U.P.B. Sci. Bull., Series C, Vol. 72, Iss. 3, 2010 ISSN 1454-234x

CELLULAR AUTOMATA WITH DYNAMICAL LOOP FUNCTION AS NOISE GENERATORS: STATISTICAL ASPECTS (1)

Ioan HĂLĂLAE[1]

Prezenta lucrare studiază pentru prima dată comportamentul automatelor celulare liniare cu 256 de celule şi buclă globală de reacţie (cu modificare dinamică pe parcursul evoluţiei). Se explorează aspectele random în evoluţia AC.

Definirea aleatoriului este o adaptare a celei folosite în teoria algoritmică a informaţiei pentru şiruri binare.

Se testează 4 funcţii de buclă diferite pentru a găsi cel mai bun operator din punctul de vedere al obţinerii unui comportament cât mai aleatoriu (zgomotos).

Ca metodă de lucru se foloseşte matematica experimentală.

The present paper is the first study of the evolution of 256 cells linear cellular automata (CA) with global loop and dynamical modification of the rule. We explore the random aspects in the evolution of CA.

For the definition of randomness we adjusted the one used in the algorithmic information theory for binary strings.

We have tested 4 different operators aiming to find the best for obtaining more randomness in the evolution of CA (more noise).

As a working tool we use experimental mathematics.

Keywords: Cellular Automata with global loop, noise generation, experimental mathematics

1. Introduction

The evolution of cellular automata (CA) is a topic which has not been systematically studied. Wolfram [1] opens the discussion and in [2] makes a more detailed study, but his explorings are exclusively graphical ones.

Ştefan [3] advances the hypothesis of using CA as noise generators, proposing several "simple machines" with a "complicated behavior".

The present paper explores the hypothesis from Ştefan [3], approaching the random aspects in the evolution of CA's. We study, for the first time, linear CA with 256 binary cells with global reaction loop (with dynamical change during the evolution). The space that we explore is of dimension 2^{256}.

[1] Assist., Department of Fundamental Sciences, Faculty of Engineering,, "Eftimie Murgu" University, Reşiţa, Romania, e-mail: i.halalae@uem.ro, i.halalae@gmail.com

The problems of that type of approach are (Ştefan [3]):

- the machines are very sensitive to their initial states, which has to be searched in a space of dimension 2^{256}, and that excludes the possibility of a significant direct generation;
- adding a global loop increases the possibility of obtaining a more "noisy" machine;
- we have no formal methods to describe the effective machine we propose, therefore the only working possibility is to use experimental mathematics (Borwein & Bailey [4], [5]).

Our complete study includes four different operators. This paper presents the common characteristics of the experiments and the results of the first experiment. In further papers, we will present the results obtained with the other three operators and compare the results.

2. Description of the common part of the experiments

2.1. Terminology

By internal status of an automaton, or structure of the automaton, we understand *the distribution of 0 / 1cells* (the physical, spatial distributions, of cells). The internal status of an automaton at any specifical moment is also called life cycle.

The evolution of the automaton means *the process of passing from an internal status to the next one*. The passing of the automaton from one status to the next one is obtained by *the simultaneous application of a calculus function to all cells*.

The evolution cycle represents *the sequence of internal statuses* the automaton passes through.

The life duration (life period) represents *the number* of subsequent life-cycles of the automaton.

The internal structuring mechanism represents the *function for computing the new value of the cell*. The new value is computed as a function of the old value of the cell and the value of the two adjacent cells.

The automaton is considered blocked if all cells have either value 0 or value 1.

We use the initial status of CA for classifications. As an indicator we use *the number of cells having value 1 in the initial distribution*. We name this indicator initial density. Further, we name current density of the automaton *the number of cells of the value 1* in the current life cycle.

2.2. The definition of randomness

The definition of randomness we work with is adapted after the definition used for binary strings in the algorithmic information theory. Chaitin [6] considers a string as random until it starts repeating himself, moment when it becomes predictable.

The definition of randomness for CA is the following: the automaton is random (noisy) until it starts to cycle. We consider here that an automaton has a random evolution if it evolves without getting blocked in a presettled number of evolution cycles that we call the *maximal lifetime duration admitted*.

We associate noise with randomness: an automaton is noisy if it has a random evolution.

2.3. Initial Hypotheses

Our experiments are taking place in a space of dimension 2^{256}, impossible to completely explore. Our search is random in itself: we are starting our study just with a few strictly intuitive hypotheses:

- the number of CA of initial density k is C_{256}^{k}. The initial hypothesis is that around the values of 127, 128, 129 (the biggest number of possible cases) we should have the greatest ratio of noisy CA;
- for each operator, we reiterate the experiments with 3 maximal life durations admitted. The initial hypothesis is that a longer maximal life duration should produce noisier CA.

The main purpose of our search is identifying more restricted areas, for a further more detailed study.

We achieved a statistical ‘constative’ exploration, aiming at:

- to obtain a first image of the evolution of CA;
- to identify, from a set of four different operators, which is the best for our experiments;
- to help us decide whether the methodology we used is good enough, or we have to ‘refine’ it);
- to identify statistically stable zones and exceptions zones; the signification of the terms is that from the paragraph above: zones which offer better chances to find noisy CA.

As we are exploring a massive parallel structure for which there are no classical statistical methods, our results will be, for the moment, just ‘constative’. These results should suffice to indicate areas where a further detailed search, eventually with a more ‘refined’ methodology, should be effectuated.

In a further stage, our search should evoluate towards ‘insight’ aspects in CA. Wolfram [2] treats the CA evolution in an exclusive graphical manner: his construction aims exclusively at the aspects which support the idea of the completely random behavior of CA. ‘Insight’, for us, means a clearer and more detailed knowledge of the CA evolution. Randomness is generally assumed for CA, but a ‘clearer’ perception of the evolution of CA could offer openings for concrete projects based on CA. Another major gain of these experiments would be finding certain aspects which could lead towards insight in CA.

We have to add a remark about our experiments. They differ in the following aspect:

- in the first three experiments, the automaton evolves independently, without being introduced any external noise. The noise which appears is exclusively internal. The differences appears from the different operators;
- in the last experiment, we introduce an external noise. The evolution of the automaton is partly modified by the external noise.

We use a software simulator of our own conception, oriented towards the architecture of the CA. We organized the experiments in four different projects, according to the operators. Further on, experiment is synonym with project.

2.4 The dynamically modified global loop

For the calculus of the new value of one cell we use a lookup table. In our experiment we use a (256,8) matrix*, containing the digits of the binary translation of the 0-255 numbers*. We further refer this matrix by val (256,8).

The actual calculus consists in *determining the position in the lookup table from where we read the new value of the cell.* Since our lookup table is a matrix, we have to establish the line and column of the matrix.

Let us present the complete details of our calculation. The experiments presented in the current literature are using a linear lookup table (technically, a vector), corresponding to one calculus rule, initially defined. The new aspect of our study is that we use 256 different rules, dynamically selected; technically, the rules are corresponding to the lines of the val(256,8) matrix, each line representing one rule. After every life cycle, we apply the current operator of the experiment to determine the number of rule we are using. Further, we denote this value by *lin(k)* for the k life cycle. This calculus is specific to each experiment. This *“by life cycle”* selection of the calculus rule is what we call **dynamical modification of the loop function**. We further name val *the rules matrix*.

After the selection of the rule, we have to determine the column corresponding to the value of the *i* cell at the *k* life cycle. The calculus is a function of the values of the *i-1* (left neighbor), i and *i+1* (right neighbor) cells in the k-1 life cycle:

$$\text{col}(i,k) = c(i-1,k-1) \times 2^0 + c(i,k-1) \times 2^1 + c(i+1,k-1) \times 2^2 \qquad (1)$$

where:
- col(i,k) denotes the *value of the column for the i cell at the k life cycle*;
- c(i,k) denotes the *value of the i cell at the k life cycle*.

For cell 1 we use cell 256 as left neighbor, and for cell 256 we use cell 1 as right neighbor.

The value of the *i* cell in the *k* life cycle is:

$$c(i,k) = \text{val}(\text{lin}(k), \text{col}(i,k)) \qquad (2)$$

where:
- **c(i,k)** denotes *the value of the i cell at the k life cycle*
- lin(k) denotes *the specifical rule number for the k life cycle* (operator-specific)
- col(i,k) denotes *the result of the (1) calculus*
- val() denotes *the rules matrix*.

This two step ((1) + (2)) calculus is common for all projects. Further, when we present the projects, we have to detail only the calculus of the lin value.

That calculus function (returning the *lin* parameter) is what we name *the global operator.* An alternative name is *loop function*.

2.5. Characteristics and parameters

We simulate linear CA with 256 binary cells.

The common characteristics for all the experiments are:

- the CA are initiated randomly; this opens the possibility that, during the simulation, some structures are repeating. To obtain relevant data we made a great number of experiments;
- the study is focused on the relevance of specific parameters. We consider two parameters as defining an automaton: *the operator* and *the maximal lifetime duration admitted*. Following this idea, we developed the experiment in two directions: we used four different operators and, for each, we reiterated the simulations with three maximal lifetime duration admitted: 500, 1000 and 2000 cycles;
- we classify the CA according to their initial status. As an indicator we use the initial density. For CA with the same initial density we make no difference as regards the geometrical distribution of 0/1 cells;
- the monitoring is statistical-constative. Relevant for us are:
  a. *the distribution of cases,*
  b. *the ratio of cyclic CA*, and
  c. *the life duration of the blocked CA*.

For each project, there are two specific parameters:

1. the global loop function (the operator)
2. the maximal lifetime duration admitted

We made a two-level monitoring:

1. the distribution cases/initial density in the total mass of experiments;
2. by density. In this case, we are interested by the distribution of cyclic/blocked CA.

We reiterated each experiment with 3 maximum lifetime duration admitted: 500, 1000 and 2000 life cycles. For each experiment we simulated a number of 10.000.000 CA.

2.6. The general structure of one experiment

The experiment has a two level structure: *by automaton* (for each one) and *by experiment* (global monitoring).

For one automaton, the structure is as follows:

Determining the life cycle of one automaton

1. we define the initial status of the automaton (we initialize the automaton);
2. we apply to all cells the operator and obtain the new status of the automaton;
3. we test whether the automaton is blocked or it reached the maximal lifetime admitted:
   - if no, goto step 2
   - if yes, the experiment with the current automaton is concluded and we archive the results

For one experiment (one operator/one maximal lifetime admitted) the general structure is as follows:

The structure of one experiment

1. we initialize the automaton;
2. we determine its life cycle;
3. we archive the result;
4. we test if we simulated 10.000.000 cases:
   - if no, goto step 1;
   - if yes, the experiment is concluded.

3. The first experiment: CA with the operator SUM

We used as loop operator SUM: to compute lin(k), we computed the density (i.e. summed the values of the cells):

- at the first life cycle, we take lin(1) the initial density of CA;

- further, lin(k), is the current density of the automaton.

The results are synthesized in the following histograms:

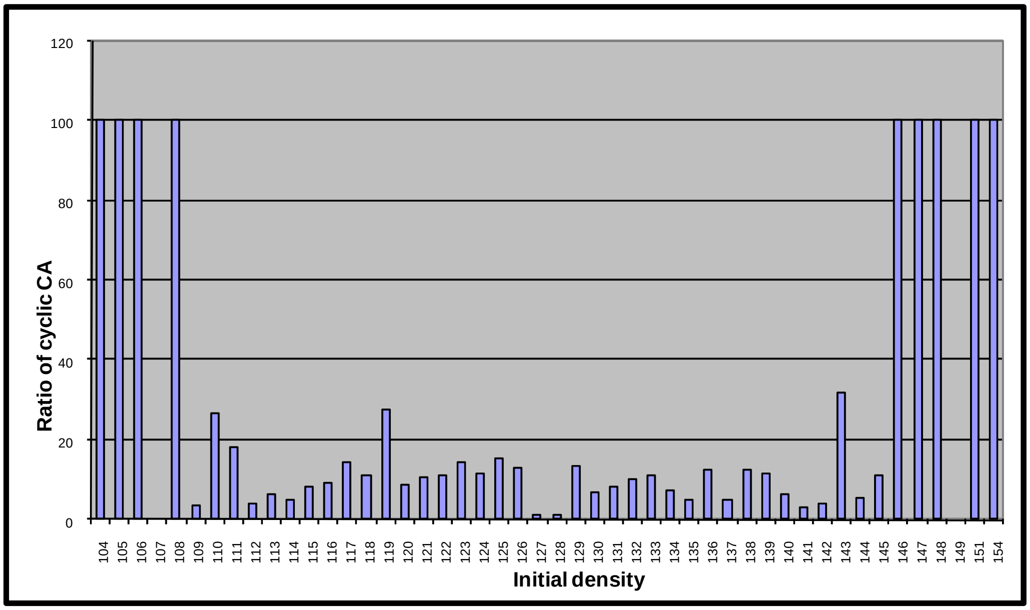

Fig 1. CA with the operator SUM and maximal lifetime duration admitted of 500 cycles. The distribution of the ratio of cyclic CA

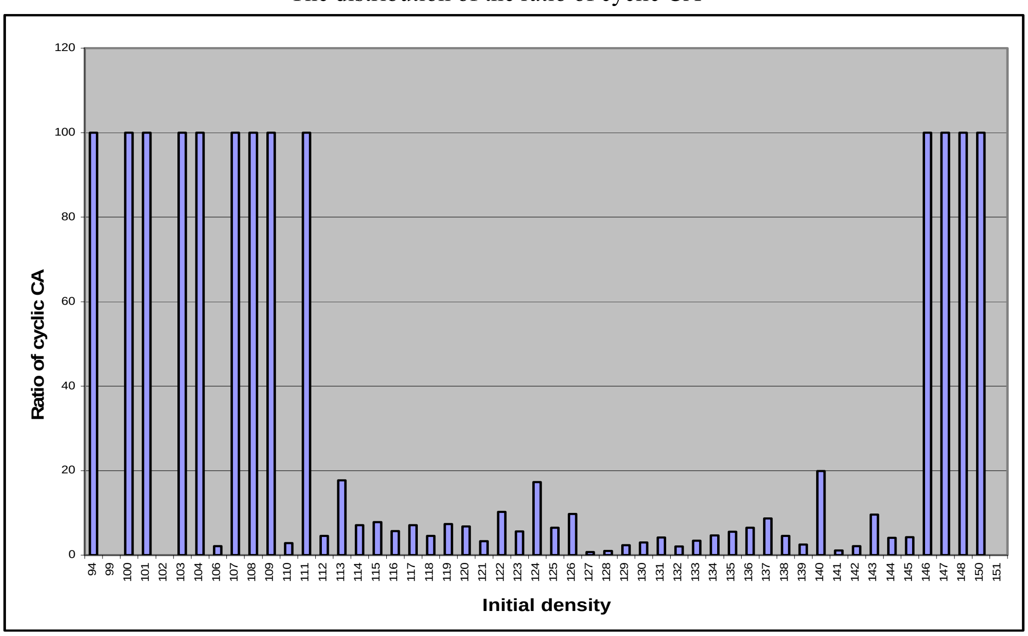

Fig. 2. CA with the operator SUM and maximal lifetime duration admitted of 1.000 cycles

The distribution of the ratio of cyclic CA

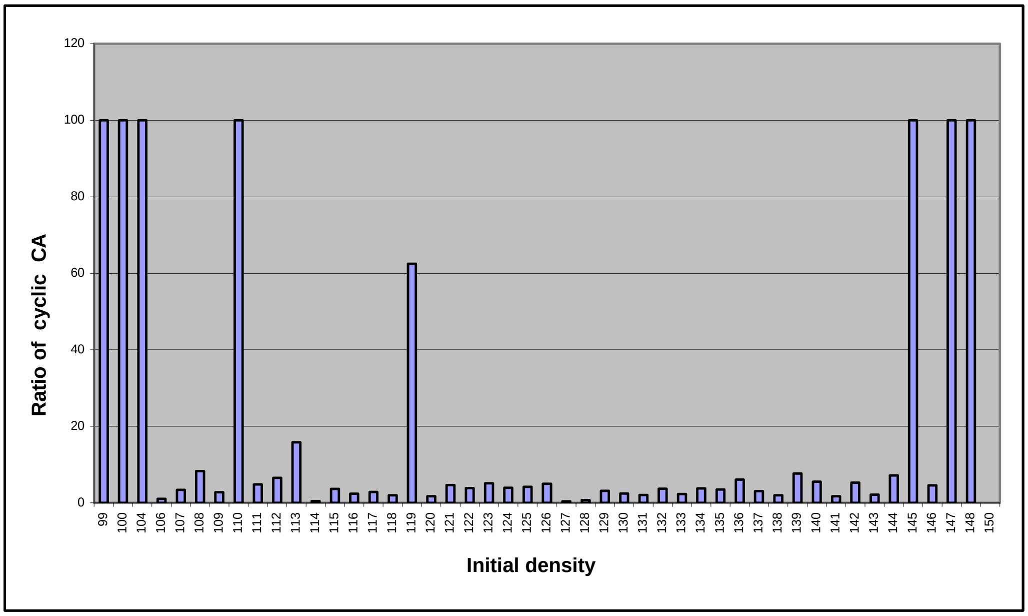

Fig. 3. CA with the operator SUM and maximal lifetime duration admitted of 2.000 cycles.
The distribution of the ratio of cyclic CA

Let us comment the results. The initial hypotheses were not confirmed by them:

- the 127, 128, 129 density area (the biggest number of possible cases) does not offer a greater percentage of cyclical CAs;
- we identify areas situated closer to the ranges of values of 100 and 150 (see histograms 1-3), where cyclical CAs appear in a percentage tending towards 100%;
- a greater maximal lifetime duration admitted does not influence significantly the percentage of cyclical CAs.

Let us take the opposite approach to the problem: to see which are the maximal performances of the blocked (noncyclic) CA. We attach the histogram of the maximal performances of the blocked (noncyclic) CA:

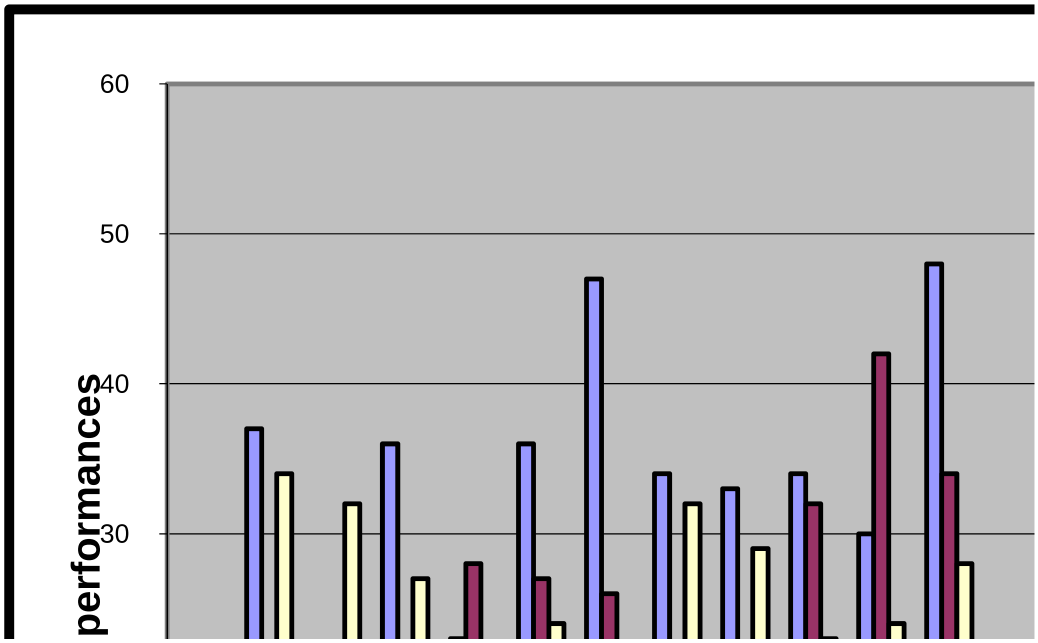

Fig. 4. CA with the operator SUM: maximal performances of the blocked (noncyclic) CA by maximal lifetime admitted

The first observation is that maximal lifetime duration admitted is not a relevant parameter for obtaining noisier CA, at least as regards this operator. The question remains open whether this aspect holds for the other operators too. We can reach a conclusion after we see the results for the other 3 operators.

On the other hand, histogram 4 raises also the first question as regards insight (and methodology). We notice that maximal performances of the blocked CA rarely pass beyond 50 life cycles. We have chosen maximal lifetime duration admitted of 500, 1.000 and 2.000 cycles. The first question is that whether in the behavior of cyclical CA (500, 1.000, 2.000 cycles) exists or not a cyclical part, or simply that they do not get blocked. Here we have a first direction of 'refining' the experiment: for a possible response, the pre-established maximal life time admitted should be replaced by a step-by-step monitoring of the evolution of the CA.

4. Conclusions for the first experiment

Let us begin with a possible reproach that can be done to our experiments. We generated the initial status of the CA randomly and we did not work on identical sets of data. This is the reason why we simulated 10.000.000 cases for every experiment: the large number of cases makes the results significant, even when the initial status of CA is partly different.

After this first experiment, we have already arrived at some conclusions.

The first one is the irrelevance, as a parameter, of the number of life-cycles when we consider the automaton cyclic and we automatically stop it (maximal lifetime duration admitted). The question that rises is whether this remains valid just for this operator. We will take again into consideration this aspect after having seen the results for the other 3 operators.

The most important result is the identification of the area of statistical exceptions: we notice, at the extremity of the interval of initially appeared density, that the automata are 100% cyclic. This points us to the area where we will make the first searches at the rerun of the experiment.

We have also identified a first direction of research at the rerun of the experiments, connected to insight. We have noticed that the life duration of the blocked CAs rarely surpasses 50 life-cycles. The question that issued was how evolve cyclical CA in the maximal lifetime admitted selected by us: they simply do not get blocked, or somehow new cycles appear (eventually repetitive). This direction suggests also a first 'refining' of methodology: it becomes necessary a step-by-step monitoring of the evolution of the CA, as well as an explicit test of cyclicity.

After seeing the results of the simulations with other operators, we will have a better image of the relevance of the operator. We will also have a better image of the working methodology we have used here.

R E F E R E N C E S

[1] *S. Wolfram,* "Random Sequence Generation by Cellular Automata", **in** Advances in Applied Mathematics, **7 (June 1986),** pp. 123-169

[2] *S. Wolfram*, A New Kind of Science**,** Wolfram Media, 2002

[3] *Gh. Ştefan*, "Looking for the lost noise", in CAS '98 Proceedings, Oct. 6 - 10, 1998, Sinaia, Romania. pp. 579 - 582

[4] *J. Borwein, D. Bailey & all.,* Experimental Mathematics in Action, A.K.Peters, 2006

[5] *J. Borwein, D. Bailey,* Mathematics by experiment: plausible reasoning in the 21'st century, 2nd ed., A.K.Peters, 2008

[6] *G. Chaitin*, INFORMATION, RANDOMNESS & INCOMPLETENESS. Papers on Algorithmic Information Theory, Second Edition, World Scientific, 1990

[7] *** C/C++ Microsoft User's Manual.