EVOLUTION OF NON-UNIFORM CELLULAR AUTOMATA USING A GENETIC ALGORITHM: DIVERSITY AND COMPUTATION

Forrest Sondahl
William Rand
forrest@northwestern.edu
wrand@northwestern.edu

Northwestern Institute on Complex Systems
http://www.northwestern.edu/nico/

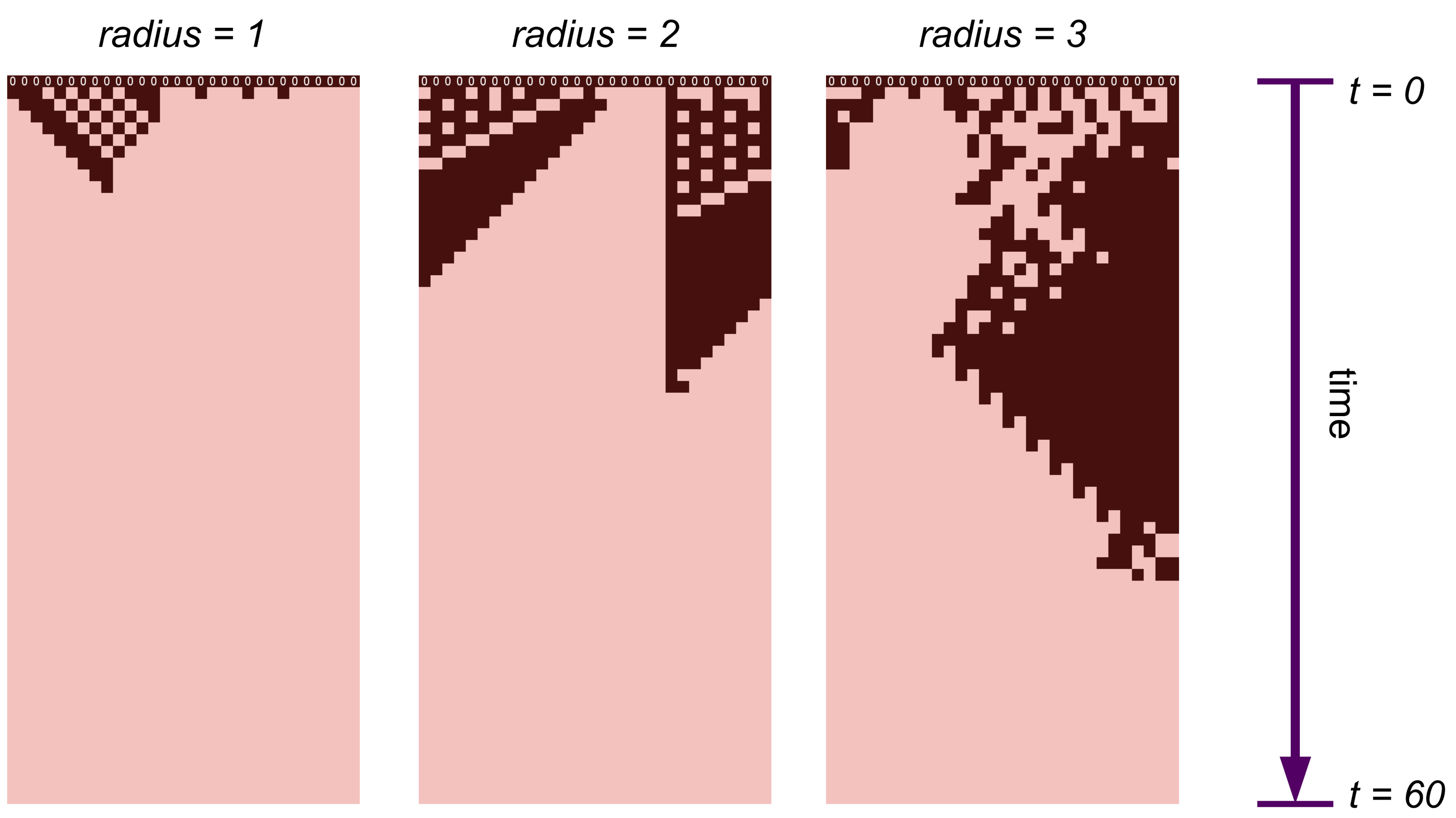
Center for Connected Learning and Computer-Based Modeling
http://ccl.northwestern.edu/

NORTHWESTERN UNIVERSITY

Motivation

To what extent is diversity beneficial when solving a problem? Increasing the diversity of problem solving strategies has two effects: (1) it allows for more complex solutions, and (2) it increases the size of the search space. Given limited computational power, there is a trade-off between these effects. Diverse strategies working together may be able to solve more challenging problems than a single strategy, but evolving strategies that can communicate with each other is often difficult. We investigate this trade-off in the density classification (DC) problem in cellular automata (CA). We formalize the concept of diversity as the number of rules in a non-uniform CA (NuCA).

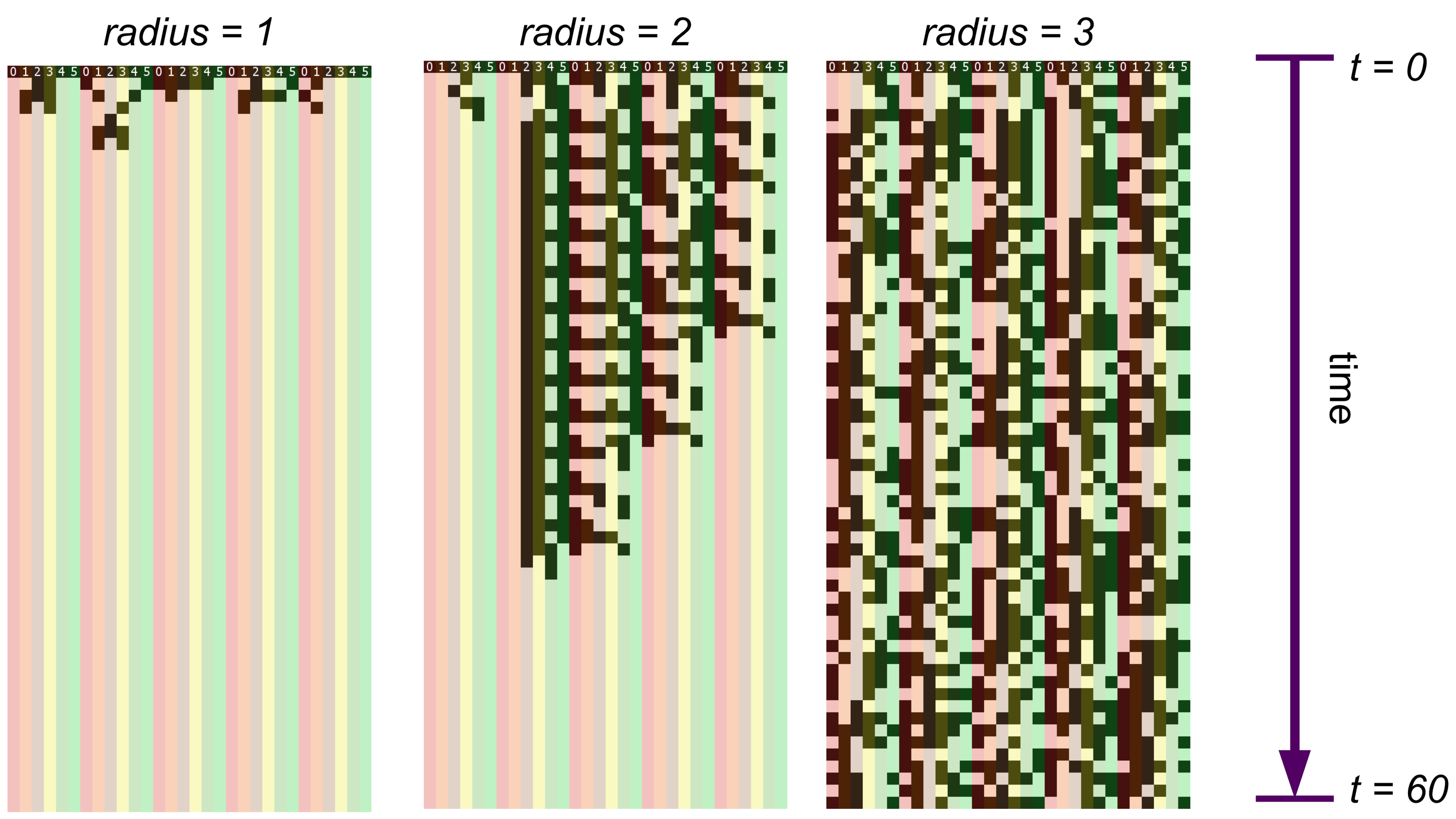
In contrast to prior work with GAs on the DC problem [1,2,3], our goal was not to discover high performance solutions, but rather to explore how different levels of diversity (numbers of rules) affect the quality of rules that the GA discovers.



Time-space diagrams for **uniform** 1-D CAs, solving the density classification problem where 18 of the 30 cells start in the ON state. Each of these is a best-of-generation ruleset from the final generation of a GA run. They each converge to a state of all ON, which is correct since the original density was greater than 0.5.

Method

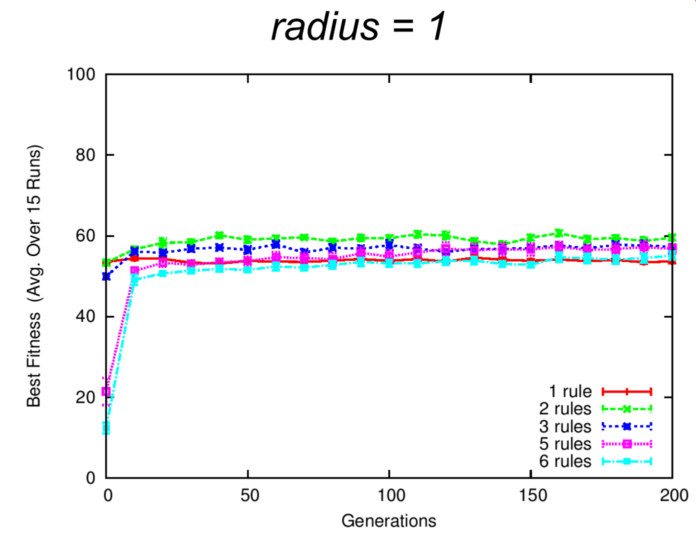
We used a simple GA with one-point crossover, per bit mutation, full population replacement, and tournament selection of size 3. We evolved a population of 100 individuals, where individuals are sets of CA rules, for 200 generations. We measured fitness by running each NuCA ruleset with 100 initial conditions (ICs) generated with the density of 1's chosen randomly from [0.0, 1.0], and counting the ICs the ruleset correctly solved (converged to the appropriate state). Our lattice was only 30 cells wide due to time/computational constraints. In our experiments we varied the number of rules used in the NuCA between 1 and 6 (omitting 4, which does not tile evenly). Rules were placed in a cyclic order across the lattice (as show below). We also varied the CA radius from 1 to 3. We used NetLogo [4] to build our model and collect experimental results.

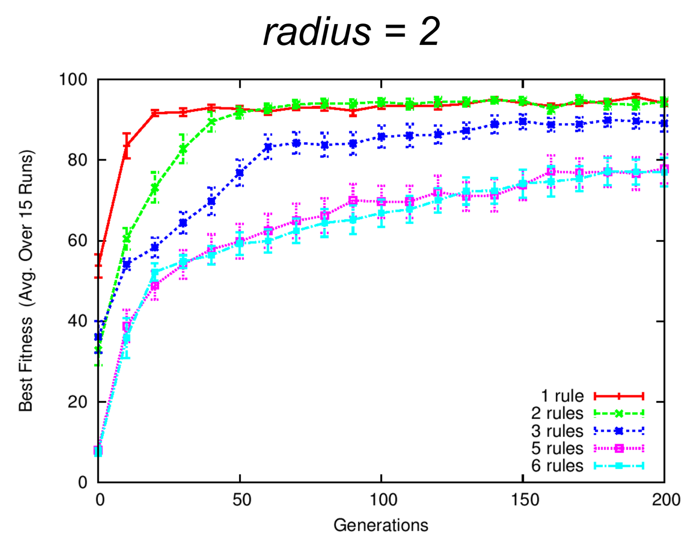


Time-space diagrams for 1-D NuCAs with 6 different rules, which are placed cyclically across the lattice. The different hues show where different rules were placed, while lightness/darkness represents the ON/OFF states of the lattice cells. All three start with 18 cells ON out of 30. The *radius=1* and *radius=2* examples correctly converge to the appropriate state in the time allotted, while the *radius=3* example fails to converge. Each of these is a best-of-generation ruleset from the final generation of a GA run.

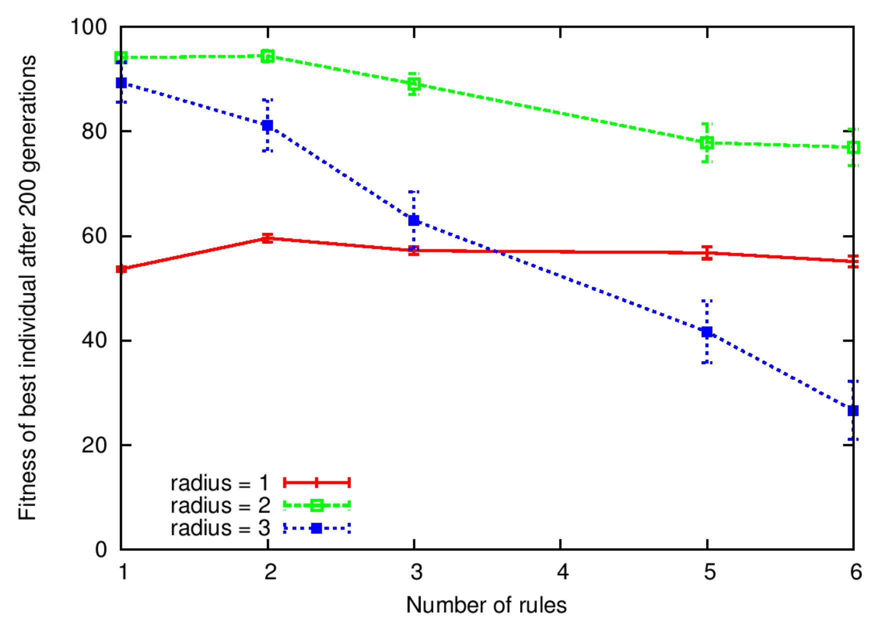
Results

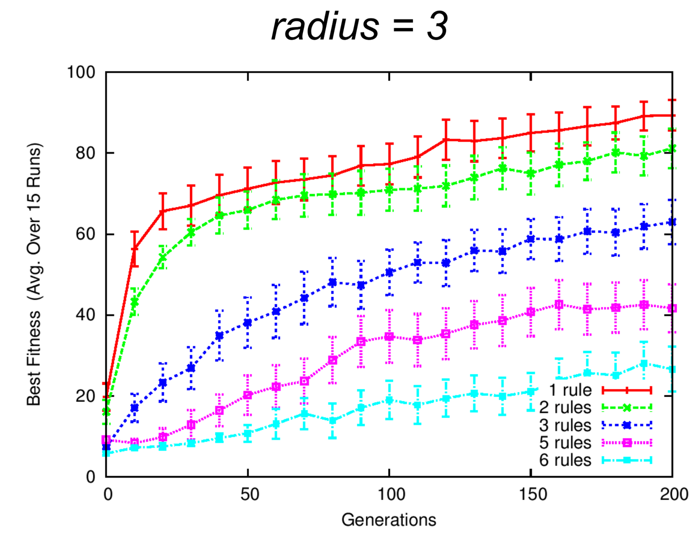
The three graphs at right show the fitness of the best rulesets found in each generation, given different amounts of diversity (numbers of rules in the NuCA). A simple ruleset which changes every cell to ON (or OFF), regardless of the previous state of the lattice, would achieve a fitness score of 50%. The best rulesets found for *radius=1* only marginally improve on this strategy. The best performance was found for *radius=2,* with either 1 or 2 rules. Each data point represents the average across 15 runs, with standard error shown with error bars.





Notice that for *radius=2* and *radius=3*, fitness is still increasing after 200 generations. Will the the more diverse cases eventually outperform the uniform case? This would make a good area for future research. Our current work is based on the premise that a very limited amount of computation time is available.





The above graph summarizes the results of our experiments, by showing more explicitly the relationship between performance and diversity. For *radius=1*, the search space is small and impoverished – there are a mere 256 choices for a single rule. A slight improvement is found by using two rules working together, but this improvement does not continue for larger numbers of rules. For *radius=2*, the best solutions evolve using 1 or 2 rules, and performance degrades slowly for more rules. At *radius=3* the search spaces are much larger, and the GA has increasing difficulty finding good solutions when the amount of diversity is increased.

Conclusion

Given a fixed amount of computational power, finding an optimal balance between diverse and simple representations is difficult. It is tempting to use more complex representations and larger search spaces, since it is likely that a better solution exists in the larger space. Even if better solutions exist, they may not be easy to find. In the NuCA domain, the expanded search spaces are generally supersets of the smaller spaces – e.g. a radius 3 solution using 6 rules can exhibit the same behavior as any radius 2 solution using one rule. However, when search time is limited, the richness of the space affects performance the most. This richness is in part determined by how often diverse strategies cooperate or interfere with each other. There may still be better solutions to the DC problem using diverse sets of rules. However, because of the size and sparseness of the search space, the cost of finding rules that work together outweighed the marginal benefit from increased expressivity.

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.

[4] WILENSKY, U. *NetLogo.* Center for Connected Learning and Computer-based Modeling, Northwestern University, Evanston, IL, 1999.

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.**