FUZZY EVOLUTIONARY CELLULAR AUTOMATA

J. NEAL RICHTER
Department of Computer Science
Utah State University
Logan , Utah
jnr@cc.usu.edu

DAVID PEAK
Department of Physics
Utah State University
Logan, Utah
peakd@cc.usu.edu

ABSTRACT
An application of adaptive genetic algorithms to find optimal cellular automata rules to solve the density classification task is presented. A study of the statistical significance of previous results of the evolutionary cellular automata, EvCA, model is detailed, showing flaws in the fitness function. A brief review of recent work in advanced GAs and fuzzy-adaptive GAs is given. These techniques are then applied to the EvCA model to show improvement in convergence speed and more effective search of the optimization landscape.

INTRODUCTION

We reintroduce an application of genetic algorithms (GAs) to cellular automata., using the GA to evolve rules for performing global computations with simple localized rules. A new more accurate fitness function is introduced to compensate for inaccuracies in the old model. In addition, we extend the model to include fuzzy-logic-controlled GA parameter adaption

EVOLUTIONARY CELLULAR AUTOMATA

The EvCA group at the Santa Fe Institute has authored many papers on using the GA to evolve cellular automata (CA) rules to perform computation (Crutchfield et al., 1997 and Mitchell et al., 1993). The intent of the research was an initial step towards using GAs to enable decentralized computation in distributed multi-processor systems.

Cellular automata are discrete space and time dynamical systems with localized parallel interaction. The universe of a CA is a grid of cells, where each cell can take on one of $\boldsymbol{k}$ states. The evolution of the CA in time is determined by a set of rules. See Wolfram (1994) for a more detailed background.

The simplest form of a CA is a binary state, one-dimensional model where the current state of the space is defined by the binary states of the individual cells. At each time step, the state is formed by applying a set of transition rules to the previous state. The neighborhood $\boldsymbol{r}$ is the number of cells on either side of the current cell that affect the cell's state in the next time step. The number of transition rules in such a system is defined as $\boldsymbol{k}^{2r+1}$.

Figure 1 displays the $\boldsymbol{k}$=2, $\boldsymbol{r}$=1 CA system and table of 8 rules. This rule is refered to as Rule 90, the conversion of the 8 bit rule outputs into a decimal number. Figure 2 displays Rule 90 in action given a single cell on in the initial condition of the lattice.

Cellular automata systems have been used to perform a variety of computational tasks, density classification, synchronization, random number generation, etc (Crutchfield et al., 1997)

Neighborhood	000	001	010	011	100	101	110	111
Output	0	1	0	1	1	0	1	0

Figure 1. Cellular Automata Rule 90

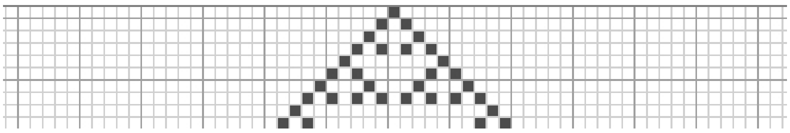


Figure 2. Cellular Automata Rule 90 in action

DENSITY CLASSIFICATION

The density classification problem is defined here. Given a CA rule applied to a randomly initialized starting state, after $\boldsymbol{M}$ time steps the output should be as follows. Let $\boldsymbol{p_0}$ be the percentage of the number of 1s in the initial state.

If $\boldsymbol{p_0} < ½$ then at t= $\boldsymbol{M}$ the system is relaxed to a state of all 0s.
If $\boldsymbol{p_0} > ½$ then at t= $\boldsymbol{M}$ the system is relaxed to a state of all 1s.
Note that $\boldsymbol{p_0} = ½$ is undefined, and is avoided here by using an odd lattice width $\boldsymbol{N}$.

Land and Belew (1995) proved that no binary, $\boldsymbol{r} \geq 1$ CA rule can perfectly classify all possible initial configurations. Fuks' (1997) and Chau et al (1999) demonstrated that using two successive CA rules, perfect density classification is possible. Capcarrere, et al. (1996) showed that with a modification of the desired output state, a system exists that can perfectly solve the density problem.

Gacs, et al, (1978) presented a hand designed $\boldsymbol{k}$=2, $\boldsymbol{r}$=3 CA rule for the density classification task (for the original output specification). It appears that as $\boldsymbol{N} \rightarrow \infty$, the error rate of the GKL rule decreases (Crutchfield et al., 1997). Later in the paper we will show a statistical analysis of the GKL rule's performance score. Figure 3 shows an example run of the GKL density classification rule.

EVCA ALGORITHM

The EvCA system of SFI is reviewed here (Crutchfield et al., 1997). The goal of the system is to use the GA to search the space of possible rules in an attempt to find rules that perform a specific computation. Using a $\boldsymbol{k}$=2, $\boldsymbol{r}$=3, $\boldsymbol{N}$=149 CA system for the density classification task with the original output specification, a fitness function is defined as follows.

1. Randomly choose I=100 Initial Conditions (ICs) uniformly distributed over $\boldsymbol{p_0} \in [0.0,1.0]$
2. Half of the ICs have $\boldsymbol{p_0} < ½$, the other have $\boldsymbol{p_0} > ½$
3. Run each rule $\boldsymbol{M}$ times where $\boldsymbol{M}$ is from a Poisson distribution with mean 320
4. Performance Fitness is fraction of I ICs that produce the correct final result

Varying $\boldsymbol{M}$ ensures that we do not evolve rules that overfit to a given $\boldsymbol{M}$. Chosing random ICs with a coin-flip for each bit would result in the ICs being binominally distributed. As the ICs near $\boldsymbol{p_0}$ = ½ are the most difficult to classify, this method would result in a more difficult fitness task especially in the early generations. Mitchell (1994) notes that the performace fitness measure produced qualitatively similar results to using proportional fitness, where partical credit is given according to the percentage of correct states in the final time step. The proportional fitness metric was used with success in other EvCA papers.

The genome is represented as the lexicographic ordering of the CA rule's bits. A $\boldsymbol{k}$=2, $\boldsymbol{r}$=3 CA rule is 2^7=128 bits long. The bits are arranged in ascending neighborhood order, from 0000000 to 1111111. The EvCA system used 100 individuals with 20% elitism. The GA is run for 100 generations. Mutation rates in the papers varried, we chose 0.03. 50 runs of this GA system were performed.

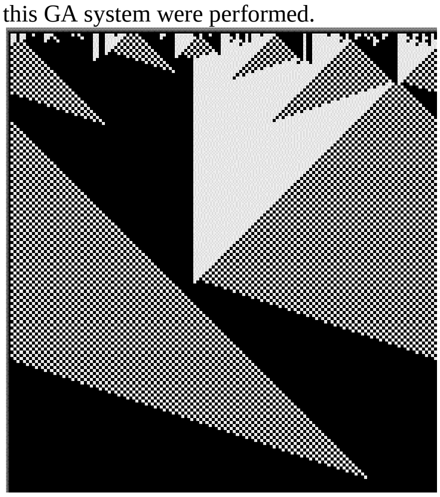


Figure 3. GKL Rule in action

RANDOMIZED FITNESS FUNCTION ANALYSIS

The total space of possible initial conditions for a N=149 CA grid is 2149. The EvCA experiments used 100 random ICs, or $100/2^{149} = 1.4\text{x}10^{-43}$ percent of the total IC space. No justification was given for this choice other than saving computation time.

We believe that the choice of I=100 gives the fitness function a statistically insignificant number of ICs to test each rule. We performed a significance analysis where 8 rules with varying fitness scores were tested 25 times with 100, 1000, 10,000 random ICs and 10 times with 100,000 ICs. Figure 4 summarizes the results with range of variance of each rule. Note that the variance decreases as the number of ICs increases!

	100 ICs	*1000 ICs*	*10000 ICs*	*100000 ICs*
GKL	98.0 **±1.0**	98.3 **±.70**	98.0 **±.03**	98.0 **±.01**
Rule 1	92.6 **±7.7**	93.4 **±.68**	93.2 **±.04**	93.1 **±.02**
Rule 2	90.2 **±5.5**	89.9 **±.58**	89.7 **±.10**	89.5 **±.03**
Rule 3	87.1 **±3.2**	87.1 **±.53**	87.0 **±.07**	87.1 **± .02**

	100 ICs	*1000 ICs*	*10000 ICs*	*100000 ICs*
Rule 4	81.6 ± **5.8**	81.0 ± **.58**	81.3 ± **.05**	81.5 ± **.01**
Rule 5	77.8 ±**14.1**	77.7 ± **1.2**	77.6 ± **.10**	77.6 ± **.05**
Rule 6	63.7 ± **4.1**	64.0 ± **.23**	63.8 ±**.03**	63.8 ± **.01**
Rule 7	61.7 ± **14**	61.6 ± **1.4**	61.3 ± **.12**	61.5 ± **.04**

Figure 4 IC significance test (scores in percentages)

We can see that 100 ICs is not enough to determine fitness to a sufficient accuracy for low-scoring rules. During the initial generations of the GA run, the population is likely to have low average fitness scores. Note it is likely that with 100 ICs, the variance of the fitness scores could exceed the overall variance of the generational average fitness. The generational average fitness is the average score at each generation step across all runs of a single EvCA experiment.

Given this, the pre-crossover ranking of the population is probably far from accurate and we have, in effect, a semi-randomized ranking of individuals! The effectiveness of elitism in the early generations is also questionable, as individuals with a good fitness score may fall out of the top 20% due to a unlucky assignment of 100 ICs. It was also noticed that a large majority of 'winning' rules in the standard EvCA model reach high scores with a lucky assignment of ICs, post-run validations of winning rule scores typically resulted in a several percentage point drop in score.

We believe that an insufficient number of ICs impedes the GA's explorative nature and the speedy ascent of average fitness score in the early generations. Notice that the highly fit rules are able to have accurate scores with low numbers of ICs. The means that the exploitive nature of the GA is reasonably unaffected, in so far as the variance of the generational average score does not exceed the individual fitness variances given here.

ADAPTIVE EVCA

We propose a new EvCA fitness function that will ensure reasonable accuracy, as well as conserve CPU time by short circuiting the fitness evaluation when a rule is shown to be sub-standard. We used a proportional fitness scoring method, although this should make no significant difference for comparison purposes. Each individual is subjected to an initial 250 ICs. At this point the fitness score is evaluated, a target number of total ICs is then determined and the fitness function continues or quits as appropriate.

If the preliminary average score f_p is 75% or below stop.
If $f_p \in [75, 85)$ perform an additional 250 ICs
If $f_p \in [75, 90)$ perform an additional 500 ICs
If $f_p \in [90, 95)$ perform an additional 750 ICs
If $f_p \geq 95$ perform an additional 2500 ICs

While the choice of 250 ICs does not fully address the randomized nature of fitness scoring in the early generations, it does improve accuracy at a reasonable cost. One goal of this fitness function is to help ensure that the elite individuals will be correctly identified early.

In addition, for highly fit individuals we chose to drastically increase the number of ICs. This should help ensure that the very highly fit rules in the elite population are correctly sorted during the latter exploitive generations of the GA. The need of 2500 ICs is not totally supported by the table above, we choose this number in the spirit of erring on the side of accuracy.

SEVCA AND MEVCA

The addition of a more accurate fitness function does let us make one computationally important improvement: it is not necessary to retest the elite population in each generation. Mitchell (1996) outlines a GA for where a percentage of the population is 'overlapping', the remaining percentage is regenerated and evaluated for fitness. While this so called 'steady state' GA should not produce qualitatively different results, we do expect that due to sublte variations in the selection procedures, there will be slightly different quantitative results.

Cantú-Paz (1998) surveys multipopulation GA models. We chose a coarse-grained multi-population model with frequent stepping-stone migration. The parameters are as follows: 5 populations with 20 individuals and 5 eligible for migration. An elite setting of 10 per population was also chosen, and these elite individuals are tested only once.

Notice that this means that there are 50 total elite individuals in each generation. While we do not claim to exactly understand the dynamics of this particular GA, the results indicate that this model performs very well in comparison to the standard EvCA at a great savings of computational time, and no sacrifice of individual fitness accuracy. Graph 1 displays the best fitness curve for the SEvCA (steady state) and MEvCA (multi-population) algorithms.

FUZZY ADAPTIVE EVCA

Here we introduce two new models for EvCA based on advanced GA models and dynamic adaption of the GA parameters. Parameter adaptation in GAs is a much talked about but seldom utilized technique. Bäck (1992) discusses a variety of parameter adaptation issues. Lee and Takagi (1993) laid early ground work for using a fuzzy logic controller to adapt population size, mutation rate, and crossover rates. Shi et al. (1999) introduced a straightforward set of fuzzy rules for adapting the mutation and crossover rates of a GA. See Figure 5 for a listing of the fuzzy rules. See Figure 6 for a diagram of a Fuzzy Adaptive GA.

We incorporated the Shi rule set into the SEvCA system. We defined the fuzzy parameter $BF \in [0.5, 1.0]$ divided into three overlapping triangular membership functions. The fuzzy parameter $UF \in [0, 10]$ is similarly configured. The fuzzy parameter $VF \in [0.0, .40]$ is divided between three overlapping triangular membership functions heavily skewed around the typical observed values (roughly .05 to .30).

Similar to FSEvCA, the FMEvCA system adds the multi-population model from MEvCA. Note that no additional multi-population parameter adaption is used here. No known fuzzy rule system exists for dynamically adjusting the additional multi-population GA parameters. Graph 1 displays the best fitness performance curves of the FSEvCA and FMEvCA systems.

If	***Then***
BF = *low*	MR = *low* & CR = *high*
BF = *medium & UN* is *low*	MR = *low* & CR = *high*

If	*Then*
BF = *medium* and UN is *medium*	MR = *medium* & CR = *medium*
BF = *high* and UN = *low*	MR = *low* & CR = *high*
BF = *high* and UN = *medium*	MR = *medium* & CR = *medium*
UN = *high* and VF = *medium*	MR = *high* & CR = *low*
UN = *high* and VF = *low*	MR = *high* & CR = *low*
UN = *high* and VF = *high*	MR = *low* & CR = *low*

BF = Best Fitness, **UN** = number of generation since last **BF** change

VF = Variance of Fitness, **MR** = Mutation Rate, **CR** = Crossover Rate

Figure 5 Fuzzy GA rule set (Shi et al., 1999)

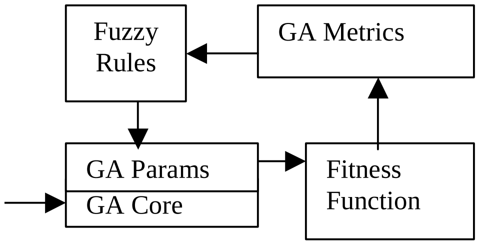


Figure 6 Fuzzy GA Diagram

CONCLUSIONS AND FUTURE WORK

We've added several new techniques to the basic EvCA system with interesting results. Both the improved EvCA and the new FEvCA system showed good results, while the MEvCA and FMEvCA system showed poor performance.

A closer examination of the effectiveness of the adaptive fitness function in relation to the variance of generational elite individual average scores may show that the fitness function could be adjusted to perform additional ICs for low ranking scores. It may also show if the current setting for elite percentage could be adjusted.

We also plan to evaluate a number of other Fuzzy GA rule sets against the density and other CA computational tasks. The exploration of fuzzy parameter adaption for multi-population GAs also looks interesting and fruitful. Other multi-population systems should be investigated. Serious modeling of the dynamics of such a system may be necessary before good fuzzy rules could be crafted.

ACKNOWLEDGEMENTS

The lead author would like to thank Dr. Nick Flann, Dr. Don Cooley, Dr. John Paxton and Dr. Heng Da Cheng and my colleges at RightNow Technologies for their support and encouragement during this research.

REFERENCES

(Crutchfield et al., 1997) J.P. Crutchfield, M. Mitchell, R. Das, Evolving Cellular Automata to Perform Computations. Handbook of Evolutionary Computation T. Bäck, D. Fogel, and Z. Michalewicz (eds), Oxford University Press. 1997

(Mitchell et al., 1993) M. Mitchell, P.T. Hraber, J.P. Crutchfield. "Revisiting the Edge of Chaos: Evolving Cellular Autmoata to Perform Compuations". Complex Systems, 7, 89--130, 1993.

(Mitchell et al., 1994) M Mitchell, J.P. Crutchfield, P.T. Hraber. "Evolving Cellular Automata to Perform computations: Mechanisms and Impediments". *Physica D* 75 (1994) 361-391

(Wolfram, 1994) S. Wolfram, Cellular Automata and Complexity: Collected Papers. Reading, MA: Addison-Wesley, 1994.

(Mitchell, 1996) M. Mitchell, An Introduction to Genetic Algorithms (Complex Adaptive Systems Series), MIT Press, 1996.

(Land and Belew, 1995) M. Land, R.K. Belew, “No Perfect Two-State Cellular Automata for Density Classification Exists” Physical Review Letters, 74, 5148, 1995

(Fuks', 1997) H. Fuks', Solution of the Density Classification Problem with Two Cellular Automata Rules, Phys. Rev. E. 55 2081R-2084R, 1997

(Chau et al., 1999) H. F. Chau, L. W. Siu, K. K. Yan, One Dimensional n-ary Classification Using Two Cellular Automaton Rules, Int. Journ. of Modern Physics C., Vol. 10, No 5, 1999.

(Capcarrere et al., 1996) M. Capcarrere, M. Tomassini, M. Sipper, A r=1, two-state Cellular Automata that Classifies density, Physical Review Letters, 77(24):4969-4971, 1996,

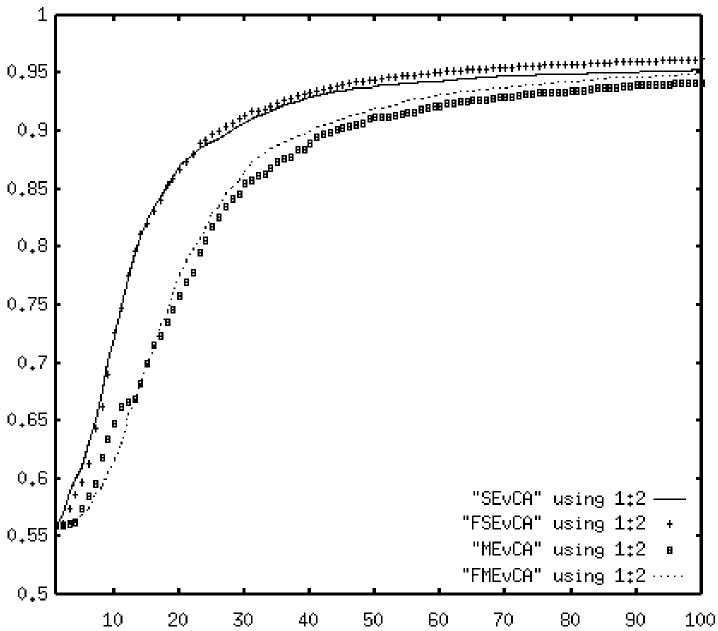
(Gacs et al., 1978) P. Gacs, G. L. Kurdyumov, and L. A. Levin. One-dimensional uniform arrays that wash out finite islands. Probl. Peredachi. Inform., 14:92--98, 1978.

(Cantú-Paz, 1998) E. Cantú-Paz,. A survey of parallel genetic algorithms. Calculateurs Paralleles, Reseaux et Systems Repartis. Vol. 10, No. 2. pp. 141-171. Paris: Hermes, 1998

(Bäck, 1992) T. Bäck, Self-adaptation in Genetic Algorithms, Proceedings Of the 1st European Conf. On Artifial Life, (P.B. F. J. Verela ed). pp 263-271, 1992.

(Lee and Takagi, 1993) M. Lee, H. Takagi. "Dynamic Control of Genetic Algorithms using Fuzzy Logic Techniques", Proc. of the Fifth Int. Conference on Genetic Algorithms (ICGA'93), Urbana-Champaign, IL, 1993, pp. 76-83.

(Shi et al., 1999) Y. Shi, R. Eberhart, Y. Chen. "Implementation of Evolutionary Fuzzy Systems" IEEE Transactions on Fuzzy Systems. Vol. 7, No. 2, April 1999. pp 109-119.



Graph 1 Performance of EvCA algorithms, fitness on Y-axis, generation on X-Axis