Research of complexity in cellular automata through evolutionary algorithms

Emmanuel Sapin

Faculty of Computing, Engineering and Mathematical Sciences,
University of the West of England
Bristol, BS16 1QY, UK

Olivier Bailleux

Jacqueline Chabrier

Universite de Bourgogne, 9 avenue A. Savary, B.P. 47870
Dijon, 21078 Cedex, France

This paper presents an evolutionary approach of the search of cellular automata accepting gliders. The proposed technique is based on a specific fitness function taking in account the spatial evolution, the number of living cells as well as the presence of gliders. The results show that the genetic algorithmic is a promising tool for the search of cellular automata with specific behaviours, and then could prove to be decisive for identification of new automata supporting universal computation.

1. Introduction

Cellular automata are discrete systems in which a population of cells evolves from generation to generation on the basis of local transitions rules. They can simulate simplified forms of life [12, 13] or physical systems with discrete time and space and local interactions [9, 10, 17].

Wolfram showed that one-dimensional cellular automata can present a large spectrum of dynamic behaviours. In "Universality and complexity in cellular automata" [22], he introduces a classification of cellular automata, comparing their behaviour with that of some continuous dynamic systems. He specifies four classes of cellular automata on the basis of qualitative criteria.

For all initial configurations, class 1 automata evolve after a finite time to a homogeneous state where each cell has the same value. Class 2 automata generate simple structures where some stable or periodic forms survive. The class 3 automata's evolution leads, for most initial states, to chaotic forms. All other automata belong to class 4. According to Wolfram, automata of the class 4 are good candidates for universal computation.

The only binary automaton currently identified as supporting universal computation is Life, which is, besides, in class 4. Its ability to simulate a Turing machine is proved in [5] in a constructive way, using gliders (i.e. periodical patterns which, when evolving alone, are reproduced identically by shifting in space) to carry information and to realize logical gates through collisions. The identification of new automata capable of generating gliders is consequently a possible lead to the search of new automata supporting universal computation.

In the spirit of the work described in [6, 7, 15] about 1D automata, this paper presents an evolutionary algorithm discovering new rules capable of spontaneously generating gliders. This algorithm uses a fitness function detecting the birth of gliders in a primordial soup.

Section 2 describes the framework, including the representation of the transition rules. Section 3 details selection, cross-over, mutation operators, and the fitness function that allowed us to obtain experimental results presented in Section 4. In Section 5, we present a synthesis of the results and several research perspectives.

2. Framework

Concerning this study, we look only into cellular automata with the following specifications:

- Cells have 2 possible values, called 0 and 1.
- They evolve in a 2D matrix, called universe.
- Transition rules only take into account the eight direct neighbours of a cell for the current generation, so as to determine its states for the next generation.

We call context of a cell the state of the cell and its 8 neighbours. A cell thus can have 512 different contexts. A transition rule is defined as a boolean function that maps each of the 512 possible contexts to the value which will be taken by the concerned cell at the next generation. Therefore the underlying space of automata includes 2^{512} rules.

Let us recall that a more limited space is known as Bays Space [1, 2, 3, 4], where a transi-tion rule is specified by a $EbEh/FbFh$ quadruplet. A cell survives to the next generation iff its number of neighbours for the current generation is included between Eb and Eh. A cell is born during the next generation iff its number of neighbours at the current generation is included between Fb and Fh. Bays Space only includes 1296 rules, which simplifies the exhaustive study of corresponding automata. However, some rules presenting complex behaviours such as the one presented in [14], where a cell survives iff it has 1 or 3 living neighbours, do not belong to Bays Space.

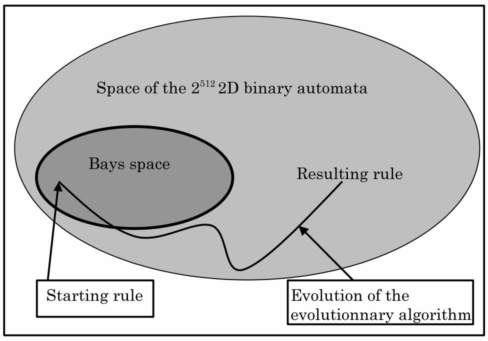

Figure 1. Representation of the evolutionary process allowing to create a new rule.

3. Genetic Evolution

This section describes the use of an evolutionary algorithm for the search of new rules capable of generating gliders (cf. figure 1). Because confusion may exist between the evolutionary algorithms and cellular automata generations, we use the word "transition" for the generations of an automaton.

3.1 Encoding

Rules have been encoded by 512-bit strings. The value of the bit related to each of the 512 possible contexts is the value taken by the concerning cell at the time of the next transition of the CA.

For example, figure 2 represents the rule 35/33 of Bays space. The rectangle pointed at by the arrow represents a context that leads to the birth of a cell at the following generation, what is shown by a point on the right of the context's representation. If the bit related to this context in the string undergoes a mutation, this context won't provoke the presence of a living cell anymore.

3.2 Initialization and Selection

The evolutionary algorithm manages a population of 50 rules. The initial population comes from a rule R of the Bays space. It includes 10 occurrences of R, 10 variants of R obtained through 1 mutation, 10 rules obtained through 2 mutations of R, 10 rules obtained through 3 mutations of R, and 10 rules obtained through 4 mutations of R. Ranking selection operator [8] is used, with conservation of the twenty best rules. At each generation, the population stemming from the selection stage is completed with 20 rules obtained through mutation and 10 obtained through cross-over.

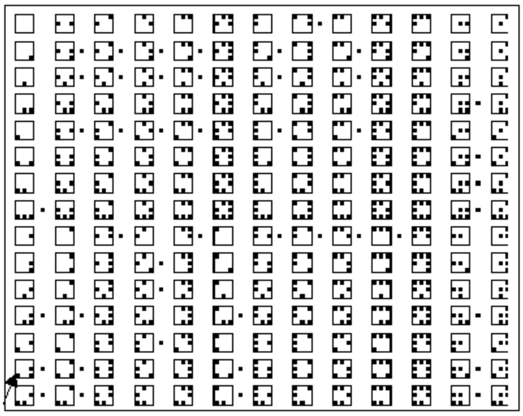

Figure 2. Relevant section of representation of the rule 35/33.

3.3 Cross over

The twenty best rules are dispatched randomly into ten couples from which stem ten new rules that are added to the population. These new rules are generated by a simple cross over operator at a median point.

3.4 Mutation

A mutation consists in modifying a randomly chosen bit, with the same weight for each of the 512 bits of a rule.

We noted that only one mutation per rule for each generation was not sufficient to observe a convergence toward automata capable of producing gliders. Therefore we chose a more aggressive mutation strategy, aimed at maintaining some diversity in the population, inspired by the research work by Lee and Takagi [17] and Sefrioui M. and P'eriaux J. [20]. For each couple used for crossing-over, Hamming's distance between both rules determines the number of mutations. If this distance is greater than or equal to 5 then only one mutation is applied to each rule. Otherwise five mutations are applied to each rule.

3.5 Fitness Function

The fitness function is based on the evolution, during three hundred transitions, of a "primordial soup", randomly generated in a square of 40*40 centered in a 200*200 space. For each transition, the dimensions of the smallest rectangle containing all the living cells, which is called "including rectangle", are measured. In the presence of gliders this rectangle grows larger over the transitions.

During evolution of the automata, the algorithm counts the number n_1 of times the area of this rectangle increases and the number n_2 of

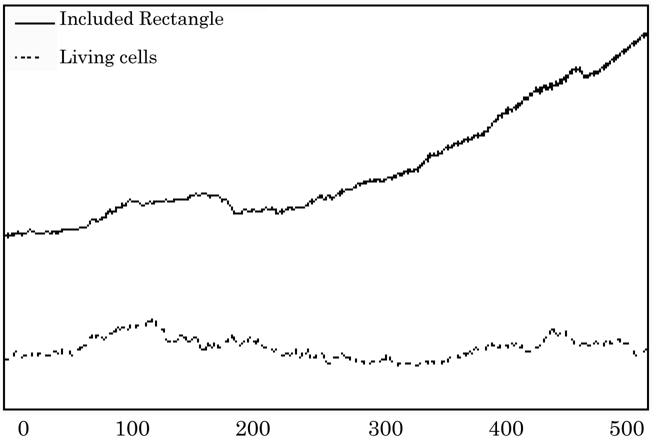

Figure 3. Generation-by-generation average of the surface of the including rectangle and the number of living cells during 500 generations for the evolution of Life.

times it decreases. For L transitions of the automata we define:

$$S_1 = \frac{n_1 - n_2 + L}{L}. \tag{1}$$

This score represents the ability of the automata to spread through the space.

At each transition, the total number of living cells is also taken into account. This number, in Life, tends to remain stable, as verified with figure 3 where the total number of cells and the surface of the including rectangle are represented for Life. Given that Life is the only automaton known as complex and supporting gliders, we choose to take into account this characteristic in our fitness function.

This propensity to have a stable number of cells is estimated by a second s_2 score. During the evolution of the automata, the algorithm counts the number m_1 of times the total number of cells decreases and the number m_2 of times it increases. For an automaton having evolved for L transitions, S_2 is defined by:

$$S_2 = \frac{m_1 - m_2 + L}{L}. \tag{2}$$

With a first fitness function $S_1 * S_2$, we observed a convergence toward rules with good S_1 and S_2 scores but accepting no gliders. In these rules, the increase of the including rectangle is due to a global move of the living cells in a privileged direction. In order to avoid such behaviour, we add a coefficient to the fitness function, reflecting the move of the gravity center of the living cells. This coefficient is defined

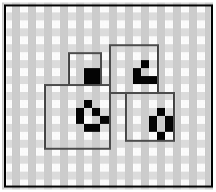

Figure 4. Group of isolated cells.

by $c_1 = 1 + \frac{dg}{sz}$, where dg is the Euclidian distance between the center of gravity and the middle of the square universe of size sz.

Moreover, we add another coefficient c_2 in order to promote the rules allowing the spontaneous apparition of gliders and periodic patterns. The presence of gliders and periodic patterns has been the subject of the following test, inspired by Bays' test [1]: after the evolution of the automaton, from a primordial soup, each group of connected cells (see figure 4) is isolated in an empty space and evolves during 20 transitions. For each transition, the original pattern is searched in the test universe. Three cases can happen:

- The initial pattern has reappeared at its first location (it is then considered to be periodic).
- It has reappeared at another location (it is then considered to be a glider).
- It has not reappeared (it's then considered evolving)

Let g denote the number of appearances of gliders and p denote the number of appearances of periodic patterns. The coefficient c_2 is then defined as follows:

$$c_2 = 1 + v + \frac{s}{100}. \tag{3}$$

The fitness function becomes:

$$F = \frac{S_1 * S_2 * c_2}{c_1}. \tag{4}$$

The number of gliders plays a part in the new fitness function, but also the number of periodical patterns, so as to promote the automata that, even if not having gliders spontaneously emerge, accept nevertheless periodical patterns. These automata are privileged because we infer that, if they accept periodical patterns, these automata are able to evolve to rules supporting gliders, thanks to the evolutionary algorithm.

256	128	64
32	16	8
4	2	1

Table 1. Table with the associated weight of the neighbors of a cell to evaluate the rank of the context of this cell

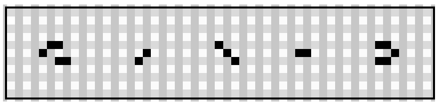

Figure 5. Stable patterns and periodic patterns of R_1.

4. Result

4.1 Notation

For conciseness and readability, the following convention has been adopted for the rules presentation. A rule R' stemming from the evolution of an initial rule R will be noted:

$$(EbEh/FbFh)\{m_k\}_{0<k<n}. \quad (5)$$

Where $(EbEh/FbFh)$ is the representation of R in Bays Space, n the number of contexts in which R and R' differ, and the set mk corresponds to the ranks of these contexts. The rank of a context corresponds to the place of the related bit in the string of the rule. It is calculated by giving the weight, shown table 1, to the living cells.

For instance, the rank of the context pointed at by the arrow in figure 2 is 13 corresponding to 1+4+8.

4.2 New complex rules

The three following rules, that accept gliders, were obtained from our evolutionary algorithm. These three rules can be found in [20] in mcl format [22].

The rule (12/33) {210, 156, 432, 312, 211, 242, 188, 179, 181, 433, 369, 372, 313, 311, 72, 73, 226, 452, 141, 142, 172, 163, 165, 166, 353, 356, 300, 227, 453, 173, 391}, denoted R_1, presents 31 mutations and was obtained after 59 generations. It allows many stable and periodic patterns to emerge. Some examples are presented figure 5.

Several gliders also appear, such as the one presented figure 6, that moves 12 squares horizontally while 4 vertically. This glider also leaves

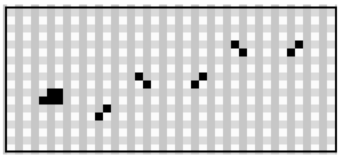

Figure 6. Puffer of R_1.

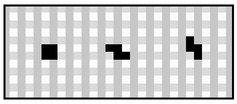

Figure 7. Stable patterns and periodic patterns R_2 rule.

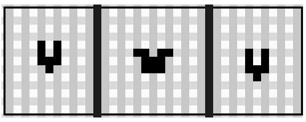

Figure 8. Evolution of a glider during one period for the R_2 rule.

a trace behind. Such a pattern is more exactly called a puffer [12](i.e. stable patterns subsisting after the passage of a periodically displacing pattern). This pattern, going through 12 squares during 40 transitions, then has a speed of $\frac{3}{10}$.

Another interesting result was obtained by letting the rule 35/33 evolved during 350 generations, ending up with the rule (35/33) {49, 83, 124, 187, 381, 164, 204, 450, 102, 488, 271, 399}, denoted R_2. This rule offers a large sample of stable and periodic patterns (see some of them presented figure 7).

A glider, which is able to move horizontally and from top to bottom following its initial direction, is represented Figure 8. It has a period of 2 and a speed of $\frac{1}{2}$. We should note that this glider exists in the rule 35/33 [18], but the mutations have strongly augmented its appearance potentiality.

Another interesting example is the rule 22/33 {144, 80, 24, 17, 18, 20, 48, 272, 154, 432, 72}, noted R_3, obtained after 30 generations. Two gliders, with a period of 11 and 4, are presented in figures 9 and 10 respectively. Many periodic patterns, such as the patterns presented in figure 11, also exist in this rule.

5. Related Works

In [6, 7], Das R., Mitchell M. and Crutchfield J. P used genetic algorithms to evolve cellular automata to perform computational tasks that

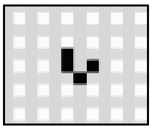

Figure 9. Glider of R_3 with a period 11, π angle, at a speed of $\frac{1}{11}$.

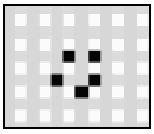

Figure 10. Glider of R_3 with a period of 4, $-\frac{\pi}{2}$ angle, at a speed of $\frac{1}{4}$.

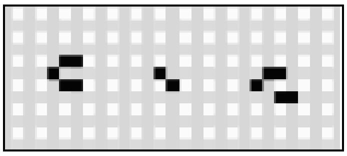

Figure 11. Stable patterns and periodic patterns of R_3.

require global information processing. They studied one-dimensional binary-state cellular automata for two computational tasks: density classification and synchronization. For the density classification task, the goal is to find a CA that decides whether or not the initial configuration contains a majority of 1s (i.e. has high density). [15] shows an embedded-particle framework capturing the main information processing mechanisms of the emergent computation presented in [6] and [7].

In [16], genetic programming is also applied to evolve CA for simple random number generation.

6. Synthesis and perspectives

In this paper, we proposed an evolutionary approach of the research of cellular automata accepting gliders, based on a classical scheme of genetic algorithm with binary representation of the transition rules. We used a specific fitness function taking in account the spatial evolution, the number of living cells as well as the presence of gliders.

Starting from initial populations of rules without the apparition of gliders, we have noticed the emergence of new rules accepting gliders. This constitutes a first experimental contribution to the research of new cellular automata supporting universal computation. Beyond this punctual result, the evolutionary approach proves to be very promising for the research of 2D complex automata presenting specific behaviours.

However, in spite of the precautions taken in the fitness function, the used mutation and crossing-over operators permit the emergence of non-isotropic rules and particularly unidirectional gliders.

We plan to adapt the mutation operator, in order to keep the isotropy of rules. The genetic algorithm would then be able to find isotropic, complex 2D automata with two states, that accept gliders.

We also plan to expand the research, starting from an initial population consisting of any rules (i.e. not only from Bays Space) for example by using randomly generated rules.

Later on, our aim is the research of new transition rules allowing the realization of glider guns, logical operators, and simulating a Turing machine. One research direction may be a co-evolution of patterns and rules.

References

[1] C. Bays, "Candidates for the game of life in three dimensions," *In Complex Systems*, **1** (1987) 373–400.

[2] C. Bays, "A note on the discovery of a new game of three-dimensional Life," *In Complex Systems*, **2** (1988) 255–258.

[3] C. Bays, "The discovery of a new glider for the game of three-dimensional Life," *In Complex Systems*, **4** (1990) 599–602.

[4] C. Bays, "A new candidate rule for the game of three-dimensional Life," *In Complex Systems*, **6** (1992) 433–441.

[5] E. Berlekamp, J. H. Conway, and R. Guy, *Winning Ways for your mathematical plays* (Academic press, New York, 1982).

[6] R. Das, M. Mitchell, and J. P. Crutghfield, *Evolving Cellular Automata with Genetic Algorithms: A Review of Recent Work*, Proceedings of the First International Conference on Evolutionary Computation and Its Applications (EvCA'96), held on 1996, edited by Russian Academy of Sciences.

[7] M. Mitchell, J. P. Crutchfield, and P. T. Hraber, "Evolving cellular automata to perform computations: Mechanisms and Impediments," *Physica D*, **75** (1994) 361–391.

[8] L. Davis, *The Genetic Algorithm HandBook* (Van Nostrand Reinhold, New York, 1991).

[9] C. Dytham, B. Shorrocks "Selection, Patches and Genetic Variation: A Cellular Automata Modeling Drosophila Populations," *In Evolutionary Ecology*, **6** (1992) 342–351.

[10] I. R. Epstein, "Spiral Waves in Chemistry and Biology," *In Science*, **252** (1991) 67.

[11] Ermentrout, G. Lotti, and L. Margara "Cellular Automata Approaches to Biological Modeling,"*In Journal of Theoretical Biology*, **60** (1993) 97-133.

[12] M. Gardner, "The fantastic combinaisons of John Conway's new solitaire game " Life ","*In Scientific American*, (1970).

[13] M. Gardner, "On Cellular Automata, Self-reproduction, the Garden of Eden, and the Game of Life,"*In Scientific American*, **224** (1971) 112-118.

[14] J. C. Heudin, "A new candidate rule for the game of two-dimensional life,"*In Complex systems*, **10** (1996) 367-381.

[15] W. Hordijk, J. P. Crutchfield, and M. Mitchell, *Evolving Cellular Automata with Genetic Algorithms: A Review of Recent Work*, Proceedings of the First International Conference on Evolutionary Computation and Its Applications (EvCA'96), held on 1996, edited by Russian Academy of Sciences.

[16] M. Mitchell, P. T. Hraber, and J. P. Crutchfield "Revisiting the edge of chaos : Evolving Cellular Automate to perform Computations,"*Complex systems*, **7** (1993) 89-130.

[17] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection* (MIT Press, Cambridge, MA).

[18] M.A Lee and H. Takagi, *Dynamics control of genetic algorithms success using fuzzy logic techniques*, Proceeding of the 5th International Conference on Genetics Algorithms, hold on 1993, edited by S.Forrest.

[19] M. Magnier, C. Lattaud, and J-C. Heudin "Complexity classes in the Two-dimensional Life Cellular automata Subspace,"*Complex systems*, **11** (1997).

[20] E. Sapin, *http : //uncomp.uwe.ac.uk/sapin/complexsystems/rules.zip.*

[21] M. Sefrioui and J. P'eriaux, *Fast convergence thanks to diversity*, Proceeding of the Fifth annual Conference on Evolutionary Programming, hold on 1996.

[22] M. Wojtowicz, *http : //psoup.math.wisc.edu/mcell/index.html.*

[23] S. Wolfram, "Universality and complexity in cellular automata,"*Physica D*, **10** (1984) 1-35.