Nonlocal Adaptation in Iterated Prisoner's Dilemma

Patrick Jordan
Dept. of Mathematics
Iowa State University
Ames, IA 50011
Email: pjordan@darcon.net

***Abstract*— This paper investigates nonlocal adaptation in iterated prisoner's dilemma, i.e., do populations of iterated prisoner's dilemma strategies adapt only to their local competitive environment or do the populations continue to develop nonlocal competitive strategies? Research for this paper suggests a statistically significant increase in a population's comparative ability relative to other populations given an increase in generations. Populations are sampled at 100, 1000, and 10000 generations for each simulation. This paper tests the hypothesis that there is no difference in median scores between highly evolved and young populations. The hypothesis is rejected with an α value of 0.05.**

I. INTRODUCTION

This paper documents the progress of evolution on populations to the degree that there is improvement of these populations when compared to a nonlocal environment. To show this 30 random populations where evolved for 10000 generations saving at 100, 1000, and 10000. A generation was compared to another through multiple round-robin tournaments used to eliminated independence due to lineage. The result of these comparisons gave paired observations that could be used to reject the null hypothesis. A probability less than α was said to reject the null hypothesis.

II. EXPERIMENTAL DESIGN

A. Iterated Prisoner's Dilemma

The set of moves $\mathcal{A}$ in prisoner's dilemma is $\{C, D\}$, that is, cooperate and defect. Table I shows that payoff a strategy receives when executing a move against another strategy.

TABLE I
PRISONER'S DILEMMA PAYOFF MATRIX.
The payoff players receive for strategies given in the form of (Player 1, Player 2)

		Player 2	
	k	C	D
Player 1	C	(3,3)	(0,5)
	D	(5,0)	(1,1)

Let $\{x_i, x_j\}$ be a pair of strategies for iterated prisoner's dilemma. Let $x_i(n)$ be the move at iteration n of strategy x_i. Define the K be the payoff of strategy x_i when playing x_j

$$K(x_i, x_j) = \frac{1}{200}\sum_{n=1}^{200} k(x_i(n), x_j(n))$$

where k is the prisoner's dilemma matrix function defined in Table I.

B. Representation

The strategy representation is given by a finite state Mealy machine of size 20. A state S_i is defined as a function that takes an action and returns an action and a state. A strategy is created by generating 20 random states. Each state has its transitions and actions assigned uniformly at random. The initial action and state are chosen randomly from the created states and action set, respectively.

C. Evolutionary Algorithm

The evolutionary algorithm has a parametric setting similar to used in Ashlock et al (see [1]). A population, $\mathcal{P}$, consists of 60 iterated prisoner's dilemma strategies. The initial population for each run is made by repeatedly creating strategies in the manor defined in Section II-B. The fitness of a strategy $p \in \mathcal{P}$, denoted $f(p)$, is given by the equation

$$f(p) = \frac{1}{|\mathcal{P}| - 1}\sum_{\hat{p}\,\in\, P - p} K(p, \hat{p})$$

This is equivalent to averaging a strategies score in a round-robin tournament which contains all members of the population at a given generation.

To be consistent, breeding was elitist with an elite size of $2/3$ of the population. Parents selected to replace the remaining $1/3$ are chosen by roulette selection over the entire population. In order to generate offspring, the parents were paired off and cloned. Crossover and mutation are applied to the clones before placing them in the next generation. Crossover is performed by exchanging a contiguous portion of the list of states with the paired clones. The selection of the two crossover points is done uniformly at random. There are five different types of mutation operators defined with their associated probabilities in Table II. The operators are as follows:

i) A state is selected uniformly at random from the list of states. A transition from that state is selected uniformly at random. That transition is reassigned to a state selected uniformly at random.
ii) A state is selected uniformly at random from the list of states. An action from that state is selected uniformly at random. That action is reassigned to an action selected uniformly at random.

TABLE II
MUTATION OPERATORS

Type	Description	Probability
i	State Transition	0.25
ii	State Action	0.50
iii	Initial Action	0.05
iv	Initial State	0.05
v	Copy State	0.15

iii) The initial action is reassigned to an action selected uniformly at random.
iv) The initial state is reassigned to an state selected uniformly at random from the list of states.
v) A state is selected uniformly at random, that state is copied into another state selected uniformly at random.

A run is defined as repeatedly evolving a random population through $10,000$ generations. The populations are saved after the 100^{th}, 1000^{th}, and $10,000^{\text{th}}$ generation. In this experiment, 30 runs where completed, generating a total of 90 populations to compare.

D. Testing Nonlocal Adaptation

Let $\hat{\mathcal{P}}(\mathcal{P}_i, \mathcal{P}_j) \subset \mathcal{P}_i \cup \mathcal{P}_j$ be a population generated by randomly selecting half of population $\mathcal{P}_i$ and concatenating with the random selection of half of population $\mathcal{P}_j$. Let ρ be defined as follows

$$\rho(\mathcal{P}_i, \mathcal{P}_j) = \sum_{p \in \hat{\mathcal{P}} \cap \mathcal{P}_i, \ \hat{p} \in \hat{\mathcal{P}} - p} K(p, \hat{p})$$

In other words, ρ is the result of summing all the scores of the strategies in $\hat{\mathcal{P}}$ in a round-robin tournament who where originally from $\mathcal{P}_i$.

For generations g_1 and g_2 and run $r \in \mathcal{R}$ let

$$\begin{aligned} s_r(g_1) &= \sum_{\hat{r} \neq r} \rho(\mathcal{P}_{g_1,r} \ , \ \mathcal{P}_{g_2,\hat{r}}), \\ s_r(g_2) &= \sum_{\hat{r} \neq r} \rho(\mathcal{P}_{g_2,r} \ , \ \mathcal{P}_{g_1,\hat{r}}). \end{aligned}$$

Here $s_r(g_1)$ represents the sum of the scores of the population given by generation g_1 and run r when played against all other populations in generation g_2 exception the population that shares its lineage. Similarly with $s_r(g_2)$.

The pair $(s_r(g_1), s_r(g_2))$ represent a paired observation, i.e., a before and after representation of the populations given by lineage of run r when compared to other populations. For each pair of generations g_1 and g_2 there are 30 paired observations that can be used for statistical tests.

III. EXPERIMENTAL RESULTS

Thirty simulations where performed in the manner described in Section II-C. The 90 populations generated were tested according to the specifications given in Section II-D. In each case, the paired data was tested for statistical significance using a Wilcoxon sign rank test. Table III show a statistically significant difference in the play of generation 10000 vs 100 and 1000.

TABLE III
SIGN RANK PROBABILITY
Probability of observing a result equally or more extreme than the one using the experimental data if the null hypothesis is true.

Generation	100	$1,000$	$10,000$
100	0.6435	0.2989	0.0196
1000		0.1589	0.0012
10000			0.3086

IV. CONCLUSION

The results from the experiment show a statistically significant increase in comparative ability as calculated in a paired statistical test. This suggests that co-evolutionary systems may acquire a general gameplay ability through evolution that was not expressly tested in local play.

ACKNOWLEDGMENT

The author would like to thank Dr. Dan Ashlock and David Doty for their helpful insights and discussions regarding nonlocal adaptation.

REFERENCES

[1] Dan Ashlock, Nicole Leahy, and Bruce Wagner. A Representational Sensitivity Study of Game Theoretic Simulations. In *Evolution 2000: Joint meeting of the Society for the Study of Evolution, Society of Systematic Biologists, Association of Tropical Biologists, and American Society of Naturalists*, Indiana University, Bloomington, Indiana, June 2000.

[2] Dan Ashlock and John E. Mayfield. Acquisition of General Adaptive Features by Evolution. In V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, editors, *Evolutionary Programming VII*, pages 75–84, Berlin, 1998. Springer.