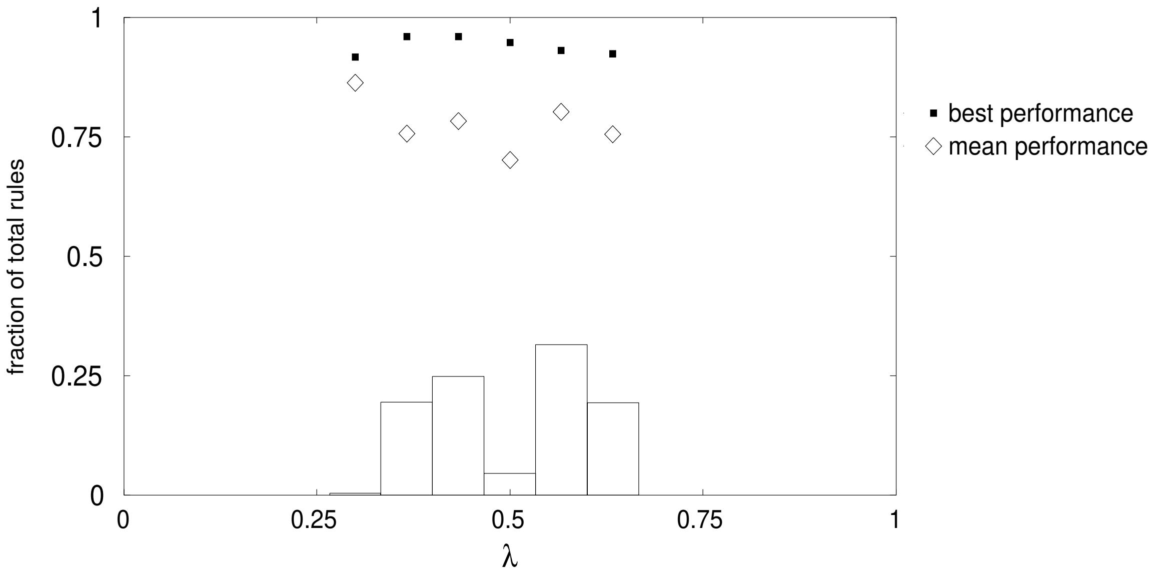

Figure 16: Performances of the rules evolved with the proportional-fitness function. Only the elite rules from generation 100 (merged together from 30 runs) are included in this histogram. The mean performance in each bin (open diamonds), and the best performance in each bin (black squares) is plotted.

7.6 Using Performance as the Fitness Criterion

Can the GA evolve better-performing rules on this task? To test this, we carried out an additional experiment in which performance as defined in the previous section is the fitness criterion. As before, at each generation each rule is tested on 300 initial configurations that are uniformly distributed over density values. However, in this experiment, a rule's fitness is the fraction of initial configurations that are correctly classified. An initial configuration is considered to be incorrectly classified if any bits in the final lattice are incorrect. Aside from this modified fitness function, everything about the GA remained the same as in the proportional-fitness experiments. We performed 30 runs of the GA for 100 generations each. The results are given in Figure 17, which gives a histogram plotting the frequencies of the elite rules from generation 100 of all 30 runs as a function of λ. As can be seen, the shape of the histogram again has two peaks centered around a dip at $\lambda = 1/2$. This shape results from the same symmetry-breaking effect that occurred in the proportional-fitness case: these runs also evolved essentially the same strategies as the epoch-3 strategies described earlier. The best and mean performances here are comparable to the best performances in the proportional-fitness case; the best performances found here are ≈ 0.95.

The performance as a function of $\rho(0)$ for one of the best rules evolved with performance fitness is plotted in Figure 18, for lattice sizes of 149 (the lattice size used for testing the rules in the GA runs), 599, and 999. This rule has $\lambda \approx 0.54$, and its strategy is similar to that shown in Figure 15: it increases sufficiently large blocks of adjacent or nearly adjacent 0's. We used the same procedure to make these plots as was described earlier for Figure 3. As can be seen, the performance according to this measure is significantly worse than that of the GKL rule (cf. Figure 3), especially on larger lattice sizes. The worst performances for the larger lattice sizes are centered slightly above $\rho = 0.5$. On such initial conditions the CA should relax to a fixed point of all 1's, but more detailed inspection of these results revealed

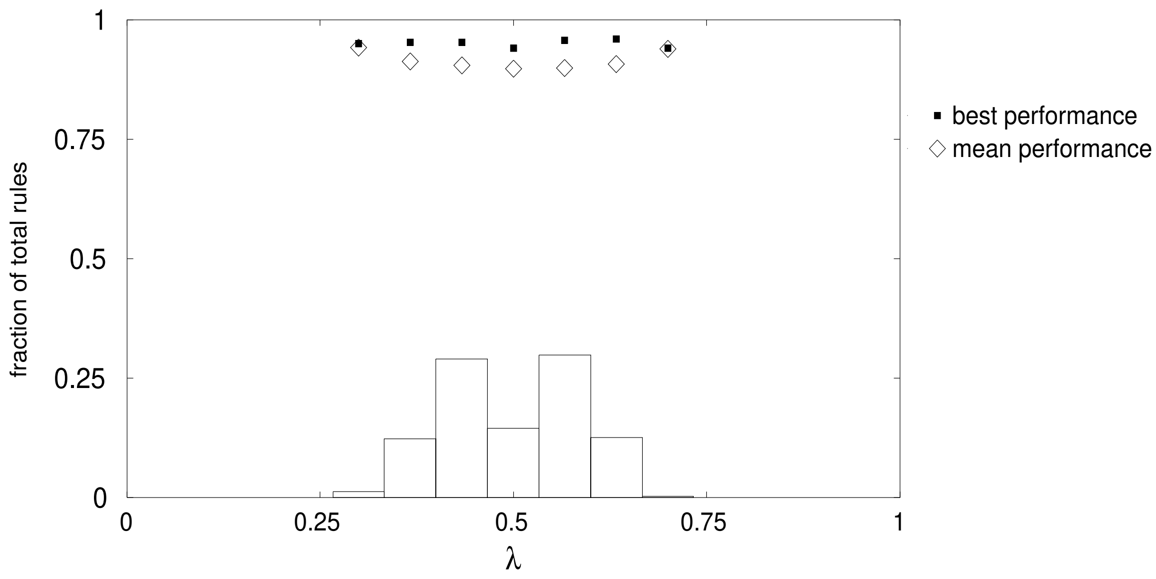

Figure 17: Results from our experiment with performance as the fitness criterion. The histogram plots the frequencies of elite rules merged from the final generations (generation 100) of 30 runs in which the performance-fitness function was used.

that on almost every initial condition with ρ slightly above 0.5, the CA is relaxing to a fixed point of all 0's. This is a result of this rule's strategy of increasing "sufficiently large" blocks of 0's: the appropriate size to increase was evolved for a lattice with $N = 149$. With larger lattices, the probability of such blocks in initial conditions with $\rho > 0.5$ increases, and the closer the ρ of such initial conditions to 0.5, the more likely such blocks are to occur. In the CA we tested with $N = 599$ and $N = 999$, such blocks occured in most initial conditions with ρ slightly above 0.5, and these initial conditions were always classified incorrectly. This shows that keeping the lattice size fixed during GA evolution can lead to over-fitting for the particular lattice size. We plan to experiment with varying the lattice size during evolution in an attempt to prevent such over-fitting.

7.7 Adding A Diversity-Enforcement Mechanism

The description given above of the four epochs in the GA's search explains the results of our experiment, but it does not explain the difference between our results and those of the original experiment reported in [24]. One difference between our GA and the original was the inclusion in the original of a diversity-enforcement scheme that penalized newly formed rules that were too similar in Hamming distance to existing rules in the population. To test the effect of this scheme on our results, in one set of experiments we included a similar scheme. In our scheme, every time a new string is created through crossover and mutation, the average Hamming distance between the new string and the elite strings—the 50 strings that are copied unchanged—is measured. If this average distance is less than 30% of the string length (here 38 bits), then the new string is not allowed in the new population. New strings continue to be created through crossover and mutation until 50 new strings have met this diversity criterion. We note that many other diversity-enforcement schemes have been developed in the GA literature; e.g., "crowding" [9].

The results of this experiment are given in Figure 19. The histogram in that figure

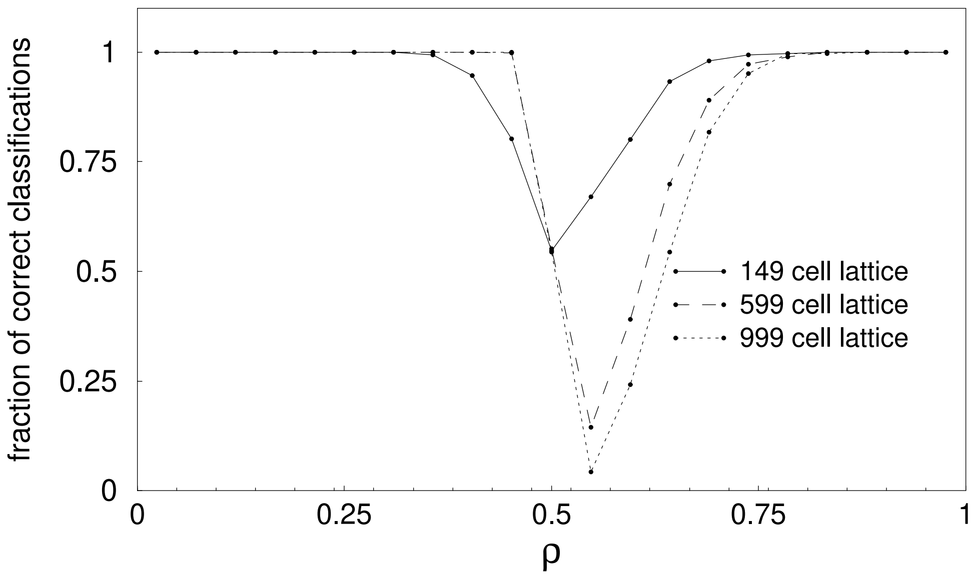

Figure 18: Performance of one of the best rules evolved using performance fitness, plotted as a function of $\rho(0)$. Performance plots are given for three lattice sizes: 149 (the size of the lattice used in the GA runs), 599, and 999. This rule has $\lambda \approx 0.54$.

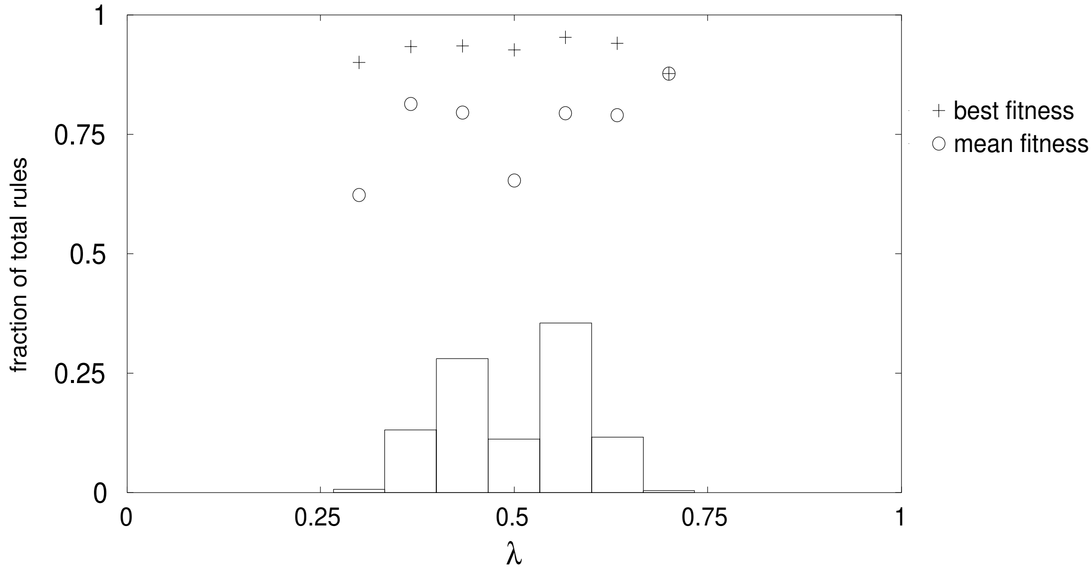

Figure 19: Results from our experiment in which a diversity-enforcement mechanism was added to the GA. The histogram plots the frequencies of rules merged from the entire population at generation 100 of 20 runs with the diversity-enforcement scheme.

represents the merged rules from the entire population at generation 100 of 20 runs of the GA, using the proportional-fitness function and our diversity-enforcement scheme. As can be seen, the histogram in this figure is very similar to that in Figure 5(b). The only major difference is the significantly lower mean fitness in the middle and leftmost bins, which results from the increased requirement for diversity in the final non-elite population. We conclude that the use of a similar diversity-enforcement scheme was not responsible for the difference between the results from [24] and our results.

7.8 Differences Between Our Results and the Original Experiment

As was seen in Figure 5(b), our results are strikingly different from those reported in [24]. These experimental results, along with the theoretical argument that the most successful rules for this task should have λ close to $1/2$, lead us to conclude that the interpretation of the original results as giving evidence for the hypotheses concerning evolution, computation, and λ is not correct. However, we do not know what accounted for the differences between our results and those obtained in the original experiment. We speculate that the differences are due to additional mechanisms in the GA used in the original experiment that were not reported in [24]. For example, the original experiment included a number of additional sources of randomness, such as the regular injection of new random rules at various λ values and a much higher mutation rate than that in our experiment [23]. These sources of randomness may have slowed the GA's search for high-fitness rules and prevented it from converging on rules close to $\lambda = 1/2$. Our experimental results and theoretical analysis give strong reason to believe that the clustering close to λ_c seen in Figure 4) is an artifact of mechanisms in the particular GA that was used rather than a result of any computational advantage conferred by the λ_c regions.

Although the results were very different, there is one qualitative similarity: the rule-frequency-versus-λ histograms in both cases contained two peaks separated by a dip in the center. As already noted, in our histogram the two peaks were closer to $\lambda = 1/2$ by a factor of 4, but it is possible that the original results were due to a mechanism similar to (i) the epoch-0 sensitivity to initial configuration and population asymmetry about $\lambda = 1/2$ or (ii) the symmetry breaking we observed in epoch 3, as described above. Perhaps these were combined with additional forces, such as additional sources of randomness, in the original GA that kept rules far away from $\lambda = 1/2$. Unfortunately, the best and mean fitnesses for the λ bins were not reported for the original experiment. As a consequence we do not know whether or not the peaks in the original histogram contained high-fitness rules, or even if they contained rules that were more fit than rules in other bins. Our results and the basic symmetry in the problem suggest otherwise.

8. General Discussion

8.1 What We Have Shown

The results reported in this paper have demonstrated that the results from the original experiment do not hold up under our experiments. We conclude that the original experiment does not give firm evidence for the hypotheses it was meant to test: first, that rules capable of performing complex computation are most likely to be found close to λ_c values and, second, that when CA rules are evolved by a GA to perform a nontrivial computation, evolution will tend to select rules close to λ_c values.

As we argued theoretically and as our experimental results suggest, the most successful rules for performing a given ρ-classification task will be close to a particular value of λ that depends on the particular ρ_c of the task. Thus for this class of computational tasks, the λ_c values associated with an "edge of chaos" are not correlated with the ability of rules to perform the task.

The results presented here do not disprove the hypothesis that computational capability can be correlated with phase transitions in CA rule space.[11] Indeed, this general phenomena has already been noted for other dynamical systems.[12] More generally, the computational capacity of evolving systems may very well require dynamical properties characteristic of phase transitions if they are to increase their complexity. We have shown only that the published experimental support cited for hypotheses relating λ_c and computational capability in CA was not reproduced.

In the remainder of this section, we step back from these particular experiments and discuss in more general terms the ideas that motivated these studies.

8.2 λ, Dynamical Behavior, and Computation

As was noted earlier, Langton presented evidence that, given certain caveats regarding the radius r and number of states k, there is some correlation between λ and the behavior of an "average" CA on an "average" initial configuration [17]. Behavior was characterized in terms of quantities such as single-site entropy, two-site mutual information, difference-pattern spreading rate, and average transient length. The correlation is quite good for very low and very high λ values, which predict fixed-point or short-period behavior. However, for intermediate λ values, there is a large degree of variation in behavior. Moreover, there is no precise correlation between these λ values and the location of a behavioral "phase transition", other than that described by Wootters and Langton in the limit of infinite k.

The remarks above and all the experimental results in [17] are concerned with the relationship between λ and the dynamical behavior of CA. They do not directly address the relationship between λ and computational capability of CA. The basic hypothesis was that λ correlates with computational capability in that rules capable of complex, and in particular, universal, computation must be, or at least are most likely to be, found near λ_c values. As far as CA are concerned, the hypothesis was based on the intuition that complex computation cannot be supported in the short-period or chaotic regimes because the phenomena that apparently occur only in the "complex" (non-periodic, non-chaotic) regimes, such as long

[11] There are some results concerning computation in CA and phase transitions. Individual CA have been known for some time to exhibit phase transitions with the requisite divergence of correlation length required for infinite memory capacity.[2]

[12] In the context of continuous-state dynamical systems, it has been shown that there is a direct relationship between intrinsic computational capability of a process and the degree of randomness of that process at the phase transition from order to chaos. Computational capability was quantified with the statistical complexity, a measure of the amount of memory of a process, and via the detection of an embedded computational mechanism equivalent to a stack automaton.[4, 5]

transients and long space-time correlation, are necessary to support complex computation. There has thus far been no experimental evidence correlating λ with an independent measure of computation. Packard's experiment was intended to address this issue since it involved an independent measure of computation—performance on a particular complex computational task—but as we have shown, these experiments do not provide evidence for the hypothesis linking λ_c values with computational ability.

One problem is that these hypotheses have not been rigorously formulated. If the hypotheses put forth in [17] and [24] are interpreted to mean that *any* rule performing complex computation (as exemplified by the $\rho = 1/2$ task) must be close to λ_c, then we have shown it to be false with our argument that correct performance on the $\rho = 1/2$ task requires $\lambda = 1/2$. If, instead, the hypotheses are concerned with generic, statistical properties of CA rule space—the "average" behavior of an "average" CA at a given λ—then the notion of "average behavior" must be better defined. Additionally more appropriate measures of dynamical behavior and computational capability must be formulated, and the notion of the "edge of chaos" must also be well-defined.

The argument that complex computation cannot occur in chaotic regimes may seem intuitively correct, but there is actually a theoretical framework and strong experimental evidence to the contrary. Hanson and Crutchfield [3, 12] have developed a method for filtering out chaotic "domains" in the space-time diagram of a CA, sometimes revealing "particles" that have the non-periodic, non-chaotic properties of structures in Wolfram's Class 4 CA. That is, with the appropriate filter applied, complex structures can be uncovered in a space-time diagram that, to the human eye and to the statistics used in [17] and [24], appears to be completely random. As an extreme example, it is conceivable that such filters could be applied to a seemingly chaotic CA and reveal that the CA is actually implementing a universal computer (with glider guns implementing AND, OR, and NOT gates, etc.). Hanson and Crutchfield's results strikingly illustrate that apparent complexity of behavior—and apparent computational capability—can depend on the implicit "filter" imposed by one's chosen statistics.

8.3 What Kind of Computation in CA Do We Care About?

In the section above, the phrases "complex computation" and "computational capability" were used somewhat loosely. As was discussed in Section 3, there are at least three different interpretations of the notion of computation in CA. The notion of a CA being able to perform a "complex computation" such as the $\rho_c = 1/2$ task, where the CA performs the same computation on all initial configurations, is very different from the notion of a CA being capable, under some special set of initial configurations, of simulating a universal computer. Langton's speculations regarding the relationship between dynamical behavior and computational capability seemed to be more concerned with the latter than the former, though the implication is that the capability to sustain long transients, long correlation lengths, and so on are necessary for both notions of computation.

If "computationally capable" is taken to mean "capable, under some initial configuration(s), of universal computation", then one might ask why this is a particularly important property of CA on which to focus. In [17] CA were used as a vehicle to study the relationship between phase transitions and computation, with an emphasis on universal computation. But for those who want to use CA as scientific models or as practical computational tools, a focus on the capacity for universal computation may be misguided. If a CA is being used as a model of a natural process (e.g., turbulence), then it is currently of limited interest to know whether or not the process is in principle capable of universal computation if universal computation will arise only under some specially engineered initial configuration that the natural process is extremely unlikely to ever encounter. Instead, if one wants to understand emergent computation in natural phenomena as modeled by CA, then one should try to understand what computation the CA "intrinsically" does [3, 12] rather than what it is "in principle capable" of doing only under some very special initial configurations. Thus, understanding the conditions under which a capacity for universal computation is possible will not be of much value in understanding the natural systems modeled by CA.

This general point is neither new nor deep. Analogous arguments have been put forth in the context of neural networks, for example. While many constructions have been made of universal computation in neural networks (e.g., [29]), some psychologists (e.g., [28]) have argued that this has little to do with understanding how brains or minds work in the natural world.

Similarly, if one wants to use a CA as a parallel computer for solving a real problem—such as face recognition—it would be very inefficient, if not practically impossible, to solve the problem by (say) programming Conway's Game of Life CA to be a universal computer that simulates the action of the desired face recognizer. Thus understanding the conditions under which universal computation is possible in CA is not of much practical value either.

In addition, it is not clear that anything like a drive toward universal-computational capabilities is an important force in the evolution of biological organisms. It seems likely that substantially less computationally-capable properties play a more frequent and robust role. Thus asking under what the conditions evolution will create entities (including CA) capable of universal computation may not be of great importance in understanding natural evolutionary mechanisms.

In short, it is mathematically important to know that some CA are in principle capable of universal computation. But we argue that this is by no means the most scientifically interesting property of CA. More to the point, this property does not help scientists much in understanding the emergence of complexity in nature or in harnessing the computational capabilities of CA to solve real problems.

9. Conclusion

The main purpose of this study was to examine and clarify the evidence for various hypotheses related to evolution, dynamics, computation, and cellular automata. We hope this study has shed some new and constructive light on these issues. As a result of our study we have identified a number of evolutionary mechanisms, such as the role of combinatorial drift, and the role of symmetry and the impediments to emerging computational strategies caused by symmetry breaking. For example, we have found that the breaking of the goal task's symmetries in the early generations can be an impediment to further optimization of individuals in the population. The symmetry breaking results in a kind of suboptimal speciation in the population that is stable or, at least, meta-stable over long times. The symmetry-breaking effects we described here may be similar to symmetry-breaking phenomena such as bilateral symmetry and handedness that emerge in biological evolution. It is our goal to develop a more rigorous framework for understanding these mechanisms in the context of evolving CA. We believe that a deep understanding of these mechanisms in this relatively simple context can yield insights for understanding evolutionary processes in general and for successfully applying evolutionary-computation methods to complex problems.

Though our experiments did not reproduce the results reported in [24], we believe that the original conception of using GAs to evolve computation in CA is an important idea. Aside from its potential for studying various theoretical issues, it also has a potential practical side that could be significant. As was mentioned earlier, CA are increasingly being studied as a class of efficient parallel computers; the main bottleneck in applying CA more widely to parallel computation is *programming*—in general it is very difficult to program CA to perform complex tasks. Our results suggest that the GA has promise as a method for accomplishing such programming automatically. In order to further test the GA's effectiveness as compared with other search methods, we performed an additional experiment, comparing the performance of our GA on the $\rho_c = 1/2$ task with the performance of a simple steepest-ascent hill-climbing method. We found that the GA significantly outperformed hill-climbing, reaching much higher fitnesses for an equivalent number of fitness evaluations. This gives some evidence for the relative effectiveness of GAs as compared with simple gradient ascent methods for programming CA. Koza [16] has also evolved CA rules using a very different type of representation scheme; it is a topic of substantial practical interest to study the relationship between representation and GA success on such tasks.

Acknowlegments

This research was supported by the Santa Fe Institute, under the Adaptive Computation, Core Research, and External Faculty Programs, and by the University of California, Berkeley, under contract AFOSR 91-0293. Thanks to Doyne Farmer, Jim Hanson, Erica Jen, Chris Langton, Wentian Li, Cris Moore, and Norman Packard for many helpful discussions and suggestions concerning this project. Thanks also to Emily Dickinson and Terry Jones for technical advice.

References

[1] E. Berlekamp, J. H. Conway, and R. Guy. *Winning ways for your mathematical plays.* Academic Press, New York, NY, 1982.

[2] M. Creutz. Deterministic Ising dynamics. *Ann. Phys.*, 167:62, 1986.

[3] J. P. Crutchfield and J. E. Hanson. Turbulent pattern bases for cellular automata. Technical Report 93-03-010, Santa Fe Institute, 1993. To appear in *Physica D.*

[4] J. P. Crutchfield and K. Young. Inferring statistical complexity. *Physical Review Letters*, 63:105, 1989.

[5] J. P. Crutchfield and K. Young. Computation at the onset of chaos. In W. H. Zurek, editor, *Complexity, Entropy, and the Physics of Information*, pages 223–269. Addison-Wesley, Redwood City, CA, 1990.

[6] P. Gonzaga de Sá and C. Maes. The Gacs-Kurdyumov-Levin automaton revisited. *Journal of Statistical Physics*, 67(3/4):507–522, 1992.

[7] D. Farmer, T. Toffoli, and S. Wolfram, editors. *Cellular Automata: Proceedings of an Interdisciplinary Workshop.* North Holland, Amsterdam, 1984.

[8] 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.

[9] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, Reading, MA, 1989.

[10] J. Guckenheimer and P. Holmes. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields.* Springer-Verlag, New York, 1983.

[11] H. A. Gutowitz, editor. *Cellular Automata.* MIT Press, Cambridge, MA, 1990.

[12] J. E. Hanson and J. P. Crutchfield. The attractor-basin portrait of a cellular automaton. *Journal of Statistical Physics*, 66(5/6):1415–1462, 1992.

[13] J. H. Holland. *Adaptation in Natural and Artificial Systems.* The University of Michigan Press, Ann Arbor, MI, 1975.

[14] S. A. Kauffman. Requirements for evolvability in complex systems: Orderly dynamics and frozen components. *Physica D*, 42:135–152, 1990.

[15] S. A. Kauffman and S. Johnson. Co-evolution to the edge of chaos: Coupled fitness landscapes, poised states, and co-evolutionary avalanches. In C. G. Langton, G. Taylor, J. Doyne Farmer, and S. Rasmussen, editors, *Artificial Life II*, pages 325–368, Redwood City, CA, 1992. Addison-Wesley.

[16] J. R. Koza. *Genetic programming: On the programming of computers by means of natural selection.* MIT Press, Cambridge, MA, 1993.

[17] C. G. Langton. Computation at the edge of chaos: Phase transitions and emergent computation. *Physica D*, 42:12–37, 1990.

[18] W. Li. Non-local cellular automata. In L. Nadel and D. Stein, editors, *1991 Lectures in Complex Systems*, pages 317–327. Addison-Wesley, Redwood City, CA, 1992.

[19] W. Li and N. H. Packard. The structure of the elementary cellular automata rule space. *Complex Systems*, 4:281–297, 1990.

[20] W. Li, N. H. Packard, and C. G. Langton. Transition phenomena in cellular automata rule space. *Physica D*, 45:77–94, 1990.

[21] K. Lindgren and M. G. Nordahl. Universal computation in a simple one-dimensional cellular automaton. *Complex Systems*, 4:299–318, 1990.

[22] N. Packard. Complexity of growing patterns in cellular automata. In J. Demongeot, E. Goles, and M. Tchuente, editors, *Dynamical Behavior of Automata: Theory and Applications.* Academic Press, New York, 1984.

[23] N. H. Packard. Personal communication.

[24] N. H. Packard. Adaptation toward the edge of chaos. In J. A. S. Kelso, A. J. Mandell, and M. F. Shlesinger, editors, *Dynamic Patterns in Complex Systems*, pages 293–301, Singapore, 1988. World Scientific.

[25] J. B. Pollack. The induction of dynamical recognizers. *Machine Learning*, 7:227–252, 1991.

[26] K. Preston and M. Duff. *Modern Cellular Automata.* Plenum, New York, 1984.

[27] A. Rosenfeld. Parallel image processing using cellular arrays. *Computer*, 16:14, 1983.

[28] D. E. Rumelhart and J. L. McClelland. PDP models and general issues in cognitive science. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing*, volume 1, Cambridge, MA, 1986. MIT Press.

[29] H. Siegelman and E. D. Sontag. Neural networks are universal computing devices. Technical Report SYCON-91-08, Rutgers Center for Systems and Control, Rutgers University, New Brunswick, NJ 08903, 1991.

[30] T. Toffoli and N. Margolus. *Cellular Automata Machines: A new environment for modeling.* MIT Press, Cambridge, MA, 1987.

[31] S. Wolfram. Universality and complexity in cellular automata. *Physica D*, 10:1–35, 1984.

[32] S. Wolfram, editor. *Theory and applications of cellular automata.* World Scientific, Singapore, 1986.

[33] W. K. Wootters and C. G. Langton. Is there a sharp phase transition for deterministic cellular automata? *Physica D*, 45:95–104, 1990.

[34] A. Wuensche and M. Lesser. *The global dynamics of cellular automata.* Santa Fe Institute Studies in the Sciences of Complexity. Addison-Wesley, Reading, MA, 1992.