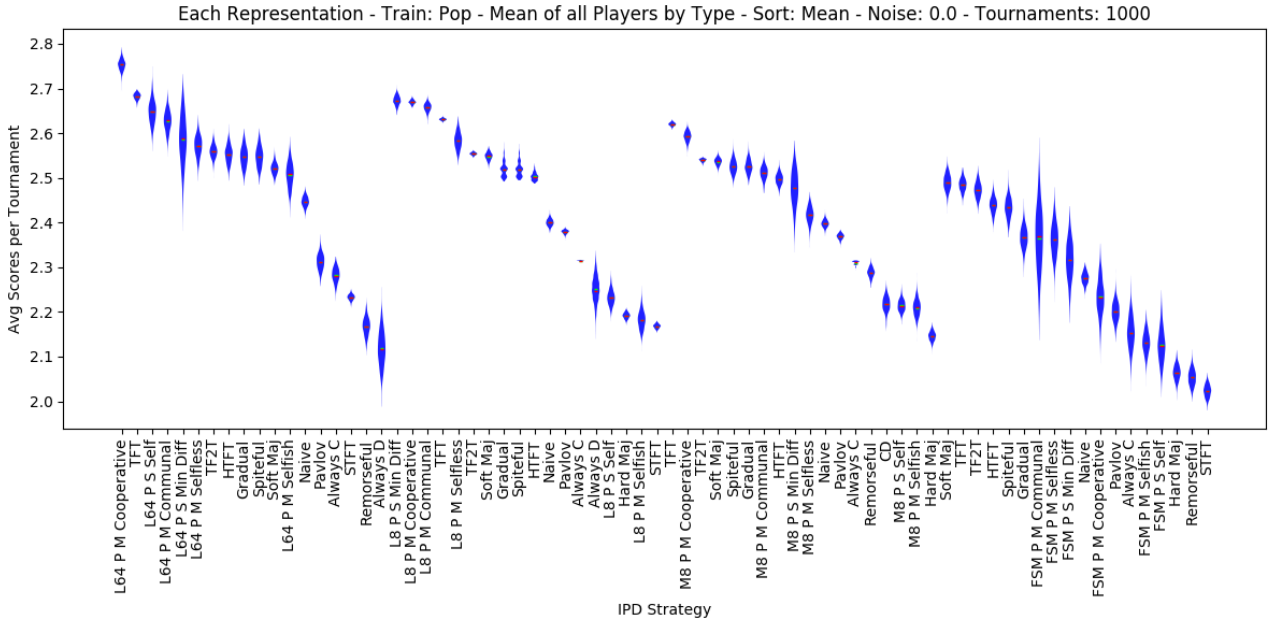

Fig. 6: Results for four round-robin competitions for players trained against the evolving population. Each competition includes evolved players from only one representation. Mean of all players of each player type in each tournament. 1000 tournaments represented.

C. The Effect of the Training Set

As described in Section IV, we run parallel sets of experiments distinguished by the population against which our evolving players are trained. The two training sets used are the evolving population itself and the benchmark strategies. Ashlock, Brown and Hingston [27] consider this question in detail. Their work with *examination boards* provides a more comprehensive examination than that provided here, as the effects of the training set is not our primary thesis.

Figure 6 depicts the same experiment types as Figure 5 but for populations trained against TRAINPOP rather than TRAINAXELROD. It is important to note that the round-robin tournaments shown in the figures have the same format independent of training set: competition within a representation is among all players, evolved and fixed. Comparing these plots allows a direct comparison of the two training sets.

TRAINPOP Players have greater variability than those trained against benchmark strategies. TRAINAXELROD players have somewhat higher scores than those trained against the evolving population, though the effect is small. This may be due to the makeup of the testing population in which there are more benchmark strategies than evolved types. Thus, TRAINPOP players are competing against more unfamiliar players than are their TRAINAXELROD counterparts.

Figure 4 allows a different comparison between training sets as these players compete directly against one another. Here we see that though the top type overall type is trained against the population, eight of the top 10 evolved player types are trained against benchmark strategies. Training set appears to have little impact on the success of evolving cooperation. This is evidenced by the similar scores for player types that differ only in training set. For example, two cooperative types (Cooperative and Communal) for each of `L64`, `L8`, and `M8` score very similarly for TRAINAXELROD and TRAINPOP.

D. The Effect of Defectors in the Training Set

For representation `L64`, we repeat the training described above with the addition of a significant number of defectors included in the training population. These players defect on every move. Playing against a defector, our Cooperative and Selfless players cannot attain a positive score for their maximizing mutual cooperation objective. In this scenario, a player must learn that it is hopeless to attempt to coax their opponent into cooperation and settle for their maximal achievable score of $(1, 1)$. The question is if learning to settle in this way affects the success of the player when competing against other strategies, particularly when the number of defectors used in training is large.

To examine this question, we evolved populations with 0, 30 and 120 defectors added to the training set. The results indicate that the addition of defectors has not significant effect on the tournament results, in terms of finishing rank or average score, for either training set. Therefore, we do not consider added defectors for the other representations.

E. The Effect of Noise

The introduction of noise dramatically changes the results of our competition. Figure 7 shows the results for a tournament with the same format as that in Figure 4. Noise significantly reduces the success of cooperative strategies as evidenced by the top 5 strategies in Figure 7. All are single or multi-objective selfish player types.

Despite its poor performance in noise-free competitions, `FSM16` Cooperative and Communal players are more robust

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TG.2020.3005124

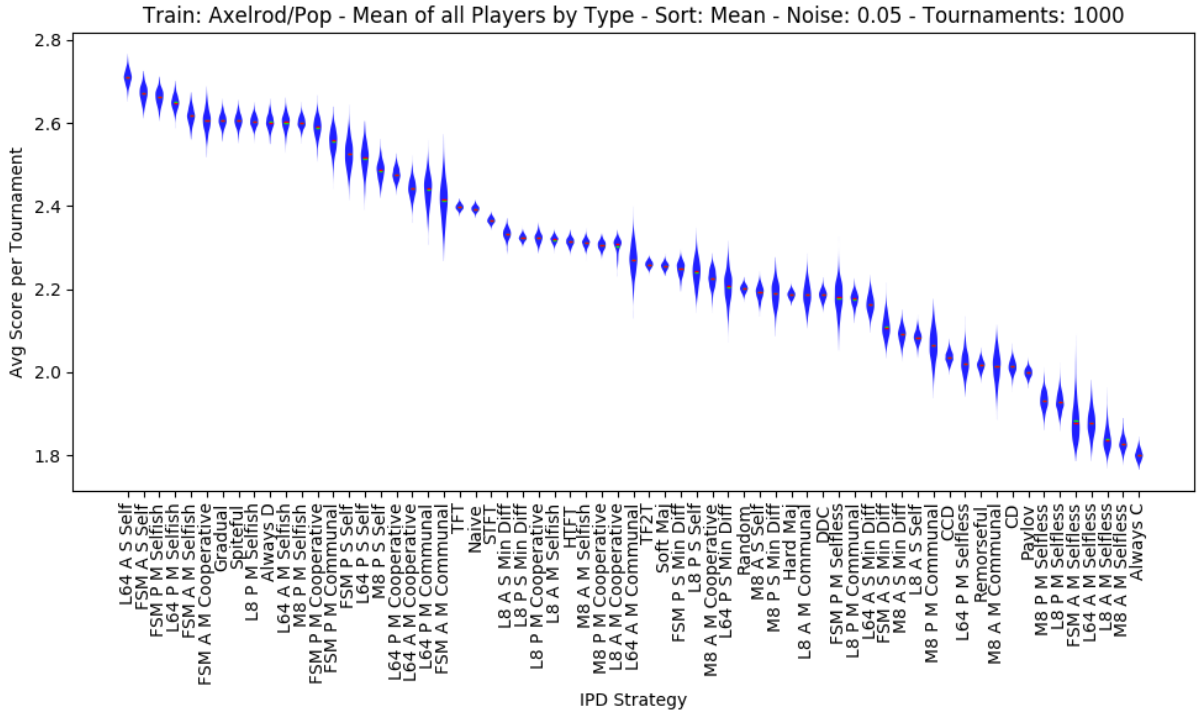

Fig. 7: Means of all players of each player type in each tournament. 1000 tournaments represented.

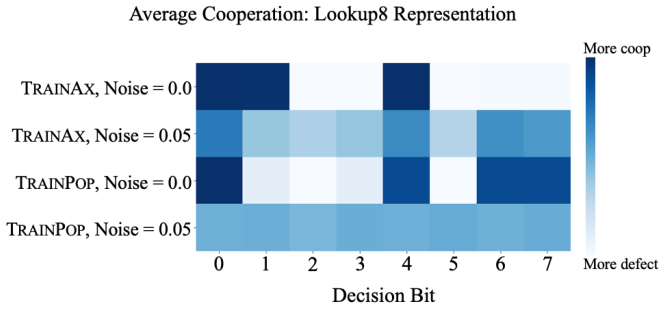

Fig. 8: Summary of decision bits for 4 different experiments using the `L8` representation.

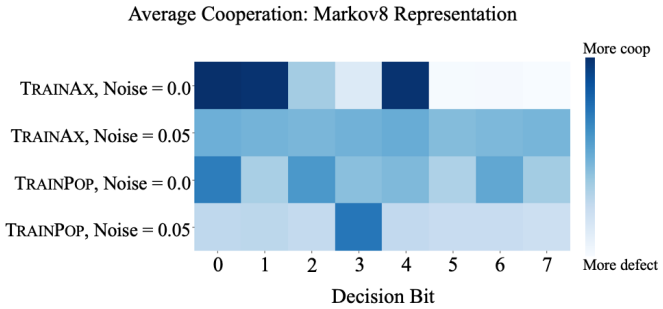

Fig. 9: Summary of decision bits for 4 different experiments using the `M8` representation.

against noise than those for other representations. The effect is particularly strong for `L8` and less pronounced for `L64`. Noise also reverses the relative rankings of the single-objective Self players and multi-objective Selfish players. Without noise, Selfish players are higher ranked while with noise Self players are higher ranked for `L64` and `FSM16` representations.

Figures 8, 9, and 10 illustrate the decrease in cooperation in the presence of noise. For representation `L8` with TRAINPOP the effect of noise is to create a strategy with an almost uniform probability of cooperation in the aggregate. Figure 10 confirms that the effect is less pronounced for representation `L64`. This is likely due to noise-caused disruption of long strings of cooperative behavior.

In a recent, large-scale IPD study, Harper, *et al.* run tournaments for a large number of strategies with and without noise [23]. Their players, trained using reinforcement learning, see a similar, though more pronounced, decline in ranking in the presence of noise. This effect is not unexpected due to the disruption mentioned above.

VI. CONCLUSION

In this work, we find that cooperation does indeed pay off: when players are trained to cooperate, they are competitive with, and often outperform, selfish players in a round-robin Iterated Prisoner's Dilemma tournament in which the sole measure of success is self score. We compare the performance of 81 different player types for the IPD, across different optimization objectives, representations and training sets both in the absence of and presence of noise. We find that although there are differences in the ranking of cooperative players in our experiments, cooperation is always a strong contender. Our results support the traditional strength of selfish players and introduce cooperative strategies that often outperform these selfish strategies.

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TG.2020.3005124

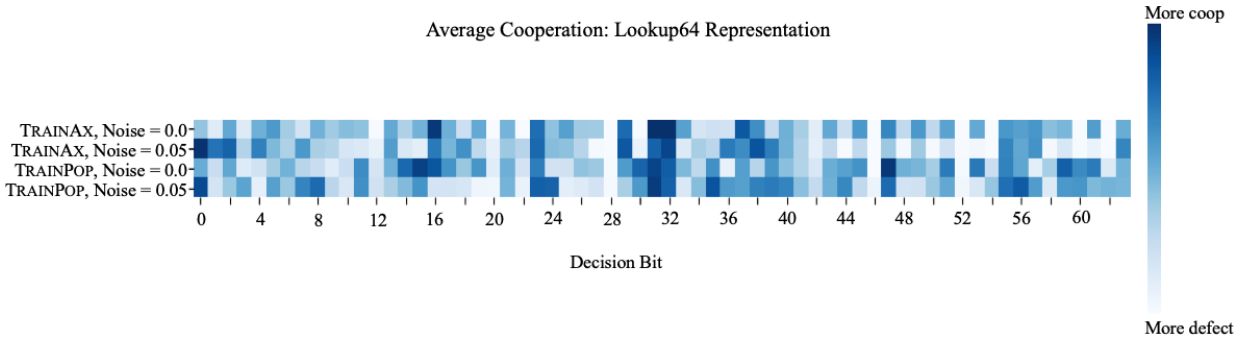

Fig. 10: Summary of decision bits for 4 different experiments using the `L64` representation.

ACKNOWLEDGEMENTS

The authors are very grateful to the anonymous reviewers for their many valuable suggestions that informed significant improvements in this work.

Jessie Finocchiaro's work is funded by an NSF Graduate Research Fellowship under Grant No. DGE 1650115.

This work utilized the Extreme Science and Engineering Discovery Environment (XSEDE) [34], which is supported by National Science Foundation grant number ACI-1548562. We used XSEDE compute resource Comet at the San Diego Supercomputer Center through allocation TG-CIE170029.

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TG.2020.3005124

REFERENCES

[1] J. F. Nash, "Equilibrium points in n-person games," *Proceedings of the National Academy of Sciences*, vol. 36, pp. 48–49, 1950.

[2] R. Axelrod, *The Evolution of Cooperation*. New York: Basic Books, 1989.

[3] D. B. Fogel, "Evolving behaviors in the iterated prisoner's dilemma," *Evolutionary Computation*, vol. 1, no. 1, pp. 77–97, 1993.

[4] M. A. Nowak, "Five rules for the evolution of cooperation," *Science*, vol. 314, no. 5805, December 2006.

[5] H. Ishibuchi, H. Ohyanagi, and Y. Nojima, "Evolution of cooperative behavior in a spatial iterated prisoner's dilemma game with different representation schemes of game strategies," in *Proceedings of the 2009 IEEE Conference on Fuzzy Systems*, 2009.

[6] G. Greenwood and D. Ashlock, "Evolutionary games and the study of cooperation: Why has so little progress been made?" in *Proceedings of the 2012 IEEE World Congress on Computational Intelligence*, 2012.

[7] H. Ishibuchi, T. Sudo, K. Hoshino, and Y. Nojima, "Effects of the number of opponents on the evolution of cooperation in the iterated prisoner's dilemma," in *Proceedings of the 2013 IEEE Conference on Systems, Man, and Cybernetics*, 2013, pp. 2001–2006.

[8] ——, "Evolution of cooperative strategies for iterated prisoner's dilemma on networks," in *Proceedings of the 5th International Conference on Computational Aspects of Social Networks*, 2013.

[9] T. LoFaro, "Crossing the threshold: the role of density dependence and demographic stochasticity in the evolution of cooperation," *Letters in Biomathematics*, vol. 2, no. 1, pp. 79–90, 2015.

[10] W. Wang, J. Hao, Y. Wang, and M. Taylor, "Toward cooperation in sequential prisoner's dilemma: a deep multiagent reinforcement learning approach," March 2018. [Online]. Available: https://arxiv.org/abs/1803.00162v1

[11] J. Nay and Y. Vorobeychik, "Predicting human cooperation," April 2016. [Online]. Available: https://arxiv.org/abs/1601.07792v2

[12] D. Madeo and C. Mocenni, "Self-regulation promotes cooperation in social networks," July 2018. [Online]. Available: https://arxiv.org/abs/1807.07848v1

[13] U. Fischbacher and S. Gächter, "Social preferences, beliefs, and the dynamics of free riding in public goods experiements," *American Economic Review*, vol. 100, no. 1, pp. 541–556, 2010.

[14] F. Guala, L. Mittone, and M. Ploner, "Group membership, team preferences, and expectations," *Journal of Economic Behavior and Organization*, vol. 86, pp. 183–190, 2013.

[15] 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.

[16] B. Beaufils, J. P. Delahaye, and P. Mathieu, "Our meeting with gradual, a good strategy for the iterated prisoner's dilemma," in *Artificial Life V: Proceedings of the 5th International Workshop Synthesis Simulation Living Systems*. MIT Press, 1996, pp. 202–209.

[17] W. Poundstone, *Prisoner's Dilemma/John von Neumann, Game Theory and the Puzzle of the Bomb*. Anchor, 1993.

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

[19] P. Mathieu and J.-P. Delahaye, "New winning strategies for the iterated prisoner's dilemma," *Journal of Artificial Societies and Social Simulations*, vol. 20, no. 4, October 2017.

[20] W. Ashlock, "Why some representations are more cooperative than others for prisoner's dilemma," in *Proceedings of the 2007 IEEE Symposium on Foundations of Computational Intelligence*, 2007, pp. 314–321.

[21] J. Li, P. Hingston, and G. Kendall, "Engineering design of strategies for winning iterated prisoner's dilemma competitions," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 4, pp. 348–360, 2011.

[22] D. Ashlock, E.-Y. Kim, and N. Leahy, "Understanding representational sensitivity in the iterated prisoner's dilemma with fingerprints," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 36, no. 4, pp. 464–475, 2006.

[23] M. Harper, V. Knight, M. Jones, G. Koutsovoulos, N. E. Glynatsi, and O. Campbell, "Reinforcement learning produces dominant strategies for the iterated prisoner's dilemma," *PLoS ONE*, vol. 12, no. 12, 2017.

[24] D. Ashlock, E.-Y. Kim, and W. Ashlock, "Fingerprint analysis of the noisy prisoner's dilemma," in *Proceedings of the 2007 IEEE Congress on Evolutionary Computation*, 2007, pp. 4073–4080.

[25] ——, "A fingerprint comparison of different prisoner's dilemma payoff matrices," in *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*. IEEE, 2010, pp. 219–226.

[26] D. Ashlock and E.-Y. Kim, "Fingerprinting: Visualization and automatic analysis of prisoner's dilemma strategies," *Evolutionary Computation*, vol. 12, no. 5, pp. 647–659, 2008.

[27] D. Ashlock, J. A. Brown, and P. Hingston, "Multiple opponent optimization of prisoner's dilemma playing agents," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 7, no. 1, pp. 53–65, 2015.

[28] M. Glomba, T. Filak, and H. Kwasnicka, "Discovering effective strategies for the iterated prisoner's dilemma using genetic algorithms," in *Proceedings of the 5th International Conference on Intelligent Systems Design and Applications*, 2005.

[29] H. Ishibuchi, H. Ohyanagi, and Y. Nojima, "Evolution of strategies with difference representation schemes in a spatial iterated prisoner's dilemma game," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 1, pp. 67–82, 2011.

[30] 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.

[31] F.-A. Fortin, F.-M. D. Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné, "Deap: Evolutionary algorithms made easy," *Journal of Machine Learning Research*, vol. 13, no. Jul, pp. 2171–2175, 2012.

[32] J. Wu and R. Axelrod, "How to cope with noise in the iterated prisoner's dilemma," *Journal of Conflict Resolution*, vol. 39, pp. 183–189, 1995.

[33] S. 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.

[34] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. R. Scott, and N. Wilkins-Diehr, "Xsede: Accelerating scientific discovery," *Computing in Science and Engineering*, vol. 16, no. 5, pp. 62–74, 2014.

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.