Proceedings of the 2007 IEEE Symposium on
Computational Intelligence and Games (CIG 2007)

# Cooperation in Prisoner's Dilemma on Graphs

Daniel A. Ashlock
Mathematics and Statistics
University of Guelph
Guelph, ON Canada N1G 2R4
dashlock@uoguelph.ca

*Abstract*— **A combinatorial graph can be used to place a geography on a population of evolving agents. In this paper agents are trained to play Prisoner's dilemma while situated on combinatorial graphs. A collection of thirteen different combinatorial graphs is used. The graph always limits which agents can mate during reproduction. Two sets of experiments are performed for each graph: one in which agents only play prisoners dilemma against their neighbors and one in which fitness is evaluated by a round robin tournament among all population members. Populations are evaluated on their level of cooperativeness, the type of play they engage in, and by identifying the type and diversity of strategies that are present. This latter analysis relies on the fingerprinting of players, a representation-independent method of identifying strategies. Changing the combinatorial graph on which a population lives is found to yield statistically significant changes in the character of the evolved populations for all the metrics used.**

**Keywords:** Prisoner's Dilemma, Spatial Algorithm, Evolutionary Computation.

## I. Introduction

The prisoner's dilemma [15], [14] is a classic model in game theory. Two agents each decide, without communication, whether to cooperate (C) or defect (D). The agents receive individual payoffs depending on the actions taken. The payoffs used in this study are shown in Figure 1. In the iterated prisoner's dilemma (IPD) the agents play many rounds of the prisoner's dilemma. IPD is widely used to model emergent cooperative behaviors in populations of selfishly acting agents and is often used to model systems in biology [29], sociology [22], psychology [28], and economics [21]. This study continues a series of experiments that seek to understand the dynamics of the evolution of agents that play iterated prisoner's dilemma. In [20], [25] the techniques for evolving finite state automata to play the iterated prisoner's dilemma are studied. In [30], [9] a mechanism for permitting agents to choose which agents they will play, and to refuse offers of play from those they find unacceptable is developed. The key point in these papers relevant to the current study is that the limitation of partner choice enabled by cooperation and refusal created a substantial increase in the level of cooperation that appeared in the evolving agents.

In [12] a curious phenomenon was studied in which agents permitted more time to evolve were found to gain in their ability to defeat agents that had evolved for a shorter time even though the agents were from separate evolutionary lineages. The phenomenon was called *non-local adaptation*. A later study [11] on the same phenomenon, with agents using tags, confirmed the result. These studies suggested that new features appear over long evolutionary time. A later study [13] demonstrated that there are strategies that only appear after tens of thousands of generations of evolution, suggesting one possible mechanism for non-local adaptation. The new agent types were very similar to the always defect strategy, but able to cooperate with copies of themselves, by using a form of "password" encoded within the finite state machine representation used. Non-local adaptation has been observed in other contexts than the prisoner's dilemma. It appears in a model of competitive exclusion by plants in [18]. Non-local adaptation is observed in agents playing the game divide the dollar in [2]. An artificial predator-prey system is shown to exhibit non-local adaptation in [8]. In [1] a population of virtual robots given the task of painting a floor competitively (in two colors) exhibit non-local adaptation.

In [31], [11] a technique called *fingerprinting* was developed for identifying which strategies were present in a population of evolved agents. This technique was used in [13], [5], [6], [7] to understand which strategies arose in different representations for agents. In [19] a continuous versions of the IPD using a neural net representation is compared with the usual discrete version of the game with finite state machines. The study in [7], continued in [5], investigates the effect of changing the representation used for a prisoner's dilemma agent. The representations covered by the two studies are two version of feed forward neural nets (one biased at the neuron level toward cooperation), Boolean parse trees [16], with and without a one-step time delay operation, a linear genetic programming representation called an ISAc list[10], look-up tables, a type of Markov chain [27], and both a direct and cellular [5] representation of finite state machines. The change of representation, with other factors held as near to constant as possible, yielded a change from 0% to 95% in the probability that final populations were cooperative.

One clear implication of the literature reviewed is that the details of how an evolutionary algorithm trains prisoner's dilemma agents can have a huge impact on the degree to which cooperation emerges as well as the strategies that arise during evolution. In this study an additional source of variation in this regard is examined: geography. In [3], [17], [4] it was shown that placing the population of an evolutionary algorithm performing optimization on a combinatorial graph can have a substantial effect on time to solution, over 9-fold faster in some cases. Such algorithms are called graph based

| | | Player 1 | |
|---|---|---|---|
| | | C | D |
| Player 2 | C | (3,3) | (5,0) |
| | D | (0,5) | (1,1) |

Fig. 1. The score matrix used for the prisoner's dilemma in this study. Scores are given for (Player 1,Player 2).

evolutionary algorithms (GBEAs). The geography specified in the form of the combinatorial graph acted to modify the flow of genetic information within the population. Different graphs were shown to be the best (and worst) for different problems. This study joins the GBEA line of investigation with that investigating the evolution of prisoner's dilemma playing agents.

The experiments in this study will perform standard experiments for evolving finite state machines to play the iterated prisoner's dilemma save that a graph will be used to restrict mating of the agents and, in half the experiments, the set of agents that they play against during fitness evaluation. Other studies, e.g. [26], [24], [23], place a spatial structure on the iterated prisoner's dilemma. These studies used far simpler representations than the eight-state finite state machines used in this study. In addition some of the analysis tools used in this study are quite recent and not available to the earlier studies. In essence none of the other studies treat the same version of spatially structured iterated prisoner's dilemma.

The remainder of the study is structured as follows. In Section II a brief introduction to graph theory is given and the combinatorial graphs used in the study are defined. The details of the evolutionary algorithm and analysis techniques are given in Section III. Results are presented in Section IV. Discussion and possible next steps are given in Section V.

## II. Mathematical Background

Some familiarity with graph theory is assumed. An excellent reference in the area is [32]. The theory required in this study is reviewed here. A *combinatorial graph* or *graph* $G$ is a collection $V(G)$ of vertices and $E(G)$ of edges where $E(G)$ is a set of unordered pairs from $V(G)$. Two distinct vertices of the graph are *neighbors* if they are members of the same edge. The number of edges containing a vertex is the *degree* of that vertex. If all vertices in a graph have the same degree, then the graph is said to be *regular*. If the common degree of a regular graph is $k$, then the graph is said to be $k$*-regular*. A graph is *connected* if one can go from any vertex to any other vertex by traversing a sequence of vertices and edges. The *diameter* of a graph is the largest number of edges in a shortest path between any two of the vertices. The diameter is, in some sense, the shortest path across the graph.

In this paper, a graph used to constrain mating and prisoner's dilemma play in a population will be called the *population*

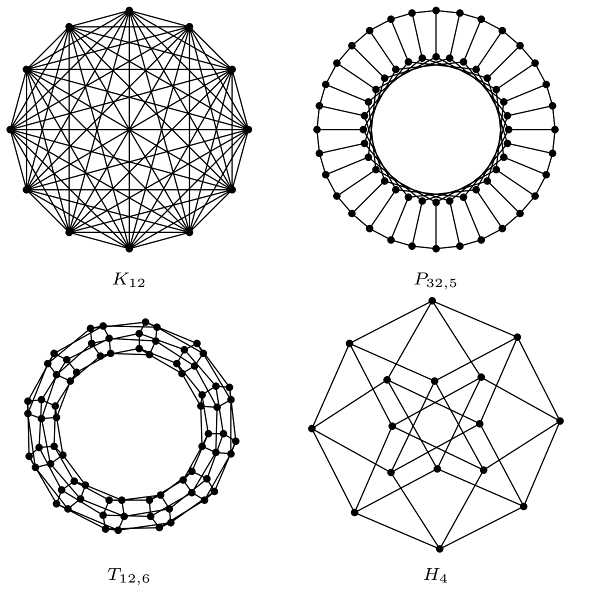


Fig. 2. Examples of complete, Petersen, Torus, and hypercube graphs. Save for the Petersen graph these examples are all smaller than the graphs actually used but are members of the same families of graphs.

*structure*. The general strategy is to use the graph to specify the geography on which a population lives, permitting mating and, in half the experiments, play of the prisoner's dilemma only between neighbors. The goal is to study the impact this has on the prisoner's dilemma agents that evolve.

TABLE I

Graphs used and their index names. Index names are used to index the graphs in figures.

| **Graph** | **Index Name** | **Size** | **Regularity** | **Diameter** |
|---|---|---|---|---|
| $C_{32}$ | **C32** | **32** | **2** | **16** |
| $C_{64}$ | **C32** | **64** | **2** | **32** |
| $P_{16,1}$ | **P16_1** | **32** | **3** | **9** |
| $P_{32,1}$ | **P16_1** | **64** | **3** | **17** |
| $P_{16,5}$ | **P32_5** | **32** | **3** | **6** |
| $P_{32,5}$ | **P32_5** | **64** | **3** | **7** |
| $T_{4,8}$ | **T4_8** | **32** | **4** | **6** |
| $T_{8,8}$ | **T8_8** | **64** | **4** | **8** |
| $T_{4,16}$ | **T4_16** | **64** | **4** | **10** |
| $H_5$ | **H5** | **32** | **5** | **5** |
| $H_6$ | **H6** | **64** | **6** | **6** |
| $K_{32}$ | **Complete 32** | **32** | **31** | **1** |
| $K_{64}$ | **Complete 64** | **64** | **63** | **1** |

### A. List of graphs

This section provides some necessary mathematical definitions and describes the combinatorial graphs used in this study.

*Definition 1:* The *complete graph* on $n$ vertices, denoted $K_n$, has $n$ vertices and all possible edges. An example of a complete graph is shown in Figure 2.

*Definition 2:* The *$n$-cycle graph*, denoted $C_n$, has vertex set $\mathbb{Z}_n$. Edges join pairs of vertices that differ by 1 $(mod\ n)$ so that the vertices form a ring with each vertex having two neighbors.

*Definition 3:* The *$n$-hypercube graph*, denoted $H_n$, has the set of all $n$-character binary strings as its set of vertices. Edges consist of pairs of strings that differ in exactly one position. A 4-hypercube is shown in Figure 2.

*Definition 4:* The *$n \times m$-torus graph*, denoted $T_{n,m}$, has vertex set $\mathbb{Z}_n \times \mathbb{Z}_m$. Edges are pairs of vertices that differ either by 1 $(mod\ n)$ in their first coordinate or by 1 $(mod\ m)$ in their second coordinate but not both. These graphs are $n \times m$ grids that wrap (as tori) at the edges. A $12 \times 6$-torus is shown in Figure 2.

*Definition 5:* The *generalized Petersen graph* with parameters $n$ and $k$, with $k$ relatively prime to $n$, is denoted $P_{n,k}$ and has vertex set $0, 1, \ldots, 2n-1$. The vertices $0, \ldots, n-1$ are connected in a standard $n$-cycle. The vertices $n, \ldots, 2n-1$ are also connected in an $n$-cycle but with the $i$th vertex connected to the $(i+k)$th (mod $n$) vertex. Finally, pairs of vertices $i, n+i$ are connected. The graph $P_{32,5}$ is shown in Figure 2.

The graphs used in the study are given in Table I. They were chosen to give a good selection of degree and diameter parameters with two population sizes, 32 and 64. The importance of degree and diameter in optimizing evolutionary algorithms is given in [17] while the impact of population size is shown to be substantial in [13].

## III. Experimental Design

For each graph used in this study two sets of evolutionary runs are performed. They differ only in the way in which fitness is evaluated. In the *neighbor fitness* runs the fitness of an agent playing iterated prisoner's dilemma is evaluated by having it play 150 rounds against each of its neighbors in the graph. In the *universal fitness* runs fitness is evaluated by having each agent play each other agent for 150 rounds. The fitness recorded is the average score per play. The use of 150 rounds of play is intended to provide consistency with [25] and other studies derived from it [30], [9].

Agents are implemented as 8-state Mealy finite state machines (FSMs). Each machine has an initial state and action and a vector of eight states. Each state specifies an action and next state in case of the opponent's cooperation or defection. A machine of the type used in this study is shown in Figure 3.

Evolution proceeds by generations. In populations of size 32 an elite of 20 machines are simply copied into the next generation. In populations of size 64 this elite has a size of 40. The remaining machines are replaced as follows. For each member of the population not in the elite a graph neighbor is selected with fitness proportional probability. The non-elite population member undergoes two point crossover of its vector of states with the selected neighbor (the neighbor remains unmodified). The initial state and action of the machine follow its first state during crossover. The machine resulting from the crossover is also subjected to a mutation. This mutation

```
--------------
States:8.
Start:C->2
   If C |If D
--------------
 0)C-> 2|D-> 4
 1)C-> 7|D-> 3
 2)C-> 2|C-> 5
 3)C-> 3|D-> 2
 4)C-> 3|D-> 1
 5)C-> 4|D-> 4
 6)D-> 0|C-> 0
 7)D-> 7|C-> 0
--------------
```

Fig. 3. A finite state machine of the type used in this study. The machines initial action is cooperation and its initial state is state 2. Actions and next states are given in the form $action \rightarrow state$ in columns corresponding to an opponent's action of cooperation or defection.

changes the initial state 5% of the time, the initial action 5% of the time, a transition to a next state 40% of the time, and an action associated with a state 50% of the time. These values provide consistency with previous experiments. Once all non-elite population members have been replaced in this fashion the algorithm continues to the next generation.

Each set of evolutionary runs contains 100 independent trials each of which is run for 250 generations. At the end of 250 generations the elite members of the final population are saved for later analysis. The population fitness is saved in each generation as well. A population is deemed *cooperative* if its average fitness is 2.8 or more. This number is derived in [30]. In play with finite state machines with 16 or fewer states, a score of 2.8 or more indicates that defections are occurring only in the transient portion of play. Finite state machines must fall into a repeating loop of states and actions at some point - the transient portion of play is the part before the repeating portion.

Three forms of analysis are performed. In the first, the probability that a population is cooperative is estimated over the 100 independent runs performed for each graph and possible fitness function. The second catalogs the elite strategies from the final generation that arise for each graph and fitness function. This cataloging uses a technique called *fingerprinting*, explained subsequently. The third analysis examines each agent's responses in its finite state transition diagram and tabulates the fraction of each type of possible response exhibited by the evolved agents. The possible types of responses are to respond to cooperation with cooperation (**CC**), to respond to defection with defection (**DD**), to respond to defection with cooperation (**DC**), and to respond to cooperation with defection (**CD**). Only those responses that are accessible from an agent's initial state are tabulated. An alternative to this analysis would have been to examine the actions that actually occurred during fitness evaluation - and this alternative is neither better nor worse, simply different. Tabulating the actions in the agent's finite state diagrams is a broader measure as it contains the agent's full potential spectrum of behavior rather than its realized behavior within its own population.

## A. Fingerprinting

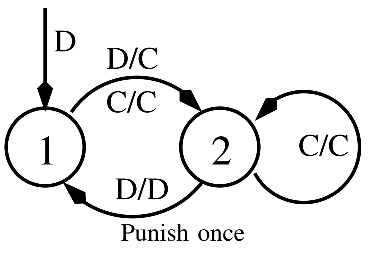
Fingerprinting is a method of extracting a functional signature from a prisoner's dilemma playing agent that is independent of the representation of the agent. Fingerprints are explained in some detail in [11] and an example of how to use Markov chains to compute exact fingerprints is given there. The fingerprint of a prisoner's dilemma strategy $A$ is the expected value per play if $A$ plays forever against a noise-modified version of Tit-for-tat. Tit-for-tat is a prisoner's dilemma strategy that returns the other player's last move as its current move. Tit-for-tat is modified by giving it a probability $x$ of cooperating no matter what it would normally have done and a probability $y$ of defecting no matter what it would normally have done. The fingerprint is thus a function from the possible values of $x$ and $y$ into the set of possible scores. It is often possible to compute the fingerprint as a rational function of the noise parameters $x$ and $y$. The fingerprint of $Tit-for-tat$, for example, is:

$$\frac{3x^2+5xy+y^2}{(x+y)^2}. \tag{1}$$

Even when fingerprints cannot be computed exactly they can be computed to a high degree of accuracy by sampling. In this study, sampled fingerprints are used to place a metric space structure on the space of prisoner's dilemma strategies: the Euclidean distance between points whose coordinates are the scores at 25 sample points of the fingerprint function. This, in turn, permits us to catalog the strategies, treating groups within a radius smaller than the error in the sampling technique as a group of identical strategies, so that we may compare the populations that evolve on each of the graphs.

Fingerprinting is used in two kinds of analysis in this study. A survey for well-known strategies was made, computing their density in the final populations for each graph. The well-known strategies used include *tit-for-tat* (TFT), *always cooperate* (AllC), and *always defect* (AllD), which appear throughout the literature. Psycho is the *opposite* of tit-for-tat, returning the opposite of its opponents previous action. The strategy *tit-for-two-tats* (TF2T) defects only if its opponent has defected on the last two actions. The strategy *two-tits-for-tat*(TTFT) defects on the two rounds after any defection by its opponent. The strategy *punish once* (Pun1) is defined in [6] while the strategy *Fortress 3* (Fort3) is defined in [13]. The finite state transition diagrams for punish-once and fortress 3 are given in Figure 4.

Fingerprints are also used to compile an empirical diversity measure based on the *entropy* of occurrence of strategies. By comparing sampled fingerprints it is possible to compute the fraction of strategies of each type that occurs. The types themselves are not identified except as a fingerprint. These fractions are then treated as probabilities of occurrence for the strategies that do appear and the entropy of this empirical



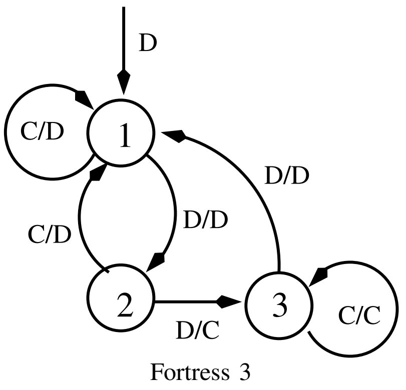


Fig. 4. Finite state transition diagrams for state-minimal representations of punish once (top) and fortress 3 (bottom).

probability distribution $\mathcal{P}$ is found using:

$$entropy(\mathcal{P}) = \sum_{p\in\mathcal{P}} -p \cdot Log_2(p) \tag{2}$$

where the $p \in \mathcal{P}$ are the probabilities of occurrence for the strategies. The set of strategies for which the entropy is computed is the aggregation of the elite part of all final populations in generation 250.

The entropy of a probability distribution is the number of bits required, on average, to describe a sample taken from the distribution. Entropy grows as the number of types present increases and it also increases with the evenness to which population members are distributed among those types. Entropy is thus a good scalar summary of the diversity of a population. Entropy is a *logarithmic* measure of diversity so that a difference of one in the entropies of two populations represents a 2-fold difference in their diversity.

## IV. Results

Figure 5 gives 95% confidence intervals on the probability that a final population will be cooperative for all graphs and both fitness functions. This probability is computed across the 100 independent populations run for each graph and fitness function. The clearest result is that populations in which the fitness is evaluated by having the entire population play one another are substantially more likely to become cooperative than ones in which fitness is evaluated by playing only neighbors. These two fitness functions are the same for the complete graphs, in which all population members are neighbors, but for all other graphs there is a statistically significant difference

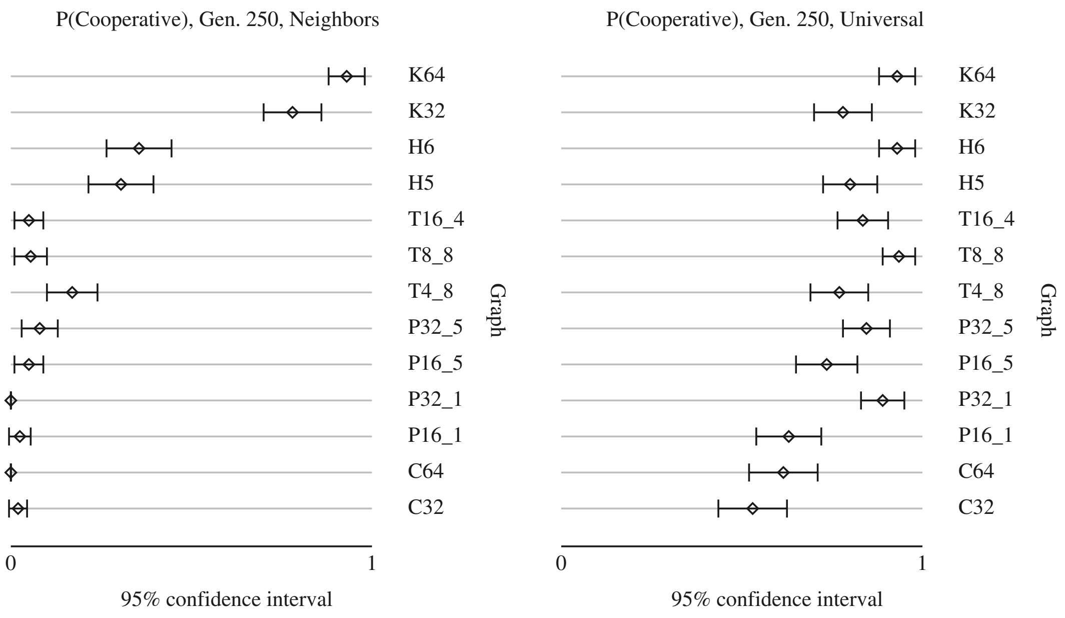


Fig. 5. Shown are 95% confidence intervals on the probability that a final population will be cooperative after 250 generations of evolution. The probabilities shown on the left are for fitness evaluation in which agents play only their neighbors in the graph while the probabilities on the right are for fitness evaluation in which all agents play one another.

in the probability of cooperative behavior when the fitness function changes.

A less dramatic but still significant effect is that the probability of cooperative behavior increases with the number of neighbors. This is true for both fitness functions. Finally, in some cases such as $P_{16,1}$ and $P_{31,1}$, larger populations are significantly more likely to cooperate than smaller ones when the fitness function and number of neighbors are held constant. There are no cases where the reverse is true, but there is not a significant difference in the majority of the cases.

The graph $T_{4,8}$ is anomalously cooperative when the fitness function that plays only neighbors is used and it is anomalously uncooperative for the other fitness function. None of the graph parameters summarized in Table I suggest an easy explanation for this and the anomaly has been noted for future study.

Table II summarizes the density of a selection of well-know strategies. For all graphs, strategies with a tit-for-tat fingerprint are the most common. As discussed in [6], fingerprints measure only the asymptotic behavior of a strategy, not its transient behavior, and so these large numbers of tit-for-tat strategies probably contain variations that end up playing tit-for-tat against most opponents after a small number of plays. If we treat the probability that a population member will have a tit-for-tat fingerprint (not exactly the same thing as playing tit-for-tat) as a Bernoulli variable then row three of Table II contains additional evidence that the graphs exhibit significantly different behaviors as geographies for evolving populations of prisoner's dilemma agents (computations not shown).

TABLE III
POPULATION DIVERSITY AS MEASURED MY EMPIRICAL ENTROPY OF STRATEGIES IN THE AGGREGATION OF ELITE FINAL POPULATIONS FOR ALL RUNS FOR EACH GRAPH. RUNS WHERE FITNESS EVALUATION WAS AGAINST NEIGHBORS ARE DENOTED WITH **Nbr** WITH POPULATION WIDE FITNESS EVALUATION IS DENOTED WITH **Uni**. THESE ARE EQUIVALENT FOR THE COMPLETE GRAPHS K32 AND K64.

| **Diversity of strategies in each graph.** | | | | | | |
|---|---|---|---|---|---|---|
| | **32-member populations** | | | **64-member populations** | | |
| **Degree** | **Graph** | **Nbr** | **Uni** | **Graph** | **Nbr** | **Uni** |
| **2** | **C32** | **8.42** | **6.95** | **C64** | **8.96** | **6.58** |
| **3** | **P16_1** | **8.21** | **7.17** | **P32_1** | **8.36** | **6.29** |
| **3** | **P16_5** | **8.10** | **6.41** | **P32_5** | **8.53** | **5.33** |
| **4** | **T8_4** | **8.01** | **7.39** | **T8_8** | **8.20** | **5.31** |
| **4** | | | | **T16_4** | **7.78** | **5.91** |
| **5/6** | **H5** | **8.06** | **7.05** | **H6** | **7.89** | **5.21** |
| **31/63** | **K32** | **7.61** | | **K64** | **5.69** | |

The entropy-based diversities of the aggregated final populations for each graph and fitness function are shown in Table III. Since the number of strategies is twice as large for the 64-member populations (with 40 member elites) as for the 32-member populations (with 20 member elites) that the diversity of corresponding graph types is slightly higher for all graphs *except* the complete graph. This reversal of diversity in the complete graph is consistent with the non-graph based results in [13] for populations of size 6, 36, and 100.

Diversity is higher for the fitness function that uses play against neighbors only in all graphs where the two fitness functions were distinct. This difference is at its lowest for

TABLE II

DENSITY OF WELL-KNOWN STRATEGIES IN THE AGGREGATION OF THE FINAL ELITE POPULATIONS FOR ALL RUNS ON EACH GRAPH.

| | C32 | C64 | H5 | H6 | P16,1 | P16,5 | P32,1 | P32,5 | T16,4 | T8,4 | T8,8 | K32 | K64 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Neighbor Play** | | | | | | | | | | | | | |
| AllC | 0.021 | 0.018 | 0.011 | 0.012 | 0.016 | 0.019 | 0.014 | 0.017 | 0.017 | 0.013 | 0.015 | 0.003 | 0.002 |
| Alld | 0.034 | 0.025 | 0.013 | 0.009 | 0.032 | 0.028 | 0.031 | 0.019 | 0.022 | 0.026 | 0.026 | 0.006 | 0.001 |
| TFT | 0.199 | 0.205 | 0.243 | 0.300 | 0.214 | 0.235 | 0.265 | 0.240 | 0.311 | 0.243 | 0.275 | 0.270 | 0.477 |
| Psycho | 0.002 | 0.004 | 0.003 | 0.004 | 0.003 | 0.003 | 0.002 | 0.005 | 0.001 | 0.004 | 0.002 | 0.001 | 0 |
| Pun1 | 0.004 | 0.002 | 0.004 | 0.006 | 0.003 | 0.004 | 0.003 | 0.004 | 0.006 | 0.005 | 0.007 | 0.005 | 0.002 |
| TF2T | 0.002 | 0.006 | 0.005 | 0.003 | 0.006 | 0.003 | 0.003 | 0.004 | 0.005 | 0.002 | 0.003 | 0.001 | 0.003 |
| 2TFT | 0.001 | 0.001 | 0.001 | 0 | 0.001 | 0 | 0 | 0.001 | 0.001 | 0.001 | 0 | 0.001 | 0 |
| Fort3 | 0 | 0.001 | 0.001 | 0 | 0.002 | 0.001 | 0 | 0 | 0 | 0 | 0 | 0.010 | 0 |
| **Universal Play** | | | | | | | | | | | | | |
| AllC | 0.004 | 0 | 0.001 | 0.002 | 0.001 | 0.004 | 0 | 0.001 | 0.001 | 0.002 | 0.001 | 0.003 | 0.002 |
| Alld | 0.003 | 0.001 | 0.002 | 0.001 | 0.002 | 0.003 | 0.001 | 0.002 | 0 | 0.005 | 0.001 | 0.006 | 0.001 |
| TFT | 0.336 | 0.403 | 0.329 | 0.532 | 0.312 | 0.399 | 0.434 | 0.526 | 0.461 | 0.296 | 0.518 | 0.270 | 0.477 |
| Psycho | 0 | 0 | 0 | 0 | 0.001 | 0 | 0 | 0 | 0.001 | 0.001 | 0 | 0.001 | 0 |
| Pun1 | 0.001 | 0.001 | 0.003 | 0.002 | 0.004 | 0.004 | 0.002 | 0.002 | 0.002 | 0.001 | 0.003 | 0.005 | 0.002 |
| TF2T | 0.002 | 0.001 | 0.001 | 0.001 | 0.003 | 0.003 | 0.002 | 0.001 | 0.001 | 0 | 0.003 | 0.001 | 0.003 |
| 2TFT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.001 | 0 |
| Fort3 | 0 | 0 | 0.001 | 0 | 0.020 | 0.002 | 0 | 0 | 0.009 | 0 | 0.004 | 0.010 | 0 |

$T_{8,4}$ where there is 53% ($2^{8.01-7.39} = 1.53$) greater diversity. The largest diversity difference is 919%, for $P_{32,5}$.q

The fraction of actions of each of the four possible types are summarized in Table IV. The "unused" portion of the finite state machines was not incorporated into this tabulation; only actions that the agent could take against some opponent were tabulated. For every graph in which the two fitness functions were different (i.e. not the complete graphs $K_{16}$ and $K_{32}$) the fraction of **CC** transitions is higher for the fitness function in which an agent plays all other agents. Since there are 11 such graphs this is significant with $p = \frac{1}{2^{11}} = 0.0005$. This is the strongest pattern in these data. It is worth noting that the complete graphs, the most cooperative, are also the least forgiving; the **DC** action is forgiving because it represents an unanswered defection.

## V. DISCUSSION AND NEXT STEPS

For all three metrics, probability of cooperative behavior in the final population, density of well-known strategies, and entropic diversity of all strategies the choice of graph made a significant difference in the character of prisoner's dilemma agents evolved. This difference made by changing the set of opponents used to evaluate fitness (neighbors as opposed to all agents) is also substantial. The choice of the contact graph for mating in a population of prisoner's dilemma agents can change the probability of cooperative behavior from zero to about 95%. This is startlingly similar to the range caused by changing the representation as demonstrated in [5], [6], [7]. This high level of change appeared only in the experiments using the fitness function that evaluated an agent's fitness by having it play only its neighbors. The impact for the other fitness function was slightly less than half as large. As in earlier studies, what might appear to be a detail of the simulation has a potentially dominant impact on its outcome.

Cooperation was found to be more likely both when agents played larger numbers of partners during fitness evaluation

TABLE IV

THIS TABLE SUMMARIZES THE FRACTION OF TRANSITIONS ACCESSIBLE FROM THE AGENT'S INITIAL STATE, TABULATED ACROSS ALL AGENTS, OF THE FOUR POSSIBLE TYPES. THIS TABULATION IS PERFORMED FOR EACH GRAPH AND FITNESS FUNCTION. THE ACTIONS ARE OF THE FORM *opponent's action agent's response* SO THAT **DC**, FOR EXAMPLE, REPRESENTS RESPONDING TO DEFECTION WITH COOPERATION. THE TWO FITNESS FUNCTIONS ARE IDENTICAL ON THE COMPLETE GRAPHS.

| | *Response Types* | | | |
|---|---|---|---|---|
| Graph | **CC** | **DD** | **DC** | **CD** |
| Neighbor Fitness | | | | |
| C32 | 0.3231 | 0.292 | 0.177 | 0.208 |
| C64 | 0.3335 | 0.288 | 0.166 | 0.212 |
| H5 | 0.3799 | 0.279 | 0.120 | 0.221 |
| H6 | 0.4053 | 0.271 | 0.095 | 0.229 |
| P16_1 | 0.3377 | 0.298 | 0.162 | 0.202 |
| P16_5 | 0.3545 | 0.281 | 0.146 | 0.219 |
| P32_1 | 0.3489 | 0.290 | 0.151 | 0.210 |
| P32_5 | 0.3761 | 0.281 | 0.124 | 0.219 |
| T16_4 | 0.3695 | 0.282 | 0.131 | 0.218 |
| T8_4 | 0.3690 | 0.284 | 0.131 | 0.216 |
| T8_8 | 0.3794 | 0.282 | 0.121 | 0.218 |
| Complete Graphs | | | | |
| K32 | 0.3978 | 0.278 | 0.102 | 0.222 |
| K64 | 0.4085 | 0.265 | 0.092 | 0.235 |
| Universal Fitness | | | | |
| C32 | 0.3904 | 0.290 | 0.110 | 0.210 |
| C64 | 0.4037 | 0.293 | 0.096 | 0.207 |
| H5 | 0.4079 | 0.271 | 0.092 | 0.229 |
| H6 | 0.4038 | 0.269 | 0.096 | 0.231 |
| P16_1 | 0.3854 | 0.288 | 0.115 | 0.212 |
| P16_5 | 0.4012 | 0.280 | 0.099 | 0.220 |
| P32_1 | 0.4178 | 0.277 | 0.082 | 0.223 |
| P32_5 | 0.4100 | 0.264 | 0.090 | 0.236 |
| T16_4 | 0.4237 | 0.279 | 0.076 | 0.221 |
| T8_4 | 0.3990 | 0.273 | 0.101 | 0.227 |
| T8_8 | 0.4193 | 0.267 | 0.081 | 0.233 |

and when they had a larger choice of available breeding partners. This observation holds for both fitness functions. It also echoes the conclusions of [23], even though this used a radically different fitness function in which agents played a new, randomly selected opponent in each round of the iterated prisoner's dilemma. The most uncooperative populations were for populations of size 64 on the two highest diameter graphs: C64 and P32_1 with diameters of 32 and 17 respectively. The very strong effect of the all agents fitness function yielding higher levels of cooperation was echoed in the higher fraction of agent actions consisting of reciprocated cooperation.

Figure 6 shows the three most common strategies, other than tit-for-tat, for K64 using fitness against all other agents. The finite state machines shown are representative; the strategies realized in these agents has many possible finite state forms. One advantage of fingerprinting is the ability to capture behavior rather than specific representation. These agents share a number of properties. Against a nice opponent (one that never defects first) they cooperate indefinitely. This cooperative behavior has small cross section to mutation occupying the initial action and one transition in the first two machines and the initial action and two transitions in the third machine. The first two machines are modest modifications of tit-for-tat. Neither will answer cooperation with defection, ever. Against an opponent that always defects they will permit an unanswered defection every fourth time (first machine) and every third time (second machine). The third machine permits three unanswered defections (more if they are mixed with cooperations in the right proportion) but when too many unanswered defections happen it falls into a state that plays tit-for-tat. All three of these machines can coexist well with copies of themselves and with the most common strategy in the experiments, tit-for-tat. These machines suggest a possible classifying principle for nice machines: the fraction of unanswered defections they will tolerate when playing against an opponent that always defects. For tit for tat and the third machine in Figure 6 the answer is "none". For the first and second machines it is one-quarter and one-third respectively.

The largest effect detected in this study is the impact of changing the fitness function. In all experiments mating was limited by a combinatorial graph. In half the experiments, opponents used to evaluate fitness were limited by the *same* graph while in the remainder all pairs of distinct agents played one another during fitness evaluation. Having different breedings and interaction graphs might yield very different results and is worth investigation.

This study clearly demonstrated a substantial impact that results from modifying the geography limiting play and/or mating for evolving prisoner's dilemma agents. This paper is one of a series that seek to understand how the details of an evolutionary algorithm changes the trajectory of evolution from overall behaviors such as the emergence of cooperation to small details like the identity and character of strategies that actually arise during evolution. A great deal of work remains to be done. First of all, many factors that might have an impact similar to that of representation, duration of evolution, or geography remain to be investigated. A systematic study of the impact of variation operators and elite size that otherwise matches the experimental design of the existing studies in this area might be valuable, for example. Before plunging ahead with such a study, however, a standard for how to report prisoner's dilemma studies is needed. Among those things that clearly should be reported are:

1) Representation or encoding. The data structure used to encode the strategies should be clearly recorded. The depth of memory and space of possible strategies should be reported when this is possible.
2) Fitness function. The fitness function should be clearly specified. In this study the sum over 150 rounds of play with each opponent was used as the fitness function. In [23] an intriguing fitness function motivated as interactions between tourists and souvenir shop owners with a new random partner selected in each play was used.
3) Model of evolution. The algorithm for selection of structures for reproduction and for placement of new structures to update the population must be specified. This includes population size, selection method, and elite fraction if any.
4) The presence of randomness, both in the form of variation operators, noise during play, and randomness in selection of opponents, should be reported.
5) The duration of the experiment, in mating events, generation, and/or fitness evaluations should be reported as well as the synchronous or asynchronous nature of population updating.
6) The exact payoff matrix used for the iterated prisoner's dilemma.

In addition to reporting the design of the experiment so that a standard set of features are reported so and to permit comparison, some metric for *evaluating* experiments are needed. A diffident suggestion of possible metrics are:

1) The fraction of populations that are cooperative together with the definition of cooperative. For the finite state machines and payoff matrix used in this study, population mean fitness in excess of 2.8 is a reasonable definition of "cooperative"[30], but for other representations this threshold may not be a good one.
2) The fraction of paired moves of the types **CC**, **DD**, **CD**, and **DC** that occurred in the final generation or in other epochs of the experiment.
3) The identity of strategies that arise. Fingerprinting [31], [11] is a method of doing this but others can be developed.
4) The level of diversity, e.g. agent count entropies of the sort used in this study.

For any such standards to be broadly used they must be community based and developing a standards group for the reporting the design and outcome of game-theoretic experiments done with evolutionary computation is probably a worthwhile undertaking.

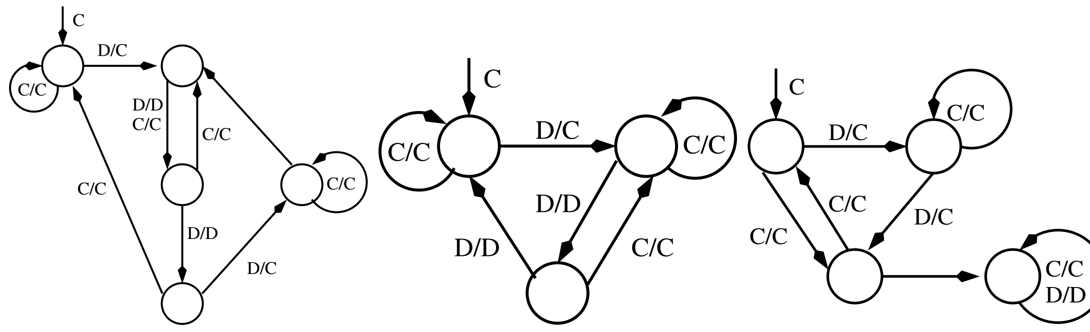


Fig. 6. The second through fourth (left to right) most common strategies in the runs for K64 with fitness evaluated against all other agents. The most common strategy was tit-for-tat.

## VI. Acknowledgments

The author would like to thank the University of Guelph for its support of this research and David Fogel for many helpful discussions of the iterated prisoners dilemma and evolution.

## References

[1] D. Ashlock, E. Blankenship, and J. D. Gandrud. A note on general adaptation in populations of painting robots. In *Proceedings of the 2003 Congress on Evolutionary Computation*, pages 46–53, Piscataway, NJ, 2003. IEEE Press.

[2] D. Ashlock and K. M .Bryden. Non-local adaptation in bidding agents. In *Smart Engineering System Design: Neural Networks, Evolutionary Programming, and Artificial Life*, volume 15, pages 193–199. ASME Press, 2005.

[3] D. Ashlock, K. M. Bryden, and S. Corns. Graph based evolutionary algorithms enhance the location of steiner systems. In *Proceedings of the 2005 Congress on Evolutionary Computation*, volume 2, pages 1861–1866. IEEE Press, 2005.

[4] D. Ashlock, K. M. Bryden, S. Corns, and J. Schonfeld. An updated taxonomy of evolutionary computation problems using graph-based evolutionary algorithms. In *Proceedings of the 2006 Congress On Evolutionary Computation*, pages 403–410, Piscataway, NJ, 2006. IEEE Press.

[5] D. Ashlock and E. Y. Kim. The impact of cellular representation on finite state agents for prisoner's dilemma. In *Proceedings of the 2005 Genetic and Evolutionary Computation Conference*, pages 59–66, New York, 2005. ACM Press.

[6] D. Ashlock and E.Y. Kim. Techniques for analysis of evolved prisoner's dilemma strategies with fingerprints. In *Proceedings of the 2005 Congress on Evolutionary Computation*, volume 3, pages 2613–2620, Piscataway, NJ, 2005. IEEE Press.

[7] 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*, 36(4):464–475, 2006.

[8] D. Ashlock and A. Sherk. Non-local adaptation of artificial predators and prey. In *Proceedings of the 2005 Congress on Evolutionary Computation*, volume 1, pages 41–48, Piscataway, NJ, 2005. IEEE Press.

[9] Dan Ashlock, Mark D. Smucker, E. Ann Stanley, and Leigh Tesfatsion. Preferential partner selection in an evolutionary study of prisoner's dilemma. *Biosystems*, 37:99–125, 1996.

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

[11] 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, pages 381–387, Piscataway NJ, 2004. IEEE Press.

[12] Daniel Ashlock and John Mayfield. Acquisition of general adaptive features by evolution. In *Proceedings of the 7th Annual Conference on Evolutionary Programming*, pages 75–84, New York, 1998. Springer.

[13] W. Ashlock and D. Ashlock. Changes in prisoner's dilemma strategies over evolutionary time with different population sizes. In *Proceedings of the 2006 Congress On Evolutionary Computation*, pages 1001–1008, Piscataway, NJ, 2006. IEEE press.

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

[15] Robert Axelrod and William D. Hamilton. The evolution of cooperation. *Science*, 211:1390–1396, 1981.

[16] W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone. *Genetic Programming : An Introduction*. Morgan Kaufmann, San Francisco, 1998.

[17] K. M. Bryden, D. A. Ashlock, S. Corns, and S. J. Willson. Graph based evolutionary algorithms. In Press, IEEE Transactions on Evolutionary Computation, 2006.

[18] David Doty. Non-loca evolutionary adaptation in grid plants. In *Proceedings of the 2004 Congress on Evolutionary Computation*, volume 2, pages 1602–1609, Piscataway, NJ, 2004. IEEE Press.

[19] D. B. Fogel and P. G. Harrald. Evolving continuous behaviors in the iterated prisoner's dilemma. *Biosystems*, 37:135–145, 1996.

[20] D.B. Fogel. Evolving behaviors in the iterated prisoners dilemma. *Evolutionary Computation*, 1(1), 1993.

[21] Michael Hemesath. Cooperate or defect? Russian and American students in a prisoner's dilemma. *Comparative Economics Studies*, 176:83–93, 1994.

[22] John M. Houston, Judy Kinnie, Bernice Lupo, Christeine Terry, and Sandy S. Ho. Competitiveness and conflict behavior in simulation of a social dilemma. *Psychological Reports*, 86:1219–1225, 2000.

[23] Hisao Ishibuchi, Naoki Namikawa, and Ken Ohara. Effects of spatial structures of iterated prisoner's dilemma game strategies in single-dimensional and two-dimensional grids. In *Proceedings of the 2006 Congress on Evolutionary Computation*, volume 1, pages 3721–3728. IEEE Press, 2006.

[24] N. Masuda and K. Aihara. Spatial prisoner's dilemma optimally played in small-world networks. *Physics LEtters A*, 313, 2003.

[25] John H. Miller. The coevolution of automata in the repeated prisoner's dilemma. *Journal of Economic Behavior and Organization*, 29(1):87–112, January 1996.

[26] H. Mühlenbein. Darwin's continent cycle theory and its simulation by the prisoner's dilemma. *Complex Systems*, 5:459–478, 1991.

[27] Sidney I. Resnick. *Adventures in Stochastic Processes*. Birkhauser, Boston, 1992.

[28] Duncan Roy. Learning and the theory of games. *Journal of Theoretical Biology*, 204:409–414, 2000.

[29] Karl Sigmund and Martin A. Nowak. Evolutionary game theory. *Current Biology*, 9(14):R503–505, 1999.

[30] E. Ann Stanley, Dan Ashlock, and Leigh Tesfatsion. Iterated prisoner's dilemma with choice and refusal. In Christopher Langton, editor, *Artificial Life III*, volume 17 of *Santa Fe Institute Studies in the Sciences of Complexity*, pages 131–176, Reading, 1994. Addison-Wesley.

[31] Warren Kurt vonRoeschlaub. Automated analysis of evolved strategies in iterated prisoner's dilemma. Master Thesis, 1994.

[32] Douglas B. West. *Introduction to Graph Theory*. Prentice Hall, Upper Saddle River, NJ 07458, 1996.