Evolution of Non-Uniform Cellular Automata using a Genetic Algorithm: Diversity and Computation

Forrest Sondahl and William Rand
Ctr. for Connected Learning & Computer-Based Modeling, Northwestern Inst. on Complex Systems
230 Annenberg Hall / 2120 N Campus Dr. / Evanston, IL, USA, 60208
{forrest,wrand}@northwestern.edu

ABSTRACT

We used a genetic algorithm to evaluate the cost / benefit of diversity in evolving sets of rules for non-uniform cellular automata solving the density classification problem.

Categories and Subject Descriptors: F.1.1 [Computation by Abstract Devices] Models of Computation

General Terms: Algorithms

Keywords: Diversity, Genetic Algorithms, Non-Uniform Cellular Automata, Search Spaces

1. SUMMARY

Our hypothesis was that given a pre-specified amount of computational power, it is beneficial to explore a minimally diverse set of rules, but exploring too much diversity will usually result in a set of rules that cannot communicate with each other, and thus are ineffective at achieving a performance goal. We designed an experiment to investigate this trade-off. We chose to examine the density classification (DC) problem in cellular automata (CA) [1]. We formalize the concept of diversity as the number of rules in a non-uniform CA (NuCA). In standard (uniform) CA, the same rule is used in each lattice location in the CA, whereas NuCA allow for different rules in each lattice location [3].

In previous work on the DC problem, high performance rules have been both handcrafted [1] and evolved using GAs [2], for uniform CA. Sipper used a modification of the GA to evolve a NuCA-based solution [3]. As opposed to this previous work, our goal was not to find a better solution to the DC problem, but instead to explore how different levels of diversity (numbers of rules) affect the GA's performance.

We used a simple GA with one-point crossover, per bit mutation, full population replacement, and tournament selection of size 3. We evolved a population of 100 individuals, where individuals are sets of CA rules, for 200 generations. We measured fitness by running the NuCA with 100 initial conditions (ICs) generated with the density of 1's chosen randomly from $[0.0, 1.0]$, and counting the ICs the NuCA correctly solved (converged to the appropriate state). The NuCA lattice was 30 cells wide, and we varied the number of different rules (n_{rules}) from the domain $\{1, 2, 3, 5, 6\}$, with rules placed in a cyclic order across the lattice.

We ran three experiments, each with a different CA radius ($r = \{1, 2, 3\}$), which determines the size of the neighborhood that each cell can look at. Results are given in Table 1. The number of rules possible to define with radius r is $2^{2^{2r+1}}$, so the search space becomes drastically larger as r increases. For $r = 1$, the rules were not powerful, and performance was poor across the board, but $n_{rules} = 2$ gave the best results, followed by $n_{rules} = 3$, $n_{rules} = 5$, $n_{rules} = 6$, and lastly $n_{rules} = 1$, although some of the differences were not statistically significant. While increased diversity improved results, only a small amount was optimal, which was consistent with our hypothesis. For $r = 2$, we found that $n_{rules} = \{1, 2\}$ yielded the highest performance, followed closely by $n_{rules} = 3$, with $n_{rules} = \{5, 6\}$ being the worst. In this larger search space, even a small amount diversity did not increase the performance. For $r = 3$, $n_{rules} = 1$ did the best, and performance degraded as n_{rules} increased. In this case any increase in diversity was harmful. There may still be good solutions to the DC problem using diverse sets of rules, but due to the vast search space, the computational cost of finding rules that work together outweighed the marginal benefit from increased expressivity. Given a fixed amount of computational power, finding an optimal balance between diverse and simple representations is difficult.

	$\mathbf{r = 1}$	$\mathbf{r = 2}$	$\mathbf{r = 3}$
$\mathbf{n_{rules} = 1}$	53.7 (*0.46*)	94.1 (*0.70*)	89.4 (*3.83*)
$\mathbf{n_{rules} = 2}$	59.6 (*0.77*)	94.5 (*1.09*)	81.2 (*4.85*)
$\mathbf{n_{rules} = 3}$	57.2 (*0.74*)	89.1 (*2.01*)	63.0 (*5.47*)
$\mathbf{n_{rules} = 5}$	56.8 (*1.16*)	77.9 (*3.59*)	41.7 (*5.97*)
$\mathbf{n_{rules} = 6}$	55.1 (*1.05*)	77.0 (*3.52*)	26.6 (*5.56*)

Table 1: Average of 15 Runs of the Best Individual after 200 Generations. Std. Err. in Parentheses.

Acknowledgments: We thank Uri Wilensky for his feedback and support for FS, NICO for support for WR, and Rick Riolo and Scott Page for early advice.

2. REFERENCES

[1] Gacs, P., Kurdyumov, G. L., and Levin, L. A. One dimensional uniform arrays that wash out finite islands. *Problemy Peredachi Informatsii 14* (1978), 92–98.

[2] Mitchell, M., Crutchfield, J. P., and Das, R. Evolving cellular automata with genetic algorithms: A review of recent work. In *Proceedings of the First International Conference on Evolutionary Computation and Its Applications* (Moscow, Russia, 1996), Russian Academy of Sciences.

[3] Sipper, M. *Evolution of Parallel Cellular Machines: The Cellular Programming Approach.* Springer-Verlag, Heidelberg, 1997.

GECCO'07, July 7–11, 2007, London, England, United Kingdom
ACM 978-1-59593-697-4/07/0007.