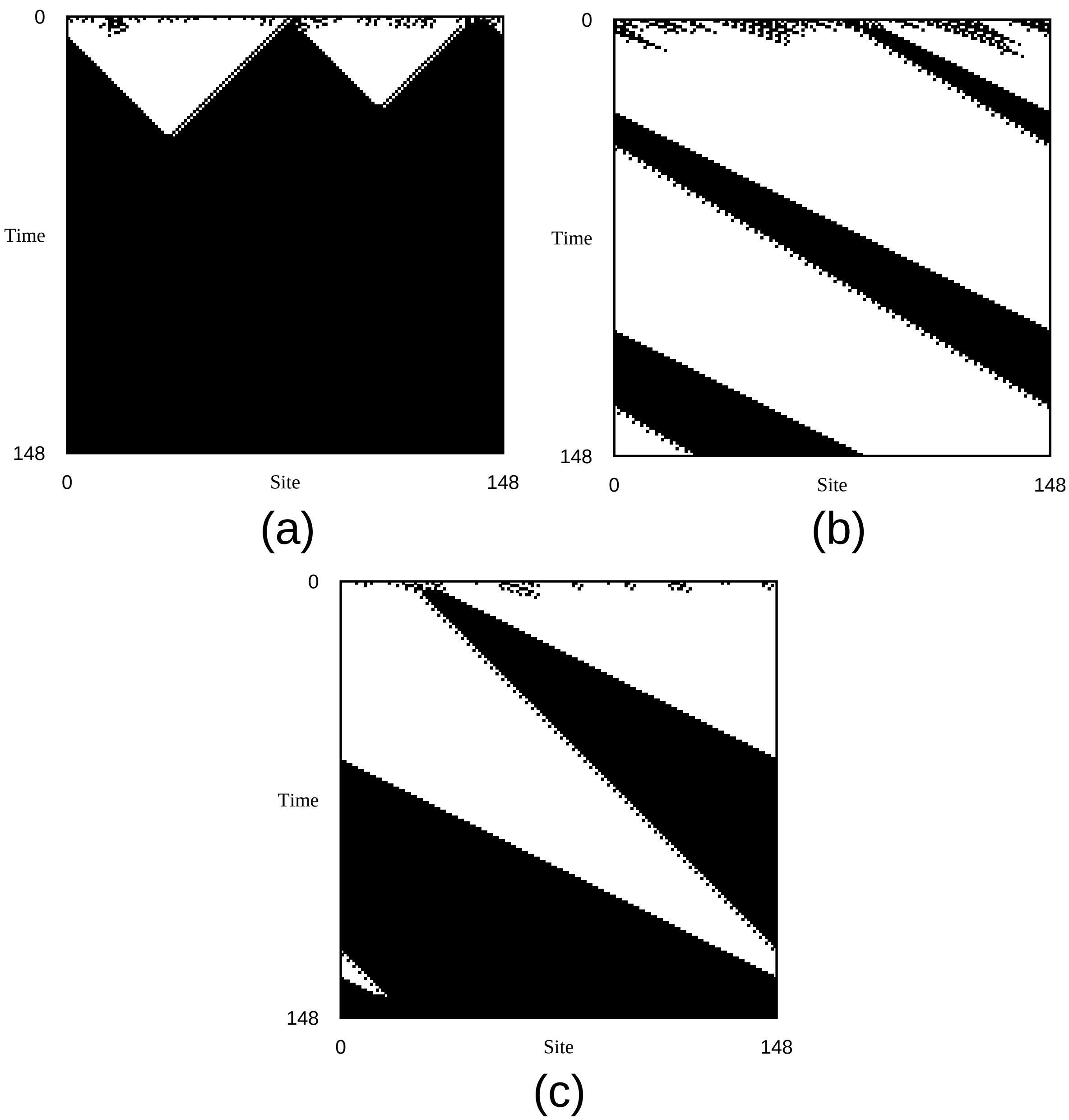


Figure 8: Three typical errors made by Epoch 3 rules. In (a), a rule with $\lambda \approx 0.49$ incorrectly expands blocks in an IC with $\rho_0 \approx 0.38$. In (b), a rule with $\lambda \approx 0.42$ expands blocks too slowly on an IC with $\rho_0 \approx 0.56$. In (c), a rule with $\lambda \approx 0.52$ *creates* a block that was not present in $\mathbf{s}_0$ with $\rho_0 \approx 0.19$, and expands it. All these examples led to incorrect classifications.

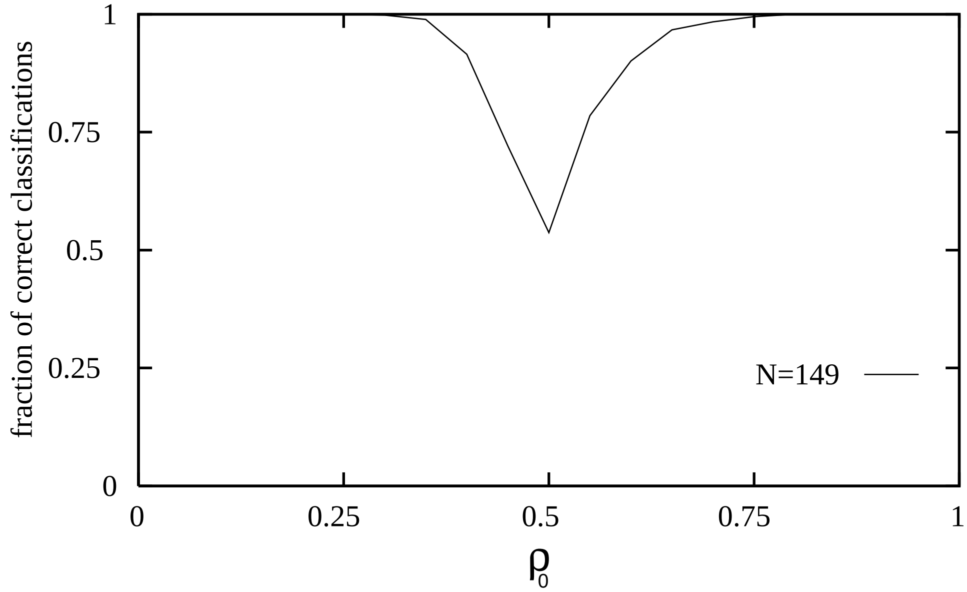


Figure 9: Performance of an Epoch 4 rule, plotted as a function of ρ_0, with $N = 149$. This rule has $\lambda \approx 0.59$.

of ICs. The experimental performance of an Epoch 4 rule as a function of ρ_0 is given in Figure 9. This plot was made in the same way as that in Figure 2. Like the GKL rule, most of the classification errors occur close to $\lambda = 1/2$, though the width of the error region is much larger here than that seen for the GKL rule in Figure 2.

8. λ, Selection, and Combinatorial Drift

Up to this point we have described the GA's behavior in terms of (i) the large-scale time history of the best fitness, (ii) the strategy epochs in this time history, and (iii) the details of the actual strategies discovered at each epoch. This description examined properties of the best individual rules rather than properties of the entire elite population. We now present an intermediate-level description of the GA's behavior in terms of the distribution of λ in the elite population over time. This will reveal how population-level structures emerge in the different epochs and will aid in understanding the mechanisms by which the GA progresses through the epochs.

The strategies described in the previous section each have two opposite instantiations—one that specializes for low ρ_0 and the other that specializes for high ρ_0. On a given run, the GA discovers one or the other class of strategies, but not both. Figure 10 displays histograms of rule frequency versus λ for the elite rules at generation 99 in two typical runs. In 10(a) all the elite rules have $\lambda < 1/2$; this run resulted in a population of low-ρ_0 specialists which implement Strategy 1. In 10(b), all the elite rules have $\lambda > 1/2$; this run resulted in a population of high-ρ_0 specialists which implement Strategy 2.

Figure 11 is a mosaic of histograms plotting the frequency of elite rules versus λ, where the elite rules from 44 different runs are merged together. Each histogram therefore contains counts from $20 \times 44 = 880$ elite rules. These 44 runs were the same ones for which statistics

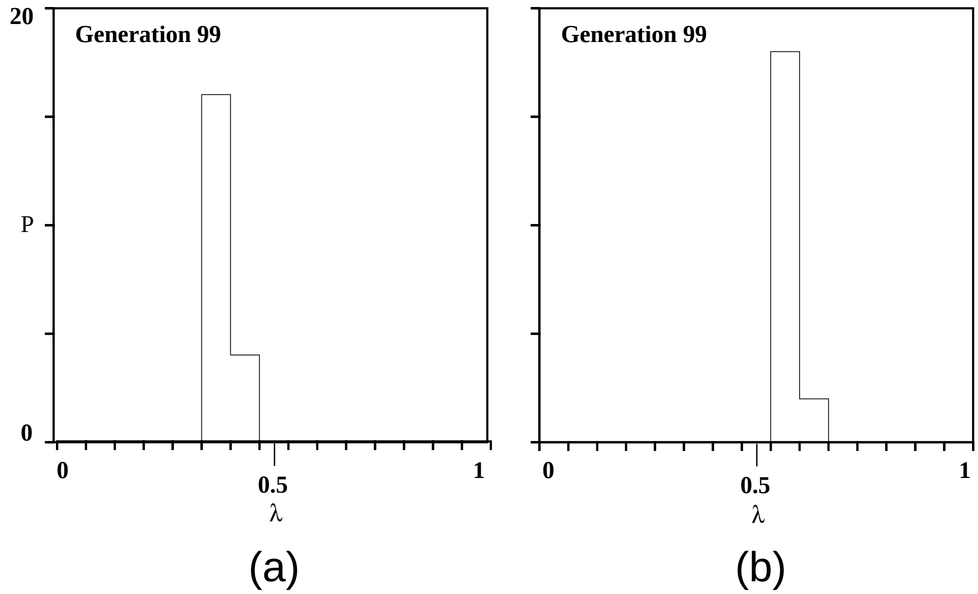


Figure 10: Histograms of the frequency of elite rules as a function of λ for two typical runs. The x-axis is divided into 15 bins of length 0.0667 each. In (a), all the elite rules have $\lambda < 1/2$; in (b), all the elite rules have $\lambda > 1/2$.

are given in Table 2. The figure shows how the structure of the elite populations changes with time. In generation 0, the elite rules are clustered close to $\lambda = 0$ and $\lambda = 1$. Why is this? Recall that in each run, the rules in the initial population are uniformly distributed over $\lambda \in [0.0, 1.0]$. Most of these rules have very low fitness; the best strategies ϕ found are those of Epoch 1 ("always relax to all 0s" or "always relax to all 1s"). These have $F_{100}(\phi) = 0.5$. At generation 0 the rules implementing these strategies have either very low or very high λ—for example, a rule with $\lambda = 0$ maps all neighborhoods to 0 and thus implements the all-0s strategy. This results in the peaks at extreme λ in the initial generation.

Very quickly, however, the elite populations move away from these extremes and towards $\lambda = 1/2$. The populations peak there between generations 5 and 10. The values for T_2 and $T_3 - T_2$ given in the first column of Table 2 indicate that the appearance of the peak roughly corresponds with Epochs 2 and 3. By generation 15 the distribution has changed again—it now has two peaks on either side of $\lambda = 1/2$. By generation 20, these two peaks have grown and the dip at $\lambda = 1/2$ has deepened. After generation 20 or so, the distribution does not change appreciably.

The two peaks on either side of $\lambda = 1/2$ result from merging together the elite rules from runs with low-ρ_0 specialists and runs with high-ρ_0 specialists. Each peak represents rules from runs of one or the other type, as seen in Figure 10. What is seen clearly in Figure 11 is a symmetry breaking on the part of the GA: as we discussed above, the $\rho_c = 1/2$ task requires certain symmetries, in particular, the 0–1 exchange symmetry $\mathcal{F}$ that requires $\lambda = 1/2$ for high performance. The GA breaks this symmetry, producing rules on either side of $\lambda = 1/2$.

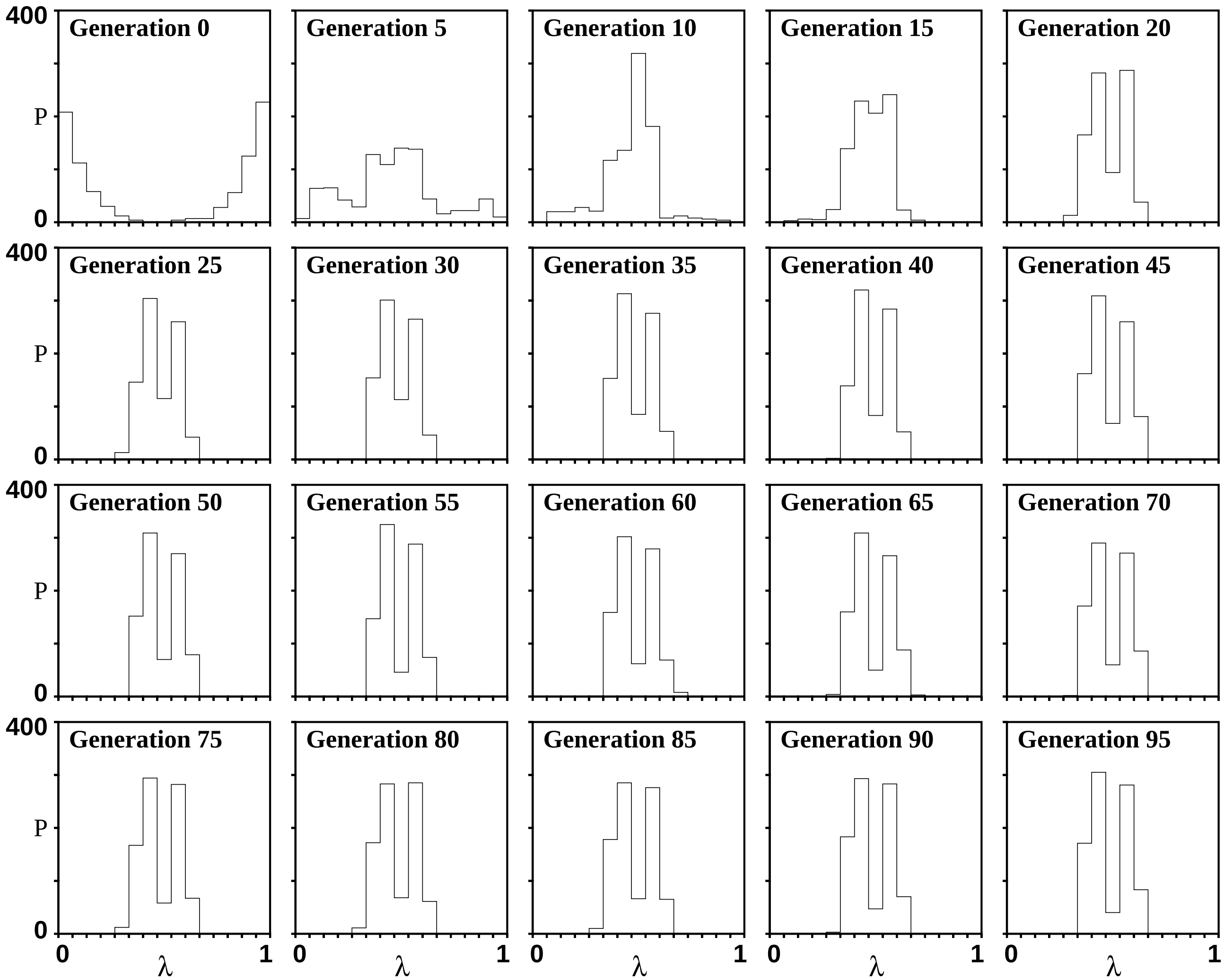


Figure 11: Frequency of elite rules versus λ given every five generations, merged from 44 GA runs with fitness function F_{100}.

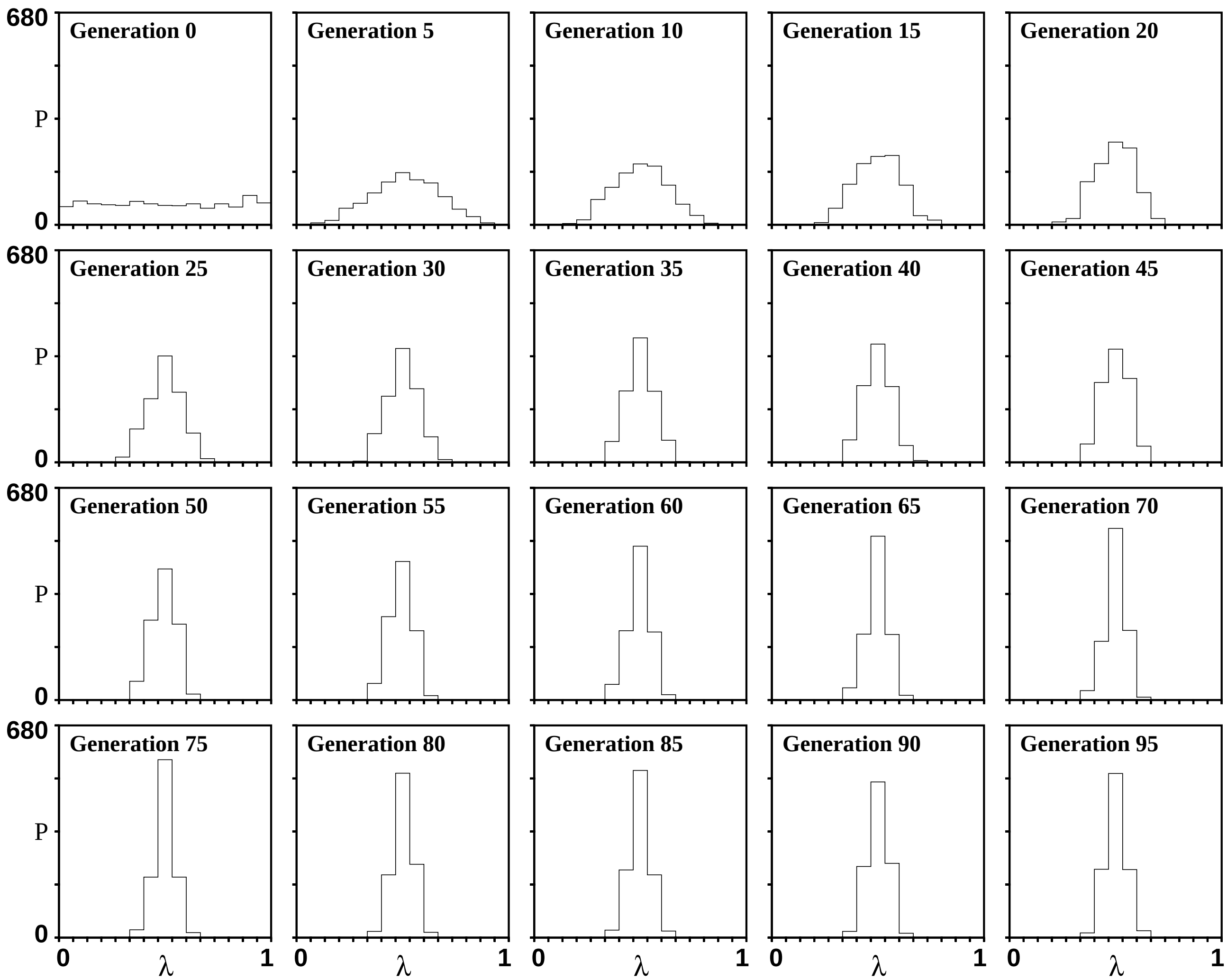


Figure 12: Frequency of elite rules versus λ given every five generations, merged from 50 GA runs with random fitnesses assigned at each generation.

The spatial-reverse symmetry $\mathcal{R}$ is also broken as seen in Figures 5(b), 6(b), 7(a), 8(b), and 8(c). Since this need not lead to a bias in λ's, it is not directly reflected in the histograms we will use here; another coordinate would be more appropriate for monitoring this symmetry breaking.

To understand the degree to which selection for performance fitness rather than intrinsic effects of crossover and mutation cause the effects seen in Figure 11, we performed 50 runs of the GA with *random* selection. Everything about the GA was the same as in the original experiment, except that F_{100} was not calculated and instead at each generation fitnesses were assigned at random. Figure 12 is a mosaic of histograms from these runs. Each histogram plots the frequency of elite rules at the given generation as a function of λ. Since the fitness function is not calculated, all effects seen in Figure 12 are due to the combined intrinsic effects of random selection, crossover, and mutation, which we term "combinatorial drift". As can be seen, by generation 10 the population has largely drifted to the region of $\lambda = 1/2$

and this clustering becomes increasingly pronounced as the run continues.

This drift to $\lambda = 1/2$ is related to the combinatorics of the space of bit strings. For binary CA with neighborhood size n $(= 2r + 1)$, the space consists of all 2^{2^n} binary strings of length 2^n. Denoting the subspace of CAs with a fixed λ and n as $\mathrm{CA}(\lambda, n)$, we see that the size of the subspace is binomially distributed with respect to λ:

$$|\mathrm{CA}(\lambda, n)| = \binom{2^n}{\lambda 2^n}$$

where $|S|$ denotes the size of set S. The distribution is symmetric in λ and tightly peaked about $\lambda = 1/2$ with variance $2^{-n}/4$. Thus, the vast majority of rules is found at $\lambda = 1/2$. Using Sanov's theorem, for example, with $r = 3$ there are about 10^{-16} fewer rules at $\lambda_c \approx 0.146$ [48] than at $\lambda = 1/2$. The steepness of the binomial distribution near its maximum gives an indication of the magnitude of the drift "force". Note that the last histogram in Figure 12 gives the GA's rough approximation of this distribution.

Drift is thus a powerful force moving the population to cluster around $\lambda = 1/2$. It is partially responsible for the initial clustering around $\lambda = 1/2$ seen in Figure 11. However, the distribution in early generations (e.g., generation 10) in Figure 11 is more sharply peaked at $\lambda = 1/2$ than that for the same generation in Figure 12, indicating that there is an additional clustering force due to selection for performance fitness. The striking difference in the two distributions in later generations shows that the symmetry breaking is due to selection forces rather than drift forces.

This completes our overview of the phenomena that were observed in the GA runs. In the remainder of this paper, we answer the following questions:

Major questions:

- How are the strategies in each epoch implemented in the rule tables?
- In what way are the macroscopic properties of the λ distributions presented in Figure 11 related to the four epochs? In particular, what causes the symmetry breaking seen at approximately generation 15 in Figure 11?
- By what mechanisms does the GA produce the behavior that we have observed? In particular, what are the mechanisms underlying the epochs of innovation?
- What impedes the GA from discovering better strategies? In particular, what prevents the GA from discovering the GKL rule or similar rules?

9. Implementation of Strategies

To investigate how the strategies in each epoch are implemented in the rule tables, we will define a new statistic over rule tables, denoted $A_s(d)$, that measures the degree of agreement of output bits with neighborhood densities. Let the density of symbol s in a neighborhood pattern η be denoted by $\rho^s(\eta)$. For example, $\rho^1(0000001) = 1/7$; $\rho^0(0000001) = 6/7$. Let the set of neighborhoods η for which $\rho^s(\eta) \geq d$ be denoted by $\mathcal{N}_s(d)$, where $d \in [0.0, 1.0]$ is some constant, then

$$\mathcal{N}_s(d) = \{\eta : \rho^s(\eta) \geq d\}$$

Note that $|\mathcal{N}_s(d)|$ is monotonically decreasing from 128 to 1 as d varies from 0 to 1. For $k = 2, r = 3$ rules, $\mathcal{N}_1(1/2)$ is the set of 64 neighborhoods with a majority of 1s in the neighborhood pattern, and $\mathcal{N}_1(6/7)$ is the set of eight neighborhoods with at least 6 1s in the neighborhood pattern. (Note that $|\mathcal{N}_1(d)| = |\mathcal{N}_0(d)|$ for all d, by the $0 - 1$ exchange symmetry.)

Then for a given rule table ϕ, consider the set $\mathcal{M}_s(d)$ of neighborhoods $\eta \in \mathcal{N}_s(d)$, that map to output symbol s:

$$\mathcal{M}_s(d) = \{\eta : \eta \in \mathcal{N}_s(d) \text{ and } \phi(\eta) = s\}$$

The "s-agreement" $A_s(d)$ is the fraction of these neighborhoods; that is

$$A_s(d) = \frac{|\mathcal{M}_s(d)|}{|\mathcal{N}_s(d)|}$$

Note that $A_1(0) = \lambda$. $A_1(1/2)$ is the fraction of rule-table neighborhoods with a majority of 1s whose output bits are 1. For the "majority" rule given in Section 2, $A_1(1/2) = 1$, since this is precisely how the rule was defined. For the GKL rule, $A_1(1/2) = 0.75$.

The temporal development of $A_s(1/2)$ and $A_s(6/7)$ for $s = 0$ and $s = 1$ averaged over the elite population helps identify how the different epochs' strategies are implemented. For rules with 7-bit neighborhoods, $|\eta| = 7$, $A_s(1/2)$ measures the degree to which $\phi(\eta)$ agrees with $\rho^s(\eta)$ in neighborhoods that have at least 4 s's ("majority agreement"). Similarly, $A_s(6/7)$ measures the degree to which $\phi(\eta)$ agrees with with $\rho^s(\eta)$ in neighborhoods that have at least 6 s's ("super-majority agreement").

Figure 13 caption: $A_0(d)$ statistics for a run that resulted in low-ρ_0 specialists. (a) Mean and standard deviation of elite 0-agreement $A_0(6/7)$ versus generation. The mean elite fitness $\overline{F}_{\text{elite}}$ is plotted for reference. (b) Mean and standard deviation of elite 0-agreement $A_0(1/2)$ versus generation. Mean elite fitness $\overline{F}_{\text{elite}}$ is plotted for reference. (c) Scatter plot of elite λ values. The generations of takeover of Epochs 2 and 3 are marked by vertical dashed lines.

Figure 13 displays plots for $\{s = 0\}$-agreement and Figure 14 displays plots for $\{s = 1\}$-agreement, for one typical run that resulted in low-ρ_0 specialists. Figures 13(a) and 14(a) plot the mean and standard deviation σ of $A_s(6/7)$ over the elite rules at each generation. For reference, the mean fitness $\overline{F}_{\text{elite}}$ of the elite rules is also plotted. The generations of takeover for Epochs 2 and 3 in the elite are marked by vertical dashed lines. Recall that the generation of takeover for a given epoch is defined as the first generation at which all or almost all elite rules implement the strategy associated with that epoch.

The $A_0(d)$ statistics reveal how the Epoch 1 strategies ("always relax to all 0s") are implemented. In Figure 13(a), $A_0(6/7)$ rises quickly and saturates at 1.0 (with $\sigma = 0.0$) at generation 9. The initial sharp rise does not indicate the onset of a new epoch, since no new strategy is discovered. Rather, the rise is due to the depletion of high-λ, high-ρ_0 specialist rules, as can be seen in Figure 13(c), a scatter plot of the elite λ values at each generation. At generation 0 the λ values are clustered at the two extremes, but they quickly consolidate

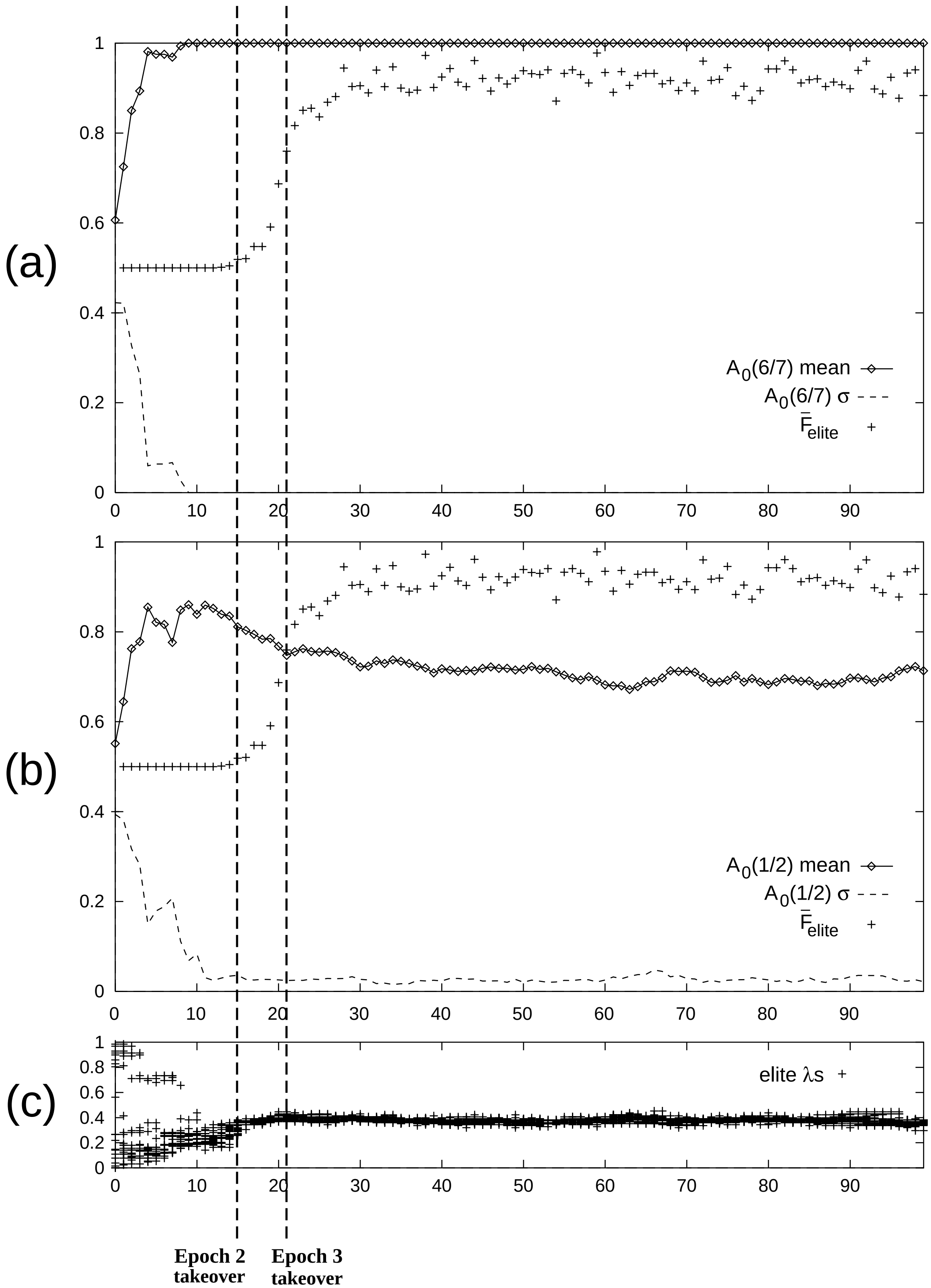


Figure 13: Caption is given in text.

at low values, since the chromosomes with $\lambda = 0$ and $\lambda = 1$, to take the extremes, are destroyed by recombination and mutation without compromising fitness. The population consists entirely of low values by generation 9, at which time $A_0(6/7)$ is essentially saturated at 1.0. This saturation indicates that in all rules in the elite, the eight neighborhood patterns consisting of at least six out of seven 0s map to $s = 0$. This is a necessary condition for implementing the Epoch 1 strategy of "always relax to all 0s".

The saturation of $A_0(6/7)$ marks the takeover of the Epoch 1 strategy. From the onset of Epoch 1 at generation 1 to the takeover at generation 9, the shape of the 0-agreement statistics indicates the rate at which the strategy spreads in the population. As Figures 13(a) and 13(b) show, this occurs without an increase in mean fitness $\overline{F}_{\text{elite}}$. The extreme λ rules die out, but their strategy lives on in later generations.

Figure 13(b) plots the mean and σ of $A_0(1/2)$ over the elite rules at each generation. The mean $A_0(1/2)$ also rises quickly during Epoch 1 and σ sharply decreases, particularly at generation 9, indicating that most neighborhoods with even four and five 0s have 0 as the output bit. In short, the GA implements the "always relax to all 0s" strategy on low-ρ_0 patterns by finding rules with high $A_0(1/2)$. The rise in mean $A_0(1/2)$ corresponds to more and more rules implementing this strategy. The sharp drop in σ corresponds to this consolidation.

The $A_1(d)$ statistics reveal how the Epoch 2 strategies are implemented. Figure 14(a) plots the mean and σ of the 1-agreement $A_1(6/7)$ averaged over the elite rules at each generation of the same run. During Epoch 1, the mean $A_1(6/7)$ is noisy, with σ remaining above 0.15. Four generations after the Epoch 1 takeover, the mean $A_1(6/7)$ begins to rise, corresponding to a slight rise in $\overline{F}_{\text{elite}}$. This is the onset of Epoch 2, at generation 13, which in this run is about three times longer than the average T_2 quoted in Table 2. At the Epoch 2 takeover — at generation 16 — the mean $A_1(6/7)$ makes a sharp jump to become saturated at 1.0 with $\sigma = 0.0$. A similar sharp jump at the Epoch 2 takeover is observed in every run in which Epoch 2 was reached. (In high-ρ_0 specialist runs, the sharp jump is seen in mean $A_0(6/7)$.) What does this sharp jump tell us? Recall that the initial Epoch 2 low-ρ_0 specialists always relax to 0 except on ICs with extremely high density (cf. Figure 5). The GA implements this strategy by finding rules with $A_1(6/7) = 1$. Most neighborhoods in an IC with very high density will thus map to 1, quickly filling the lattice with 1s as the CA is iterated.

Figure 14(b) plots the mean and σ of $A_1(1/2)$ over the elite rules at each generation. Again the mean $A_1(1/2)$ is noisy over most of Epoch 1, though there is a sharp drop in σ at generation 9, again corresponding to the sudden disappearance of high-ρ_0 specialists. But close to the Epoch 2 onset, $A_1(1/2)$ begins to rise—at the same time as, though less sharply than $A_1(6/7)$—and rises significantly at the takeover as σ falls to near 0. (Again, a similar rise was seen in every run.) This rise is partially due to the saturation of $A_1(6/7)$, which reflects a uniform mapping of $\rho^1(\eta) \geq 6/7$ neighborhoods to 1s. But it is also due to the additional mapping of some $1/2 \leq \rho^1(\eta) < 6/7$ neighborhoods to 1s. This rise coincides with a rise in $\overline{F}_{\text{elite}}$. By mapping many of the $1/2 \leq \rho^1(\eta) < 6/7$ neighborhoods to 1s, the GA is discovering rules that are doing an increasingly good job of implementing the Epoch 2 strategies. That is, it is finding rules that correctly classify an increasing number of high-density ICs, and thus obtain increasingly higher fitness.

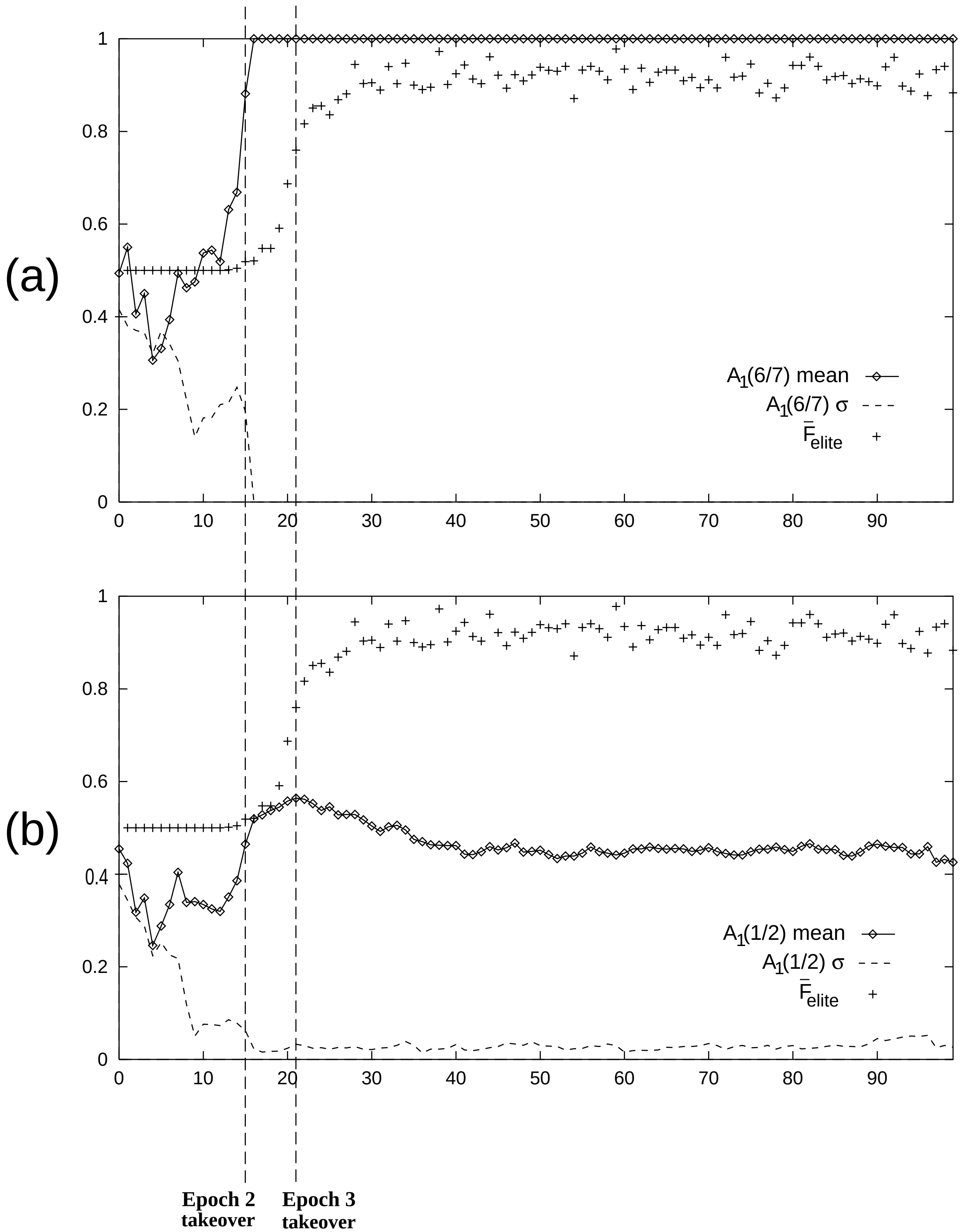


Figure 14: $A_1(d)$ statistics for the same run that was represented in Figure 13. (a) Mean and standard deviation of elite 1-agreement $A_1(6/7)$ versus generation. Mean elite fitness $\overline{F}_{\mathrm{elite}}$ is plotted for reference. (b) Mean and standard deviation of elite 1-agreement $A_1(1/2)$ versus generation. Mean elite fitness $\overline{F}_{\mathrm{elite}}$ is plotted for reference. The generations of takeover of Epochs 2 and 3 are marked by vertical dashed lines.

At the Epoch 3 takeover, the mean $A_1(1/2)$ begins to fall, and continues to fall appreciably for several generations. (A similar fall was seen in every run.) Note that this is different from the behavior of the mean $A_1(6/7)$, which remains saturated at 1.0. The reason for this decrease is the following. The elite rules at Epoch 3 implement a wholly new strategy—relaxing to all 0s by default but expanding sufficiently large blocks of 1s if they are present in the IC. Once $A_1(6/7) = 1.0$, implementing the block-expanding strategy only requires setting a few additional neighborhoods to 1s—this does not affect $A_1(1/2)$ appreciably. This block-expanding strategy accomplishes the same thing as the Epoch 2 trick of increasing $A_1(1/2)$, but in a different way. In the later part of Epoch 2, the elite rules correctly classify many high-density ICs because $\rho^1(\eta) \geq 1/2$ neighborhoods map to 1. However, a high-density IC is very likely to contain at least one, if not many, sufficiently large blocks of 1s, and so the Epoch 3 rules do not need this trick of immediately mapping most of the $\rho^1(\eta) \geq 1/2$ neighborhoods to 1—they can rely on expanding blocks to do the job instead. Thus many of the $\rho^1(\eta) \geq 1/2$ neighborhoods are not required have output 1s for high fitness, so under mutation, some of these 1s drift to 0s. The latter also enhances fitness since it reduces the creation of spurious 1-blocks as shown in Figure 8(c).

Let us summarize this section briefly. For runs resulting in low-ρ_0 specialists, Epoch 1 strategies ("relax to all 0s") are implemented by mapping almost all $\rho^0(\eta) \geq 1/2$ neighborhoods to 0. Epoch 2 strategies ("relax to all 0s, unless the IC has very high density, in which case relax to all 1s") are implemented by mapping many of $\rho^1(\eta) \geq 1/2$ neighborhoods to 1s. The more such mappings, the more high-density ICs will be correctly classified. Epoch 3 strategies ("relax to all 0s, unless the IC contains a sufficiently large block of 1s, in which case expand it") are implemented, once all the $\rho^1(\eta) \geq 6/7$ neighborhoods map to 1s, by mapping a small number of specific neighborhoods to 1s.

Which bits in the rule table need to be set in order to expand 1-blocks? This can be determined by direct enumeration. To expand a 1-block in both directions at equal velocities in a sea of 0s, for example, a $\ldots 111000 \ldots$ wall must be propagated to the right and a $\ldots 000111 \ldots$ wall must be propagated to the left. (Note that walls can be more complicated than this, as seen in Figures 5(b), 6(b), and 7(a), for example.) The neighborhoods which participate in this are those patterns of length $2r + 1$ that contain one or both types of wall. The required output bit for each such neighborhood is simply read off the space-time diagram from the cell below the pattern's center at the next time step. From this it can be seen that a bi-directional expansion of 1-blocks of length greater than the neighborhood size requires 14 bits in the chromosome to be properly set. Presumably, these bits or similar constellations that support the observed strategies are set during Epochs 2 and become fixed in Epochs 3 and 4. In light of this, a better statistic for Epoch 3 would be based not on $A_1(6/7)$ but on the appearance of the constellations of output bits supporting walls that expand blocks.

In any case, the discovery of a strategy to expand s-blocks relaxes the constraints on many of the $\rho^1(\eta) \geq 1/2$ neighborhoods that were set to 1 in Epoch 2; many of these drift back to 0, possibly reducing the tendency to create spurious blocks.

This account of how strategies are implemented applies to runs that evolve low-ρ_0 specialists. A similar account applies to runs that evolve high-ρ_0 specialists with the roles of 0 and 1 reversed.

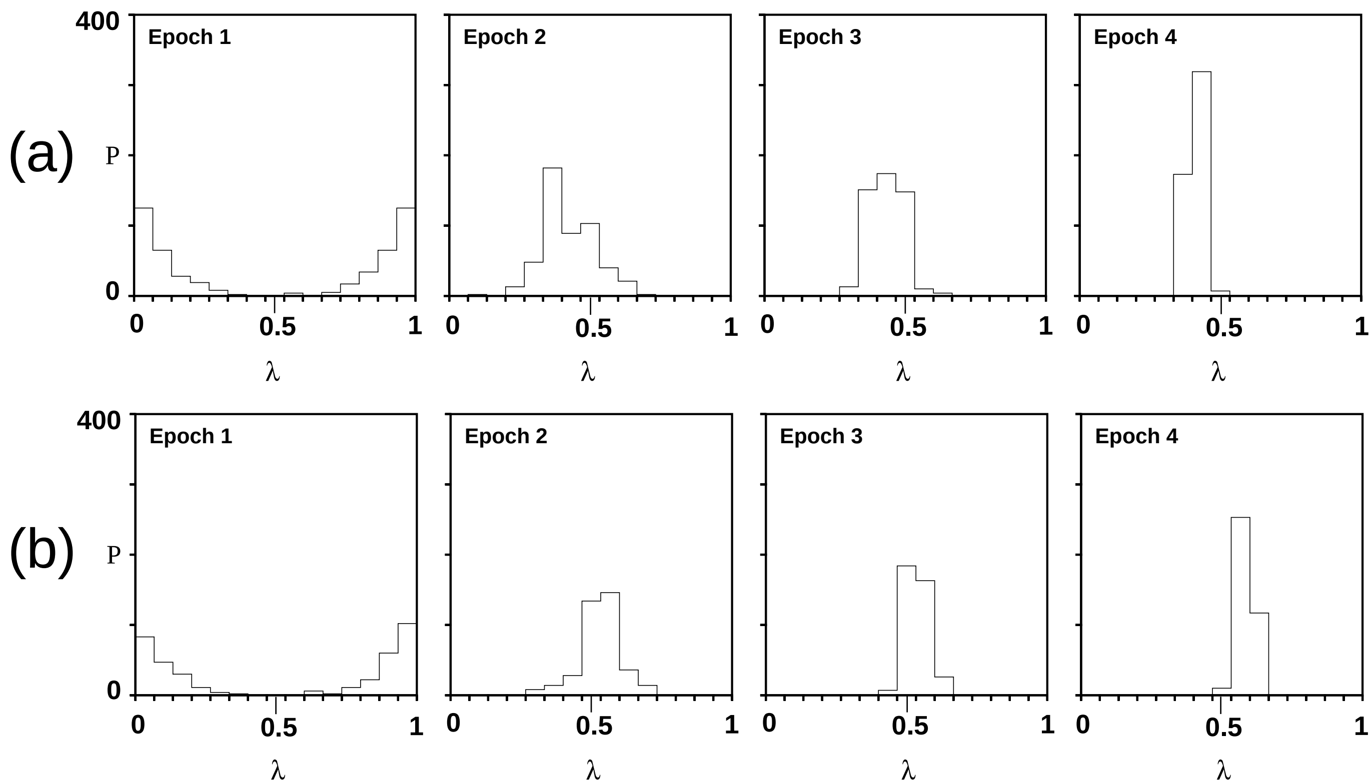


Figure 15: (a) Histograms of elite rule frequency versus λ at each epoch, with rules merged from 25 runs that evolved low-ρ_0 specialists. (b) Histograms of elite rule frequency versus λ at each epoch, with rules merged from 19 runs that evolved high-ρ_0 specialists.

10. Epochs and λ Distributions

We have now described in some detail how strategies at different epochs are implemented in the rule table. This answers the first of our major questions. This understanding helps us to answer the second question: In what way are the macroscopic properties of the λ distributions presented in Figure 11 related to the four epochs? In particular, what causes the symmetry breaking seen in Figure 11?

Figure 15 gives two sets of histograms similar to those in Figure 11. Each histogram in Figure 15(a) represents elite rules merged from 25 runs that evolved low-ρ_0 specialists. The "Epoch 1" histogram plots the elite rules from each run at generation 0. The "Epoch 2" histogram plots the elite rules from each run at the generation of Epoch 2 takeover—defined, as above, as the first generation at which all or almost all the elite rules are implementing Epoch 2 strategies. This generation is different for each run, so the rules represented in this histogram are from different generations on different runs. In short, the runs are lined up with respect to epoch's generation of takeover. The "Epoch 3" histogram plots rules at the generation of Epoch 3 takeover in each run, and the "Epoch 4" histogram plots the elite rules at generation 99 in each run. Figure 15(b) gives the same histograms for 19 runs that evolved high-ρ_0 specialists. (These are the same $19 + 25 = 44$ runs for which statistics are given in the first column of Table 2.)

Both Epoch 1 histograms show most elite rules to be clustered at very low and very high λ values. As was noted above, these are the rules that are selected in the first generation because they are the ones that initially implement the Epoch 1 strategies.

Both Epoch 2 histograms show the elite population clustered much closer to $\lambda = 1/2$—on the low side in 15(a) and on the high side in 15(b). This movement of the elite population towards $\lambda = 1/2$ has two sources. The first is combinatorial drift, which moves even Epoch 1 rules closer to $\lambda = 1/2$. The second is the selection of rules implementing Epoch 2 strategies. In runs that evolve low-ρ_0 specialists, most Epoch 1 rules have low λ (e.g., see Figure 13(c)). The innovation at Epoch 2 is to increase the number of $\rho^1(\eta) \geq 1/2$ neighborhoods with output bit 1. These two trends result in an increase in λ. The opposite is true for runs that evolve high-ρ_0 specialists. In both cases, the result is to move closer to $\lambda = 1/2$.

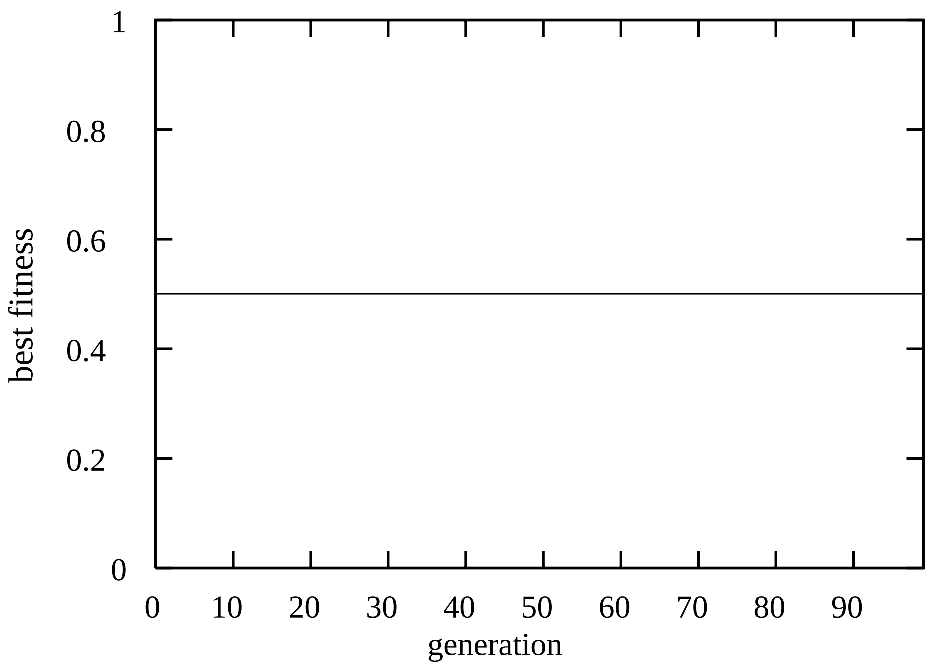
Both Epoch 3 histograms show even narrower distributions, now close to being peaked at $\lambda = 1/2$. As was said above, the Epoch 3 block-expanding strategies require only relatively few bits to be set in the rule table, so the discovery of these strategies does not appreciably change the λ distribution. (Generally, as epoch 3 is reached the utility of λ declines as an information projective coordinate for monitoring changes in the population structure.) Drift continues to move rules closer to $\lambda = 1/2$, and rules implementing Epoch 3 strategies can be found at these λ values. However, as was seen in Figure 8, these rules make a number of errors, such as expanding blocks that are too small, or creating and expanding blocks that were not in the IC. The GA can correct such errors without destroying the new strategy by setting bits so as to increase the minimum block size required for expansion (Correction A), and by ensuring that if there are no sufficiently large blocks present in the IC, that the CA very quickly relaxes to the default fixed-point configuration (Correction B). For low-ρ_0 specialists, both these corrections require mapping more neighborhoods to 0s. This is a way to ensure that the all-0s fixed point is reached quickly on ICs without sufficiently large blocks. For high-ρ_0 specialists, they require mapping more neighborhoods to 1s. For low-ρ_0 specialists, the corrections increase λ; for high-ρ_0 specialists, they decrease it. This is seen in the Epoch 4 histograms: the low-ρ_0-specialist runs are now clustered below $\lambda = 1/2$ and the high-ρ_0-specialist runs are now clustered above $\lambda = 1/2$. Thus, the symmetry breaking in Epoch 3 results from improvements in the block-expanding strategies. The result is clearly seen in Epoch 4 where the $\lambda = 1/2$ rules are largely suppressed.

11. GA Mechanisms of Innovation

We have now answered the first two of our major questions. In this section we address the third: What GA mechanisms underlie the epochs of innovation? In particular, we investigate the roles of crossover and mutation in producing the behavior that we have observed.

To better understand the role of crossover, we performed a set of 50 GA runs with the same parameter values as were described in Section 6, but with crossover turned off. In these runs, the new 80 rules at each generation were created from the 20 elite rules by mutation only—pairs of parents were chosen at random from the elite as before, but no crossover was performed and each offspring was a copy of its parent with exactly two mutations.

(a)



(b)

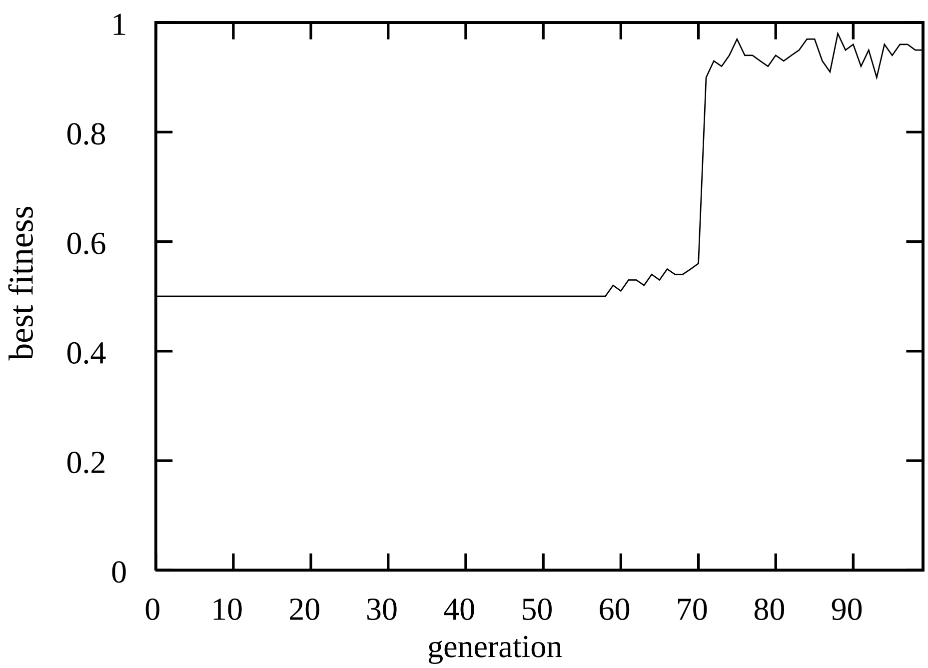


Figure 16: Best fitness versus generation for two of the 50 runs performed with no crossover.

Figure 16 displays the best fitness at each generation for two of the runs without crossover. In the run displayed in Figure 16(a), the GA never found a rule with fitness greater than 0.5. This occurred in 37 out of the 50 runs, compared with 4 out of 50 runs when crossover was turned on. These statistics are given in Table 2.

The other 13 runs were similar to Figure 16(b). More detailed examination of these runs showed that the GA made the same progression through strategy epochs as in the runs with crossover, but the onset of Epoch 2 was, on average, much later. However, once Epoch 2 rules were discovered, the GA moved on to Epoch 3 rules very quickly. The first two columns of Table 2 compare these times for the 44 runs with crossover and for the 13 runs with no crossover that reached Epoch 3.

Crossover clearly plays a role in speeding up the onset of Epoch 2. However, its role in the move from Epoch 2 to Epoch 3 is much less pronounced. The analysis we gave above of how Epoch 2 strategies are implemented in the rule tables helps to explain why. Consider, for example, a run that evolves low-ρ_0 specialists. To get to Epoch 2, the GA must discover a low-λ rule with $A_1(6/7) = 1$. The lexicographic ordering of neighborhoods in the rule-table chromosome happens to allow single-point crossover to create such a rule in one time step. This is because, under our encoding, most of the eight $\rho^1(\eta) \geq 6/7$ neighborhoods are at the extreme "right-hand" side of the chromosome. A crossover between an Epoch 1 low-λ rule and an Epoch 1 high-λ rule thus has a fair chance of yielding an Epoch 2 rule. And, since low and high λ rules are in the initial population to begin with, it does not take much time to discover an Epoch 2 rule when crossover is in effect. However, when crossover is turned off, the GA must rely on mutation alone to set the $\rho^1(\eta) \geq 6/7$ neighborhood bits correctly. The waiting time for this is reflected in the "GA, no xover" statistics given in Table 2. In 37 out of 50 runs, the no-crossover waiting time was greater than 99 generations.

Once an Epoch 2 rule is discovered, a small number of mutations can turn it into an Epoch 3 rule. This is seen in the $T_3 - T_2$ statistics given in Table 2. The mean length of Epoch 2 is small for both the crossover and no-crossover runs. Thus, mutation alone suffices to quickly move to Epochs 3 and 4 and to discover the associated strategies. Crossover does not play a large role, though it does appear to shorten the times.

We performed an additional experiment without crossover in which, for each run, the initial population was not uniformly distributed over $\lambda \in [0.0, 1.0]$, but rather each initial rule had $\lambda \approx 1/2$ ("GA, no xover, initpop 1/2"). Our hypothesis was that there would be more rules in the initial population with, say, low λ but high $A_1(6/7)$, and thus the rules in the population would be closer than in the original no-crossover experiment to the conditions necessary for the discovery of Epoch 2 strategies. The results, given in column 3 of Table 2 supported this hypothesis: the number of runs reaching Epoch 3 and the mean values for T_2 and $T_3 - T_2$ were intermediate between those measured in the crossover and no-crossover experiments. In a sense, the original uniform-λ initial population is responsible for a long transient; it substantially slows down the GA.

We performed a final experiment in which we used a simple stochastic hill climbing method instead of a GA to search the space of rules. A run of this method is the following. A bit string is chosen at random and its fitness is evaluated. A random bit is flipped, and if the new fitness is equal to or higher than the original fitness, the mutation is retained; if not, the original string is retained. This process continues for $9,900$ total evaluations—

the same total number of evaluations as performed in one run of the GA. We performed 22 such runs, each with a different random-number seed. The purpose of this experiment was, again, to test the hypothesis that crossover confers an advantage for reaching Epoch 2. The results of 22 runs of the hill climbing method, given in column 4 of Table 2, further support this hypothesis. Under Monte Carlo search Epoch 2 was reached in only 8 out of 22 runs and T_2 (given here in generations, where each generation equals 100 fitness evaluations) is close to that of the "GA, no xover, initpop 1/2" experiment. It is interesting that the duration of Epoch 2 with stochastic hill climbing search is about half that with the $\lambda \approx 1/2$ initial-population runs and closer to that of the basic experiment ("GA, xover" in Table 2).

12. GA Impediments

With this understanding of the GA's behavior on the $\rho_c = 1/2$ task, we now can address the last of our major questions: What impedes the GA from discovering better strategies? In particular, what prevents the GA from discovering the GKL rule or similar rules? Here we list a number of impediments that are, or might be, faced by our GA on this problem. We will discuss their relevance to GAs in general and propose ways in which they could be overcome. Perhaps surprisingly, most of the impediments we identify are also forces that help the GA in the initial stages of its search. What is needed is a theory of how the costs and benefits of these various forces trade off. Such a theory would allow for active monitoring of the change from beneficial to harmful effects.

Symmetry breaking

A primary impediment is the GA's tendency to break the task's symmetries by producing low-ρ_0 or high-ρ_0 specialists. A pressure towards symmetry breaking is effectively built into our fitness function, since specializing on one half of the ICs is an easy way to obtain a higher fitness than that of a random rule. This kind of symmetry breaking occurs in generation 0 with the selection of the two types of Epoch 1 rules and, in subsequent generations, the entire elite population naturally drifts into one or the other specialist "camps". The Epoch 2 and Epoch 3 strategies are simply elaborations of these original symmetry-broken Epoch 1 strategies. Symmetry breaking thus produces a short-term gain for the GA, but later prevents it from making improvements beyond Epoch 4 strategies, as the long periods of stasis seen in Figures 3, 11, 13, and 14 over generations 20 to 99. We hypothesize that this propensity to break symmetries for short-term gain is a general feature of GAs and even of natural evolution. This implies that when one wants to apply a GA to a particular problem, one should first determine all the relevant symmetries in the optimization, and then restrict the GA's search space to candidate solutions with those symmetries. This can be done either by having the fitness function penalize asymmetric candidate solutions or by building the desired symmetries into the representation. This is akin to the general problem of using domain knowledge to assist the GA's search (e.g., see [19]). We plan to investigate the effect of the latter approach on the GA's performance on the $\rho_c = 1/2$ task. As pointed out in our description of $\mathbf{T}_{1/2}$, the task has a number of symmetries in addition to the ρ_0 symmetry that requires $\lambda = 1/2$ for high performance. One possible problem in imposing symmetries on the GA, though, is that this could make innovation substantially more difficult to achieve. In other words, it could be the case that broken-symmetry solutions can lead to symmetry-respecting ones more quickly than in a symmetry-restricted chromosome space. This is exactly what has not happened in our experiments, however.

Drift

A second possible impediment is the force due to combinatorial drift. As was seen in Figure 12, the intrinsic effects of crossover and mutation, apart from selection, produce a strong drift force moving the population close to $\lambda = 1/2$. This stochastic drift is the force that produces conditions necessary for Epoch 2 strategies to be discovered. For example, it creates low-ρ_0 specialists with higher λs, in some cases creating low-ρ_0 specialists with $A_1(6/7) = 1$. However, later in the run drift also restricts the GA's search to one part of the chromosome space. In the absence of strong selection, it is difficult for the GA to maintain candidate solutions far away from $\lambda = 1/2$. This may present a problem for some GA applications, though not necessarily for the $\rho_c = 1/2$ task. The force due to drift is something GA practitioners should take into account when designing a GA for a particular application, and it may be necessary to design operators to counteract this force.

$\rho_c = 1/2$ fitness landscape

A third possible impediment is the effective fitness landscape of the $\rho_c = 1/2$ task. We have seen that there is a ready path for the GA to take from the easily discovered Epoch 1 rules to Epoch 4 rules—almost every run of the GA follows this path by generation 20 or so. Following this path leads to one or the other of two relatively high-fitness "potential wells"—to make a physical analogy—by a breaking of symmetries. But if the GA could avoid this symmetry-broken potential, is there another readily accessible path that the GA could follow to discover GKL-rule-like behavior?

We performed some preliminary experiments that indicate that such a path could exist. We ran the GA on populations of mutants of the GKL rule and found that many different rules have GKL-like behavior, using signals such as those described in Section 5 to classify IC density. Such rules were found at Hamming distances of up to 30 bits or more from the GKL rule. They had $F_{10^4} \approx 0.96$ and thus indicate an intermediate fitness plateau between that of the Epoch 4 rules with maximum $F_{10^4} \approx 0.945$ and that of the GKL rule with $F_{10^4} \approx 0.972$. Further investigations of the landscape around the GKL rule will be reported in future work.

Finally, a more detailed analysis using F_{10^4} of the generation-99 populations revealed that a fairly sophisticated rule had been evolved in one run. This rule had $F_{10^4} \approx 0.945$, whereas most Epoch 4 rules had $F_{10^4} < 0.930$. This rule exhibited signaling mechanisms similar to that of the GKL ($F_{10^4} \approx 0.972$) and the mutants mentioned above. More details of this rule's structure will be reported elsewhere. But it does suggest the existence of yet another fitness plateau between Epoch 4 and the GKL CA.

Stochastic nature of F_{100}

A fourth impediment is the stochastic nature of F_{100}. The small sample of ICs used to compute fitness limits the resolution available to the GA for distinguishing among competing rules. This limited resolution obscures differences in fitness that might be significant. This was observed in our experiments with mutations of the GKL rule mentioned above. Even when the initial population consisted of rules that were each one bit different from the GKL rule, the GA did not rediscover and retain the GKL rule because F_{100} could not reliably distinguish the mutated rules with $F_{10^4} \approx 0.96$ from the GKL rule with $F_{10^4} \approx 0.972$. This problem of low resolution could be solved—at considerable computational cost—by using a much enlarged sample of ICs. An intermediate solution would be for the GA to retain and use accumulated fitness information for individuals over many generations; for example, it could keep a running fitness average for rules that survive. This differs from the present method which discards fitness information from previous generations, determining the elite rules only from the fitnesses calculated on the given generation. More experiments need to be performed to determine what level of fitness resolution is needed to obtain improved performance.

Structure of IC sample

A fifth impediment is presented by the structure of the IC sample chosen at each generation. Our current method is to choose a sample uniformly distributed over $\rho_0 \in [0.0, 1.0]$, with exactly half the sample having $\rho_0 < \rho_c$ and exactly half having $\rho_0 > \rho_c$. This distribution was meant to present some "easy-to-classify" extreme-ρ_0 ICs to the evolving rules in order to allow evolution to get off the ground. However, aside from the above-mentioned pressure towards very early symmetry breaking arising from this distribution, there is another impediment. After a small number of generations, this IC distribution does not present a sufficient challenge to the evolving rules. For example, all Epoch 2 rules correctly classify half the distribution in addition to the extreme-ρ_0 cases of the other half. This means that the fitness differences among Epoch 2 rules are being judged on the basis of the remaining ICs—less than half of the original 100—which exacerbates the fitness-resolution problems discussed above. That is, the reduction in the fraction of informative test ICs reduces the number of useful fitness evaluations and so increases the variance in the mean fitness. Epoch 4 rules, for example, routinely achieve 100% correct classification on some set of ICs during a run of the GA, whereas under F_{10^4}, they never reach fitnesses above ≈ 0.95. One possible solution to this problem is to co-evolve a population of IC samples with the population of CA rules, with the fitness of an IC sample being inversely related to the classification performance of the current CA population on this sample. In principle, such co-evolution—analogous to biological "arms races" seen in nature—should produce sets of ICs that are tuned expressly to present challenges to rules in the current population. In this situation, the absolute meaning of the fitness function F_{100} would change over the generations. This approach should help alleviate the problem of fitness accuracy without requiring computationally intractable sample sizes. Such a co-evolutionary approach has been studied in the context of using GAs to discover efficient sorting networks [39]. Another alternative would be to use modern statistical evaluation methods that make more efficient use of the available fitness evaluations.

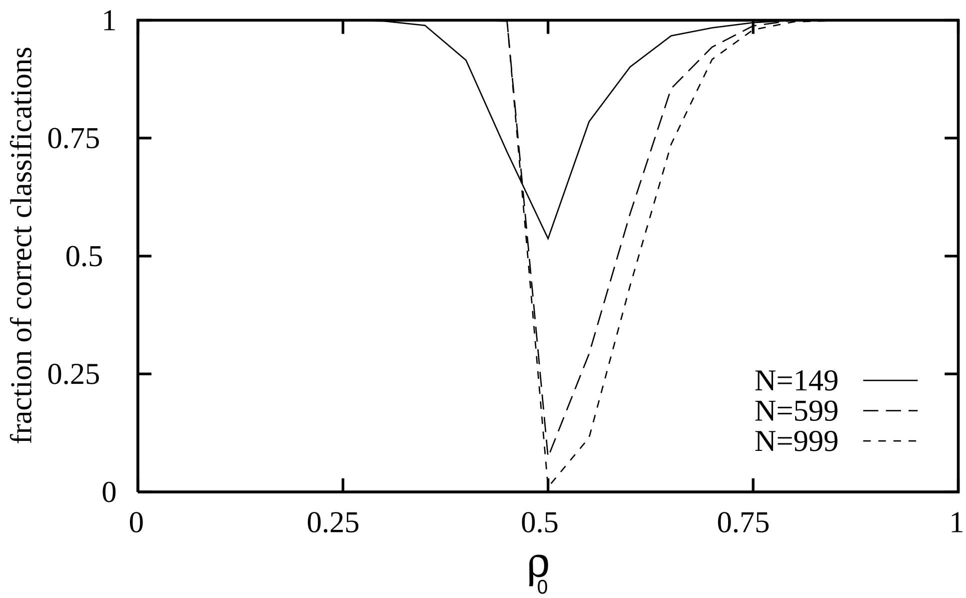


Figure 17: Performance of an Epoch 4 rule (the same one whose performance was plotted in Figure 9) plotted as a function of ρ_0. Performance plots are given for three lattice sizes: 149, 599, and 999. This rule has $\lambda \approx 0.59$.

Fixed lattice size

A sixth impediment, due to our particular method, is the restriction of fitness evaluation to a fixed lattice size—here, $N = 149$. As was shown in Figure 2, the GKL rule's classification performance improves as lattice size increases. The opposite is true of the fittest evolved rules in our experiments. The performance of one Epoch 4 rule as a function of ρ_0 is plotted in Figure 17 for lattice sizes of $N = 149$, 599, and 999. (This is the same rule whose performance was plotted in Figure 9.) This rule has $\lambda \approx 0.59$; it increases sufficiently large blocks of adjacent or nearly adjacent 0s. We used the same procedure to make these plots as was described for Figure 2. As can be seen, the performance according to this measure is not only significantly worse than that of the GKL rule on $N = 149$ lattices, but also decreases dramatically for larger N. The worst performances for $N = 599$ and $N = 999$ are centered slightly above $\rho_0 = \rho_c$. (Since we used only odd N, the actual ρ_0s plotted at 0.5 are slightly above 0.5.) With $\rho_0 > \rho_c$, the CA should relax to a fixed point of all 1s. Detailed inspection, however, revealed that on almost every IC with ρ_0 slightly above ρ_c, the CA is relaxing to a fixed point of all 0s. This is a result of this rule's strategy of expanding "sufficiently large" blocks of 0s. The appropriate block size b to expand was evolved to be a good predictor of ρ_0 for $N = 149$. With larger lattices the probability of b-length 0-blocks in ICs with $\rho_0 > \rho_c$ increases. And so the closer high ρ_0s are to ρ_c, the more likely such blocks are to occur. In the CA we tested with $N = 599$ and $N = 999$, such blocks occurred in most ICs with ρ_0 slightly above ρ_c, always leading to incorrect classifications. This shows that keeping the lattice size fixed during GA evolution can lead to overfitting for the particular lattice size. We plan to experiment with varying the lattice size during evolution in an attempt to prevent such overfitting.

Representation

A seventh impediment is the lexicographic bit-string representation used for the CA rules. This ordering has the initially beneficial effect of grouping together most of the $\rho^s(\eta) \geq 6/7$ neighborhoods, with $s = 0$ neighborhoods at the "left" extreme and $s = 1$ neighborhoods at the "right" extreme. As pointed out above, this ordering enables crossover to quickly produce Epoch 2 rules—low-ρ_0 specialists with $A_1(6/7) = 1$ or high-ρ_0 specialists with $A_0(6/7) = 1$. However, the lexicographic ordering of output bits may hinder the GA's progress in later generations, since to produce the kinds of coordinated signals used in the GKL rule, a number of neighborhood output bits must work in concert. These co-active neighborhoods are unlikely to be adjacent in a lexicographic ordering, and thus cannot be moved together from a parent to an offspring via simple crossover. More disruptive crossover operators such as uniform crossover [83] run the risk of destroying the necessary structures. The problem of designing a representation that will work well with genetic operators is a general one for GAs. One solution that has been explored is adapting the representation to suit the operators (e.g., see [32]). Phenomena in natural genetics such as inversion and jumping genes may be a form of representation adaptation. These have inspired some work in GAs along these lines (e.g., see [31, 41]).

Loss of diversity

Finally, an eighth impediment is the loss of diversity over time in the population. When a new strategy is discovered, it sweeps through the population and quickly all the elite rules are representatives of that strategy. This convergence aids the rapid moves from Epoch 2 to Epoch 3 to Epoch 4. However, convergence, like drift, limits the region of chromosome space that the GA is searching. Controlling convergence in GAs has been the subject of much research. (See [31] for a review of work in this area.) In our experiments with $E = 20$, diversity (measured as the mean pairwise Hamming distance in the elite) falls quickly—more rapidly, in fact, than it did in our previous experiments with $E = 50$. However, the smaller E also sped up the onsets of the different epochs, since the newly discovered strategies were able to invade the population more quickly. We also performed experiments in which a minimum level of diversity was explicitly maintained. This scheme did not yield improved performance [60]. But in spite of these results, it may be the case that the rapid decrease in diversity is an impediment for moving beyond Epoch 4 strategies. The need to balance the level of population diversity with the need to quickly propagate newly discovered innovations to the rest of the population is discussed in several places in GA literature (e.g., see [59]).

As was noted above, most of these impediments are also forces that help the GA in the initial stages of its search. None are specific to the $\rho_c = 1/2$ task or even to the problem of evolving CA. Rather, they are general issues in any GA application, and some of them are relevant to any machine-learning method. In this work our analysis tools have enabled us to observe some of these forces (e.g., symmetry breaking) quite clearly, and to study them carefully. Going beyond this to develop a predictive theory of the tradeoffs these forces produce in GA efficiency is one of our long-term objectives.

13. Conclusion

As was said in the introduction, the goals of our research are (i) to better understand the ways in which CAs can perform computations; (ii) to learn how to best use GAs to evolve computationally useful CAs and (iii) to understand the mechanisms by which GAs can produce complex and innovative behavior in systems with simple components and local interactions.

This paper has reported progress on these goals obtained by analyzing in detail a GA's behavior on evolving CAs to perform a particular computation: $\rho_c = 1/2$ density classification. We analyzed the strategy of the GKL rule for performing this task, and used it as a benchmark with which to compare the rules evolved by the GA. We have described the epochs of innovation in each run of the GA and the strategies corresponding to these epochs, and have understood in detail how these strategies are implemented and how these epochs manifest themselves in large-scale population structures. We then explained the respective roles of crossover and mutation in the discovery of new strategies and identified several impediments for the GA in achieving higher computational capability in CA. Primary among the impediments is the GA's breaking of task symmetries in the pursuit of short-term gains in fitness. We believe that this type of detailed analysis is essential in order to understand and improve the GA's behavior and to develop predictive theories of the tradeoffs among different evolutionary forces. The results are relevant to the application of GAs in general, and they point the way to a more general analysis of the evolutionary forces we have identified. This work is also a first step in developing methods for automatic programming of CAs and other spatially-distributed parallel computers. Success in this area should have significance for the field of parallel computation and for nonlinear spatial modeling.

Acknowledgments

We thank Jonathan Amsterdam, Rajarshi Das, Jim Hanson, and Dan Upper for helpful comments on an earlier draft of this paper. 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.

References

[1] D. H. Ackley and M. L. Littman. Interactions between learning and evolution. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial Life II*, pages 487–507, Reading, MA, 1992. Addison-Wesley.

[2] D. H. Ackley and M. L. Littman. A case for Lamarckian evolution. In C. G. Langton, editor, *Artificial Life III*, Reading, MA, 1993. Addison-Wesley.

[3] J. Andreoni and J. H. Miller. Auctions with adaptive artificial agents. Technical Report 91-01-004, Santa Fe Institute, Santa Fe, New Mexico 87501, 1991.

[4] J. C. Bean. Genetics and random keys for sequencing and optimization. Technical Report 92-43, Dept. of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109, 1992.

[5] M. A. Bedau and N. H. Packard. Measurement of evolutionary activity, teleology, and life. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial Life II*, pages 431–461, Reading, MA, 1992. Addison-Wesley.

[6] M. A. Bedau, F. Ronneburg, and M. Zwick. Dynamics of diversity in an evolving population. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature 2*, pages 95–104, Amsterdam, 1992. North Holland.

[7] R. K. Belew. Evolution, learning, and culture: Computational metaphors for adaptive algorithms. *Complex Systems*, 4:11–49, 1990.

[8] R. K. Belew, J. McInerney, and N. N. Schraudolph. Evolving networks: Using the genetic algorithm with connectionist learning. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial Life II*, Santa Fe Institute Studies in the Sciences of Complexity, pages 511–547, Reading, MA, 1992. Addison-Wesley.

[9] A. Bergman and M. W. Feldman. More on selection for and against recombination. *Theoretical Population Biology*, 38(1):68–92, 1990.

[10] M. F. Bramlette and E. E. Bouchard. Genetic algorithms in parametric design of aircraft. In L. D. Davis, editor, *Handbook of Genetic Algorithms*, pages 109–123. Van Nostrand Reinhold, 1991.

[11] J. Campbell, B. Ermentrout, and G. Oster. A model for mollusk shell patterns based on neural activity. *The Veliger*, 28:369, 1986.

[12] D. J. Chalmers. The evolution of learning: An experiment in genetic connectionism. In D. S. Touretzky et al., editor, *Proceedings of the 1990 Connectionist Models Summer School*, San Mateo, CA, 1990. Morgan Kaufmann.

[13] R. J. Collins and D. R. Jefferson. Selection in massively parallel genetic algorithms. In R. K. Belew and L. B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 249–256, San Mateo, CA, 1991. Morgan Kaufmann.

[14] R. J. Collins and D. R. Jefferson. The evolution of sexual selection and female choice. In F. J. Varela and P. Bourgine, editors, *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, pages 327–336, Cambridge, MA, 1992. MIT Press/Bradford Books.

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

[16] J. P. Crutchfield and J. E. Hanson. Turbulent pattern bases for cellular automata. *Physica D*, in press, December 1993. Santa Fe Institute Report SFI-93-03-010.

[17] T. Dandekar and P. Argos. Potential of genetic algorithms in protein folding and protein engineering simulations. *Protein Engineering*, 5(7):637–645, 1992.

[18] Y. Davidor. *Genetic algorithms and robotics*. Robotics and Automated Systems. World Scientific, Singapore, 1991.

[19] L. D. Davis, editor. *The Handbook of Genetic Algorithms*. Van Nostrand Reinhold, 1991.

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

[21] D. d'Humières, P. Lallemand, J. P. Boon, D. Dab, and A. Noullez. Fluid dynamics with lattice gases. In R. Livi, S. Ruffo, S. Ciliberto, and M. Buiatti, editors, *Workshop on Chaos and Complexity*, pages 278–301, Singapore, 1988. World Scientific.

[22] M. Dorigo and E. Sirtori. Alecsys: A parallel laboratory for learning classifier systems. In R. K. Belew and L. B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 296–302, San Mateo, CA, 1991. Morgan Kaufmann.

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

[24] D. B. Fogel and J. W. Atmar. Comparing genetic operators with Gaussian mutations in simulated evolutionary processes using linear search. *Biological Cybernetics*, 63:111–114, 1990.

[25] J. F. Fontanari and R. Meir. The effect of learning on the evolution of asexual populations. *Complex Systems*, 4:401–414, 1990.

[26] S. Forrest, B. Javornik, R. Smith, and A. Perelson. Using genetic algorithms to explore pattern recognition in the immune system. *Evolutionary Computation*, in press.

[27] U. Frisch, B. Hasslacher, and Y. Pomeau. Lattice-gas automata for the Navier-Stokes equation. *Physical Review Letters*, 56:1505, 1986.

[28] P. Gacs. Nonergodic one-dimensional media and reliable computation. *Contemporary Mathematics*, 41:125, 1985.

[29] 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.

[30] H. Gerola and P. Seiden. Stochastic star formation and spiral structure of galaxies. *Astrophysical Journal*, 223:129, 1978.

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

[32] D. E. Goldberg, B. Korb, and K. Deb. Messy genetic algorithms: Motivation, analysis, and first results. *Complex Systems*, 3:493–530, 1990.

[33] F. Gruau. Genetic synthesis of Boolean neural networks with a cell rewriting developmental process. In L. D. Whitley and J. D. Schaffer, editors, *International Workshop on Combinations of Genetic Algorithms and Neural Networks*, pages 55–72, Los Alamitos, CA, 1992. IEEE Computer Society Press.

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

[35] A. B. Hadj-Alouane and J. C. Bean. A genetic algorithm for the multiple-choice integer program. Technical Report 92-50, Dept. of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109, 1992.

[36] 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.

[37] S. A. Harp and T. Samad. Genetic synthesis of neural network architecture. In L. D. Davis, editor, *Handbook of Genetic Algorithms*, pages 202–221. Van Nostrand Reinhold, 1991.

[38] I. Harvey, P. Husbands, and D. Cliff. Issues in evolutionary robotics. In J.-A. Meyer, H. L. Roitblat, and S. W. Wilson, editors, *From Animals to Animats 2: Proceedings of the second international conference on simulation of adaptive behavior*, pages 364–373, Cambridge, MA, 1993. MIT Press.

[39] W. D. Hillis. Co-evolving parasites improve simulated evolution as an optimization procedure. *Physica D*, 42:228–234, 1990.

[40] G. E. Hinton and S. J. Nowlan. How learning can guide evolution. *Complex Systems*, 1:495–502, 1987.

[41] J. H. Holland. *Adaptation in Natural and Artificial Systems.* MIT Press, Cambridge, MA, 1992. Second edition (First edition, 1975).

[42] J. H. Holland. Echoing emergence: Objectives, rough definitions, and speculations for Echo-class models. Technical Report 93-04-023, Santa Fe Institute, 1993. To appear in *Integrative Themes*, G. Cowan, D. Pines and D. Melzner, Reading, MA: Addison-Wesley.

[43] J. H. Holland and J. H. Miller. Artificial adaptive agents in economic theory. Technical Report 91-05-025, Santa Fe Institute, Santa Fe, New Mexico, 1991.

[44] D. Jefferson, R. Collins, C. Cooper, M. Dyer, M. Flowers, R. Korf, C. Taylor, and A. Wang. Evolution as a theme in artificial life: The Genesys/Tracker system. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial Life II*, pages 549–577, Reading, MA, 1992. Addison-Wesley.

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

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

[47] 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.

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

[49] K. Lindgren. Evolutionary phenomena in simple dynamics. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial Life II*, pages 295–312, Reading, MA, 1992. Addison-Wesley.

[50] K. Lindgren and M. G. Nordhal. Artificial food webs. In C. G. Langton, editor, *Artificial Life III*, Reading, MA, 1993. Addison-Wesley.

[51] A. Mackay. Crystal symmetry. *Phys. Bull.*, 27:495, 1976.

[52] B. Madore and W. Freedman. Computer simulations of the Belousov-Zhabotinsky reaction. *Science*, 222:615, 1983.

[53] F. Menczer and D. Parisi. A model for the emergence of sex in evolving networks: Adaptive advantage or random drift? In F. J. Varela and P. Bourgine, editors, *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, Cambridge, MA, 1992. MIT Press/Bradford Books.

[54] M. Meriaux. A cellular architecture for image synthesis. *Microprocess. and Microprogram*, 13:179, 1984.

[55] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Artificial Intelligence Series. Springer-Verlag, 1992.

[56] G. F. Miller and P. M. Todd. Exploring adaptive agency I: Theory and methods for simulating the evolution of learning. In D. S. Touretzky et al., editor, *Proceedings of the 1990 Connectionist Models Summer School*, San Mateo, CA, 1990. Morgan Kaufmann.

[57] G. F. Miller, P. M. Todd, and S. U. Hegde. Designing neural networks using genetic algorithms. In J. D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 379–384. Morgan Kaufmann, San Mateo, CA, 1989.

[58] M. Mitchell, J. P. Crutchfield, and P. T. Hraber. Dynamics, computation, and the "edge of chaos": A re-examination. In G. Cowan, D. Pines, and D. Melzner, editors, *Integrative Themes*, Reading, MA, 1993. Addison-Wesley. In press.

[59] M. Mitchell and J. H. Holland. When can a genetic algorithm outperform hill climbing? To appear in J. D. Cowan, G. Tesauro, and J. Alspector (editors), *Advances in Neural Information Processing Systems 6*. San Mateo, CA: Morgan Kaufmann.

[60] M. Mitchell, P. T. Hraber, and J. P. Crutchfield. Revisiting the edge of chaos: Evolving cellular automata to perform computations. *Complex Systems*, 1993. In press.

[61] D. J. Montana and L. D. Davis. Training feedforward networks using genetic algorithms. In *Proceedings of the International Joint Conference on Artificial Intelligence*, San Mateo, CA, 1989. Morgan Kaufmann.

[62] S. Nolfi, J. L. Elman, and D. Parisi. Learning and evolution in neural networks. Technical Report CRL 9019, Center for Research in Language, University of California, San Diego, 1990.

[63] Y. Oono and M. Kohmoto. A discrete model of chemical turbulence. *Phys. Rev. Lett.*, 55:2927, 1985.

[64] N. H. Packard. Lattice models for solidification and aggregation. In Y. Katoh, R. Takaki, J. Toriwaki, and S. Ishizaka, editors, *Proceedings of the First International Symposium for Science on Form*. KTK Scientific Publishers, 1986.

[65] 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.

[66] N. H. Packard. Intrinsic adaptation in a simple model for evolution. In C. G. Langton, editor, *Artificial Life*, pages 141–155, Reading, MA, 1989. Addison-Wesley.

[67] D. Parisi, S. Nolfi, and F. Cecconi. Learning, behavior, and evolution. In *Proceedings of the First European Conference on Artificial Life*, Cambridge, MA, 1992. MIT Press/Bradford Books.

[68] R. Parsons, S. Forrest, and C. Burks. Genetic algorithms for dna sequence assembly. In L. Hunter, D. Searls, and J. Shavlik, editors, *Proceedings of the first international conference on intelligent systems for molecular biology*, pages 310–318, Menlo Park, CA, 1993. AAAI Press.

[69] J. Pecht. On the real-time recognition of formal languages in cellular automata. *Acta Cybernetica*, 6:33, 1983.

[70] D. J. Powell, M. M. Skolnick, and S. S. Tong. Interdigitation: A hybrid technique for engineering design optimization employing genetic algorithms, expert systems, and numerical optimization. In L. D. Davis, editor, *Handbook of Genetic Algorithms*, pages 312–331. Van Nostrand Reinhold, 1991.

[71] K. Preston. Basics of cellular logic with some applications in medical image processing. *Proc. IEEE*, 67:826, 1979.

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

[73] T. S. Ray. Is it alive, or is it GA? In R. K. Belew and L. B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 527–534, San Mateo, CA, 1991. Morgan Kaufmann.

[74] T. S. Ray. An approach to the synthesis of life. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial Life II*, pages 371–408, Reading, MA, 1992. Addison-Wesley.

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

[76] J. D. Scargle, D. L. Donoho, J. P. Crutchfield, T. Steiman-Cameron, J. Imamura, and K. Young. The quasi-periodic oscillations and very low frequency noise of Scorpius X-1 as transient chaos: A dripping handrail? *Astrophy. J. Lett.*, 411:L91 – L94, 1993.

[77] J. D. Schaffer and L. J. Eshelman. On crossover as an evolutionarily viable strategy. In R. K. Belew and L. B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 61–68, San Mateo, CA, 1991. Morgan Kaufmann.

[78] S. Schulze-Kremer. Genetic algorithms for protein tertiary structure prediction. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature 2*, pages 391–400, Amsterdam, 1992. North Holland.

[79] K. Shahookar and P. Mazumder. A genetic approach to standard cell placement using metagenetic parameter optimization. *IEEE Transactions on Computer-Aided Design*, 9(5):500–511, 1990.

[80] A. R. Smith. Real-time language recognition by one-dimensional cellular automata. *J. Comput. System Sci.*, 6:233, 1972.

[81] S. A. Smith, R. C. Watt, and S. R. Hameroff. Cellular automata in cytoskeletal lattices. *Physica D*, 10:168–174, 1984.

[82] S. Sternberg. Language and architecture for parallel image processing. In *Proceedings of the Conference in Pattern Recognition in Practice*, Amsterdam, 1980.

[83] G. Syswerda. Uniform crossover in genetic algorithms. In J. D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 2–9, San Mateo, CA, 1989. Morgan Kaufmann.

[84] C. E. Taylor, D. R. Jefferson, S. R. Turner, and S. R. Goldman. RAM: Artificial life for the exploration of complex biological systems. In C. G. Langton, editor, *Artificial Life*, pages 275–295, Reading, MA, 1989. Addison-Wesley.

[85] P. M. Todd and G. F. Miller. Exploring adaptive agency III: Simulating the evolution of habituation and sensitization. In H.-P. Schwefel and R. Männer, editors, *Parallel Problem Solving from Nature*, Berlin, 1990. Springer-Verlag (Lecture Notes in Computer Science).

[86] P. M. Todd and G. F. Miller. Exploring adaptive agency II: Simulating the evolution of associative learning. In J.-A. Meyer and S. W. Wilson, editors, *From animals to animats: Proceedings of the first international conference on simulation of adaptive behavior*, pages 306–315, Cambridge, MA, 1991. MIT Press.

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

[88] R. Unger and J. Moult. A genetic algorithm for 3d protein folding simulations. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 581–588, San Mateo, CA, 1993. Morgan Kaufmann.

[89] G. Y. Vichniac. Simulating physics with cellular automata. *Physica D*, 10:96–116, 1984.

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

[91] D. A. Young. A local activator-inhibitor model of vertebrate skin patterns. *Math. Biosciences*, 72:51, 1984.

[92] P. Zamperoni. Some reversible image operators from the point of view of cellular automata. *Biological Cybernetics*, 54:253–261, 1986.