Using a Genetic Algorithm to Evolve Behavior in Multi Dimensional Cellular Automata

Breukelaar and Bäck

GECCO 2005

Introduction

- Science has long been fascinated by how simple localized hehavior can produce very complex global behavior.
- Cellular automata (CA) is an abstract representation of individuals.

1D Cellular Automata

- A one-dimensional cellular automata (CA) looks like an array of bits.
- For example, $C = \{a_1, a_2, \ldots, a_N\}$ is a CA of width N and a_N is adjacent to a_1.
- s_k denotes the neighborhood of a_k. If we define the neighbor radius $r = 3$, $s_2 = \{a_{N-1}, a_N, a_1, a_2, a_3, a_4, a_5\}$.
- The transition rule $\Theta : \{0,1\}^{2r+1} \rightarrow \{0,1\}$ takes neighborhood as inputs and output a bit.
  Assume $r = 5$. There are 2^{11} different combination of neighbors. Each neighbor is mapped to zero or one, so there are $2^{2^{11}}$ different rules.

Multi dimensional Cellular Automata

- There are two definitions of neighborhood: the von Neumann neighborhood and the Moore neighborhood.

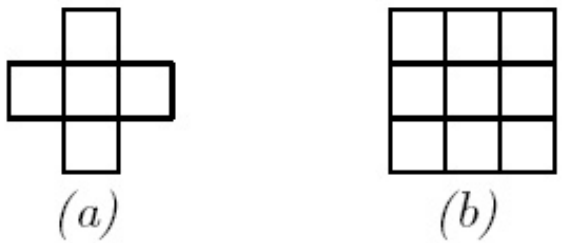

Figure 1: Two often used and well known two dimensional neighborhoods. *(a)* the von Neumann neighborhood and *(b)* the Moore neighborhood.

- The neighborhood used in experiments is von Neumann neighborhood.

Genetic Algorithm

- λ: the number of individuals.
- Single-point crossover with crossover rate c.
- Mutation rate m.
- Random initialization. The number of ones distributes over normal distribution.
  Prevent the algorithm from specializing in a particular area of the search space at the beginning of the algorithm.
- D: maximum generation.
- Tournament selection with size q.
- Fitness: the success probability.

Majority Problem Description

- The majority problem is often used to show the power of cellular automata.
- The problem is:
  Given a set $A = \{a_1, \ldots, a_n\}$ with n odd and $a_m \in 0, 1$ for all m. Answer the question: are there more ones than zeros in A?
- Convert the problem to 1D CA:
  Find a transition rule that result in an all zero state if $\lambda < 0.5$, and an all one state otherwise.

Methods for majority problem

- Intuitive method:
  If the number of ones is more than the number of zeroes in the neighborhood, the output value is 1.
- 1978 GKL rule: 81.6% for $N = 149$.
- 1995 Davis: 81.8% for $N = 149$.
- 1995 Das: 82.178% for $N = 149$.
- 1996 David, Forrest, Koza: 82.326% for $N = 149$.

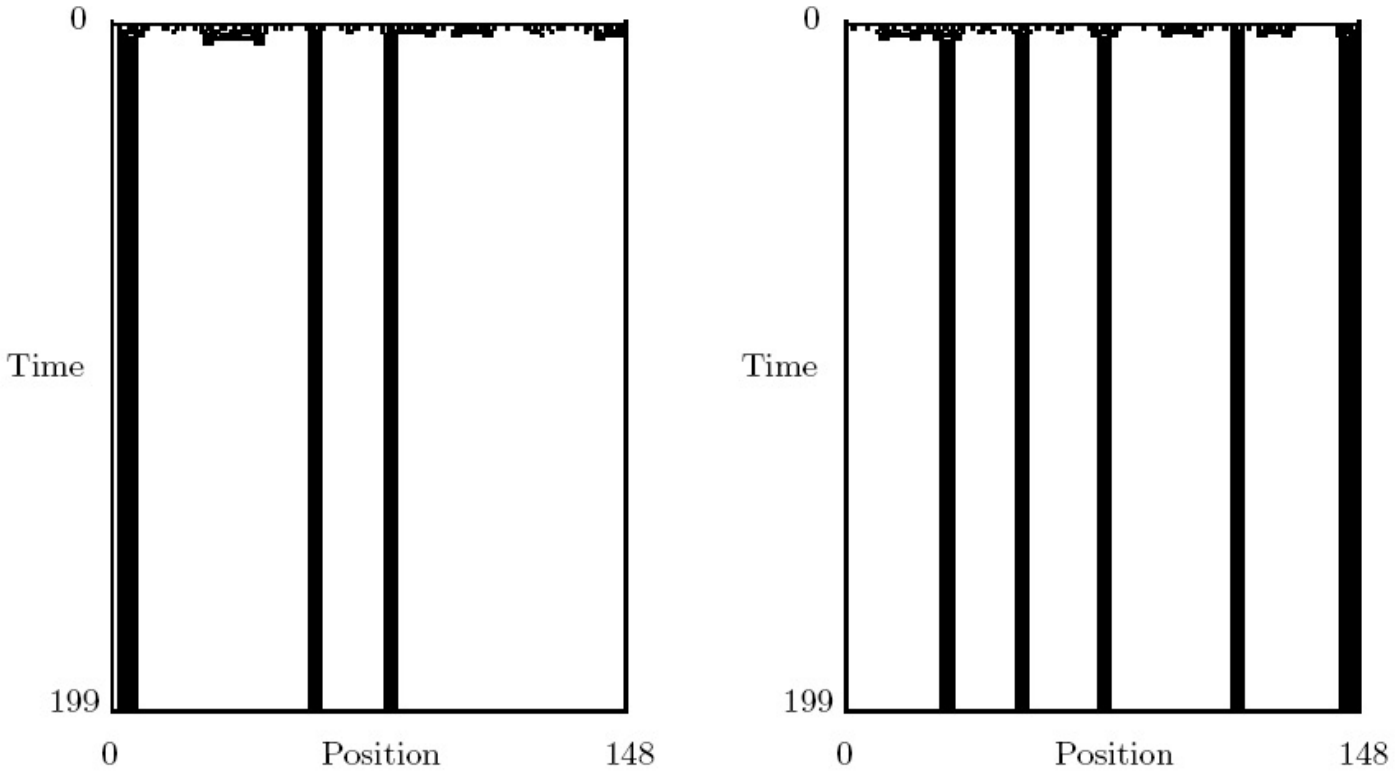

Figure 2: These are examples of majority problem classification by the "majority rule". The pictures show how the rule gets stuck on "thick lines" in the time plot. Time t proceeds from top to bottom and every row corresponds to C^t.

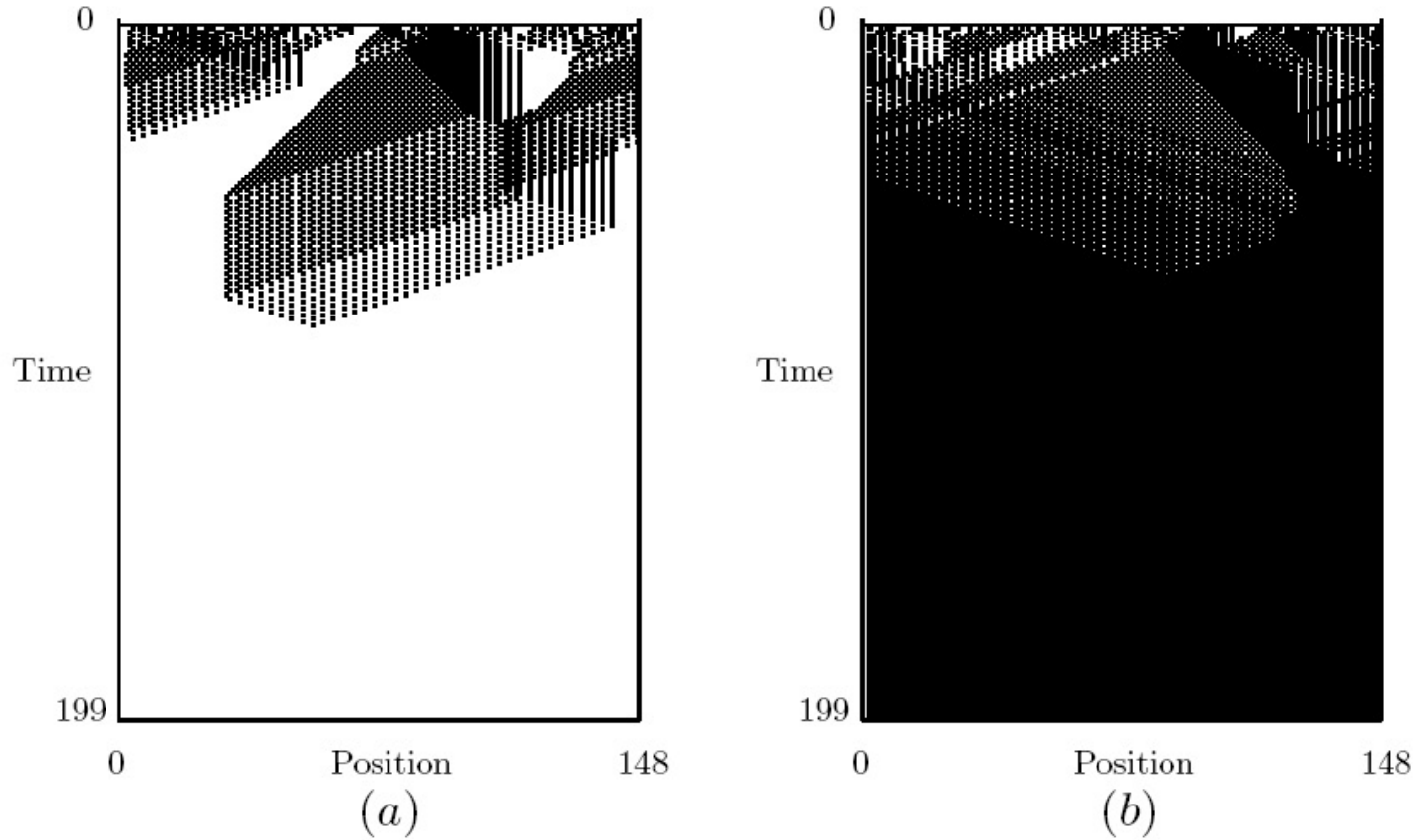

Figure 3: These are examples of majority problem classification by the rule found by David, Forrest and Koza. Both are correct classifications (a) with 74 ones in the initial state, (b) with 75.

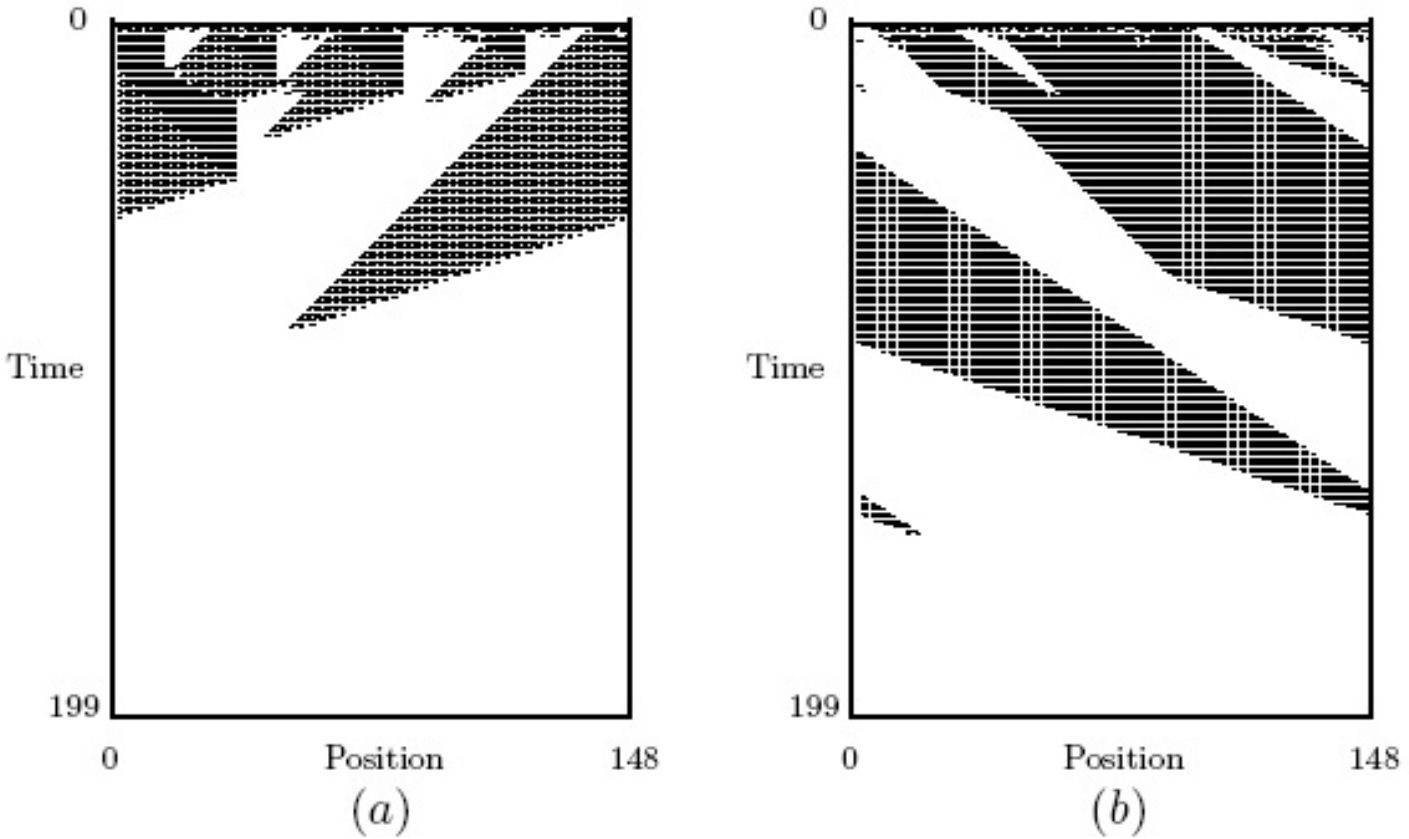

Figure 4: This figure displays two correct classification of the majority problem by two different particle based rules generated with the GA. (a) has $F^B_{149,10^4} \approx 0.76$ and (b) has $F^B_{149,10^4} \approx 0.75$. with $N = 149$.

The parameter of GA

- $r = 3$ and $d = 1$. The length of chromosome is 128 bits.
- Population size $\lambda = 100$, tournament size $q = 20$.
- Crossover probability $c = 0.6$, mutation probability $m = \frac{2}{128} = 0.015625$.
- Max generation $D = 100$.
- Select the distribution of the number of ones:
  It is very difficult to evolve good transition rules with a GA while using a binomial distribution over the number of ones in the initial state.
  The distribution starts as uniform distribution and ends as binomial distribution.
- Each experiment is runned 100 times.
- $I = 320$: the maximum iterations of CA.
- $M = 10^4$: the number of test cases.

$F^B_{N,10^4}$	d		
	1	2	3
0.0 - 0.5	0	0	1
0.5 - 0.55	2	0	0
0.55 - 0.6	1	4	0
0.6 - 0.65	54	17	14
0.65 - 0.7	37	72	55
0.7 - 0.75	3	1	30
0.75 - 0.8	3	0	0
0.8 - 1.0	0	0	0

Table 4: This table shows the fitness distribution using different number of dimensions d. Note that $N = 149$ for $d = 1$, $N = 169$ for $d = 2$ and $N = 343$ for $d = 3$. $q = 20$, $c = 0.6$ and other settings are the same as the initial values proposed in section 4.1.

Some problems of the experiment

- The tournament size is too large and the population size is too small.
- The performance is not as good as GP ($F < 0.8$).

Problem Description

- Find a transition rule that, given an initial state of a CA, it iterates this CA to a "checkerboard pattern" within I iterations.
- Checkerboard pattern: $1, 0, 1, 0, 1, 0, \ldots$
- The same GA that was used in majority problem is used here.

Results

- For $M = 10^5$, $r = 3$, $d = 1$, $N = 150$, the fitness is 0.99834.
- For $M = 10^5$, $r = 1$, $d = 3$, $N = 6^3$, the fitness is 0.99997.

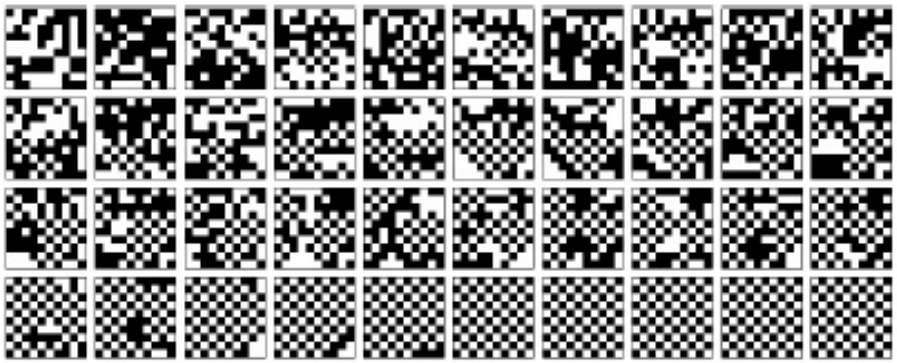

Figure 7: This figure shows a correct two dimensional CA iteration for the checkerboard problem. It start top-left with a random initialization of a 10×10 CA, iterates from left to right, top to bottom and ends up with a perfect checkerboard pattern in the end state.

Problem Description

- Given an initial state and a specific desired end state: find a rule that iterates from the initial state to the desired state in less than I iterations.
- This is a generalized version of the checkerboard problem.

Results

Table 5: Number of successful rules found per bitmap.

Bitmap	*Successful rules (out of a 100)*
"square"	100
"hourglass"	96
"heart"	55
"smiley"	29
"letter"	18

Results

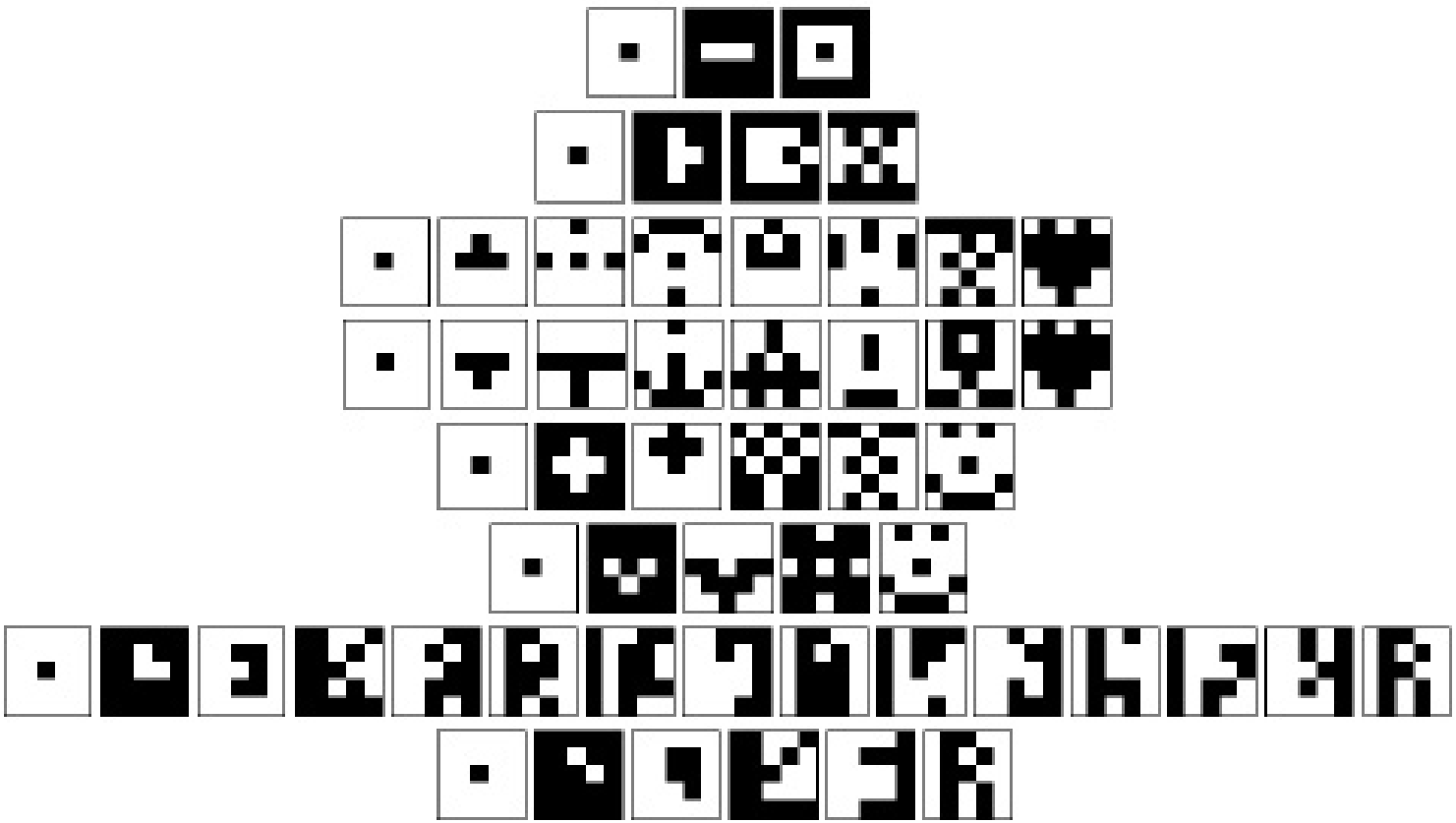

Figure 10: This figure shows some iteration paths of successful transition rules.

Conclusions

- The GA exceeds the performance found in [9,10] and does this for different topologies of CA.
- Multi dimensional CA can solve problems faster than one dimensional CA.
- The global behavior of a CA is not limited to checkerboards and majority problems.

Discussion

- The applications of CA: bot.
  Insufficient information, multi dimensional, distributed environment.
- Improve the process of GA.