ELSEVIER

Parallel Computing 23 (1997) 1673–1682

PARALLEL COMPUTING

CA-like error propagation in fuzzy CA

Paola Flocchini [a,*], Frédéric Geurts [b], Nicola Santoro [c]

[a] *Département d'Informatique et de Recherche Opérationelle, Université de Montréal, Montréal, Canada H3C 3J7*

[b] *Département d'Ingénierie, Université Catholique de Louvain, FSA / INGI, Place Sainte Barbe 2, B-1348 Louvain-la-Neuve, Belgium*

[c] *School of Computer Science, Carleton University, Ottawa, Canada K1S 5B6*

Received 15 July 1996; revised 7 March 1997

Abstract

We describe and analyze the surprising evolution of an overflow error which has occurred during the simulation of a class of continuous complex systems called 'fuzzy cellular automata'. This error not only perturbs the global behavior of the system but it generates its own interesting dynamics; in fact, it propagates over the original evolution as a binary cellular automaton in a structured, fractal-like way, regardless of the underlying system evolution. The analysis of the errors behavior suggests interesting observations about the difficulty of detecting errors and verifying results using computer simulations. © 1997 Elsevier Science B.V.

Keywords: Fuzzy cellular automata; Error propagation; Continuous–discrete states interaction; Diseased–healthy bodies interaction; Simulation

1. Introduction

Cellular automata (CA) have been introduced by Von Neumann and Ulam as models of self-organizing and self-reproducing behaviors [8]. Their versatility and simplicity have permitted their use in many disciplines, ranging from ecology and biology to theoretical computer science [5,7,9].

CA are discrete-time discrete-space models: the local space of each component is discrete and finite, the local function is defined by a look-up table. The global evolution is synchronous: at each step, all cells update their value according to their local

* Corresponding author. E-mail: flocchini@iro.umontreal.ca.

PII S0167-8191(97)00077-X

transition table and the values of their direct neighbors. Binary CA are such that the local state space of all cells is {0,1}.

A *continuous* version of CA has been defined in [1,2] to classify complex CA behaviors, and it has been used to simulate and to study the interaction between knowledge and uncertainty, the influence of noise in distributed information, and the propagation of diseases in healthy organisms (e.g. [3,4]). This model is called *fuzzy* CA since its local transition function is obtained by 'fuzzifying' the local rule and states of a classical binary CA.

Our interest in using this model was to observe how continuous values interact with binary ones. For example, in the context of evolutionary computing, this interaction can be interpreted as the effect of diseased cells in a healthy body: some organisms are self-healing, others are completely destroyed, and some others evolve symbiotically with the disease [3,4].

To study those behaviors, we have observed the space–time diagram of fuzzy CA, associating different colours to different states. At a certain time, an overflow error occurred in the calculation of a cell state. The presence of an error was visible in the evolution since one cell assumed a colour different from all the other cells, and, from that moment on, it fastly propagated forming very interesting and peculiar patterns. In fact, the error propagated with the same structure as a binary CA, regardless of and destroying the underlying system evolution.

The error and its evolution are fully reproducible, and as shown in this paper, its behavior can be fully characterized. The error appears and behaves in a CA-like way, giving rise to a rich fractal-like dynamics.

The paper is organized as follows. in Section 2, we present the model we use; in Section 3, we informally describe the phenomenon and illustrate it with some figures; in Section 4, we analyze the propagation structure of errors; in Section 5, we give an evolutionary interpretation of the analysis; and finally, in Section 6, we conclude by discussing the relevance of the reported phenomenon.

2. Model

A fuzzy CA is obtained by 'fuzzification' of the local evolution rule of a classical binary CA. An elementary fuzzy CA is thus a linear array of cells which evolves in time. Each cell of the array assumes a state in [0,1] and changes its state according to a *local evolution* function on its state and the states of its two neighbours. The *global evolution* results from the synchronous application of the local rule to all cells of the array. The model can thus be seen as a particular case of Coupled Map Lattice [6].

The *local rule* of a binary CA is given by 8 binary tuples, corresponding to the 8 possible local configurations a cell can detect in its direct neighborhood:

$$(000, 001, 010, 011, 100, 101, 110, 111) \rightarrow (r_1, \dots, r_8).$$

Each triplet represents a local configuration of the left neighbour, the cell itself, and the right neighbour. In general, we convert the binary representation $(r_1, \dots r_8)$ to the decimal one; the rule is then called $\sum_{i=1:8} 2^{i-1} r_i$.

Example 1. *Let us consider rule* $14 = 2 + 4 + 8$:

$$(000, 001, 010, 011, 100, 101, 110, 111) \rightarrow (0, 1, 1, 1, 0, 0, 0, 0).$$

The local rule is canonically expressed as a *disjunctive normal form*:

$$f(x_1, x_2, x_3) = \vee_{i \mid r_i = 1} \wedge_{j=1:3} x_j^{d_{ij}},$$

where d_{ij} is the jth digit, from left to right, of the binary expression of i, and x^0 (respectively x^1) stands for $\neg x$ (respectively x).

Example 2. *The canonical expression of rule* 14 is:

$$f_{14}(x_1, x_2, x_3) = (\neg x_1 \wedge \neg x_2 \wedge x_3) \vee (\neg x_1 \wedge x_2 \wedge \neg x_3) \vee (\neg x_1 \wedge x_2 \wedge x_3).$$

The *fuzzification* process works as follows: we replace $(a \vee b)$ by $(a + b)$, $(a \wedge b)$ by (ab), and $(\neg a)$ by $(1 - a)$. The local rule becomes a real function simulating the original function on $\{0,1\}^3$:

$$f(x_1, x_2, x_3) = \sum_{i=1:8} r_i \prod_{j=1:3} l(x_j, d_{i,j})$$

where $l(a,b) = 1 - a - b + 2ab$.

Example 3. Again, in the case of rule 14, simplifying the expression, we have:

$$f_{14}(x_1, x_2, x_3) = (1 - x_1) \cdot (x_2 + x_3 - x_2 \cdot x_3).$$

The function l correctly implements the previous logical exponentiation: $l(a,0) = 1 - a$ and $l(a,1) = a$. However, we will see that its expression strongly influences the error propagation structure.

3. Experiment

We fix a CA rule that we fuzzify to obtain a fuzzy CA, the lattice of which contains 1500 cells. The initial configuration is randomly chosen: cells 1–700 and 801–1500 are set to binary values; cells 701–800 take real values in [0,1]. We observe the evolution of the 500 central cells during 500 steps, which makes the boundary condition unimportant. We now consider what happens when, during the global evolution, an overflow error occurs locally, due to some faulty computation.

A single error propagates in a structured fractal-like way, and the introduction of more errors leads to the formation of complex patterns, following the natural evolution of a binary CA. These strong errors let the system change its dynamics, as genetic mutations in organisms. The original system is progressively destroyed, giving its place to the new evolving system. Let us illustrate the phenomenon with some figures. Each one of them shows the evolution of a one-dimensional fuzzy CA in which the vertical axis represents time going downwards, and the horizontal axis represents space.

- The first example in Fig. 1 shows the effect of one error on the global evolution of rule 184. The error propagates in a structured way, developing fractal-like patterns.

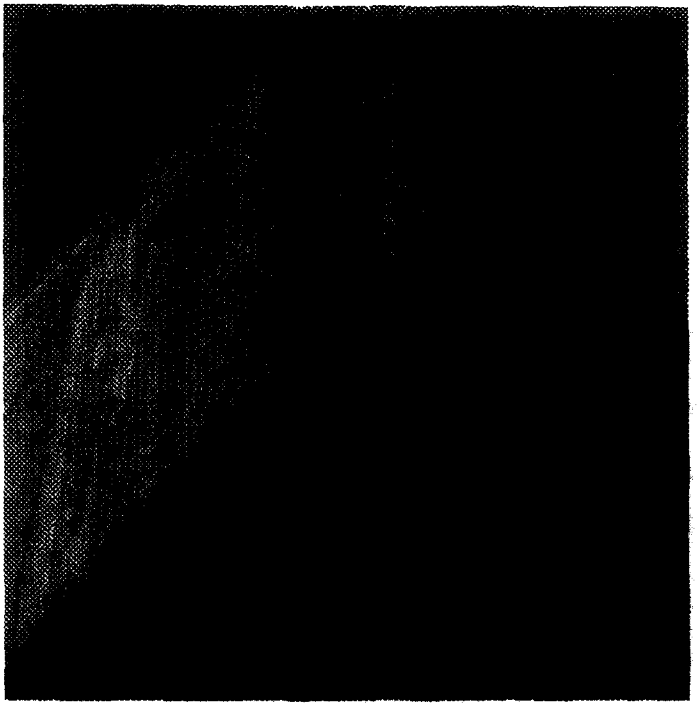

Fig. 1. Effect of one error in the evolution of rule 184.

Moreover, the error region is perfectly symmetric, whatever the local rule is.

- In Fig. 2, we have four occurrences of the error at different times in the evolution of rule 105. Before any interaction happens, the evolution is as structured as in the first case. When the error regions collapse, the structure progressively disappears, though the underlying order of the error is preserved. Note the error region very resembles the normal evolution (see Section 4).
- In Fig. 3, we observe the propagation of two consecutive blocks of random errors in the evolution of rule 18, and their interaction; each block contains 50 errors. At the beginning, there is no apparent structure but some small patterns are generated as the system proceeds, as if the order was present under the original disorder.

Note that the pattern generated by the interaction of only four independent errors (Fig. 2) is more complex than the one generated by the interaction of one hundred errors grouped into two segments (Fig. 3).

In each situation, the errors cover more and more cells, eventually invading the whole lattice, in a predatory way.

4. Structural analysis: CA-like behavior

The occurring error does not come from nowhere, it results from an error in the object code of a simulation program, from the treatment of overflow errors by the

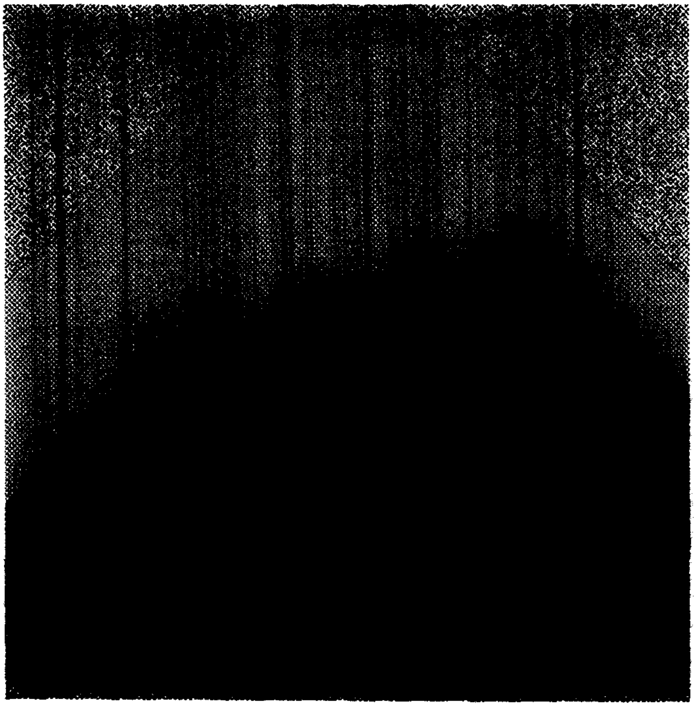

Fig. 2. Effect of separate errors in the evolution of rule 105.

compiler, and from the way the local rule is implemented. In this case, the function l presented hereabove (Section 2) plays a very important role.

The error (let us call it E) is fully reproducible. By simulation, it is easy to verify that its inverse exists, $-E$, and $-(-E)=E$. Moreover, its effect on simple computations is the following: $\forall x \in [0,1]$, $y \in [0,1] \cup \{E\}$,

$$E(-E) = -E, \quad Ex = E, \quad \pm E \pm y = \pm E.$$

Then we note that its effect on l is a kind of negation: $l(\pm E, \cdot) = \mp E$.

The local function of fuzzy CA is implemented as follows:

$$f(x_1,x_2,x_3) = \sum_{i=1}^{8} r_i \prod_{j=1}^{3} l(x_j,d_{ij}).$$

If at least one argument is erroneous, the second argument of l does not matter anymore; $\prod_{j=1}^{3} l(x_j,d_{ij})$ does not depend on i and is erroneous for all i. Since $\forall x \in [0,1]$, $Ex = E$, the coefficients r_i can be removed, and by the third observation, i.e. $E \pm y = E$, the first term of $\sum_{i=1}^{8} \prod_{j=1}^{3} l(x_j, \cdot)$ determines the result:

$$f(x_1,x_2,x_3) = \prod_{j=1}^{3} l(x_j, \cdot).$$

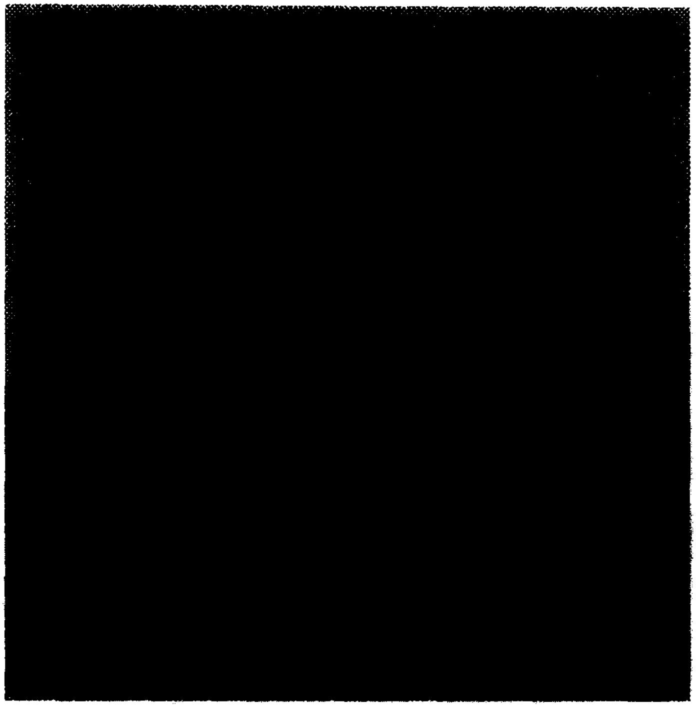

Fig. 3. Effect of two groups of errors in the evolution of rule 18.

Remark that the ordering of the arguments does not influence the result; it only determines the propagation direction of the error: in this sense,

$$f(\cdot,E,\cdot) = \pm E \neq f(\cdot,\cdot,E) = \pm E.$$

Using these facts, we are able to compute the effect of E on the local function of our model; since only additions and multiplications are required, we can find out the rules which describe the propagation of the error.

Four general cases must be considered, regarding the number of errors in the arguments of the local function. In Fig. 4, the evolution starts at the top with one error in the central cell, and the next configurations are computed according to the local rule.

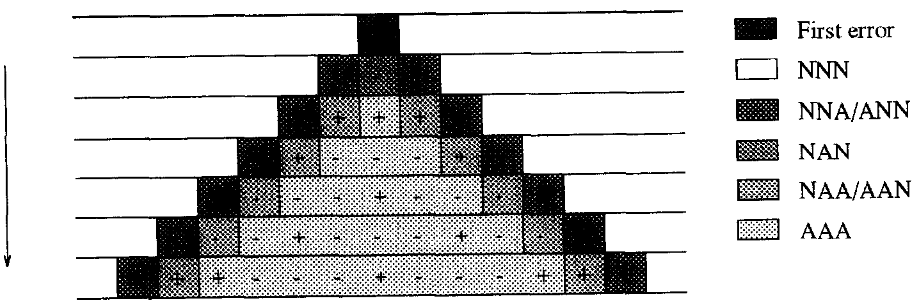

Fig. 4. Space–time propagation structure of errors.

Different shades of gray correspond to different kinds of configurations, from no errors (NNN) to three (AAA). In the following, x and y are normal, viz. healthy, values of the interval [0,1], and e_1, e_2, $e_3 \in A = \{E, -E\}$ are abnormal, viz. erroneous, values.

Case NNN. If all arguments are normal, the cell behaves as prescribed by the normal (or healthy) local rule.

Case N²A. In case of one error and two normal arguments, we have an inverter:

$$f(e_1, x, y) = -e_1.$$

The rule table describing this behavior is the following, independently of the location of the error in the neighborhood:

Nbd	f
$E\ x\ y$	$-E$
$-E\ x\ y$	E

If the arguments are disposed as *NNA*, a left shift effect is added. A right shift comes from the symmetric disposition *ANN*. In case of a single error introduced in the lattice, the configuration *NAN* is observed once, and it inverts the error without any shift. When several errors are introduced, this last situation never occurs.

Case NA². If we have two errors and one normal argument, the result is essentially a multiplication of signs:

$$f(e_1, e_2, x) = e_1 \cdot e_2.$$

Again, the location of the error in the neighborhood do not influence the behavior. The local rule table is the following:

Nbd	f
$E\ \ E\ x$	E
$E\ -E\ x$	$-E$
$-E\ \ E\ x$	$-E$
$-E\ -E\ x$	E

Interpreting E as 0 and $-E$ as 1, this table describes an exclusive disjunction. In case of NAA, we have a left shift, and a right shift in the symmetric case AAN.

Case A^3. When all arguments are erroneous, which happens in the central part of the error pattern, the local function is slightly more complex:

$$f(e_1, e_2, e_3) = -e_1 \cdot e_2 \cdot e_3.$$

Its rule table reads:

Nbd			f
E	E	E	$-E$
E	E	$-E$	E
E	$-E$	E	E
E	$-E$	$-E$	$-E$
$-E$	E	E	E
$-E$	E	$-E$	$-E$
$-E$	$-E$	E	$-E$
$-E$	$-E$	$-E$	E

Interpreting E as a 0 (respectively 1) and $-E$ as a 1 (respectively 0), the behavior of the error between the two borders is described by rule 105.

In summary, the error follows three different types of CA-rules. At the external border between the error and the 'normal' evolution of the fuzzy CA, the behavior of the error is described by a precise CA-rule for the configurations of the form $(x, y, \pm E)$ and $(\pm E, y, z)$, and by another CA-rule for the configurations of the form $(x, \pm E, \pm E)$ and $(\pm E, \pm E, z)$, where $x, y, z \in [0,1]$; finally, inside the error region, the error propagates following the local rule of an elementary binary CA with states E and $-E$. We can thus explain how these nicely structured patterns can arise from the propagation of overflow errors. The evolution law of the error is increasingly richer from outside to inside the error region. We have a shifting inverter on $\{E, -E\}$, a shifting exclusive disjunction on $\{E, -E\}^2$, and a CA rule 105 on $\{E, -E\}^3$.

Remark. The symmetric pattern created by the error in Fig. 1 cannot be generated by a classical CA rule 105 alone. The boundaries imposed by the external behaviors N^2A and

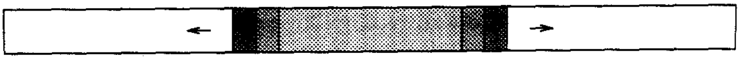

Fig. 5. One-dimensional organism growing in another organism.

NA^2 play a very important role in the resulting global regularity, as they strongly destroy and replace the original underlying dynamics.

More precisely, suppose $(x_0,\dots,x_n)$ is a given initial configuration of a CA and f is the local rule; the dynamics of the evolution of the overflow error can be seen as a binary CA with the following border conditions:

$$f(x_{-2},x_{-1},x_0) = -x_0, \quad f(x_n,x_{n+1},x_{n+2}) = -x_n,$$
$$f(x_{-1},x_0,x_1) = x_0 \oplus x_1, \quad f(x_{n-1},x_n,x_{n+1}) = x_{n-1} \oplus x_n,$$

where $\oplus$ is the binary exclusive-or.

5. Evolutionary interpretation

Once the error occurs, it behaves like a growing body enclosed in a destructive expanding *membrane* (see Fig. 5).

This membrane has two layers. The *external* membrane replaces one normal cell at each step by an error, alternating in time positive and negative values. The *internal* membrane follows the external one and computes its next state using an exclusive disjunction between itself and the previous external membrane state.

Inside the developing organism, the evolution is very close to the behavior of a classical binary CA, computing a next state according to the value of a direct neighborhood.

The structure present in Fig. 1 depends on the combination of these factors, plus the fact that it originates from a single error. As shown in the other figures, the structure resulting from a collapse of two membranes is richer and apparently less ordered. However, the order progressively appears when the growing organism has enough time to impose its own structure onto the underlying structure.

The interaction between error and fuzzy CA can be easily summarized in evolutionary terms. A complex organism A (the fuzzy CA) is penetrated by a strongly destructive organism B (the error), which develops itself inside A, and uses an impermeable membrane to attack A's body and propagate. The behaviors of A and B are different. Their evolutions are independent. Eventually, A totally disappears, leaving B alive in its body. This behavior is typical of a *predator-prey interaction*.

6. Concluding remark on simulation

Simulation is very useful to understand the evolution of complex systems. It serves as intuitive guide to more formal results. Very often, experiments are the base of statistical results. It is thus very important to know the influence errors can have on such results.

In this paper, we have described a surprising phenomenon illustrating the way computing errors can behave in the evolution of complex systems like coupled map lattices (CML) and, more specifically, fuzzy CA. We have observed that a single error not only perturbs the global behavior of the system, which weakens the consequences of simulation results, but it generates a very interesting dynamics by itself. If the error is not detected, its patterns can be mistaken for the evolution of a classical CA.

One error propagates in a structured fractal-like way, and the introduction of more errors leads to the formation of complex patterns, following the natural evolution of a binary CA.

Acknowledgements

Paola Flocchini is partially supported by an International Fellowship from the Italian CNR. Frédéric Geurts is supported by the Belgian *Fonds National de la Recherceh Scientifique*, and Nicola Santoro by grant #A2415 from NSERC.

References

[1] G. Cattaneo, P. Flocchini, G. Mauri, C. Quaranta Vogliotti, N. Santoro, Cellular automata in fuzzy backgrounds, Physica D vol. 105, 1997, pp. 105–120.

[2] G. Cattaneo, P. Flocchini, G. Mauri, N. Santoro, Fuzzy cellular automata and their chaotic behavior, in: Proc. International Symposium on Nonlinear Theory and its Applications, volume 4, Hawaii, IEICE, 1993, pp. 1285–1289.

[3] P. Flocchini, F. Geurts, N. Santoro, Dynamics and algebraic properties of moss reinforced shifts, in: Proc. International Symposium on Nonlinear Theory and its Applications, volume 2, Las Vegas, IEICE, 1995, pp. 1149–1152.

[4] P. Flocchini, N. Santoro, The chaotic evolution of information in the interaction between knowledge and uncertainty, in: R.J. Stonier, X.H. Yu (Eds.), Complex Systems, Mechanism of Adaptation, IOS Press, 1994, pp. 337–343.

[5] M. Garzon, Models of Massive Parallelism – Analysis of Cellular Automata and Neural Networks, Springer-Verlag, 1995.

[6] K. Kaneko, Theory and Application of Coupled Map Lattices, John Wiley and Sons, 1993.

[7] C.G. Langton, Studying artificial life with cellular automata, in: Evolution, Games, and Learning, North-Holland, Amsterdam, 1986.

[8] J. Von Neumann, Theory of Self-Reproducing Automata, University of Illinois Press, Urbana, 1966.

[9] S. Wolfram, Theory and Applications of Cellular Automata, World Scientific, 1986.