

Evolving One Dimensional Cellular Automata to Perform a Non-Trivial Collective Behavior Task: One Case Study

F. Jiménez-Morales¹, M. Mitchell², and J.P. Crutchfield².

¹ Departamento de Física de la Materia Condensada. Universidad de Sevilla.
P. O. Box 1065, 41080-Sevilla, Spain.

² Santa Fe Institute, 1399 Hyde Park Road
Santa Fe, New Mexico, 87501, USA.

Abstract. Here we present preliminary results in which a genetic algorithm (GA) is used to evolve one-dimensional binary-state cellular automata (CA) to perform a non-trivial task requiring collective behavior. Using a fitness function that is an average area in the iterative map, the GA discovers rules that produce a period-3 oscillation in the concentration of 1s in the lattice. We study one run in which the final state reached by the best evolved rule consists of a regular pattern plus some defects. The structural organization of the CA dynamics is uncovered using the tools of computational mechanics.

PACS: 82.20Wt Computational modeling; simulation.

1 Introduction

A cellular automata (CA) is a regular array of N cells, each of whose state $s_i(t)$ is taken from a finite number of values, and which evolves in discrete time steps according to a local rule ϕ . CAs provide simple models of complex systems in which collective behavior can emerge out of the actions of simple, locally connected units. This collective behavior obeys laws that are not easily deduced from the local rule. However it is not well established how to design a CA to exhibit a specific behavior.

The application of genetic algorithms (GA) to the design of one dimensional CA that perform useful computations has both scientific and practical interest [4, 13, 14]. In the original work of Crutchfield and Mitchell [4] a GA was able to discover CAs with high performance on tasks requiring cooperative collective behavior, namely the density and the synchronization tasks. Figure 1 shows two space-time diagrams of two evolved CA. A successful CA for the density classification task decides whether or not the initial configuration contains more than half 1s. If it does, the whole lattice eventually iterates to the fixed point configuration of all cells in state 1; otherwise it eventually iterates to the fixed-point configuration of all 0s [6, 4]. For the synchronization task, a successful CA will

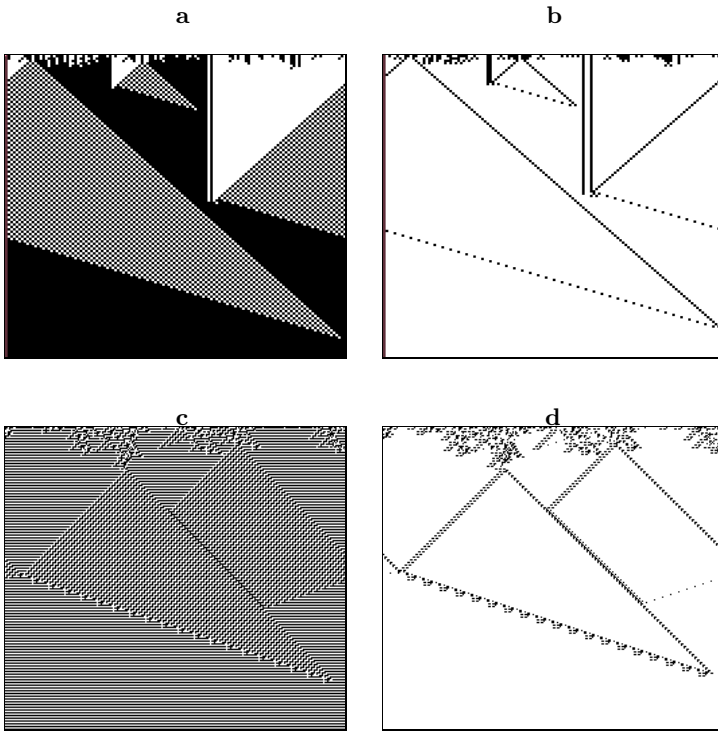


Fig. 1. Space-time diagrams illustrating the behavior of two CAs that perform the density classification task (a) and the synchronization task (c) [5, 13]. Time starts on $t = 0$ and goes from up down and space is displayed on the horizontal axis. Black represents a cell in state 1 while white is a cell in state 0. The corresponding space-time diagram after filtering out regular domains are shown in (b) and (d).

reach a final configuration in which all cells oscillate between all 0s and all 1s on successive time steps [5, 9].

The computational task that we study in this paper is the QP3(P3) task in which the goal is to find a CA that, starting from a random initial condition, reaches one final configuration in which the concentration $c(t) = \frac{1}{N} \sum_i^N s_i(t)$ oscillates among three different values. This task was previously studied in $d = 3$ where the concentration of some CAs show many non-trivial collective behaviors (NTCB).

Figure 2 shows the iterative map, i.e. the graph of $c(t + 1)$ versus $c(t)$, for a family of totalistic CA rules [10]. Here the different collective behaviors are represented by distinct clouds of points. A few years ago a generic argument was given against the existence of collective behaviors with period larger than 2 in extended systems with local interactions [1]. Nonetheless, larger-period collec-

tive behaviors have been observed in such systems. The most interesting NTCB is the quasiperiod three behavior (QP3). Several attempts have been made to understand its phenomenology and have addressed the possible mechanisms by which this puzzling collective behavior emerges [2, 3, 8] but at the moment there is not an answer to the question of how NTCB can be predicted from the local rule. In [11] a GA was used to evolve a population of three dimensional CAs to perform a QP3(P3) task, i.e. under an appropriate fitness function the GA selected rules with P3 or QP3 collective behavior. In this work we evolve a population of one-dimensional CAs rules to perform the QP3(P3) task.

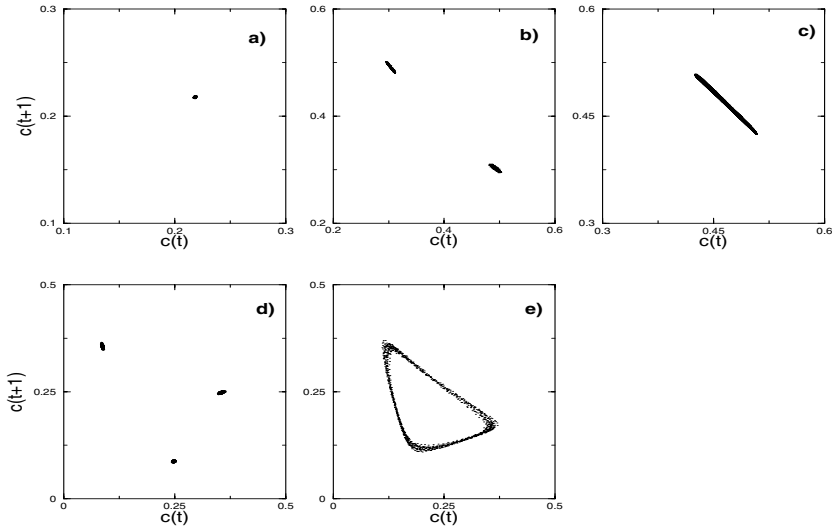


Fig. 2. Iterative map for five totalistic cellular automata rules [10]. The different collective behaviors are shown by distinct clouds of points: (a) period-1; (b) period-2; (c) intermittent period-2; (d) period-3; (e) quasiperiod-3.

2 The Genetic Algorithm

Our GA begins with a population of $P=20$ randomly generated chromosomes. Each chromosome is a bit string representing the output bits for all possible neighborhood configurations in a binary state CA of radius 3, listed in lexicographic order. ("Radius 3" means that each neighborhood of a cell consists of the cell itself plus the three nearest neighbors on either side.) Since there are $2^7 = 128$ such neighborhood configurations, the chromosomes are 128-bits long. Thus there is a huge space of 2^{128} possible rules in which the GA is to search. The fitness evaluation for each CA rule is carried out on a lattice of N cells

starting from a random initial condition of concentration 0.5. After a transient time of $N/2$ time steps, we allow each rule to run for a maximum number of $M = N/2$ iterations. From the iterative map of Figure 2 we observe that the area of period-1 and period-2 behaviors are smaller than the area of period-3 and QP3. This heuristic lead us to define the fitness $F(\phi)$ of a rule ϕ as the average area covered in the iterative map corresponding to ϕ . It is worth pointing out that the final state the CA must reach for this task is not necessarily a periodic one such as it is for the synchronization and the density tasks.

The values of concentration are assembled in groups of 4 consecutive values (c_1, c_2, c_3 and c_4) and the fitness function $F(\phi)$ is defined by:

$$F(\phi) = \frac{4}{M} \sum_i^{M/4} \frac{1}{2} \text{abs}[(c_2 - c_1)(c_4 - c_2) - (c_3 - c_2)(c_3 - c_1)]_i$$

In each generation: (i) $F(\phi)$ is calculated for each rule ϕ in the population. (ii) The population is ranked in order of fitness. (iii) A number $E = 5$ of the highest fitness (“elite”) rules is copied without modification to the next generation. (iv) The remaining $P - E = 15$ rules for the next generation are formed by single-point crossover between randomly chosen pairs of elite rules. The offsprings from each crossover are each mutated with a probability $m = 0.05$. This defines one generation of the GA; it is repeated $G = 10^3$ times for one run of the GA.

3 Results

We performed more than 100 different runs of the GA, each with a different random-number seed. In the density and in the synchronization task the usual lattice size was 149 cells, but here to define clearly the collective behavior we need a greater lattice size. We used a lattice of $N = 10^3$ cells.

Symbol	Rule Table Hexadecimal Code	NTCB	Fitness	λ
ϕ_a	21088418-01091108-41038844-10c18080	P3	0.048	0.211
ϕ_b	ffbe84bc-10874438-c6a08204-9d1b800b	P3	0.186	0.414
ϕ_c	146157d1-fbb53fec-7dfbeffc-eaf0fa28	QP3(P3)	0.066	0.625
ϕ_d	f193800-c06b0eb0-e000461c-80659c11	P3	0.031	0.336

Table 1. The best evolved rules, the rule table hexadecimal code, the type of non-trivial collective behavior, the fitness function averaged over 100 initial conditions and the lambda parameter. To recover the 128-bit string giving the output bits of the rule table, expand each hexadecimal digit to binary. The output bits are then given in lexicographic order.

Generation	Rule Table Hexadecimal Code	Fitness	λ
10	21008100-00200500-4001a000-1080c000	0.004	0.117
25	21008108-20200d20-41412000-1090521b	0.006	0.203
48	21008908-20200500-40298008-1081c082	0.012	0.180
104	21088118-20200108-41098008-1081c142	0.015	0.195
140	21088518-01091108-41018040-10c1a002	0.043	0.203
415 (ϕ_a)	21088418-01091108-41038844-10c18080	0.048	0.211

Table 2. CA look-up table output bits given in hexadecimal code, value of the fitness function and the lambda parameter for some ancestors of rule ϕ_a .

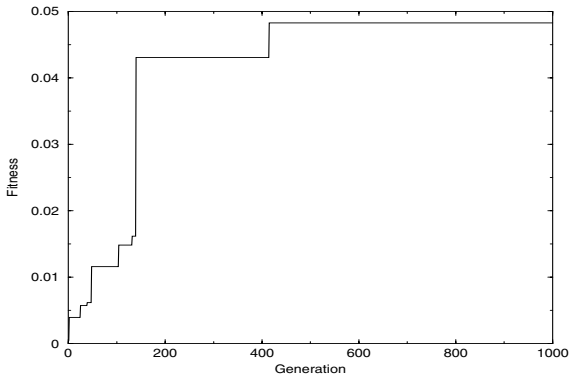


Fig. 3. Best fitness rule versus generation in the run in which rule ϕ_a was found at generation 415. Lattice size is 10^3 cells. The rules of the initial population were randomly selected.

Table 1 shows some of the best evolved rules, the rule table hexadecimal code, the kind of collective behavior observed, the fitness of the rule averaged over 100 initial configurations and the parameter λ , which is the fraction of 1s in the rule table's output bits [12]. Under the fitness function $F(\phi)$ the GA was able to find many rules with the desired behavior and about 30% of the runs ended up with a rule that showed a P3 collective behavior. Only one rule ϕ_c showed a QP3 collective behavior that after a long time decays into a P3, but this will be studied elsewhere.

The progression of a typical evolutionary run is depicted in Figure 3 which plots the fittest rule of each generation in the run in which rule ϕ_a was found. It is observed that the fitness of the best CA rules increases in jumps. Qualitatively the rise in fitness can be divided into several epochs, each one corresponding to the discovery of a new improved strategy. The generational progression of the GA can give important information about the design of CA rules with a specific behavior. Table 2 shows some ancestors of rule ϕ_a . Their iterative map and the time series of the concentration are shown in Figures 4 - 5. In the initial gener-

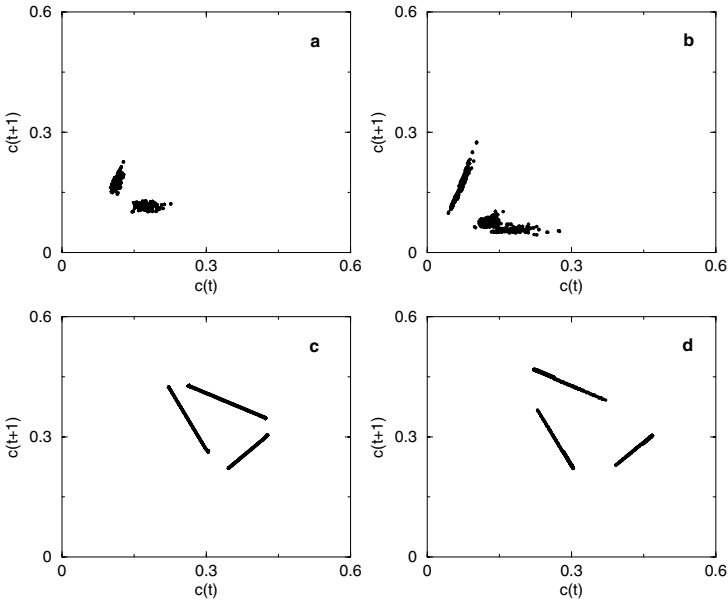


Fig. 4. Iterative map of the concentration of the fittest rule in generation: (a) 10; (b) 48; (c) 140; (d) 415 at which ϕ_a was discovered.

ations the GA discovers a rule (ϕ_{10}) that displays a P2 behavior, Figures 4a-5a, and the fitness of the rule is very small $F(\phi_{10}) = 0.039$. In generation 48 there is a jump in $F(\phi_{48}) = 0.116$ and now the iterative map, Figure 4b, shows two clouds of points with a greater size than in generation 10. In generation 140 there is a big jump in fitness $F(\phi_{140}) = 0.430$, and the iterative map now shows a triangular object, while the time series of the concentration shows clearly three branches. Finally, when rule ϕ_a is discovered at generation 415, a further improvement in the fitness is attained.

Under the fitness function $F(\phi)$ the evolutionary process has selected some rules that, starting from a random initial condition, synchronize the concentration to a three-state cycle. To see how such a synchronization is obtained we use the tools of the computational mechanics developed by Crutchfield and Hanson [7]. This point of view describes the computation embedded in the CA space-time configuration in terms of domains, defects, and defect interactions. Figure 6 shows a space-time diagram of rules ϕ_{10} , ϕ_{48} , ϕ_{140} , and ϕ_a . The space-time diagrams of ϕ_{10} and ϕ_{48} , Figures 6a-b, show irregular regions of fractal structures separated by straight lines. While for rules ϕ_{140} and ϕ_a the space-time diagrams show patterns in which there is an easily recognized spatio-temporally periodic background -a domain- on which some dislocations move. In the simplest case, a domain A consists of a set of cells in the space-time diagram that are always repeated. For example, the domain for rule ϕ_a and ϕ_{140} is shown in

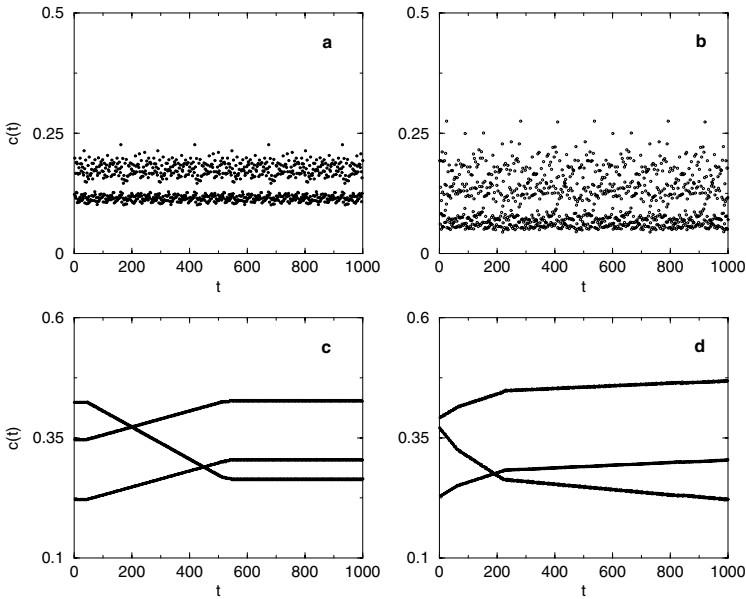


Fig. 5. Time series of the concentration of the fittest rule in generation: (a) 10; (b) 48; (c) 140; (d) $415(\phi_a)$.

Table 3. The concentration of activated cells in Λ oscillates among three values $1/2$, $1/3$, and $1/6$. The computational strategy used by ϕ_a and ϕ_{140} seems to be their capacity to discover a domain that fulfills the requirements imposed by the task. In many cases, two adjacent synchronized regions left and right are out of phase i.e., there is some displacement of the right domain along the temporal axis when compared with the left domain. At the boundaries between them there are several defect cells, or particles that propagate with a given velocity. The particles, named with Greek letters, can be seen in Figure 7 which is a filtered space-time diagram of Figure 6d. The filtered diagrams reveal defect cells that interact among them and are transmitted from distant parts of the lattice until more synchronized regions are obtained.

Though the concentration of the CA can attain a P3 collective behavior, rules like ϕ_a fail to synchronize the whole lattice to a single domain: there are some particles like α and μ that appear at the boundary between two Λ domains which are in phase. Figure 8 shows in a log-log plot the decaying of the density of defects ρ_d versus the time for ϕ_a . Data points have been averaged over ten different initial condition with the system running for 5×10^4 time steps. It can be observed that the density of defects ρ_d decreases with time to a non-zero asymptotic value. This decay of the number of defects means that there are increasingly large homogeneous domains and the particles become increasingly less important to the collective behavior over time.

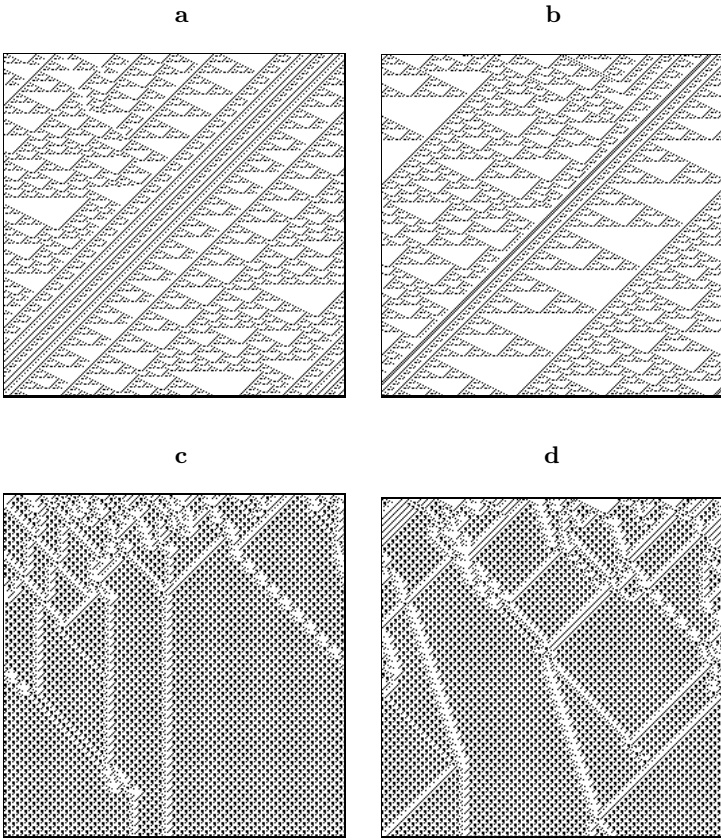


Fig. 6. Space-time diagrams of the fittest evolved rules in the run in which ϕ_a was discovered. For each space-time diagram a window of 256x256 cells is displayed. Generations: (a) 10. (b) 48. (c) 140. (d) 415 (ϕ_b).

Domain Λ	Particles (velocities)
1 1 0 1 0 0	$\alpha \sim \Lambda\Lambda$ (-1)
1 1 0 0 0 0	$\beta \sim \Lambda\Lambda^{+1}$ (1/3)
0 1 0 0 0 0	$\delta \sim \Lambda\Lambda^{-1}$ (0)
1 0 0 1 1 0	$\epsilon \sim \Lambda\Lambda^{+2}$ (1)
0 0 0 1 1 0	$\gamma \sim \Lambda\Lambda^{+1}$ (1)
0 0 0 0 1 0	$\mu \sim \Lambda\Lambda$ (1)
Main Particle Interactions	
$\beta + \gamma \rightarrow \alpha + \delta$	$\alpha + \beta \rightarrow \delta + \epsilon$
$\alpha + \gamma \rightarrow \mu$	$\alpha + \epsilon \rightarrow \gamma$

Table 3. Domain Λ for rule ϕ_a , the particle catalog with their boundaries and velocities and several of the interactions.

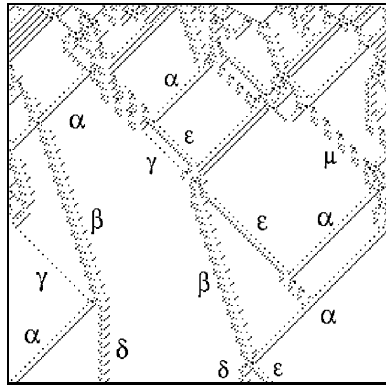


Fig. 7. Filtered space-time diagram of rule ϕ_a corresponding to Figure 6d.

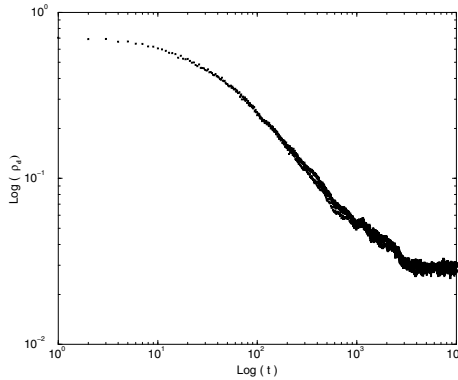


Fig. 8. The density of defects versus time for rule ϕ_a

4 Conclusion

The emergence of collective behaviors that arise from the local interaction of a collection of individuals is a phenomenon observed in many natural systems. To investigate emergent properties in spatially extended systems a theoretical framework was proposed in [4], which requires a decentralized system such as a CA, an idealized computational model of evolution such as a genetic algorithm and finally a computational task that necessitates global information processing.

The computational task that we studied was the QP3(P3) collective behavior task in spatial dimension $d = 1$, where the entire system has to cooperate in order that the concentration of activated cells is oscillating among three values. Using a fitness function that was the size of the attractor in the iterative map, the GA discovers many CA rules whose space-time dynamics displays the desired behavior. It is worth pointing out that for this task the system does not need to be globally synchronized. Here we have focused on a run in which the

best evolved rule's strategy used domains and particles to produce the desired behavior. Domains and particles were also shown to be central to the GA's solutions for the density and synchronization tasks, described in [4, 5].

To our knowledge there are no systematic studies of this computational task in $d = 1$, and we expect that this work will motivate further investigations such as the study of globally synchronized P3 behavior.

Acknowledgments

This work was partially supported by Grant No.PB 97-0741 of the Spanish Government.

References

1. T. Bohr, G. Grinstein, Y. He, and C. Jayaprakash. Coherence, Chaos, and Broken Symmetry in Classical, Many-Body Dynamical Systems. *Phys. Rev. Lett.*, 58:2155-2158, 1987.
2. H. Chaté, G. Grinstein, and P. Lei-Hang Tan. Long-range correlations in systems with coherent(quasi)periodic oscillations. *Phys. Rev. Lett.*, 74:912-915, 1995.
3. H. Chaté and P. Manneville. Collective behaviors in spatially extended systems with local interactions and synchronous updating. *Progress Theor. Phys.*, 87(1):1-60, 1992.
4. J. P. Crutchfield and M. Mitchell. The evolution of emergent computation. *Proceedings of the National Academy of Science U.S.A.*, 92:10742-10746, 1995.
5. R. Das, J. P. Crutchfield, M. Mitchell, and J. E. Hanson. Evolving globally synchronized cellular automata. In L. J. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 336-343, San Francisco, CA, 1995. Morgan Kaufmann.
6. R. Das, M. Mitchell, and J. P. Crutchfield. A genetic algorithm discovers particle-based computation in cellular automata. In Y. Davidor, H.-P. Schwefel, and R. Männer, editors, *Parallel Problem Solving from Nature—PPSN III*, volume 866, pages 344-353, Berlin, 1994. Springer-Verlag (Lecture Notes in Computer Science).
7. J. E. Hanson and J. P. Crutchfield. Computational mechanics of cellular automata: An example. *Physica D*, 103:169-189, 1997.
8. J. Hemmingsson. A totalistic three-dimensional cellular automaton with quasiperiodic behaviour. *Physica A*, 183:225-261, 1992.
9. W. Hordijk. Dynamics, Emergent Computation, and Evolution in Cellular Automata. *Ph.D. dissertation*, Univ. New Mexico, 1999.
10. F. Jiménez-Morales and K. Hassan. Non-trivial collective behavior in three-dimensional totalistic illegal cellular automata with high connectivity. *Physica Letters A*, 240:151-159, 1998.
11. F. Jiménez-Morales. Evolving three-dimensional cellular automata to perform a quasiperiod-3(p3) collective behavior task. *Phys. Rev. E*, 60(4):4934-4940, 1999.
12. C. G.Langton. Studying Artificial Life with Cellular Automata. *Physica D*, 22:120-149, 1986.
13. M. Mitchell, J. P. Crutchfield, and P. T. Hraber. Evolving cellular automata to perform computations: Mechanisms and impediments. *Physica D*, 75:361 - 391, 1994.
14. M. Sipper. *Evolution of Parallel Cellular Machines*. Springer, Germany, 1997.