

# Investigating Chaos as a Source of Innovation in a Simple Model of a Co-Evolutionary System

**Grant W. Braught**

Nova Southeastern University  
braught@dickinson.edu

**Abstract:** A simple model of a co-evolutionary system is introduced in which the effect of co-evolution is modeled as a gradually changing fitness function. Two such gradually changing fitness functions are created and used to evolve populations cellular automata to perform two simple computations. The role of cellular automata with behavior classified as chaotic in the ability of the evolving population to track the moving fitness function is investigated. No evidence was found to support the hypothesis that the presence of chaotic cellular automata is significant in this way. However, an interesting averaging property of the population of cellular automata in the co-evolutionary model was discovered.

**Keywords:** Evolution; Co-evolution; Genetic Algorithm; Cellular Automata;

## I. Introduction:

In a natural co-evolutionary system two or more species are evolving in an environment in such a way that the survival of each species is dependent on the others. For example, in a predator-prey pair, if the prey evolves spots to hide more effectively the predator may evolve better sensory capabilities to continue to capture the prey. In some circumstances this type of relationship can escalate into a co-evolutionary “arms race.” (Dawkins, 1999) dedicates a full chapter of his book “The Extended Phenotype” to the discussion of such arms races in nature. (Cliff & Miller, 1995) point out that (van Vallen, 1973), (Ridley, 1993) as well as (Dawkins, 1986) have suggested that these co-evolutionary arms races are the source of many “innovations and adaptations” in natural evolutionary history.

Significant attention has been given to the idea of using co-evolution and arms races to improve innovation and adaptation in artificial evolutionary systems. (Hillis, 1991) showed that the performance of an evolving population of sorting networks is less likely to stall at local maxima when a co-evolving population of parasites is present. (Koza, 1991) found that co-evolution in a genetic programming framework enabled two populations of co-evolving programs to discover minimax strategies for playing a simple game. (Miller & Cliff, 1994) and (Cliff & Miller, 1996) have shown that the effectiveness of evolved pursuit and evasion strategies is enhanced by co-evolution in predator-prey simulations. Extensive work by (Nolfi & Floreano, 1998), (Floreano & Nolfi, 1997a, 1997b) and (Floreano, Nolfi & Mondada, 1998) has generated similar results using physical robots.

A shared property of all these co-evolutionary systems is that the fitness landscape on which individuals are judged is constantly changing. Not only is the fitness landscape changing but how it changes is determined by the other co-evolving species. In artificial co-evolutionary

systems this effect manifests itself by stasis in many of the common measures of progress because they are measured with respect to the changing fitness landscape. This effect has often been called “Red Queen Effect” after the character in Lewis Carroll’s “Through the Looking Glass” who runs continuously on a landscape that is also moving, thus making it impossible to perceive progress. To address this issue (Cliff & Miller, 1995) have begun to develop new techniques for performing selection and measuring progress in co-evolutionary systems. (Rossin & Belew, 1997) demonstrated the effectiveness of using direct competition between individuals in the selection of co-evolving strategies for playing Nim and 3-D Tic-Tac-Toe. (Gomez & Miikkulainen, 1997) have shown that making incremental changes in the selection criterion, allows co-evolving predators and prey to develop more sophisticated strategies in a pursuit and evasion game.

Taken in total, the work cited above illustrates that co-evolution appears to produce significant innovations and adaptations that may not be possible using a more traditional evolutionary system. The work presented here seeks to better understand co-evolution by examining the presence and effect of chaotic behavior in a simplified model of a co-evolutionary system. More specifically, populations of rule tables for an elementary one dimensional cellular automata (CA) are evolved to perform two distinct computations. To emulate the red queen effect, the fitness functions used in the evolutions are changed slightly between generations. The amount of chaotic behavior is then measured by counting the number of rule tables in the population that are classified as chaotic or locally chaotic using the classification reported by (Li & Packard, 1990). The work of Li and Packard also shows that mutation of a chaotic or locally chaotic rule table is more likely to produce a rule table with qualitatively different behavior than mutation of a rule table from the other classes. This suggests that chaotic or locally chaotic rule tables in a population may provide a source of easily modifiable behavior that can change quickly via mutation or crossover. Thus, the ability of an evolving population of CA to track a moving fitness function may be enhanced by the presence of chaotic and locally chaotic rule tables.

The experiments conducted were designed to investigate the hypothesis that populations of elementary CA evolving in the presence of a moving fitness function will naturally contain higher than expected concentrations of chaotic or locally chaotic rule tables. Preliminary results seemed to suggest that a maximal number of chaotic rules appear in the population at a critical rate of change in the fitness function. However, further analysis of the data, revealed an alternative explanation for the rise in the number of chaotic rules. This alternative explanation appears to be correct and lead to the discovery of an interesting averaging phenomenon in the CA population. Specifically, if a population of CA are evolved with the moving fitness function  $F(g)$ , where  $g$  is the generation, then the distribution of rule tables among Li & Packard’s classes is predicted at generation  $n$  by averaging the distributions of rule tables in separate populations evolved using the static fitness functions  $F(1), F(2), \dots F(n)$ .

The remainder of this paper has the following organization. Section II presents some general background information on CA, classification of CA behavior and related work on evolving CA. Section III describes the computational tasks and evolutionary environments that were used in the experiments. Section IV presents the results of the experiments. This section discusses why the data fails to support the hypothesis, provides an alternative explanation for the results and documents the averaging phenomenon that was discovered. Section V highlights a collection of

reasons why the performed experiments may have failed to support the hypothesis and suggests additional experiments to address these reasons. General observations regarding the observed averaging phenomenon are expressed and directions for clarifying this phenomenon are suggested.

## **II. Background:**

As stated above the work presented here is based on investigation of the hypothesis that chaotic rule tables in an evolving population of CA may serve as a source of readily mutable behavior, making the population better suited for a co-evolutionary environment. Before discussing this work a brief introduction to CA and the classification of CA behavior is presented. Some related work on the relationship between evolved CA and the complexity of their behavior is also reviewed.

### *A. Cellular Automaton*

A cellular automaton (CA) is a collection of identical processing cells each of which assumes one of a specified number of states based on the states of a number of nearby cells. In a one dimensional CA the processing cells are arranged in a straight line and the next state of each cell is determined by some number of cells to its left and right. Therefore, a one dimensional CA can be characterized by the number of possible states,  $S$ , that a cell can assume and the radius,  $R$ , of a cell's neighborhood. The radius of a cell's neighborhood indicates the number of cells to the left and right that influence its next state. For example, simplest interesting CA is the elementary CA and has  $R=1$  and  $S=2$ . Thus in an elementary CA a cell can take on one of 2 possible states  $\{0,1\}$  and the next state of the cell is determined by its current state, and the current states of the cell to its immediate left and right. Typically, a lookup table, called the rule table, is used to specify the next state of a cell for each of the  $S^{(2R+1)}$  possible current neighborhood states. The qualitative behavior of a CA can then be observed by specifying an initial state for all of the cells and simultaneously iterating each cell using the CA's rule table.

### *B. Classification of Cellular Automaton Behavior*

(Wolfram, 1984) was the first to attempt to classify CA by their behavior. He identified four qualitatively different classes of CA based on the pattern of cell states observed as the CA is iterated from many random initial conditions. Class I contains CA that converge leaving all cells in a single state. Class II contains CA that fall into a fixed pattern of multiple states or a periodic cycle of states. Class III contains CA that produce a chaotic pattern of states. Class IV contains CA that produce complex localized patterns of states. Wolfram as well as (Langton, 1990) suggest that the properties of Class IV CA make them capable of universal computation. Unfortunately, determining the class of a particular CA requires iterating it on a variety of inputs and observing its behavior.

(Langton, 1986) attempted to quantify the behavior of a CA based on the composition of its rule table and thus allow the behavior of a CA to be classified without iteration. He defines the  $\lambda$  parameter, to be the percentage of next states in a rule table that are not a designated quiescent state. For example, in an elementary CA if state 1 is the quiescent state, then  $\lambda$  is the percentage of 0 states in the rule table. A nice property of the  $\lambda$  parameter is that it allows a CA to be

constructed with a specified  $\lambda$  value and thus a known classification. He has shown that for  $S \geq 4$  and  $R \geq 3$  gradually increasing the  $\lambda$  value of a CA by altering its rule table tends to modify the behavior of the CA such that it moves through Wolfram's classes in the order I-II-IV-III. However, when  $S < 4$  or  $R < 3$   $\lambda$  only very roughly predicts CA behavior.

(Li & Packard, 1990) provide an exhaustive classification of all  $S^{S^{(2R+1)}} = 256$  elementary CA rule tables based on their behavior when iterated. They identify five qualitatively different classes of behavior and investigate the relationship between the rules in the different classes. Class A contains *null* CA and corresponds to Wolfram's Class I. Classes B & C split Wolfram's class II into *fixed-point* CA that settle to a fixed pattern of states and *periodic* CA that settle to a periodic pattern of states<sup>1</sup>. Class D contains *locally chaotic* CA where chaotic patterns appear but are contained within well-defined boundaries. Class E corresponds to Wolfram's class III and contains *globally chaotic* CA. Note that Class D here does not correspond directly to any of Wolfram's classes nor does Wolfram's Class IV correspond to any of these classes. In fact, Wolfram reports that no elementary CA exhibit class IV behavior while, Li and Packard contend that there are three such rules which they place in class D. Appendix A shows an example CA that is typical of each of Li & Packard's five classes. Each horizontal line in the figures represents an iteration of the CA. Appendix B lists the classification of all rule tables from the elementary CA rule space.

Li and Packard also investigated the how the behavior of an elementary CA is likely to change when a single next state value in its rule table is mutated. They calculated the probability of intra-class and inter-class transitions resulting from a single next state mutation. They found that a single mutation is less likely to change the classification of a null, fixed-point or periodic CA than that of the chaotic rules. Specifically, the probability that a mutated CA will remain in the same class is: 0.417 for null CA, 0.526 for fixed-point CA, 0.511 for periodic CA and 0.337 for both classes of chaotic CA lumped together. Thus, 66.3% of the time the rule table of a chaotic CA is mutated it will exhibit behavior that is not of the same class as compared to 48.9% for periodic CA, 47.4% for fixed-point CA and 58.3% for null CA. Therefore, chaotic CA (Class D and E together) are significantly more likely to exhibit a change in qualitative behavior when their rule tables are mutated. This fact forms the basis for the hypothesis under investigation.

### C. Evolving Cellular Automaton

(Packard, 1988) evolved a population of  $R=3$ ,  $S=2$  CA to classify the density of their initial configuration. When iterated, the fittest CA converge such that if greater than 50% of the cells had initial state 1 (state 0) then all cells would be in state 1 (state 0). Packard's data showed that the rule tables in the population tended to cluster around critical values of Langton's  $\lambda$  parameter that were interpreted to indicate that the evolved CA were likely to be in Wolfram's Class IV. These results coupled with Wolfram and Langton's arguments that CA in Class IV are capable of universal computation, suggested that this evolutionary system tended to select for general computational ability over specific "hard-wired" solutions.

---

<sup>1</sup>Note that CA with a finite number of cells will always fall into a periodic pattern of states. Class C contains those CA whose period grows at most linearly as the number of cells is increased.

However, Packard's results were vigorously attacked by (Mitchell, Hraber & Crutchfield, 1993) who argue not only that Packard misapplied Langton's  $\lambda$  parameter to quantify CA behavior but also that his results were due not to interesting phenomenon but rather to simulation artifacts. Mitchell, Hraber & Crutchfield were wholly unable to reproduce Packard's results. They provide theoretical and empirical arguments that validate their results showing that the evolved CA should and do cluster around  $\lambda=0.5$  as opposed to the critical value found by Packard. They also point out, as mentioned above, that the  $\lambda$  parameter is known to be a poor predictor of CA behavior when  $R<3$  or  $S<4$ . This is however a double edged criticism. If the evolved CA had clustered around the critical  $\lambda$  value this would not mean that they were in fact Class IV CA but conversely the fact that they did not cluster around the critical  $\lambda$  value does not mean that they are not Class IV CA either. Perhaps more convincing is the fact that the evolutionary task selected by Packard requires the CA to converge to a single state and thus it must belong to Wolfram's Class I regardless of its  $\lambda$  value.

Unreported preliminary results by this author have verified the results of Mitchell, Hraber and Crutchfield for  $R=3, S=2$  CA and found similar results for  $R=1, S=2$ , and  $R=1, S=4$  CA. Extending this work to  $R=3, S=4$  CA in which the  $\lambda$  parameter would gain meaning requires  $2*4^{16536}$  bits of storage for each CA rule table in the population and far exceeds the storage capacity of the available computing resources. Additional preliminary experiments attempted to evolve elementary CA that solved Packard's density classification problem not by exhibiting Class I behavior but by exhibiting qualitatively different behavior based on the density of the input. This requires a single rule to produce responses in two different behavior classes based on the density of the input. It appears based on these experiments that the elementary CA space is insufficiently rich to support such a computation. Experiments performed to investigate the role of chaotic rules in a co-evolutionary system are described in detail in the following sections.

### III. Methods

The following describes experiments that investigate the role of chaotic behavior in a simplified co-evolutionary system. These experiments evolved populations of elementary CA to complete gradually changing computational tasks. For generality, two tasks of qualitatively different difficulty were investigated independently. The rule tables in each evolving population were recorded at each generation to give a complete record of the evolutionary history of the populations. The recorded rule tables were then classified using Li and Packard's scheme and the distribution of rule tables in each class (A-E) were computed for each population. The rule table distributions for the populations evolved in the presence of the gradually changing tasks (the co-evolutionary system) were compared to the distributions found when similar ranges of static tasks were used. It was this comparison that lead to the discovery of the interesting averaging phenomenon that is reported.

The easier of the two computational tasks is a *density mapping task*. This task required the CA to map a random initial configuration through 100 iterations to a final configuration with a desired number of cells in the 1 state. The desired number of cells in state 1 in the final configuration was indicated by specifying a *target density*. In these experiments the fitness of a CA was determined by averaging, across 30 random initial configurations, the difference between the target density and the density of 1's in the CA's final configuration. To simulate the

moving fitness function of a co-evolutionary system the target density was changed at a specified rate between each generation. For comparison, populations of CA were also evolved using static fitness functions based on fixed target densities covering the same range as the moving fitness function.

The more difficult computational task was a generalization of the *density classification task* used by Packard. This task required the CA to map all cells to state 1 or state 0 though 100 iterations based on whether or not the density of 1's in the initial configuration exceeded a specified *density threshold*. In these experiments the fitness of a CA was calculated by averaging the percentage of correct cells in the final configuration across 30 initial configurations with uniformly distributed densities. Simulation of the moving fitness function was achieved by changing the density threshold at a specified rate between each generation. As with the previous task, populations of CA were also evolved using a range of static fitness functions based on fixed density thresholds.

In all of the experiments the population consisted of 60 CA rule tables which were evolved for 100 generations. In each generation the 15 fittest CA's plus 5 other CA's chosen at random were selected as parents. Reproduction was accomplished using a monogamous mechanism in which parents were paired and each pair of parents produced 4 offspring. Offspring were produced from the parents using crossover and mutation. The probability of crossover between each bit was 0.25 and the probability of mutating each bit was 0.1. While a mutation rate of 0.1 may seem high, an elementary CA rule table has 8 bits, thus this mutation rate results in less than one mutation per offspring. Each successive generation consisted of all parents of the current generation plus the newly generated offspring.

#### IV. Results

The first experiment performed evolved populations of CA to perform the density mapping task. To simulate the co-evolutionary environment the target density of a population was increased from 0.25 to 0.75 at a specified rate of change. Populations were evolved for 100 distinct rates of change from 0.0000 to 0.0100 in increments of 0.0001. For rates of change greater than 0.005 the change in the target density reversed directions after reaching 0.75. Thus, at the maximum rate of change of 0.01 the target density would reach 0.75 in the 50<sup>th</sup> generation and reverse directions, returning to 0.25 in the 100<sup>th</sup> generation. For each rate of change 200 populations were evolved and the observed distributions of CA rule tables were averaged.

Figure 1 shows a plot of the number of chaotic CA (Class D and Class E combined) in the co-evolutionary population as a function of the rate of change of the target density. Included in the plot are a trace that includes all CA in all generations, a trace that counts just the 15 fittest CA of each generation (the elite CA) and a trace that includes only the fittest CA from each generation. The traces for the elite and best CA have been scaled for comparison purposes. This plot shows a peak in the number of chaotic CA at a rate of change of approximately 0.0038. At first glance this suggests that chaotic CA may be enhancing the evolutionary process at critical rates of change in the target density between 0.0025 and 0.0045.

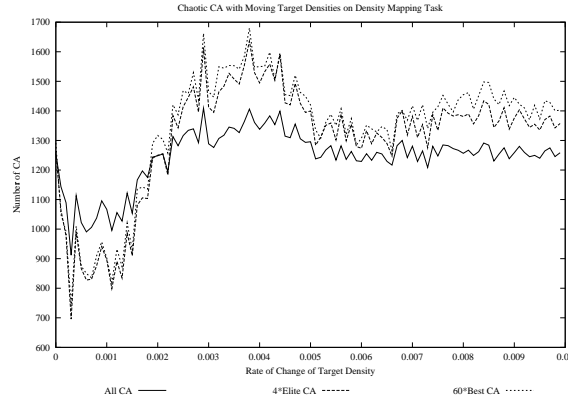


Figure 1: Distribution of chaotic CA (Class D and Class E) in the population summed across all generations and averaged over 200 trials. Shown are traces that include the sums of all CA, the elite CA (15 fittest CA) and the best CA. The traces for the elite and best CA are scaled to appear on the same scale as the trace for all CA.

Before concluding that the peak in chaotic CA observed in figure 1 is caused by an evolutionary benefit from the mutability of chaotic CA, several other possible explanations needed to be examined. If the number of chaotic CA in figure 1 was not in excess of the number of chaotic CA expected by chance, then the evolutionary process would seem to be limiting rather than increasing the number of chaotic CA. However, the percentage of chaotic CA in figure 1 exceeds the number expected by chance. Li and Packard classify 18.0% of all elementary CA as chaotic while the peak values of figure 1 show that approximately 28% of the best CA, 27.8% of the elite CA and 23.3% of all evolving CA are classified as chaotic.

Knowing that the number of observed chaotic CA supports the hypothesis is still insufficient evidence to validate it. It must also be shown that the increased number of chaotic CA is not due to the particular structure of the evolutionary task. To investigate how the structure of the density mapping task affects the distribution of CA, populations of CA were evolved to solve the density mapping task for a range of fixed target densities. Figure 2 shows the distribution of all CA from all generations averaged over 20 populations for each target density in the range 0.2 to 0.8 in increments of 0.02. It can be seen that the number of chaotic CA in the population increases as the target density approaches 0.5 and decreases as it approaches 0.25 and 0.75. Thus, it is possible that the peak observed in figure 1 could be caused by this increase in chaotic CA as the target density approaches 0.5.

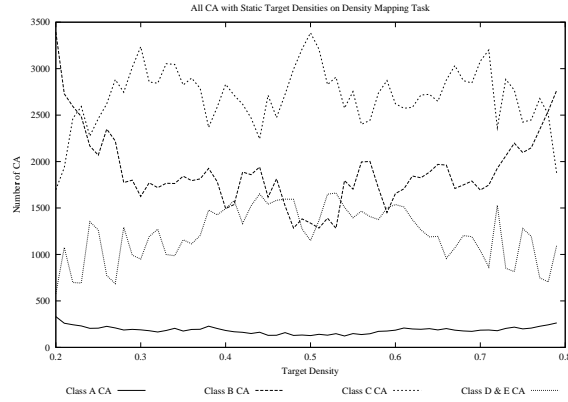


Figure 2: Distribution of all CA in all generations when evolved to solve the density mapping task for static target densities.

Attempts to relate the distributions of chaotic CA from evolutions with static fitness functions in figure 2 to those with moving fitness functions in figure 1 have led to the discovery of an interesting averaging property. Figure 3a shows the distribution of all chaotic CA in all generations as a function of the rate of change in the target density (the same one that appeared in figure 1). Figure 3a also shows a plot of the distribution of chaotic CA from figure 2 averaged from a target density of 0.25 up to a target density of 0.25 plus 100 times the rate of change. For example, at a rate of change of 0.002 the average trace was computed by averaging the distribution of chaotic CA in figure 2 from 0.25 to 0.45. Clearly, the average of the distribution of chaotic CA from the evolutions with static target densities is a good predictor of the distribution of chaotic CA in the co-evolutionary system. Figure 3b shows that this type of averaging is also a good predictor of the distribution of Class B and C CA in the co-evolutionary system. Examination of elite and best rules showed very similar results.

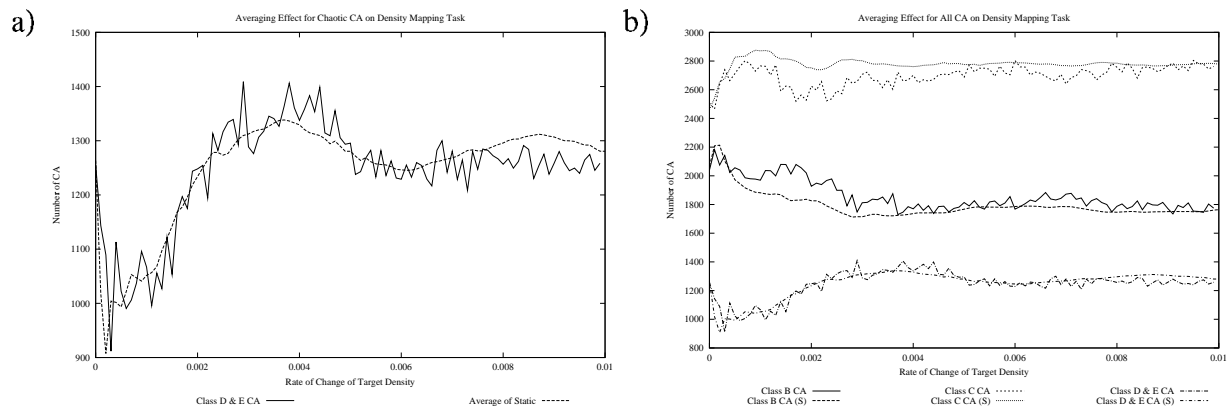


Figure 3: (a) Distribution of chaotic CA in the co-evolutionary system as compared to the average number of chaotic CA in populations evolved with static fitness functions covering the same range of target densities. b) Similar comparison for Class B, C, D & E CA.

A second experiment was performed to look again for an effect related to the presence of chaotic CA and also to test the generality of the averaging effect that was discovered. In this

experiment, CA were evolved to perform the density classification task. The co-evolutionary environment was simulated by increasing the density threshold from 0.25 to 0.75 at a specified rate of change. Again, 100 distinct rates of change, varying from 0.0000 to 0.0100 in increments of 0.0001 were used. For each rate of change, 20 populations were evolved and the observed distributions of CA were averaged. Figure 4a compares the distribution of Class A and Class B CA in all generations of the co-evolutionary system to the average distributions of these CA classes across the corresponding static density thresholds shown in Figure 4b. Class C, D and E CA were omitted from figure 4a because the populations contain only small numbers of these CA and thus evidence for the hypothesized role of chaotic CA fails to be found in this task as well. However, as with the density mapping task the averages derived from the evolutions using the static fitness functions accurately predict the composition of the population in the co-evolutionary system.

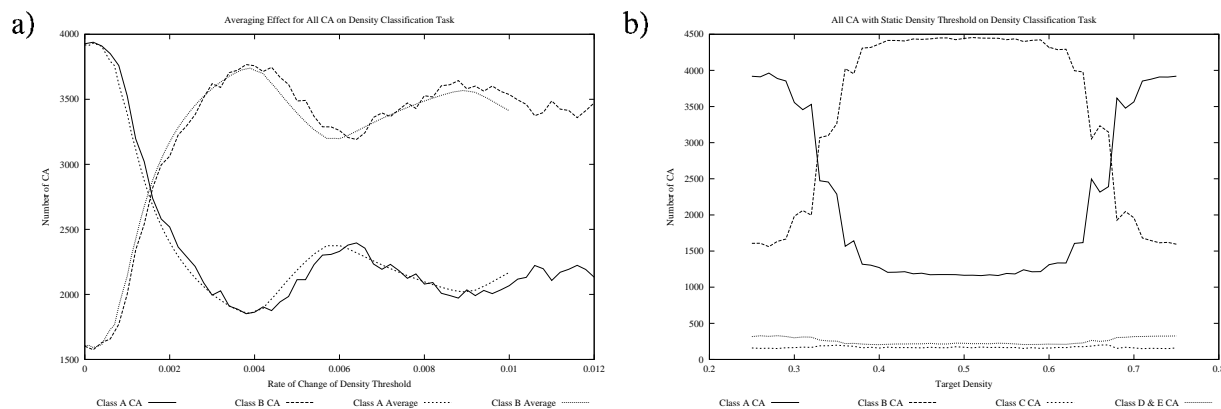


Figure 4: (a) Comparison of the distribution of Class A and B CA evolved to perform the density classification task in a co-evolutionary environment to the average distribution of Class A and B CA evolved using a corresponding range of static density thresholds. (b) The distribution of CA evolved using static density thresholds averaged to produce the average trace in (a).

## V. Discussion:

The experiments performed were unable to show any evidence that populations of CA evolving in a simple model of a co-evolutionary system contain higher than expected proportions of chaotic rule tables. These experiments have shown that an overabundance of chaotic CA do not appear naturally in this simple co-evolutionary system. However, they have not shown conclusively that the presence of chaotic CA cannot play a significant role in the generation of innovation. Experiments could be performed to monitor the ability of a population to track a moving fitness function as additional chaotic CA are inserted into the population or existing chaotic CA are removed from the population. These experiments would serve to demonstrate the effect of excesses or shortages of chaotic CA in the populations. Particularly relevant would be an examination of the number of offspring from chaotic parents that become the elite CA in the following generations.

The averaging phenomenon that was discovered needs to be better understood. At first glance it may seem obvious that this averaging should occur. The populations in the co-evolutionary

system are exposed to the same fitness functions that are experienced by the populations used to produce the average. However, a closer examination reveals that the phenomenon is likely to be more subtle. While it is true that the populations in the co-evolving system are exposed to the same fitness functions as the populations that are averaged, they are exposed to each function for only a single generation. Therefore, for this to be a simple phenomenon it is necessary that the evolutionary tasks selected allow a population to reach its final distribution in approximately one generation. This would mean that during each generation in the co-evolutionary system produces the same population distribution as the population using the corresponding static fitness function. Alternatively, it could be the case that the fitness function exerts no force on the population distributions. Both of these explanations seem unlikely and will not be hard to dismiss by investigation.

It seems more likely that the moving fitness function prevents the population from clustering around a single solution as occurs when a static fitness function is used. With the static fitness function the distribution of the clustering is related to the specifics of the selection, crossover and mutation genetic operators. In the co-evolutionary system the distribution is affected by all of those factors as well as the movement of the fitness function. How these effects interact to produce the observed averaging effect is unclear at this time.

What is clear is that some form of historical information is being maintained in the distribution of the population. However, it is surprising that this information seems to be dependent on the entire history of the population. Reinforcing this view, were failed efforts to model the co-evolutionary population distribution using moving averages of the past 10, 20, 30 and 40 generations.

If further investigation reveals that this averaging effect is truly interesting and not an artifact of the simulation, the classification scheme or the selected tasks, then efforts to see if it generalizes are warranted. A true co-evolutionary system in which two populations of CA are co-evolving and affecting each others fitness functions should be investigated to see if the effect appears. If the effect persists in this environment it should be investigated in a co-evolutionary system that does not involve CA but a more general genotype.

In a much broader sense, continued research in this area may help provide an effective path to high performance computing. CA are often used to model complex system dynamics and have been proposed by (Sterling, 2001) as a potential path to trans-petaflop computing systems. However, CA are traditionally difficult to program and therefore efficient mechanisms for evolving CA to perform computations may play a role in making high performance computing based on CA a reality.

### **Acknowledgements:**

The author would like to thank Dickinson College for the 20 machines that provided the nearly 2500 hours of cpu time required to collect the data presented in this paper.

## References:

(Cliff & Miller, 1995)

Cliff, D. & Miller, G. (1995). Tracking the red Queen: Measures of Adaptive Progress in Co-Evolutionary Simulations. In Third European Conference on Artificial Life, F. Moran, A. Moreno, J. Merelo & P. Chacon (eds.), (Springer-Verlag, Berlin) pp. 200-218.

(Cliff & Miller, 1996)

Cliff, D. & Miller, G. (1996). Co-Evolution of Pursuit and Evasion II: Simulation Methods and Results. In From Animals to Animats IV: Proceedings of the Fourth International Conference on the Simulation of Adaptive Behavior, (MIT Press, Cambridge), pp. 506-515.

(Dawkins, 1986)

Dawkins, R. (1986). The Blind Watchmaker. (Longman, Essex).

(Dawkins, 1999)

Dawkins, R. (1999). The Extended Phenotype: The long reach of the gene. (Oxford University Press, Oxford).

(Floreano & Nolfi, 1997a)

Floreano, D. & Nolfi, S. (1997). God Save the Red Queen! Competition in Co-Evolutionary Robotics. In Genetic Programming 1997: Proceedings of the Second Annual Conference, J. Koza, D. Kalyanmoy, M. Dorigo, D. Fogel, M. Garzon, H. Iba & R. Riolo (eds.) (Morgan Kaufman, San Francisco), pp. 398-406.

(Floreano & Nolfi, 1997b)

Floreano, D. & Nolfi, S. (1997). Adaptive behavior in competing co-evolving species. In Proceedings of the Fourth European Conference on Artificial Life, P. Husband & I. Harvey (eds.), (MIT Press, Cambridge), pp. 378-387.

(Floreano, Nolfi & Mondada, 1998)

Floreano, D., Nolfi, S. & Mondada, F. (1997). Competitive Co-Evolutionary Robotics: From Theory to Practice. In Proceedings of the Fifth International Conference of the Society for Adaptive Behavior, R. Pfeifer, B. Blumberg & H. Kobayashi (eds.) (MIT Press, Cambridge).

(Langton, 1986)

Langton, C. (1986). Studying Artificial Life with Cellular Automata. *Physica D*, Vol. 22, pp. 120-149.

(Langton, 1990)

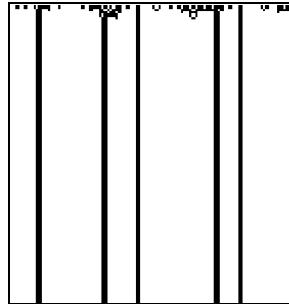
Langton, C. (1990). Computation at the Edge of Chaos. *Physica D*, Vol. 42, pp. 12-37.

- (Li & Packard, 1990)  
Li, W. & Packard, N. (1990). The Structure of the Elementary Cellular Automata Rule Space. *Complex Systems*, Vol. 4, pp. 281-297.
- (Miller & Cliff, 1994)  
Miller, G. & Cliff, D. (1994). Protean behavior in dynamic games: Arguments for the co-evolution of pursuit-evasion tactics. In *From Animals to Animats 3: Proceedings of the Third International Conference on the Simulation of Adaptive Behavior*, D. Cliff, P. Husbands, J. Meyer & S. Wilson (eds.). (MIT Press, Cambridge), pp. 411-420.
- (Mitchell, Hraber & Crutchfield, 1993)  
Mitchell, M., Hraber, P. & Crutchfield, J. (1993). Revisiting the Edge of Chaos: Evolving Cellular Automata to Perform Computations. *Complex Systems*, Vol. 7, pp. 89-130.
- (Nolfi & Floreano, 1998)  
Nolfi, S. & Floreano, D. (1998). Co-evolving predator and prey robots: Do 'arms races' arise in artificial evolution? *Artificial Life*, Vol. 4, No. 4, pp. 1-25.
- (Packard, 1988)  
Packard, N. (1988). Adaptation Toward the Edge of Chaos. In *Dynamic Patterns in Complex Systems*, J. Kelso, A. Mandell & M. Shlesinger (eds.). (World Scientific, Singapore) pp. 293-301.
- (Ridley, 1993)  
Ridley, M. (1993). *The Red Queen: Sex and the evolution of human nature*. (Viking, London).
- (Rossin & Belew, 1997)  
Rossin, C. & Belew, R. (1997). New Method for Competitive Coevolution. *Evolutionary Computation*, Vol. 5, No. 1, pp. 1-29.
- (Sterling, 2001)  
Sterling, T. (2001). How to Build a HyperComputer, *Scientific American*, Vol. 285, No. 1, pp. 38-45.
- (van Valen, 1973)  
van Valen, L. (1973). A new evolutionary law. *Evolutionary Theory*, Vol. 1, pp. 1-30.
- (Wolfram, 1984)  
Wolfram, S. (1984). Universality and Complexity in Cellular Automata. *Physica D*, Vol. 10, pp. 1-35.

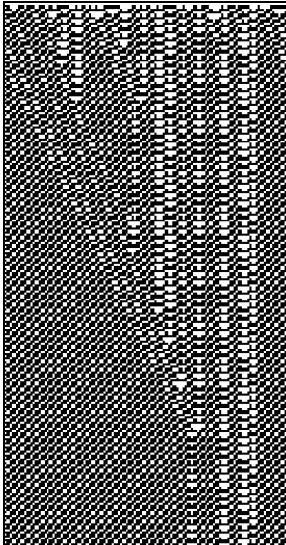
## Appendix A: Examples of Li and Packard's CA Classification



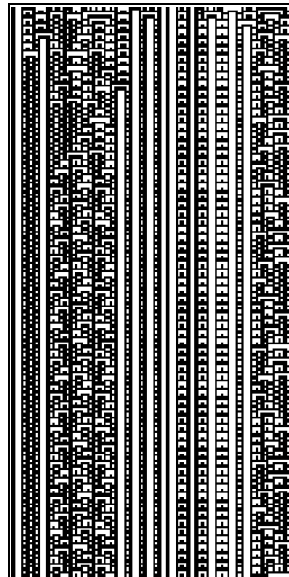
**Class A: Null**  
Initial configurations  
converge leaving all cells  
in the same state.  
Rule #224



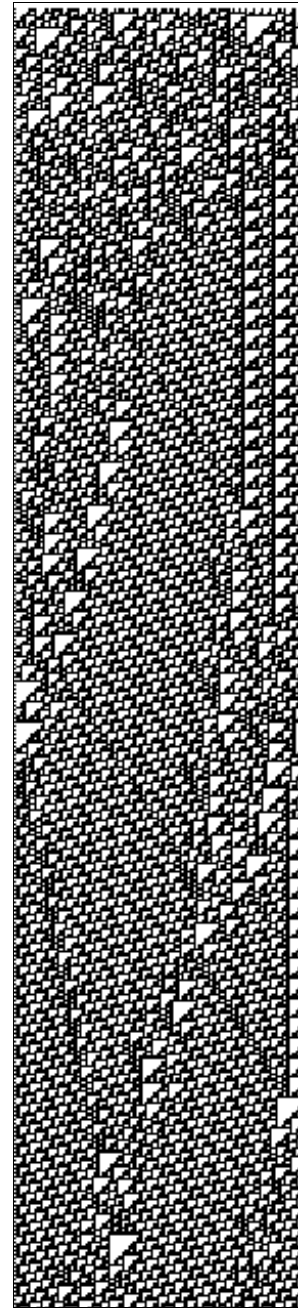
**Class B: Fixed Point**  
Initial configurations  
settle to a fixed pattern of  
states  
Rule #104



**Class C: Periodic**  
Initial configuration settle  
to repeating patterns of  
states.  
Rule #62



**Class D: Locally Chaotic**  
Chaotic patterns of states  
appear in well defined and  
bounded areas.  
Rule #109



**Class E: Globally Chaotic**  
Chaotic patterns of states  
appear both temporally and  
spatially.  
Rule #110

## Appendix B: Li & Packard's Classification of Elementary CA Rule Tables

Rule tables are expressed as decimal numbers corresponding to a binary interpretation of the next state for each neighborhood taken in the order, 111, 110, ... 000. For example the rule table 00010010 is rule table number 18 and maps neighborhoods 100 and 010 to a next state of 1 and all other neighborhoods to a next state of 0. For each class the number of rules in that class is indicated along with the average lambda value of those rules and the percentage of all rules that are contained in the class.

### Class A:

# of Rules: 24  
% of Rules: 0.094  
Ave. Lambda: 0.500  
Rules: 0, 8, 32, 40, 64, 96, 128, 136, 160, 168,  
192, 224, 234, 235, 238, 239, 248, 249, 250, 251,  
252, 253, 254, 255

### Class B:

# of Rules: 97  
% of Rules: 0.379  
Ave. Lambda: 0.500  
Rules: 2, 4, 10, 12, 13, 16, 24, 34, 36, 42,  
44, 46, 48, 56, 57, 58, 66, 68, 69, 72,  
76, 77, 78, 79, 80, 92, 93, 98, 99, 100,  
104, 112, 114, 116, 130, 132, 138, 139, 140, 141,  
144, 152, 162, 163, 164, 170, 171, 172, 174, 175,  
176, 177, 184, 185, 186, 187, 188, 189, 190, 191,  
194, 196, 197, 200, 202, 203, 204, 205, 206, 207,  
208, 209, 216, 217, 218, 219, 220, 221, 222, 223,  
226, 227, 228, 230, 231, 232, 233, 236, 237, 240,  
241, 242, 243, 244, 245, 246, 247

### Class C:

# of Rules: 89  
% of Rules: 0.348  
Ave. Lambda: 0.500  
Rules: 1, 3, 5, 6, 7, 9, 11, 14, 15, 17,  
19, 20, 21, 23, 25, 27, 28, 29, 31, 33,  
35, 37, 38, 39, 41, 43, 47, 49, 50, 51,  
52, 53, 55, 59, 61, 62, 63, 65, 67, 70,  
71, 74, 81, 83, 84, 85, 87, 88, 91, 94,  
95, 97, 103, 107, 108, 111, 113, 115, 117, 118,  
119, 121, 123, 125, 127, 131, 133, 134, 142, 143,  
145, 148, 155, 156, 157, 158, 159, 173, 178, 179,  
198, 199, 201, 211, 212, 213, 214, 215, 229

### Class D:

# of Rules: 10  
% of Rules: 0.039  
Ave. Lambda: 0.500  
Rules: 26, 73, 82, 109, 154, 166, 167, 180, 181, 210

### Class E:

# of Rules: 36  
% of Rules: 0.141  
Ave. Lambda: 0.500  
Rules: 18, 22, 30, 45, 54, 60, 75, 86, 89, 90,  
101, 102, 105, 106, 110, 120, 122, 124, 126, 129,  
135, 137, 146, 147, 149, 150, 151, 153, 161, 165,  
169, 182, 183, 193, 195, 225