Proceedings of the 2007 IEEE Symposium on
Foundations of Computational Intelligence (FOCI 2007)

Why Some Representations Are More Cooperative Than Others For Prisoner's Dilemma

Wendy Ashlock
University of Guelph
Guelph, ON Canada N1G 2R4
washlock@uoguelph.ca

ABSTRACT

In [4] it was shown that the representation used has a large impact on the cooperativeness of evolved prisoner's dilemma strategies. Why is this? This paper examines the look-up table representation, the finite state machine representation, and the neural net representation to gain insight into this somewhat surprising result. A tool called a *prisoner's dilemma fingerprint* is used to compare the strategies produced by the different representations, and a Voronoi tiling (based on which of 12 reference strategies is the closest neighbor) of the strategy space is done. The initial random populations are shown to have significantly different distributions, and the evolved populations are shown to favor different parts of the strategy space.

I. INTRODUCTION

This paper explores strategies for playing prisoner's dilemma. Prisoner's dilemma is a two-player simultaneous game. The game is explored in detail with many excellent examples in [7]. Two players simultaneously decide either to defect or cooperate. If both defect, they receive a score of D. If both cooperate, both receive a score of C. If one defects and the other cooperates, then the defector receives a score of T (temptation) and the cooperator receives a score of S (sucker). Play is repeated many times. (In this study, C = 3; D = 1; T = 5; and S = 0 and play is repeated 150 times.) This game is interesting because there is no single "best" strategy; which strategy is best depends strongly on the opponent's strategy.

Many studies have been done using evolutionary computation with prisoner's dilemma. Some examples are [14], [12], [15], [11], [8], [10]. These studies either use simple strategies that are easy to analyze or they analyze their results in terms of population level statistics. This study uses a technique called fingerprinting to characterize complex strategies at the individual level so that they can be compared across representations. Then, it compares strategies in evolved populations using three different representations in an attempt to understand why two of the representations produce more cooperative populations than the other does.

In [4] ten representations for iterated prisoner's dilemma were studied. This paper looks in detail at three of these representations, neutral neural nets (NNN), look-up tables (LKT), and directly encoded finite state machines (AUT). In [4] it was found that evolved neutral neural nets were not at all cooperative and that evolved look-up tables and finite state machines were highly cooperative. Out of 400 populations, none of the neural net populations were cooperative; 333 (85%) of the look-up table populations were cooperative, and 351 (88%) of the finite state machine populations were cooperative. (Populations were deemed cooperative if their average score was greater than 2.8.) This paper analyzes the strategies found using the different representations in an attempt to understand this difference.

II. FINGERPRINTS

In order to analyze prisoner's dilemma strategies one needs some way to define and name each individual strategy. Table I lists twelve strategies with names. However, this naming method quickly becomes intractable as the number of strategies increases. Also, it gives no objective way to compare the strategies. A prisoner's dilemma fingerprint is a way of assigning a real-valued function to each strategy, or as with the simplified version used in this study, a point in 25-space which consists of 25 values of that function.

A *prisoner's dilemma fingerprint* is a function in two variables, x and y ranging from 0 to 1, whose value is the strategy's expected score when playing against a set of strategies called Joss-Ann strategies. These strategies were chosen to be as representative as possible of all possible strategies. If $x + y \leq 1$, the Joss-Ann strategy plays C with a probability of x and D with a probability of y and TFT otherwise. If $x + y \geq 1$, the

TABLE II
TABLE OF FINGERPRINT FUNCTIONS FOR SOME WELL-KNOWN STRATEGIES

Strategy	Fingerprint Function
TFT	$\frac{y^2+5xy+3x^2}{(x+y)^2}$
ALLD	$4x+1$
ALLC	$3(1-y)$
TF2T	$\frac{3x^2y+5xy^2-3xy+3x+y^2}{x^2y+2xy^2+x+y^3}$
2TFT	$\frac{3x^2+5x^2y+4xy+xy^2+y}{x^3+2x^2y+xy^2+y}$
PSYCHO	$\frac{4(y-1)(x-1)+5(y-1)^2}{2(x-1)(y-1)+(x-1)2+(y-1)^2}$

Joss-Ann strategy plays C with a probability of $1-y$, D with a probability of $1-x$ and PSYCHO otherwise.

Note that in either case when $x+y=1$, the strategy is just the random strategy which cooperates with probability x and defects with probability y. For more complete details, see [16], [5], [3], [4]. See Table II for examples of some fingerprint functions. This study uses an approximation to the fingerprint function consisting of 25 numbers which are the values of the fingerprint function on a 5 by 5 evenly-spaced grid from 1/6 to 5/6.

III. DATA GENERATION

All the evolutionary algorithms used to generate data for this paper use a population size of 36 with an elite of 24. Fitness is determined by a round robin tournament in which each pair of players plays for 150 rounds. Twelve pairs of parents are chosen by fitness proportional selection. They are crossed over and mutated. Their children replace the non-elite. 400 populations were run for each representation, each for 250 generations.

These parameters were chosen because they were the ones used in [4]. The authors of [4] chose them to be consistent with previous work. Tweaking these parameters could easily have an effect on cooperativeness. This study focuses on the effect of representation on cooperativeness by keeping the other parameters fixed.

The finite state machines used are 8-state Mealey machines. Each state has an action (C or D) and state transition (1-8) associated with each of the opponent's possible actions on the previous round. For each round, the machine plays the action in its current state based on the opponent's last action and moves to the state indicated. For each machine, an initial state and initial action are designated. For more information about using finite state machines in evolutionary computation, see Chapter 6 of [2].

In [4] 16-state machines were used. 8-state machines were used in this study to make them more similar to the neural net and look-up table representations used in terms of strategy complexity. They are stored as linear chromosomes with the initial state and action stored at the beginning. Two-point crossover is used (preserving whole states). The mutation operator changes either a state transition, an action, the initial state, or the initial action. This is a many-one representation.

The look-up tables consist of eight actions (coded 0 for cooperate, 1 for defect) indexed by the eight possibilities for the opponent's last three moves. Two-point crossover is used and point mutation. Each look-up table is unique. However, 14 have behavior similar enough to have the same fingerprint. Out of the 256 possible look-up tables, there are three with the TFT fingerprint, three with the PSYCHO fingerprint, four with the UC fingerprint, and four with the UD fingerprint; the other 242 look-up tables have unique fingerprints. (These strategies are described in Table I.) It is proved in [4] that look-up tables can be specified as finite state machines. When you convert the look-up tables to finite state machines, they have at most four states in their finite state machine minimal form.

The neural nets used are those presented in [13]. They have a hidden layer of three neurons and a single output neuron. The neurons are 0-1 threshold neurons. There are three inputs which represent the opponent's last three moves. Each input is connected to each neuron in the hidden layer and the neurons in the hidden layer are all connected to the output neuron. So, twelve connection weights are needed: 9 for the connections of the input neurons to the hidden layer and 3 for the connections of the hidden layer to the output neuron. These connection weights make up the player's chromosome. The connection weights are initialized in the range $-1 \le c \le 1$. Two-point crossover is used and a point mutation which adds a random number in the range $-0.1 \le x \le 0.1$ to a connection weight. Note that it is highly unlikely that any connection weight will ever be exactly 0 or exactly 1, so the output always depends in some nontrivial way on all of the inputs. Also, note that very often a mutation will make no change in the behavior of the neural net. Like look-up tables, neural nets can be converted to finite state machines. In this paper they are displayed that way in figures making them easier to understand.

IV. VORONOI TILING

In order to analyze the strategies, twelve reference strategies were selected. These strategies are all the possible look-up tables based on the opponent's last two moves. These are all equivalent to one-state or two-state finite state machines. A non-linear projection of the reference strategy fingerprints is shown in Figure 1. Nonlinear projection is a technique which uses an evolutionary algorithm to project multi-dimensional

TABLE I
STRATEGIES USED AS REFERENCE STRATEGIES FOR VORONOI TILING.

Abbrev.	Name	Description
ALLD	Always defect	This strategy always defects.
2TFT	Two-tits-for-tat	This strategy defects twice in response to defection and otherwise cooperates.
TFT	Tit-for-tat.	This strategy does whatever its opponent did last time.
TF2T	Tit-for-two-tats	This strategy cooperates except after a sequence of two defections.
UC	Usually cooperate	This strategy cooperates except after a C following a D.
ALLC	Always cooperate	This strategy always cooperates.
(2TFT)	Inverse two-tits-for-tat	This strategy does the opposite of what 2TFT would do.
PSYCHO	Psycho	This strategy does the opposite of what its opponent did last time.
TFT-PSY	Tit-fot-tat-psycho	This strategy plays like TFT until its opponent defects; then it plays like PSYCHO until its opponent cooperates.
PSY-TFT	Psycho-tit-for-tat	This strategy plays like PSYCHO until its opponent defects; then it plays like TFT until its opponent cooperates.
(TF2T)	Inverse tit-for-two-tats	This strategy does the opposite of what TF2T would do.
UD	Usually defect	This strategy defects except after a D following a C.

points onto a 2-dimensional picture which preserves the distance relationships between the points as closely as possible. It is defined in [6] and applied in [1] to RNA folding and in [2] to evolutionary robotics.

Notice that ALLD and ALLC are at either end of the figure. They have the largest fingerprint distance possible for any two strategies. (Proof available upon request.) In general, strategies near ALLD score well against strategies which are different from them, and poorly against strategies similar to themselves. ALLD gets the maximum possible score of 5 against ALLC but only a score of 1 against itself. In general, strategies near ALLC score best when playing strategies similar to themselves and are easily exploited by strategies far from them. ALLC gets a score of 3 when playing itself and a score of 0 when playing ALLD. Strategies in the middle of the diagram, like TFT, have greater potential to score well or poorly both against strategies similar to them and strategies far from them.

Strategies are analyzed in terms of which of these strategies they are closest to. This is called a Voronoi tiling or a Dirichlet tessellation. Explanations can be found in many places; one good one is [9]. If a strategy is closer to Strategy S than it is to any other reference strategy, then it is said to be in the Strategy S bin. The assumption is that strategies with fingerprints that are close together behave similarly, so we can group them together and analyze them in terms of the simple strategies that we understand well and in terms of where they fall in fingerprint space.

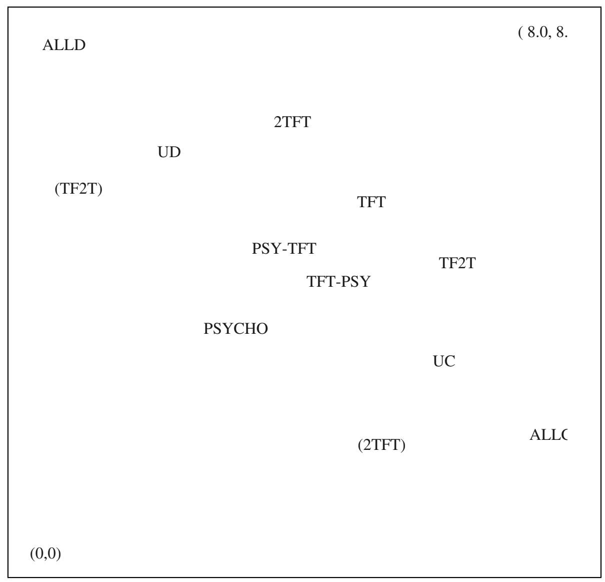

Fig. 1. Non-linear projection of reference strategies.

A. Understanding the Strategy Space

The strategies in the top left part of the space, ALLD, 2TFT, (TF2T), and UD, are highly uncooperative. When they play the Joss-Ann strategies represented by the 25 points used in the fingerprint, the average of their scores and the scores of the Joss-Ann strategies are 1.75, 1.96, 2.09, and 1.97, respectively. The strategies in the lower right part of the space, ALLC, (2TFT), UC, and TF2T are highly cooperative. Their averages when playing the 25 Joss-Ann strategies are 2.75, 2.50, 2.53, and 2.46. The strategies in the middle, PSYCHO, TFT, PSY-TFT, and TFT-PSY are sometimes cooperative. Their Joss-Ann averages are 2.28, 2.22, 2.19, and 2.31. Notice that none of these numbers are high enough (> 2.8) to be deemed cooperative according to the standard set in [4]. These numbers represent the degree of cooperativeness the strategies have against a wide range of opponents. Scores of play against a more uniform population could show a much higher or lower degree of cooperation.

The other extreme is to measure cooperativeness with a population consisting entirely of copies of the same strategy. Since evolved populations are inbred, this measure is closer to the measure used in [4]. We

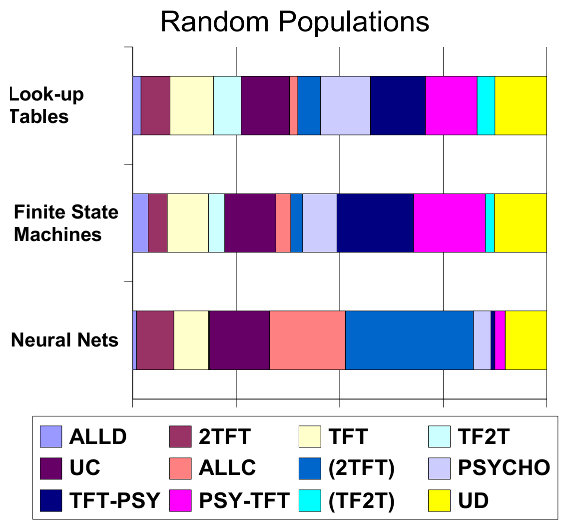

Fig. 2. Distribution of random strategies using the three different representations.

calculate this value as the average over all possible combinations of initial actions, since changing the initial action is a single mutation for all representations. Using this measure, the strategies in the top left have self-cooperativeness ALLD 1.01, 2TFT 1.51, (TF2T) 2.08, and UD 1.88. The strategies in the lower right have self-cooperativeness ALLC 3.00, (2TFT) 2.42, UC 2.63, and TF2T 3.00. The strategies in the middle have self-cooperativeness PSYCHO 2.25, TFT 2.25, PSY-TFT 2.33, and TFT-PSY 2.33.

Another interesting measure is the average score against all the Joss-Ann strategies. (Before we averaged this with the scores the Joss-Ann strategies got.) This is a measure of how good the strategy is against a variety of opponents. The values using this measure are: ALLD 3.00, 2TFT 2.47, (TF2T) 2.72, UD 2.67, ALLC, 1.50, (2TFT) 1.87, UC 1.83, TF2T 1.95, PSYCHO 2.28, TFT 2.22, PSY-TFT 2.34, and TFT-PSY 2.16. For these simple strategies, it is true that the less cooperative the strategy, the more effective it is.

V. RANDOM POPULATIONS

Figure 2 shows the distributions of strategies chosen at random (as in an initial population) using Voronoi tilings of each of the three representations. The neural net representation is clearly very different from the other two. Both look-up tables and finite state machines have significant numbers of random strategies in all twelve strategy bins. Neural nets have a significant number of representatives in only eight bins. The ALLD, TF2T, TFT-PSY, and (TF2T) bins are mostly empty. There are also larger concentrations in the ALLC bin and the (2TFT) bin than in those bins for random finite state machines and look-up tables.

Finite state machines and look-up tables have distributions that are more like each other than they are like the neural net distribution. However, a chi-square test shows them to be significantly different ($\alpha << 0.005$). Finite state machines have larger proportions of ALLD- and ALLC-like strategies, and look-up tables have a larger proportion of TFT-PSY- and PSY-TFT-like strategies (the strategies in the middle of fingerprint space). Also, since finite state machines are a many-one representation and look-up tables are mostly unique (UC, UD, TFT, and PSYCHO are the only strategies represented by more than one look-up table) the distributions of strategies within a tile are different.

Why does this matter? Since fitness in prisoner's dilemma depends on the population which the strategy is in, the composition of the initial population is very important. Since ALLC and (2TFT), both highly exploitable strategies, are so common in random neural net populations, a non-cooperative strategy capable of exploiting them is highly successful early in evolution. On the other hand, since strategies with fingerprints in the central region are more common with finite state machine and look-up table representations, there is a greater chance for strategies to evolve which are cooperative but not exploitable.

VI. EVOLVED POPULATIONS

The composition of the evolved populations is shown in Figures 3-5. The inner rings show the bins random strategies fall into, and the outer rings show the bins evolved strategies fall into. Notice that for the look-up tables nearly half and for the finite state machines nearly three-quarters of the strategies fall in the TFT bin, and that for the neural nets hardly any do (0.08%). For the evolved finite state machines, 9 bins have a noticeable number (more than 1%) of strategies in them (the PSYCHO, (2TFT), and (TF2T) bins are nearly empty). Evolved neural nets have strategies in 6 bins (UD, ALLD, (2TFT), PSYCHO, (TF2T), and UC), and evolved look-up tables have strategies in 5 bins (TFT, 2TFT, TF2T, TFT-PSY, and PSY-TFT). The neural nets and look-up tables are finding strategies in completely different parts of fingerprint space; the finite state machines and neural nets have some overlap; and the part of fingerprint space with strategies found by the look-up tables is a subset of the part of fingerprint space with strategies found by the finite state machines.

A. Population Diversity

Two measures of population diversity were used: entropy and variation from the mean fingerprint. Entropy is a measure used by biologists for diversity of species in a region. It is:

$$\text{Entropy} = -\sum_{i=1}^{N} \frac{N_i}{N} \cdot \log_2 \frac{N_i}{N}$$

where N = the total number of individuals and N_i = the number of individuals of species i. It gives a better measure of diversity than just counting the number of species, because it distinguishes an ecology with 91 individuals of one species and 1 individual from each of 9 other species from an ecology with 10 individuals from each of 10 different species. (The second ecology has more entropy.) For measuring entropy in this study, two strategies are considered to be of the same species if they have the same fingerprint.

The look-up tables have the most entropy of the three representations, and the neural nets have the least. When averaged over the 400 populations, the look-up tables have an entropy of 1.87; the finite state machines 1.14, and the neural nets 0.63. This means that the look-up tables have an average of roughly 2^{entropy} or 3.7 "species" in each population, the finite state machines roughly 2.2, and the neural nets roughly 1.5. (Besides these species, there may be some singletons who don't contribute much to the entropy.) The population with the most entropy (4.25 or roughly 19 species) was found among the finite state machines. The neural nets had 191 populations with zero entropy (meaning only one fingerprint present); the finite state machines had 83, and the look-up tables had only 10.

The other diversity measure is the variance from the mean fingerprint. Using this measure, the neural nets were the most diverse with an average variance of 1.29. The look-up tables had an average variance of 0.67, and the finite state machines had an average variance of 0.69. To give you a scale for this, the average distance from a bin center to the next closest center is 1.70, and the farthest apart any two fingerprints can be (the distance from ALLD to ALLC) is 9.54. So, on average the fingerprints within a single population are pretty close together with the neural net strategies more spread out than those of the other two representations. The population with the highest fingerprint variance was one which used the neural net representation. It had a variance of 4.17 which is comparable to the distance between TFT-PSY and ALLC. The most the finite state machines varied was 2.93 which is comparable to the distance between TFT-PSY and (2TFT). The look-up tables' maximum variance was 1.60, about the distance between TFT and TF2T.

It is interesting that the representation which produces populations with the lowest entropy has the highest fingerprint variance (neural nets). This means that, when a neural net population has a variety of different fingerprints, they are often *very* different. Likewise, the representation with the highest entropy had the lowest fingerprint variance (look-up tables). This is because look-up table populations almost always have a variety of fingerprints, but those fingerprints are usually close together.

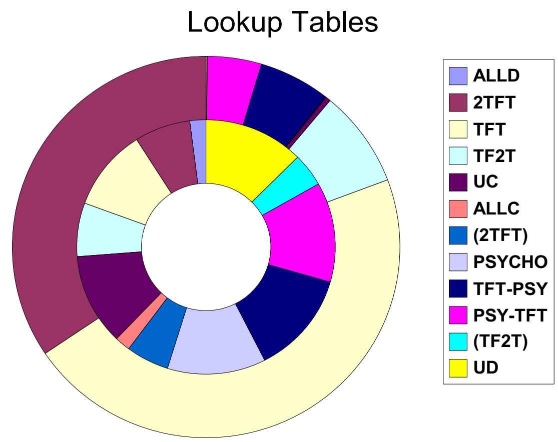

Fig. 3. Distribution of Strategies for Look-up Table Representation. Inner ring shows distribution of random strategies; outer ring shows distribution of strategies evolved for 250 generations.

B. Look-up Tables

Figure 3 shows the strategy distribution for the evolved look-up tables. 375 out of 400 populations (94%) have some strategies from the TFT bin. 17 (4%) of the populations are made up of strategies entirely from the TFT bin. (The only other bin which contains entire populations is 2TFT which has 16.) Only about 4% of the strategies in the TFT bin are exactly TFT. The most common is the strategy shown in Figure 6. This strategy is like TFT, but less prone to get stuck in a mutual defection or mutual cooperation loop. It does not appear in either the finite state evolution or the neural net evolution. Likewise, in the 2TFT bin, the exact strategy 2TFT is rare, occurring only 0.2% of the time. Half the strategies in this bin are the one shown in Figure 7 as a finite state machine. This is a variation on 2TFT in which the second tit is delayed by one move. When converted to finite state machines reduced to their minimal representation, the evolved look-up tables use an average of 3.60 states. This means they are finding the most complex (at least by this measure) strategies of the three representations.

C. Finite State Machines

Figure 4 shows the strategy distribution for the evolved finite state machines. About one-fourth of the finite state populations have strategies contained entirely within a single bin, and two-thirds of the populations are almost entirely contained within one bin (20 or more

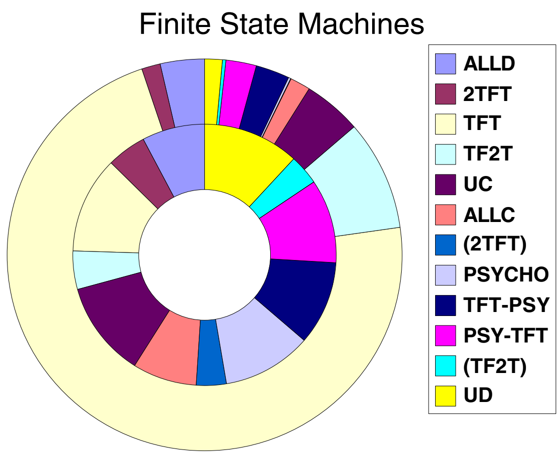

Fig. 4. Distribution of Strategies for Finite State Machine Representation. Inner ring shows distribution of random strategies; outer ring shows distribution of strategies evolved for 250 generations.

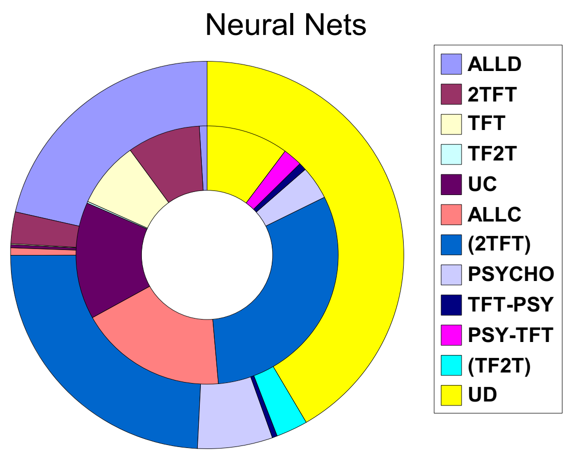

Fig. 5. Distribution of Strategies for Neural Net Representation. Inner ring shows distribution of random strategies; outer ring shows distribution of strategies evolved for 250 generations.

individuals out of 24). Eight of the twelve bins contain at least one of these populations. The TFT bin contains 98 entire populations and 136 nearly entire populations for a total of 234 populations or 59%. The ALLD bin contains 3 entire populations and 8 nearly entire populations, a total of 3% of the populations. Only 29 populations (7%) contain no strategies from the TFT bin. The TFT bin, the ALLD bin, and the ALLC bin are filled almost entirely with their exact reference strategy (more than 90%). The UD reference strategy makes a significant showing in its bin (43%), the UC strategy in its bin (20%), and the 2TFT strategy in its bin (17%). However, the exact TF2T strategy barely shows up (6%) in its bin nor does the PSY-TFT strategy (3%) in its bin, and the TFT-PSY strategy doesn't appear at all.

D. Neural Nets

Figure 5 shows the strategy distribution for the evolved neural nets. Neural net populations are even more prone than finite state machine populations to be entirely contained within a single bin. More than half (52%) are, and all six of the bins with a noticeable number of strategies contain entire populations. There are 127 such populations in the UD bin. The exact strategy UD does not appear in this bin, but the strategy in Figure 8 is the most commonly found strategy in this bin, representing 86% of the strategies in it. Two other strategy bins dominate the strategy distribution, ALLD and (2TFT). ALLD has 17 populations entirely contained within it, and all of the strategies in the ALLD bin are exactly ALLD. (2TFT) has 43 populations entirely contained within it, and 52% of the strategies contained within it are exactly (2TFT). The other three bins, 2TFT, (TF2T), and PSYCHO, all contain a variety of strategies. The 2TFT bin contains 8 entire populations; the (TF2T) bin contains 2 entire populations, and the PSYCHO bin contains 10 entire populations.

Opponent	Action	TFT's Action
CCC	C	C
CCD	D	D
CDC	C	C
CDD	C	D*
DCC	D	C*
DCD	D	D
DDC	C	C
DDD	D	D

Fig. 6. Strategy in the TFT bin for evolved look-up tables occurring 2444 times compared with TFT (starred actions are different).

It is surprising that there are virtually no TFTs in the evolved neural nets since this strategy is so common among the evolved strategies for the other two representations. They do exist in the initial random population, but they do not survive the evolutionary process. In the initial random population there are three distinct strategies that show up in the TFT bin, none of them exactly TFT. 90% of the random strategies in the TFT bin are the strategy shown in its finite state machine form in Figure 9.

The bin which is most full for the random strategies, (2TFT) a very cooperative strategy, is dominated by two distinct strategies, (2TFT) itself and a 3-state variation of it. (2TFT) is about 30% more common. Both these strategies also dominate the evolved (2TFT) bin in roughly equal numbers. Even though this strategy is at the cooperative end of the fingerprint diagram and thus cooperates with many strategies, it does not cooperate well with itself. Against itself, it plays CDCDCDCD... yielding an average score of 2.33. So, populations containing it are

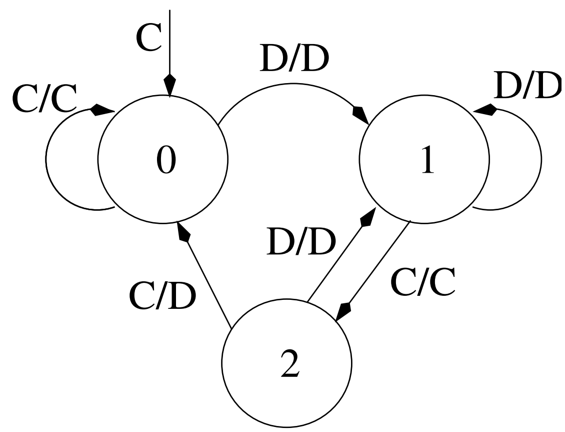

Fig. 7. Finite state representation of the strategy in the 2TFT bin for evolved look-up tables which occurs 1688 times.

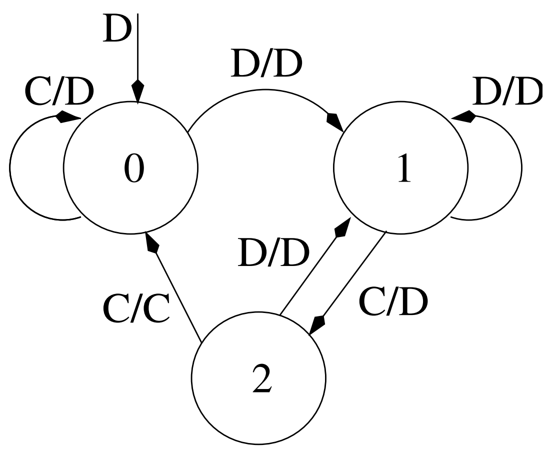

Fig. 8. Finite state representation of the strategy in the UD bin for evolved neural nets which occurs 3420 times.

not cooperative.

E. The Importance of TFT

The strategy TFT seems to be key in terms of evolving cooperative populations. An inability to easily produce a TFT strategy makes evolving cooperative populations difficult (perhaps impossible). Of the three representations studied the two which produced cooperative behavior, finite state machines and look-up tables, can easily represent TFT, while the neural net representation that can't evolved uncooperative populations. A single mutation can turn a finite state strategy into TFT by making a state which cooperates in response to cooperation and defects in response to defection loop to itself. Because neutral mutations are common with finite state machines (just mutate a state which is never reached), this strategy is also quite robust to further mutation. This is why so many of the evolved strategies are exactly TFT. A look-up table requires more mutations

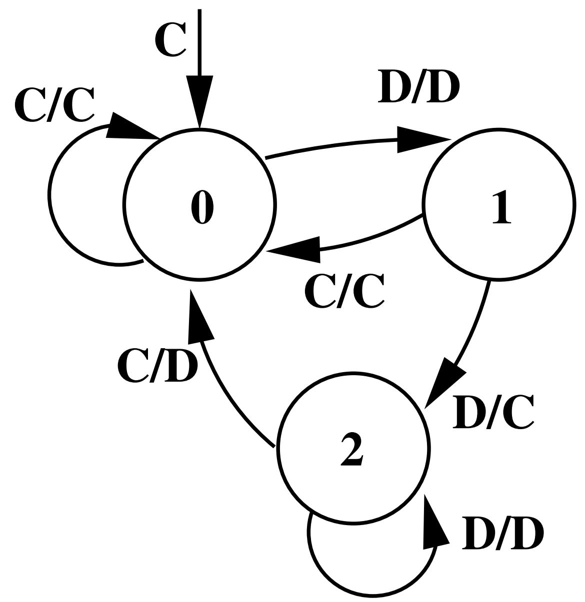

Fig. 9. Finite state machine form of the neural net strategy which fills 90% of the random neural net TFT bin.

but can also easily form TFT. However, since it is not a many-one representation, any mutation varies the strategy. As a result, the evolved populations contain many strategies close to but not exactly TFT. The neural net representation is extremely unlikely to produce an exact TFT strategy through mutation. To do this it would have to ignore its opponent's next-to-last and third-from-last moves which means having connection weights of 0 or 1 for those inputs. There are virtually no strategies close to TFT in the evolved neural net populations. The evolved strategies are mostly ALLD, the strategy in Figure 8 which is not far from ALLD, or at the ALLC end of the strategy spectrum, close to (2TFT). This last is not a particularly good strategy (it hardly shows up at all in the evolved populations for the other two representations), but it is common in random neural net populations and seems to be able survive evolution.

VII. CONCLUSION

Representation is one of the many parameters which can be tweaked in an evolutionary algorithm to change its behavior. It defines the search space and how the algorithm moves through it. In a co-evolutionary algorithm it is an even more important parameter because of the effect it has on fitness evaluation. How fit an individual is depends on the composition of the rest of the population. Strategies whose success depends on exploiting other strategies are only fit when in the company of exploitable strategies. Other strategies can only be successful when there is a threshold number of other strategies like them present in the population.

It was shown in [4] that finite state machine and look-up table representations evolved cooperative populations and that the neural net representation evolved uncooperative populations. This study shows that not only are the strategies different using this crude measure, but they are also different using the more precise fingerprint measure. The differences result from differences in the initial populations, differences in the way the variation operators work, and differences in which strategies compete with which other strategies.

The strategies were classified using a Voronoi tiling. The representations were significantly different from each other in terms of the distributions of strategies in their initial populations and in their evolved populations. The finite state machine and look-up table representations started with strategy distributions in their initial populations that were more alike than either were to the strategies in the neural net initial populations. After evolution, the strategy distributions were even more different. There were some strategies that were common among the neural nets which were nonexistent in the populations evolved using the other two representations.

The look-up table strategies were the most diverse using an entropy measure; the neural net strategies were the most diverse using a variance from the mean measure; and the finite state machines found strategies in a larger proportion of the total search space. Look-up tables produce strategies that are similar but not identical which coexist well. This is related to the fact that it is not a many-one representation. Neural nets produce strategies that are at either end of the strategy spectrum, either close to ALLD or close to ALLC, but not in the middle close to TFT. Their populations tend to be uniform, but when there is a difference it is usually a large difference in terms of fingerprint distance. This is a result of which strategies are easiest to produce using the representation and of the fact that TFT is hard to represent. Finite state machines also tend to produce uniform populations, but they search a larger proportion of the space. Eight of the twelve possible strategy bins contained finite state populations. However, TFT dominated the strategies found, even more so than for the populations using the look-up table representation. This is due to the many-one nature of the finite state representation together with its greater flexibility and the ease by which the variation operators can create TFT.

The results of this study suggest a way to change the neural net representation to make it more cooperative. In [4] a modification was made to the threshold level of the neurons making cooperation more likely. This did not result in an increase in cooperativeness. This study shows that what is needed instead is a modification making strategies in the middle of the strategy distribution (like TFT) easier to represent. One possible way to do this would be a mutation that allowed the strategy to ignore one or more of its inputs. Testing this modification is a next step for this research as well as analyzing the strategies produced by the other representations in [4].

REFERENCES

[1] D.A. Ashlock and J. Schonfeld. Depth annotation of rna folds for secondary structure motif search. In *Proceedings of the 2005 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, page 3845, 2005.

[2] Daniel Ashlock. *Evolutionary Computation for Opimization and Modeling*. Springer, New York, 2006.

[3] Daniel Ashlock and Eun-Youn Kim. Techniques for analysis of evolved prisoners dilemma strategies with fingerprints. In *Proceedings of the 2005 Congress on Evolutionary Computation*, volume 3, page 26132620, 2005.

[4] Daniel Ashlock, Eun-Youn Kim, and Nicole Leahy. Understanding representational sensitivity in the iterated prisoners dilemma with fingerprints. *IEEE Transactions on Systems, Man, and CyberneticsPart C: Applications and Reviews*, 36(4):464475, July 2006.

[5] Daniel Ashlock, Eun-Youn Kim, and Warren vonRoeschlaub. Fingerprints: Enabling visualization and automatic analysis of strategies for two player games. In *Proceedings of the 2004 Congress on Evolutionary Computation*, volume 1, page 381387, 2004.

[6] Daniel Ashlock and Justin Schonfeld. Nonlinear projection for the display of high dimensional distance data. In *Proceedings of the 2005 Congress on Evolutionary Computation*, volume 3, page 27762783, 2005.

[7] Robert Axelrod. *The Evolution of Cooperation*. Basic Books, New York, 1984.

[8] Siang Y. Chong and Xin Yao. Behavioral diversity, choices and noise in the iterated prisoners dilemma. *IEEE Transactions on Evolutionary Computation*, 9(6):540551, 2005.

[9] M. de Berg, M. van Kreveld, M. Overmans, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, 2000.

[10] D. B. Fogel and P. G. Harrald. Evolving continuous behaviors in the iterated prisoners dilemma. *Biosystems*, 37:135145, 1996.

[11] Nelis Franken and Andries P. Engelbrecht. Particle swarm optimization approaches to coevolve strategies for the iterated prisoners dilemma. *IEEE Transactions on Evolutionary Computation*, 9(6):562579, 2005.

[12] Patrick Grim. Spatialization and greater generosity in the stochastic prisoners dilemma. *Biosystems*, 37:317, 1996.

[13] Simon Haykin. *Neural Nets, a Comprehensive Foundation*. Macmillan College Publishing, New York, 1994.

[14] Enda Howley and Colm ORiordan. The emergence of cooperation among agents using simple fixed bias tagging. In *2005 IEEE Congress on Evolutionary Computation*, volume 2, page 10111016, Piscataway, NJ, 2005. IEEE Press.

[15] Hisao Ishibuchi and Naoki Namikawa. Evolution of iterated prisoners dilemma game strategies in structured demes under random pairing in game playing. *IEEE Transactions on Evolutionary Computation*, 9(6):552561, 2005.

[16] Eun-Youn Kim. *Fingerprinting: Automatic Analysis of Evolved Game Playing Agents*. PhD thesis, Iowa State University, 2005.