Uniform and Non-uniform Cellular Automata

This content is based on the book chapter: Romano, A. L. T. ; Villanueva, W. J. P. ; Zanetti, M. ; Von Zuben, F. J. . Synthesis of Spatio-temporal Models by the Evolution of Non-uniform Cellular Automata. In: A. Abraham; A.-E. Hassanien; A.P.L.F. Carvalho (Eds.). Foundations of Computational Intelligence - Bio-Inspired Data Mining. Springer, Chapter 4, pp. 85-104, 2009.

1 Introduction

- Cellular automata are idealizations of physical systems where time and space are discrete and the physical quantities, or cellular states, can assume a finite set of values.
- The transition rules can be seen as an expression of a microscopic dynamics that leads to a desired macroscopic behavior (FRISCH *et al.*, 1986).
- The existence of a regular lattice, a well-defined neighborhood, a transition rule and an initial state for the cells in the lattice, taken from a finite set of possible states, gives rise to the cellular automata mathematical model.
- The transition rule is a function that depends on the states of the neighboring cells and which leads each cell to the next state at each time step, synchronously or asynchronously (SCHÖNFISCH & DE ROOS, 1999) with the other ones in the lattice. So, each cell state is a fixed function of its previous state and of the previous state of the neighboring cells.

- The dimension of the lattice will certainly depend on the spatial dimension of the physical phenomenon under investigation. In the literature, we can find one-dimensional (TOMASSINI & PERRENOUD, 2001), two-dimensional (RABINO & LAGHI, 2002), and three-dimensional (BASANTA *et al.*, 2004) lattices.
- The neighborhood of each cell is previously defined and is generally assumed to have the same shape for each cell in the lattice, even in the case of non-uniform cellular automata. Three usually adopted two-dimensional configurations for the neighborhood are illustrated in Figure 1.

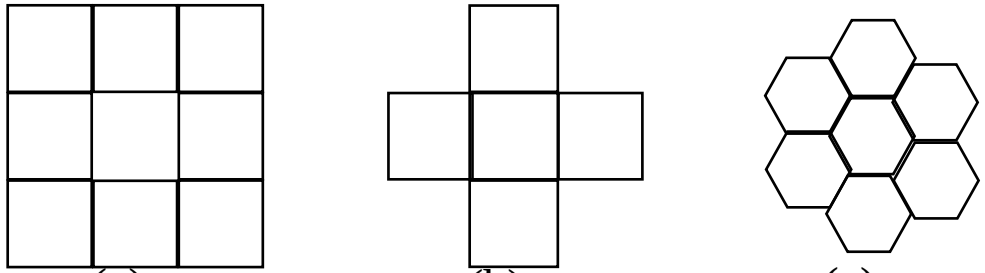
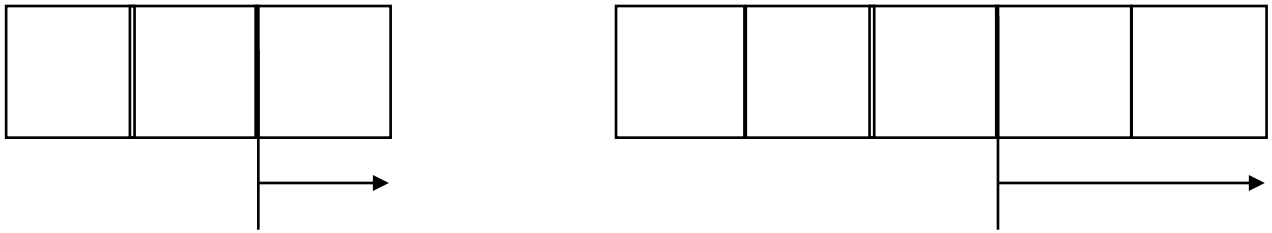


Figure 1 – Examples of neighborhoods for a two-dimensional lattice: (a) Moore; (b) von Neumann; and (c) Hexagonal.

- The simplest version of cellular automata is the binary-state one-dimensional one, in which each cell can only assume the values 0 or 1. Examples of the neighborhood of this CA are presented in Figure 2.



(a) Radius 1 (b) Radius 2

Figure 2 – One-dimensional lattice: (a) Neighborhood with radius 1; and (b) Neighborhood with radius 2.

- In the one-dimensional lattice, the number of cells involved in the updating of the state of a given cell is 2*Radius+1. For the binary-state one-dimensional cellular automata, the neighborhood can assume $2^{2*Radius+1}$ possible configurations, and

the number of transition rules for a given cell is $2^{2^{2*Radius+1}}$. This is the size of the search space for the simplest cellular automata that can be conceived, supposing that each cell will obey the same transition rule.

- For the non-uniform case, the cardinality of the set of all possible transition rules is given by $\left(2^{2^{2*Radius+1}}\right)^n$, where n is the number of cells in the one-dimensional lattice.

2 Non-uniform cellular automata

- Non-uniform (NunCA) or inhomogeneous cellular automata (SIPPER, 1994) are spatio-temporal models for dynamical systems in which space and time are discrete, and there is a distinct transition rule for each cell, with a finite number of states. The cells are in a regular lattice and the transition from one state to another is performed synchronously.

- The next state of a given cell will then be provided by a local and fixed transition rule that associates its current state and the current state of the neighbouring cells with the next state.
- The neighbourhood could also be specific for each cell, but here will be considered the same, except for the cells at the frontiers of the regular lattice. So, the only distinct feature between NunCA and the traditional uniform cellular automata (CA) (TOFFOLI & MARGOLUS, 1987, VON NEUMANN, 1961) is the adoption of a specific transition rule for each cell instead of a single transition rule for all the cells in the lattice.
- Both CA and NunCA have been applied to a wide variety of scenarios, including (but not restricted to):
  - ✓ **CA**: physical systems modeling (CHOPARD, 1998; NAGEL & HERRMANN, 1993), ecological studies (COLLASANTI & GRIME, 1993; JAI, 1999), computational applications (TOFFOLI & MARGOLUS, 1987; WOLFRAM, 1994);

- ✓ **NunCA**: VLSI circuit design (KAGARIS & TRAGOUDAS, 2001; TSALIDES, 1990), computational applications (SIPPER, 1994; TOMASSINI & PERRENOUD, 2001; TOMASSINI *et al.*, 1999; VASSILEV *et al.*, 1999).

- NunCA has one predominant advantage over CA, i.e. the greater flexibility to define the transition rules, which can be explored to produce dynamic behaviors not (easily) obtainable by means of a single rule.

- So, the possibility of updating the state of each cell following local and distinct rules can be explored to conceive synthetic universes from simple rules, with the emergence of complex spatio-temporal structures.

- Instead of investigating and/or exploring the computational power of NunCA, the purpose here is to provide **a systematic procedure to achieve a mathematical model for the NunCA framework capable of reproducing a sequence of spatio-temporal behaviors**. Two scenarios will be considered:

1. For one-dimensional lattices: given a desired sequence of state transitions, the aim is to determine one of the possibly multiple set of fixed, though distinct, rules that is capable of driving the sequential transition of states according to the desired profile. This has already been performed in the case of uniform CA (MITCHELL *et al.*, 1993), and the main purpose here is to indicate that some profiles cannot be achieved when a single transition rule is defined for all cells in the lattice. In such a case, only the NunCA framework can fulfill the task.

2. For two-dimensional lattices: given the initial and final states of each cell, the aim is to determine one of the possibly multiple set of fixed, though distinct, rules that is capable of driving the sequential transition of states from the initial to the final one, with the final state as a stationary configuration. The trajectory between the initial and final states, denoted transitory phase, can be of interest or not. A case study will be considered in

which the transitory phase is left unrestricted, and another case study will impose some restrictive conditions to the intermediary states.

- The great challenge of such a formulation is the necessity of defining the whole set of transition rules, one rule for each cell in the regular lattice. The necessity of as many rules as cells has precluded a wider dissemination of similar approaches. In what follows, we will present the necessary steps toward the synthesis of an evolutionary design of these transition rules.
- After a successful determination of an appropriate set of transition rules, the interpretation of the resulting NunCA may take place, even though the obtained set of rules is generally just one of the possible solutions.
- The interpretation is easier in the case of one-dimensional lattices, because the states are binary, but relevant information can be extracted from the resulting two-dimensional lattices too, where multivalued states are considered.

- The use of multivalued states can be interpreted as a quantization of the continuous state case, where the NunCA would then be equivalent to a non-uniform coupled map lattice (KANEKO, 1993) and a discrete-time cellular neural network (CHUA & YANG, 1988).

3 The motivation for non-uniform lattices

- Every dynamical event whose description involves the evolution of variables in time and space is called a spatio-temporal phenomenon. Examples of these dynamics include dispersion, expansion, contraction, and local interrelation of groups of elements, and can be associated with living and other physical phenomena in nature (CAMAZINE *et al.*, 2001; CHOPARD, 1998; NICOLIS & PRIGOGINE, 1977).
- Most of these spatio-temporal systems are continuous in space and time. However, a computational model will necessarily require the quantization of space, in the

form of regular lattices, and the use of a discrete-time dynamics to update the state of the cells in the lattice, denoted a transition rule.

- The use of cellular automata models is generally associated with one of two purposes:

    1. Classification of the spatio-temporal behavior;
    2. Reproduction of a predefined behavior from the simplest transition rule that can be defined.

- The second approach is the one of interest here and has been explored in the literature in distinct ways, as:

    - ✓ An architecture for fast and universal computation (WOLFRAM, 1994);
    - ✓ An alternative paradigm for the investigation of computational complexity (WOLFRAM, 1994);
    - ✓ Pattern recognition tools (MAJI *et al.*, 2002);
    - ✓ Modeling devices (GREGORIO & SERRA, 1999).

- Our primary concern here is the last item: **cellular automata as powerful models of actual physical phenomena**. The main motivation is the possibility of reproducing specific behaviors in space and time, always based on simple transition rules.
- GUTOWITZ & LANGTON (1988) have pointed out that lattices with interesting behavior are the ones that achieve a tradeoff between high-level and low-level of dependence among neighboring cells.
- With distinct transition rules for each cell, the level of inter-cell dependence may be established with more flexibility, when compared with the existence of a single transition rule to be followed by every cell. **In fact, a uniform CA can be interpreted as a particular case of a non-uniform CA, here denoted NunCA**.
- To achieve a proper tradeoff capable of reproducing the desired spatio-temporal behavior, powerful search devices should be conceived to determine a proper set of transition rules.

- Evolutionary computation has already been demonstrated to provide effective procedures to optimize parameters of a single transition rule in uniform and binary-state cellular automata (Mitchell *et al.*, 1993).

- That is why we are going to extend the already proposed evolutionary approaches to search for an optimal set of parameters for each transition rule in the non-uniform case.

- As a transition rule of a given cell will represent a local binding to the neighboring cells, multiple equivalent transition rules can be capable of reproducing the same global behavior, so that we will be interested in finding just one of them.

4 Formalism for non-uniform bidimensional cellular automata

- In the case of a two-dimensional $n \times m$ lattice, only von Neumann neighborhoods will be implemented, with multivalued states. Only non-uniform cellular automata

will be considered, and the definition of the transition rule is based on the notation of Figure 3.

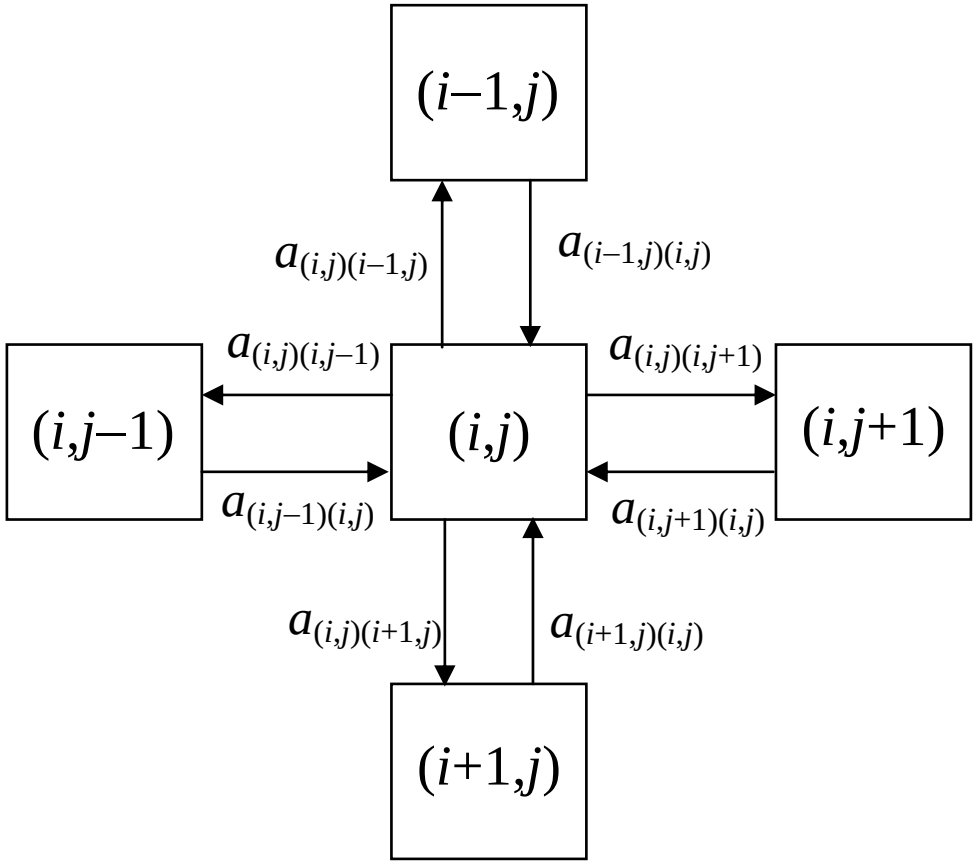


Figure 3 – Parameters of the transition rules.

- The spatio-temporal behavior will be associated with the flux of some material from cell to cell, according to the von Neumann neighborhood. The state of each cell is the concentration of that material and the set of four parameter values $\left\{a_{(i,j)(i,j+1)}, a_{(i,j)(i+1,j)}, a_{(i,j)(i,j-1)}, a_{(i,j)(i-1,j)}\right\}$ defines the amount that will be transferred from cell (i,j) to cell (k,p), where the indices k and p are defined according to the corresponding neighbor cell.

- Being interpreted as rate of flux, the following restrictions are imposed:
  - ✓ $a_{(i,j)(i,j+1)} \geq 0;\ a_{(i,j)(i+1,j)} \geq 0;\ a_{(i,j)(i,j-1)} \geq 0;\ a_{(i,j)(i-1,j)} \geq 0;$
  - ✓ $a_{(i,j)(i,j+1)} + a_{(i,j)(i+1,j)} + a_{(i,j)(i,j-1)} + a_{(i,j)(i-1,j)} \leq 1;$
  - ✓ $a_{(i,j)(k,p)} = 0$ when (k,p) is an absent neighbor, motivated by the fact that (i,j) is a cell at the frontier of the lattice.

- When dealing with uniform cellular automata, the following additional restrictions are necessary:

- ✓ $a_{(i,j)(i,j+1)} = a_{(i,j-1)(i,j)}$;
- ✓ $a_{(i,j)(i+1,j)} = a_{(i-1,j)(i,j)}$;
- ✓ $a_{(i,j)(i,j-1)} = a_{(i,j+1)(i,j)}$;
- ✓ $a_{(i,j)(i-1,j)} = a_{(i+1,j)(i,j)}$.

- So, given that $c_{(i,j)}(t)$ is the concentration of material at cell (i,j) in the instant t, the transition rule for cell (i,j) is given by:

$$c_{(i,j)}(t+1) = \left(1 - a_{(i,j)(i,j+1)} - a_{(i,j)(i+1,j)} - a_{(i,j)(i,j-1)} - a_{(i,j)(i-1,j)}\right)c_{(i,j)}(t) +$$
$$a_{(i,j+1)(i,j)}c_{(i,j+1)}(t) + a_{(i-1,j)(i,j)}c_{(i-1,j)}(t) + a_{(i,j-1)(i,j)}c_{(i,j-1)}(t) + a_{(i+1,j)(i,j)}c_{(i+1,j)}(t)$$

where $i \in \{1,...,n\}$ and $j \in \{1,...,m\}$. Notice that n can be taken equal to m in a square lattice. When a non-toroidal neighborhood is considered, every time that i=1 and/or j=1, the terms involving indices i–1 and j–1 are null, and the same happens with the terms involving i+1 and j+1 when i=n and/or j=m.

5 Methodology for the evolutionary design

5.1 Genetic algorithms for one-dimensional and binary-state cellular automata

- Genetic algorithms (GAs) (GOLBERG, 1989) have been successfully applied to the synthesis of uniform cellular automata (MITCHELL *et al.*, 1996; OLIVEIRA *et al.*, 2001).
- Inspired by the process of natural selection, a GA maintains a population of candidate solutions in a genotypic representation, and mutation and recombination operators (GOLBERG, 1989) are then conceived to promote a proper exploration of the search space in a population-based mechanism.
- Selection is performed to implement the principle of the survival of the fittest, and individuals with higher fitness values have a high probability of being selected to spread their genetic material to the next generation of individuals.

- The recursive application, generation after generation, of selection and genetic operations, together with local search procedures when available, tends to promote an increase in the average fitness of the population, at least in the fitness of the best individual at each generation.
- Better fitness means a candidate solution with better quality. Every problem will have its own fitness function.
- In one-dimensional lattices composed of *n* cells, each individual will be a binary vector describing the single transition rule, in uniform cellular automata, and the whole set of transition rules, in non-uniform cellular automata. In fact, the codification will interpret each possible configuration of the neighborhood (given by a sequence of 2**Radius*+1 bits, where *Radius* is the order of the neighborhood) as an integer index, and this index will indicate the position of its corresponding next state in the transition rule.

Table 1 – An example of transition rule (3rd column) for a neighborhood of Radius = 1

Configuration	Index	Next State
000	0	0
001	1	1
010	2	1
011	3	0
100	4	1
101	5	0
110	6	0
111	7	1

- As an example, taking *Radius* = 1, Table 1 presents in the third column a possible transition rule, so that every configuration of neighborhood has an indication of next state, e.g. when the neighborhood achieve 100 then the next state of the cell under analysis will suffer a transition from 0 to 1, and for a neighborhood 010, the state remains the same (equal to 1).

- In the non-uniform case, the genetic codification of a transition rule will be given by a binary vector whose size is *n* times the size of the binary vector in the uniform case, because each cell can have a distinct next state for each configuration of the neighborhood.
- The fitness function will be simply given by the inverse of 1 plus the Hamming distance between ⟨the observed evolution of states in time⟩ and ⟨the desired one⟩.
- When a given transition rule is capable of exactly reproducing the spatio-temporal behavior, then the Hamming distance will be zero and the fitness will achieve the maximum value. The highest the Hamming distance, the smallest the fitness value, so that the fitness is restricted to fit in the range (0,1].
- The initial condition of the automata is arbitrarily defined and is considered fixed.
- The flowchart in Figure 4 depicts the main steps of the adopted GA. A local search is also applied every time a new individual is obtained. This local search

consists in definitely changing one of the next states suggested by the transition rule if this change turns to improve the overall performance of the cellular automata.

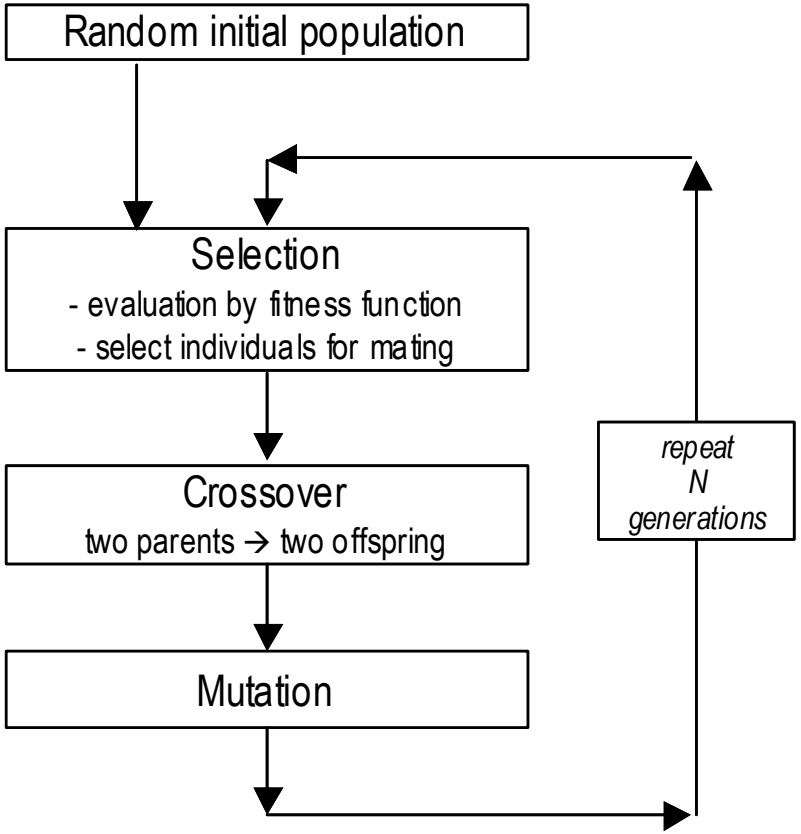


Figure 4 – Flowchart of a Simple GA

5.2 Evolution strategy for two-dimensional and multivalued-state cellular automata: the non-uniform case

- Evolution Strategies (ESs) (BÄCK *et al.*, 1991; SCHWEFEL, 1981) have primarily been proposed to serve as a searching device for the optimization of continuous-valued parameters in a wide variety of applications.
- The mutation operator is the basic genetic operator and the next generation is obtained from the current population by means of one of two strategies: (μ,λ) or $(\mu+\lambda)$.
- In the (μ,λ) conception, the population is composed of μ individuals and λ new individuals are generated from each one of the μ ancestors. Then μ individuals are selected solely from the offspring.
- On the other hand, in the $(\mu+\lambda)$ framework, the same happens except for the way the μ individuals are selected to compose the next generation: the ancestors and

the λ offspring are candidates to the next generation, and the μ individuals with the highest fitness are then selected.

- The (1+1) is the simplest version of evolution strategy, where one parent creates one single offspring via Gaussian mutation.
- Parameters of the Gaussian distribution may be evolved together with the individuals, incorporated into the genetic code. The recombination may be implemented as done in genetic algorithms, as here we have adopted uniform crossover (GOLDBERG, 1989).
- In the flowchart in Figure 5, describing the basic steps of the algorithm, the individuals are formed by the attributes of the solution candidate and the variance to be used by the Gaussian mutation operator (BEYER & SCHWEFEL, 2002). Every time that the search space is composed of feasible and unfeasible candidate solutions, additional procedures should be incorporated to deal with feasibility issues.

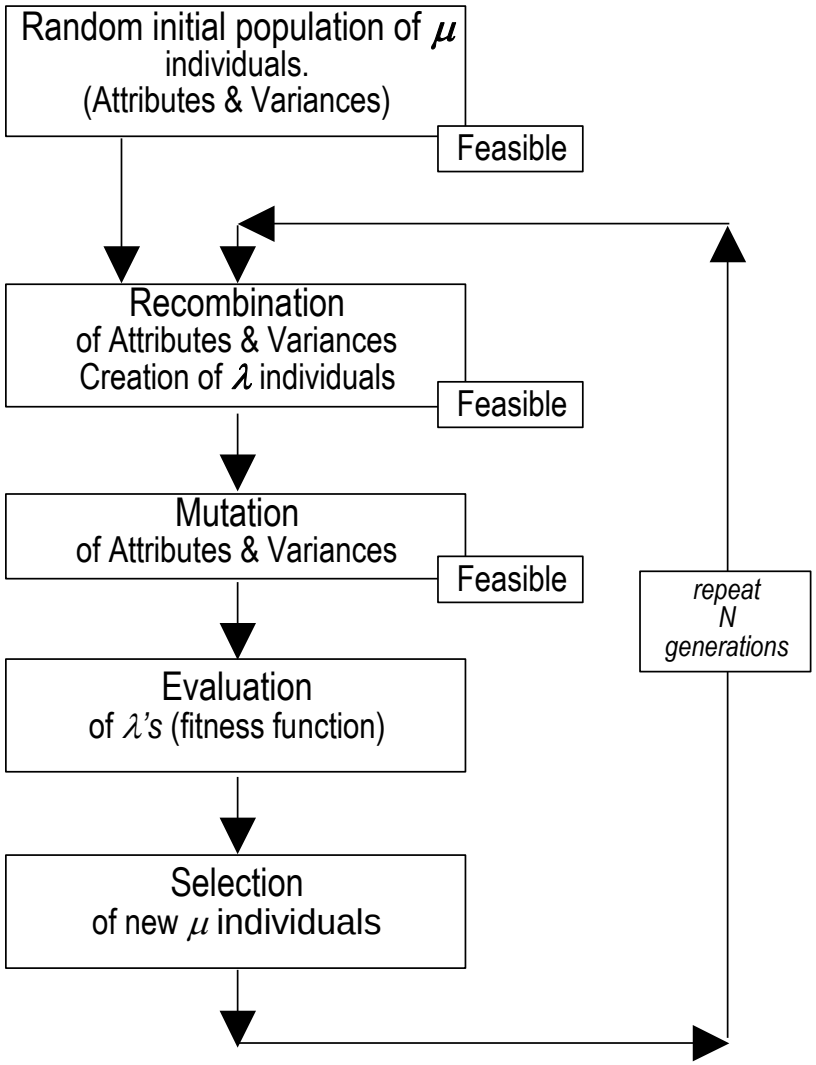


Figure 5 – Flowchart for the Evolution Strategy

- In terms of codification of the attributes, Figure 3 indicates that, in a two-dimensional $n \times m$ lattice, each cell (i,j), $i=1,...,n$ and $j=1,...,m$, will require four parameters in the genetic codification. So, in the NunCA framework, the size of the chromosome will be $4*n*m$.

- The fitness will be given by the inverse of one plus the sum of the squared difference between the desired final state of each cell and the obtained final state. When intermediary states are of concern, additional terms will be included in the fitness function. As in the one-dimensional lattice, here the initial condition of the automata is arbitrarily defined and is considered fixed.

6 Related approaches and possible extensions

- SIPPER (1994) proposes an evolutionary-like and local procedure to update transition rules for binary states, including the possibility that one cell changes the state of a neighbor cell and copies itself onto that neighbor cell.

- Vacant cells, i.e. cells without a transition rule, are also accepted. However, the applicability was restricted to binary NunCA and requires specific operators to evaluate the fitness of individual rules, according to its local success, when applied to updating the state of its corresponding cell. Such methodology can hardly be directly extended to deal with global description of the intended spatio-temporal behavior.
- VASSILEV *et al*. (1999) proposed a co-evolutionary procedure to deal with transition rules for binary states, and the spatio-temporal event under investigation was global synchronization.
- LI (1991) investigated partially and totally wiring (non-local CAs) and pointed out that the connection profile is decisive in the emergence of certain dynamical behaviors, like edge of chaos and attractors of convergent dynamics.

- Structurally dynamic cellular automata (SDCA) are generalizations of uniform and non-uniform CA such that the lattice itself is part of the optimization process (HILLMAN, 1995; ILACHINSKI & HALPERN, 1987).

7 Experimental results

- Our experiments aim to show the flexibility of the NunCA approach when compared to the conventional uniform CA. We are going to consider two scenarios: a one-dimensional lattice with binary-state cells, and a two-dimensional lattice with multivalued-state cells.
- In the former case, the purpose is to reproduce a sequence of state transitions in time, and in the latter case the intent is to obtain a non-uniform cellular automata capable of converging to a predefined final state, starting from an initial state and having the intermediary states submitted to some restrictive conditions or not. In both cases, the initial condition was set arbitrarily.

- In the two-dimensional lattice, the transition rules admit an interpretation in terms of a local pattern of dispersion of a given material. This is one of the possible physical interpretations of the spatio-temporal dynamical model.
- The set of binary rules for each cell in the one-dimensional CA was obtained via GA with a binary code (see Fig. 4), and the real values of the rules for each cell in the two-dimensional CA were provided by an evolution strategy (see Fig. 5).
- The individuals in the population are transition rules, and to evaluate each individual the corresponding cellular automata should be implemented and executed along time. Every discrete instant of time is relevant in the one-dimensional lattice, but in the two-dimensional lattice the final state may be the only relevant information or it may be considered together with the intermediary states. With the restrictions imposed to the parameter values of cell (i,j), presented in section 4, the dynamic of the two-dimensional non-uniform cellular automata is guaranteed to be convergent.

- The fitness function for the one-dimensional case will be given by:

$$F = \frac{1}{1 + d_{Hamm}(SS_{des}, SS_{obt})}$$

where $d_{Hamm}(\cdot,\cdot)$ is the Hamming distance between matrices SS_{des} and SS_{obt}, which contain respectively the desired and obtained sequence of states of the one-dimensional cellular automata. The number of columns equals the number of cells in the lattice, and the number of rows equals the number of state transitions along time.

- Though you will see two-dimensional pictures in Figures 6, 7, 8 and 9, they are just the representation of matrices SS_{des} and SS_{obt}, with the time evolution being represented by the sequence of rows. The gray (green) represents state 0 and the black (dark blue) corresponds to state 1.

- On the other hand, in the two-dimensional lattice the fitness has two alternative expressions. When the final state is the only relevant information, the fitness function is expressed as follows:

$$F = \frac{1}{1 + \sum_{i=1}^{n} \sum_{j=1}^{m} \left(c_{(i,j)}^{end} - \hat{c}_{(i,j)}^{end} \right)^2}$$

where $c_{(i,j)}^{end}$ and $\hat{c}_{(i,j)}^{end}$ are respectively the desired and the obtained final states of cell (i,j) in the lattice, with the state being associated with the concentration of a given material.

- When intermediary states are also relevant, one possibility is to express the fitness function in the form:

$$F = \frac{1}{\left(1 + \sum_{k=1}^{N} \sum_{i=1}^{n} \sum_{j=1}^{m} \left(c_{(i,j)}^{k} - \hat{c}_{(j,i)}^{k} \right)^2 \right)}$$

where $c_{(i,j)}^{k}$ and $\hat{c}_{(i,j)}^{k}$ are respectively the desired and the obtained states of cell (i,j) in the lattice, at instant k, and N is the number of intermediary states under consideration.

- In the experiments to be presented in what follows, we will adopt an alternative fitness function that emphasizes the necessity of a symmetrical dispersion, so that cells in opposite sides of a two-dimensional lattice should have similar concentrations along time. The expression is given by:

$$F = \frac{1}{\left(1+\sum_{i=1}^{n}\sum_{j=1}^{m}\left(c_{(i,j)}^{end}-\hat{c}_{(i,j)}^{end}\right)^2\right)\left(1+\sum_{k=1}^{N}\sum_{i=1}^{n-1}\sum_{j=i+1}^{m}\left(\hat{c}_{(i,j)}^{k}-\hat{c}_{(j,i)}^{k}\right)^2\right)}$$

7.1 One-dimensional lattice

- All the results in this subsection have been obtained with a genetic algorithm.

7.1.1 Experiment 1: One-dimensional CA – "synchronization"

- In the synchronization task, the objective is to alternate the one-dimensional lattice between states 1 and 0, so that every cell in the lattice share the same state at a given instant, and simultaneously change to the complementary state in the next instant. As already emphasized along the text, the initial configuration of states is arbitrary, though fixed. The *Radius* of the neighborhood is 1.

- Figure 6(a) presents the results obtained with the NunCA approach, with Figure 6(b) depicting the set of rules for each cell in the lattice. The set of transition rules for each cell is represented in each column of Figure 6(b). Given that the neighborhood is 1, we have eight possible configurations for the binary states of neighbor cells, and the transition rules should indicate the next state to every possible configuration. Figure 7 presents the best result obtained with a uniform CA. Figure 7(b) shows the unique transition rule for the uniform CA, and Figure 7(a) indicates that the uniform CA was incapable of solving the task.

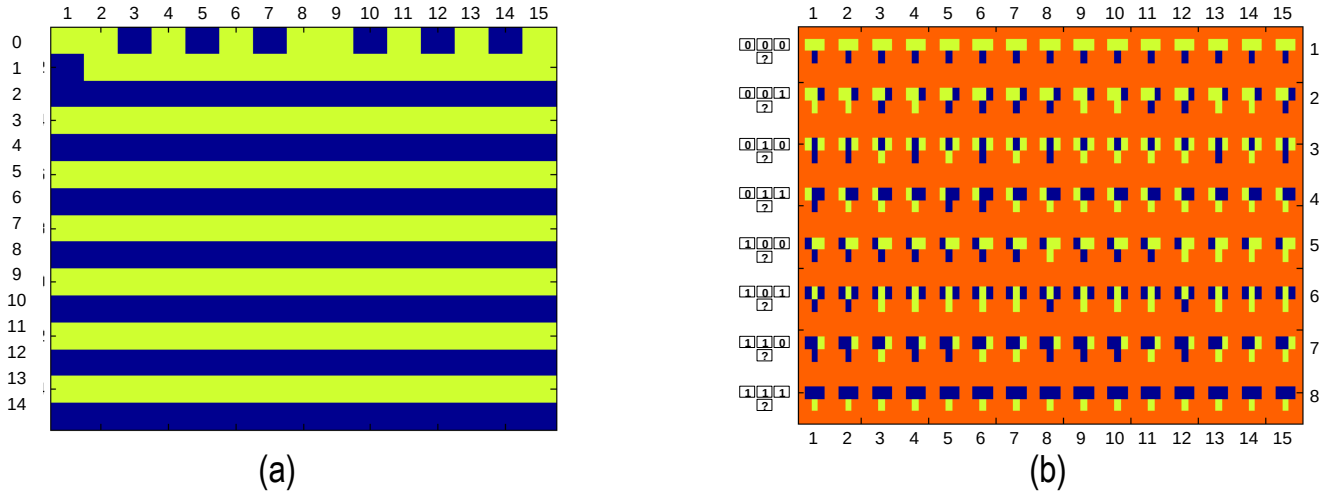


Figure 6 – Results for Experiment 1 using NunCA

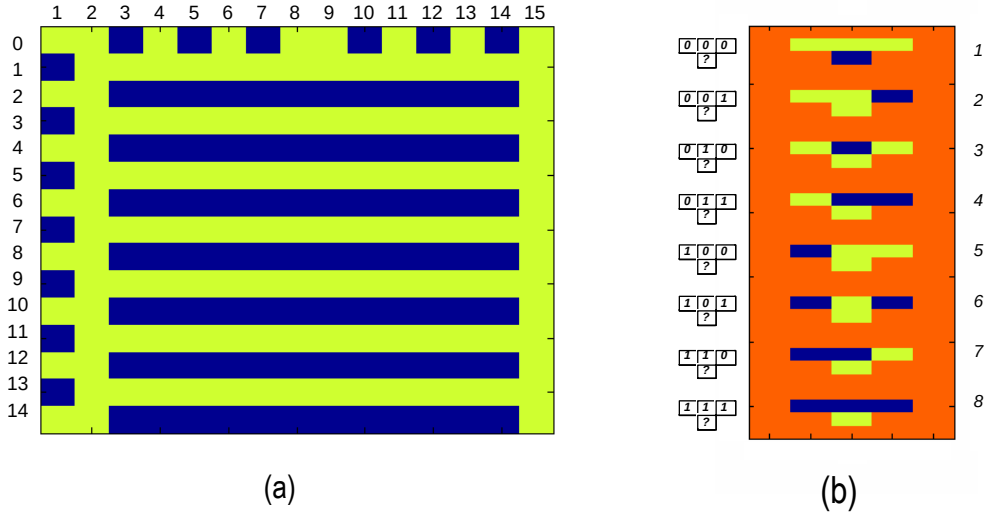


Figure 7 – Results for Experiment 1 using uniform CA

- Returning to Figure 6(b), which represents a successful implementation of the synchronization effect, we can see that no pair of cells shares the same transition rule, indicating that non-uniformity is a necessity here. Notice that this set of non-uniform transition rules may not be the only one capable of reproducing the desired behavior.

7.1.2 Experiment 2: One-dimensional CA – "waves"

- Experiment 2 consists in reproducing the temporal pattern that resembles the behavior of sinusoidal waves. Again, Figure 8 shows a successful performance of NunCA, and Figure 9 indicates that uniform CA fails to achieve the desired spatio-temporal behavior, because the best obtained behavior is far from the desired one.

- Figure 8(b) shows that each cell is associated with a distinct transition rule, again a strong indication of the complexity of the task and of the flexibility inherent to the NunCA framework.

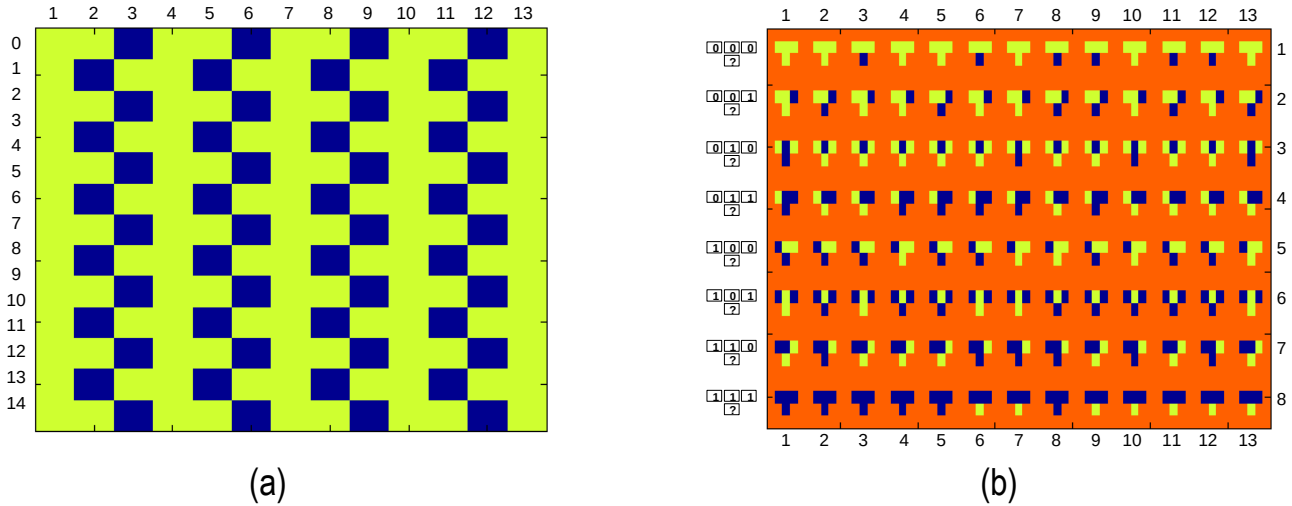


Figure 8 – Results for Experiment 2 using NunCA

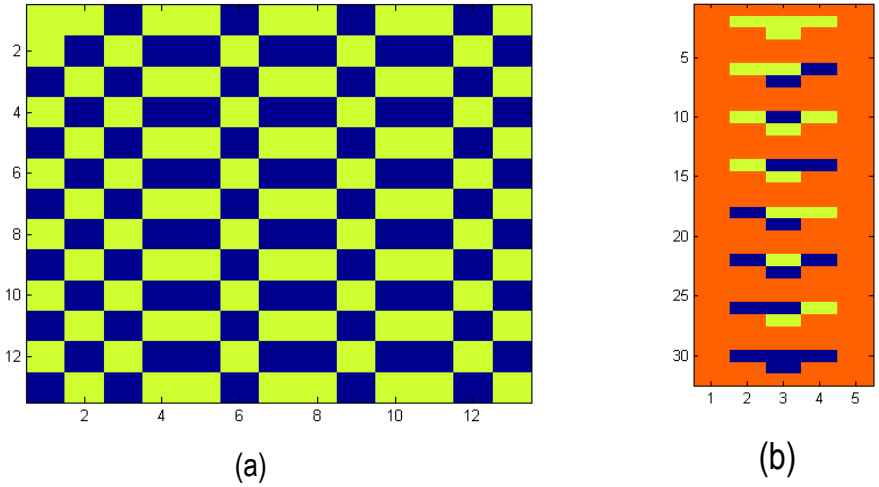


Figure 9 – Results for Experiment 2 using uniform CA

- In Experiment 2, using a neighborhood with *Radius*=2, a uniform CA gains enough representation power to accomplish the task that was successfully executed by a NunCA with *Radius*=1.

7.2 Two-dimensional lattice

- Now we will analyze some experiments involving two-dimensional lattices and multivalued states. The synthesis of the desired spatio-temporal behavior will now be implemented by an evolution strategy.

- The motivation for such experiments is the possibility of emulating dispersion phenomena in a great range of applications. In Experiments 3 and 4 we adopted a toroidal neighborhood, i.e. right-most cell is a neighbor of the left-most one, in the same row, and the up-most cell is a neighbor of the bottom-most one, in the same column. However, in Experiments 5 and 6 cells at the frontier of the lattice can not promote dispersion to the outside world, so that the dispersion is restricted to happen in a compact two-dimensional space.

7.2.1 Experiment 3: Two-dimensional CA – "homogeneous distribution"

- Experiment 3 is illustrated in Figure 10 and the purpose is to start with a maximum concentration of material at the cell in the centre of the lattice. The final convergent state will be a homogeneous distribution of concentration along the cells. So, if we start with 100 in the central cell of the lattice (see Figure 10(a)), and the lattice has a 5×5 dimension, the desired final concentration per cell will be 4.

- Here, each candidate NunCA should be put in operation and the convergence of the dynamics is measured by means of a threshold. When the sum of the square distance between two consecutive states (each term in the summation corresponds to a cell in the lattice) is below a predefined threshold, the convergence is detected and the fitness of that proposal is then evaluated. The best NunCA, obtained by the evolutionary search procedure based on an evolution strategy, produces the behavior illustrated in Figure 10 when put in operation.

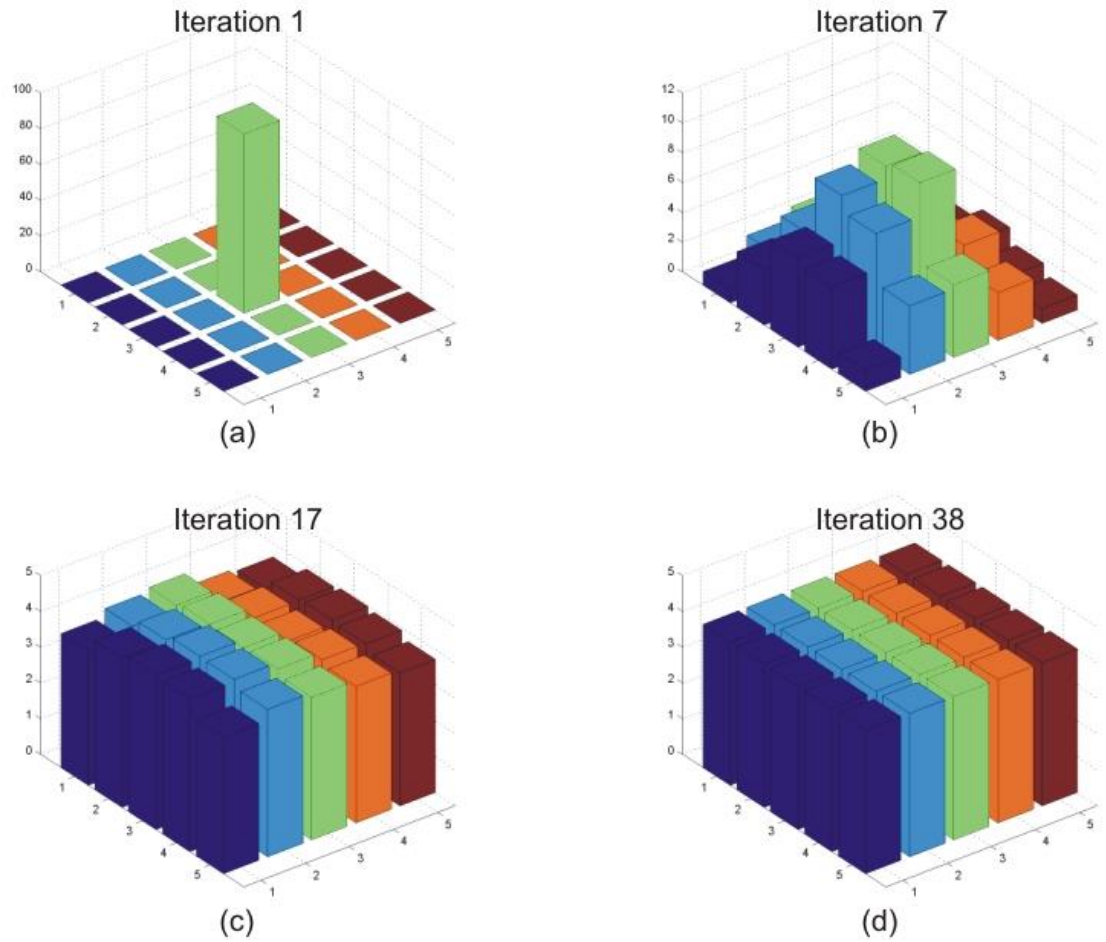


Figure 10 – Convergence of the dynamic for Experiment 3, produced by the best evolved NunCA.

- Figure 14(a) shows the gradient of the dispersion for this experiment, extracted from the interpretation of the resulting set of parameters for each transition rule. As expected, there is no preferential direction of dispersion. Even with an unbalanced profile for the obtained gradient of dispersion, we have the emergence of a homogeneous equilibrium.

7.2.2 Experiment 4: Two-dimensional CA – "contour"

- Experiment 4 is illustrated in Figure 11. In this experiment, the objective was to equally distribute all the initial mass at the frontier of the lattice, so dividing the initial mass by 16. As a consequence, starting with 100 at the central cell, we want to obtain 6,25 in each of the sixteen cells at the frontier. Figure 11(d) presents the convergent state, indicating the ability of the best evolved NunCA to reproduce the desired spatio-temporal behavior.

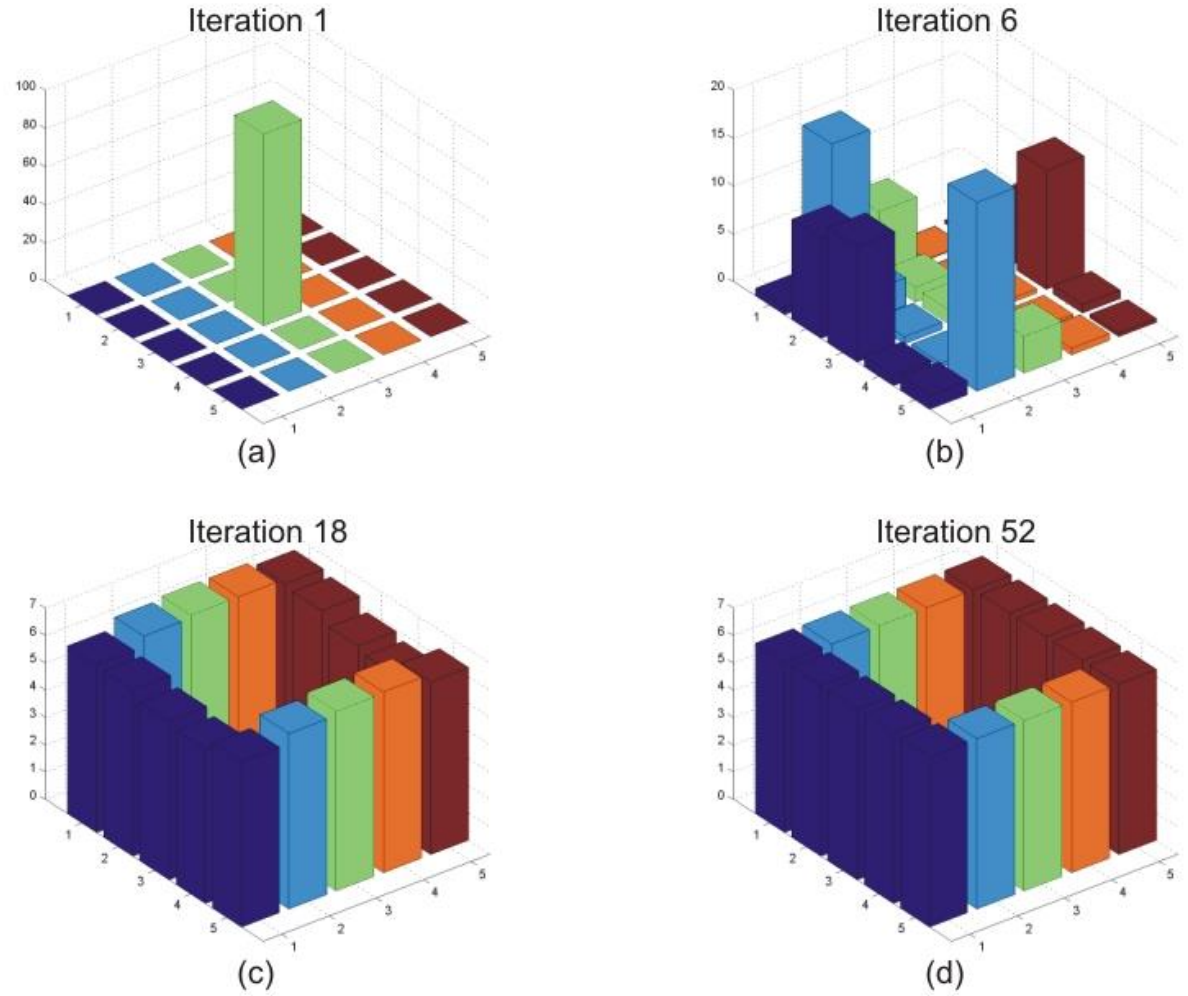


Figure 11 – Convergence of the dynamic for Experiment 4, produced by the best evolved NunCA.

- Figure 14(b) shows the gradient of dispersion for this experiment. We can see that there is a preferential direction of dispersion pointing from the centre to the borders of the lattice.

7.2.3 Experiment 5: Two-dimensional CA – "barrier"

- Experiments 5 and 6 are the most complex to be considered here, and Experiment 5 is illustrated in Figure 12. In this experiment, the purpose was to move all the initial mass in cell (1,1), the one at the top-left corner, to cell (5,5), the one at the bottom-right corner. However, there is a barrier at cells (4,2), (3,3) and (2,4), so that the gradual transfer of mass must be accomplished avoiding the obstacle at the centre of the lattice.
- Notice that cells (4,2), (3,3) and (2,4) has no transition rule and can not receive or deliver any amount of mass. Figure 12 shows the result and Figure 14(c) shows the gradient of dispersion for this experiment.

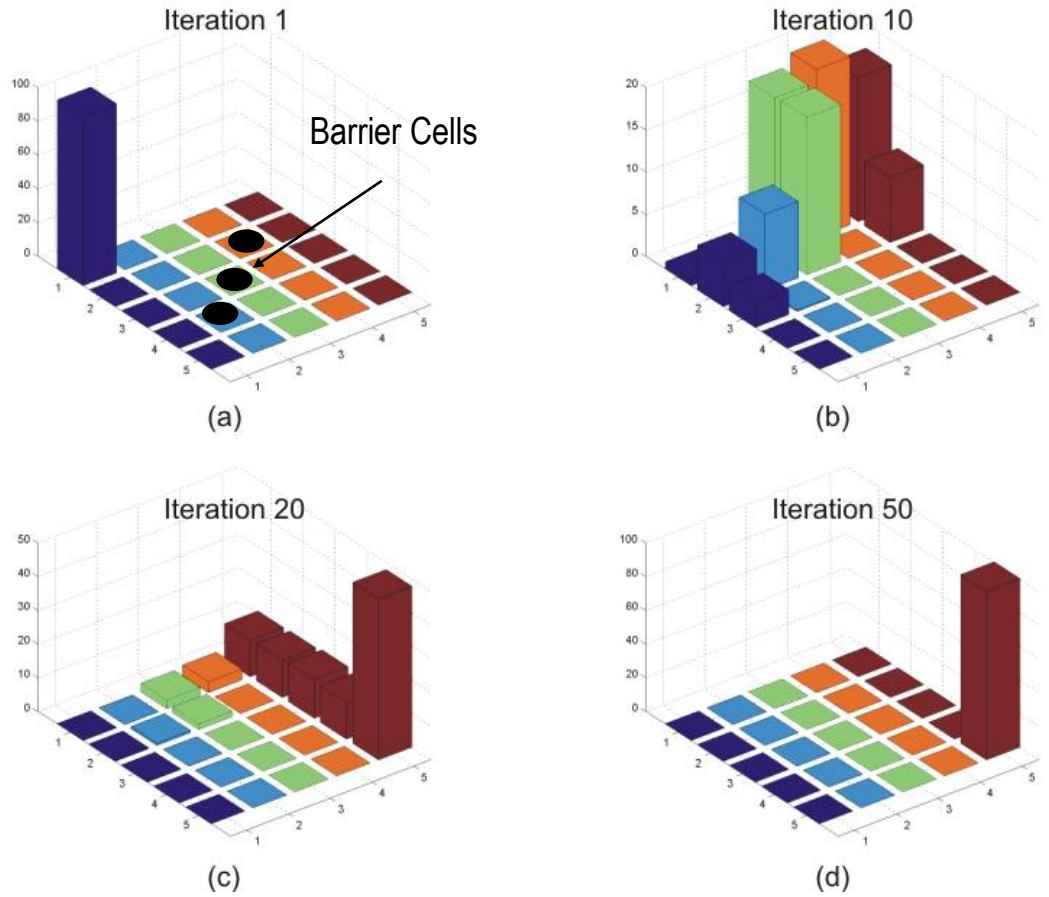


Figure 12 – Convergence of the dynamic for Experiment 5, produced by the best evolved NunCA.

- It can be inferred from Figure 14(c) that the obtained solution forces the dispersion to follow the path through the top-right corner only. Of course, the bottom-left corner could have been considered as well, and Experiment 6 will impose an additional restriction requiring that the dispersion be symmetrical between both corners.

7.2.4 Experiment 6: Two-dimensional CA – "barrier with symmetrical dispersion"

- Experiment 6 involves the same scenario already presented in Experiment 5, with the additional restriction of having a symmetrical dispersion along both sides of the barrier. Figure 13 indicates that the best evolved NunCA was capable of producing the intended spatio-temporal behavior, and Figure 14(d) shows the gradient of dispersion for this experiment.

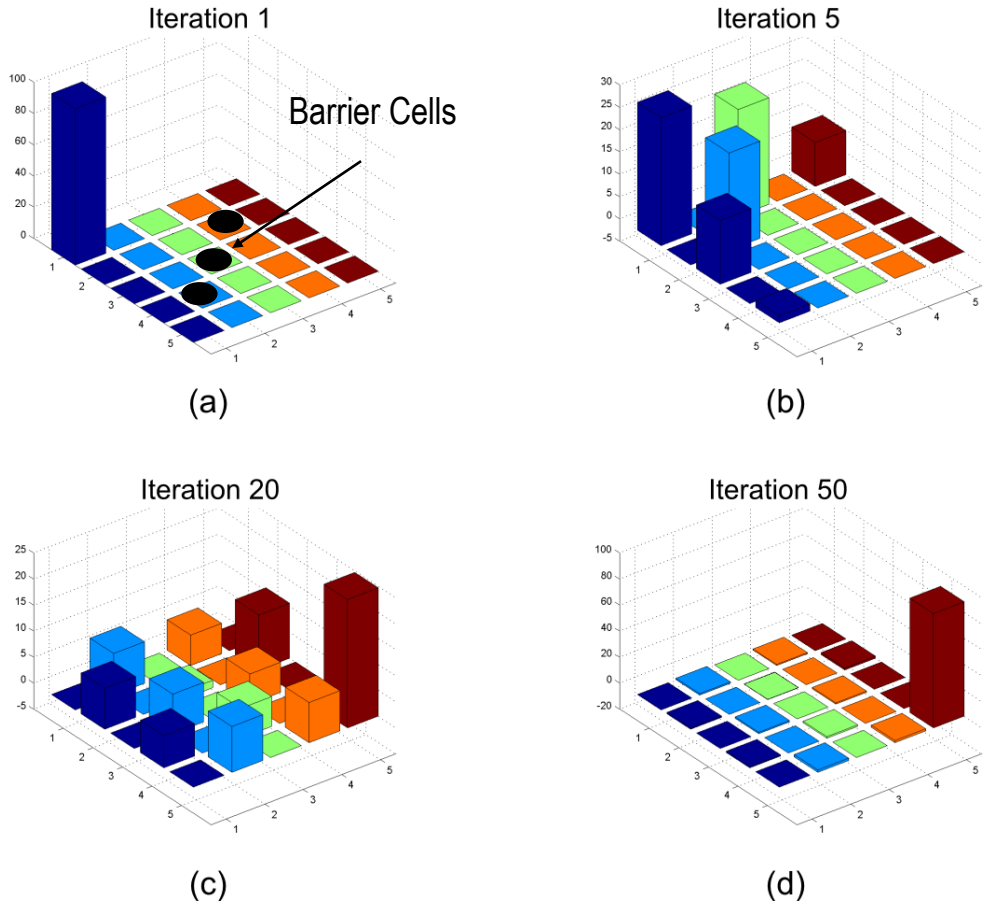


Figure 13 – Convergence of the dynamic for Experiment 6, produced by the best evolved NunCA.

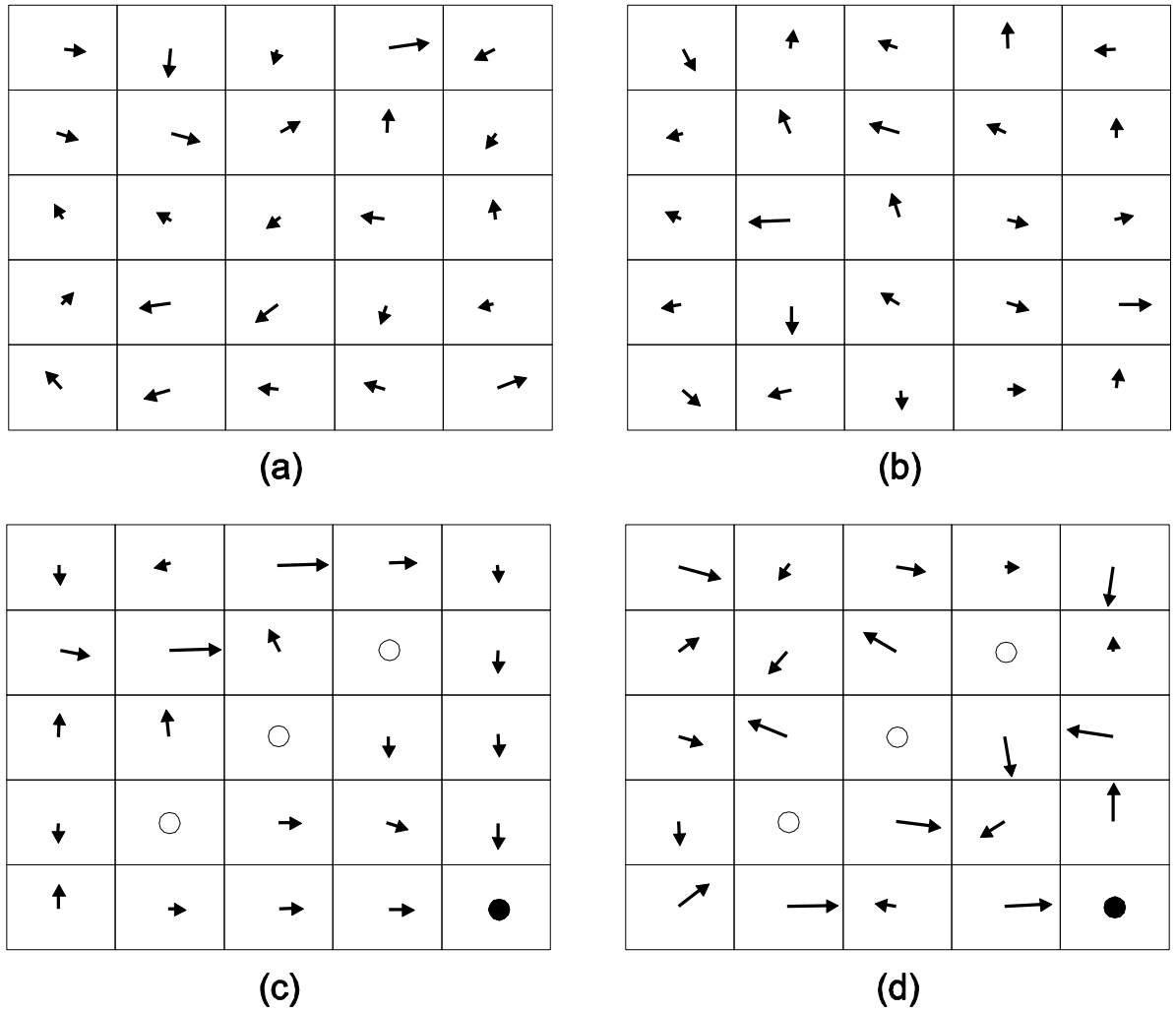


Figure 14 – The gradient of dispersion throughout the lattice. (a) Experiment 3; (b) Experiment 4; (c) Experiment 5; and (d) Experiment 6.

8 Concluding Remarks

- Non-uniform cellular automata (NunCA) have been proposed here as mathematical models capable of reproducing desired spatio-temporal behaviors. The necessity of defining one transition rule per cell in the regular lattice motivated the application of evolutionary algorithms, due to the impossibility of performing an exhaustive search.
- Evolutionary algorithms have already been proposed to design uniform and non-uniform cellular automata. However, none of these previous applications were devoted to spatio-temporal modelling using NunCA. When the purpose was the same, the cellular automata were taken to be uniform (MITCHELL *et al.*, 1993). When the cellular automata were non-uniform, other purposes were involved in the evolutionary design, as the straight classification of the obtained transition rules according to the qualitative nature of the spatio-temporal behavior produced by the cells in the regular lattice (SIPPER, 1994).

- When the transition rules involve a binary codification, a genetic algorithm has been designed to properly search for a feasible solution. In this scenario, the cellular automata are restricted to be one-dimensional lattices, and the purpose is to reproduce some specific and periodic profiles along time. The increment in flexibility provided by the NunCA framework was demonstrated to be essential to allow the reproduction of the intended spatio-temporal behavior. Very simple profiles have been defined, and even under these favorable circumstances (including a fixed initial condition for the cells in the lattice) there is no uniform CA capable of accomplishing the task, while multiple equivalent solutions have been obtained with NunCA.
- A more challenging scenario is characterized by two-dimensional lattices with transition rules that implement dispersion of a given material, where the state of each cell is associated with the concentration of material at that position in space and at a given instant of time.

- Here, the cellular automata is characterized by transition rules each one obeying a difference equation with 4 parameters to be independently determined, once a set of physical restrictions is not violated. Due to the continuous nature of the parameters to be optimized, an evolution strategy has been conceived. Four distinct experiments have been implemented, and the last two ones incorporate spatial restrictions to the dispersion process. The spatial restrictions may be interpreted as a physical barrier to the flux of material. The single objective in the first three experiments was to design a two-dimensional NunCA capable of achieving a predefined final state from a predefined initial state, no matter the transitory behavior between the two configurations.
- The fourth experiment incorporates a temporal restriction associated with symmetrical flux, and here the intermediary states do matter, besides the initial and final states.

- The NunCA framework represents a significant increment in the computational demand of the design phase. However, the additional flexibility in implementing a distinct transition rule per cell in the regular lattice opens the possibility of multiple solutions and gives rise to an additional step in the investigation of means to reproduce spatio-temporal phenomena: the obtained transition rules for the NunCA can be interpreted and can be used to raise hypothetical explanations for complex spatio-temporal events in nature.

9 References

Bäck, T., Hoffmeister, F. and Schwefel, H.-P. (1991) "A Survey of Evolution Strategies", Proceedings of the 4th International Conference on Genetic Algorithms, pp. 2-9.

Basanta, D., Miodownik, M.A., Bentley, P.J. and Holm, E.A. (2004) "Investigating the Evolvability of Biologically Inspired CA", Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems (ALIFE9).

Beyer H. and Schwefel, H. (2002) "Evolution strategies: A comprehensive introduction", Natural Computing.

Camazine, S., Deneubourg, J.-L., Franks, N.R., Sneyd, J., Theraulaz, G. and Bonabeau, E. (2001) "Self-Organization in Biological Systems", Princeton University Press.

Chua, L.O. and Yang, L. (1988) "Cellular Neural Networks: Theory", IEEE Transactions on Circuits and Systems, vol. 35, no. 10, pp. 1257-1272.

Chopard, B. (1998) “Cellular automata modeling of physical systems”, Cambridge University Press.

Collasanti, R. L. and Grime, J.P. (1993) “Resource dynamics and vegetation process: a deterministic model using two-dimensional cellular automata”. Functional Ecology, vol. 7, pp. 169-176.

Frisch, U., Hasslacher, B. and Pomeau, Y. (1986) “ Lattice-gas automata for the Navier-Stokes equation”, Physical Review Letters 56, pp. 1505-1508.

Golberg, D.E. (1989) “Genetic Algorithms in Search, Optimization & Machine Learning”, Addison Wesley.

Gregorio, S. D. and Serra, R. (1999) “An empirical method for modelling and simulating some complex macroscopic phenomena by cellular automata”, Future Generation Computer Systems, vol. 16, pp. 259-271.

Gutowitz, H. and Langton, C. (1988) “Methods for Designing ‘Interesting’ Cellular Automata”, CNLS News Letter.

Hillman, D. (1995) “Combinatorial Spacetimes”, Ph.D. Thesis, University of Pittsburgh.

Ilachinski and Halpern, P. (1987) “Structurally dynamic cellular automata”, Complex Systems 1, pp. 503-527.

Jai, A. E. (1999) “Nouvelle approche pour la modélisation des systèmes en expansion spatiale: Dynamique de végétation, Tendences nouvelles en modélisation pour l´environnement”, Elsevier, pp. 439-445.

Kagaris, D. and Tragoudas, S. (2001) “Von Neumann Hybrid Cellular Automata for Generating Deterministic Test Sequences”, ACM Trans. On Design Automation Of Electronic Systems, vol. 6, no. 3, pp. 308-321.

Kaneko, K. (ed.) (1993) “Theory and Applications of Coupled Map Lattices”, Wiley.

Li, W. (1991) “Phenomenology of Non-Local Cellular Automata”, Santa Fe Institute Working Paper 91-01-001.

Maji, P., Ganguly, N., Saha, S., Roy, A. K. and Chaudhuri, P.P. (2002) “Cellular Automata Machine for Pattern Recognition”, Lecture Notes In Computer Science, vol. 2493. Springer-Verlag, pp. 270-281.

Mitchell, M., Hraber, P. and Crutchfiled, J. (1993) “Revisiting the edge of chaos: Evolving cellular automata to perform computations”, Complex Systems, vol. 7, pp. 89-130.

Mitchell, M., Crutchfield, J.P. and Das, R. (1996) “Evolving Cellular Automata with Genetic Algorithms: A Review of Recent Work.” Proceedings of the First International Conference on Evolutionary Computation and Its Applications.

Nagel, K. and Herrmann, H.J. (1993) “Deterministic models for traffic jams”, Physica A 199, pp. 254-269.

Nicolis, G. and Prigogine, I. (1977) "Self-organization in non-equilibrium systems", Wiley.

Oliveira, G.M.B., de Oliveira, P.P.B. and Omar, N. (2001) "Definition and Application of a Five-Parameter Characterization of Unidimensional Cellular Automata Rule Space", Artificial Life, vol. 7, pp. 277-301.

Rabino, G.A. and Laghi, A. (2002) "Urban Cellular Automata: The Inverse Problem", Lecture Notes in Computer Science, vol. 2493, pp. 349-356, Springer.

Schönfisch, B. and de Roos, A. (1999) "Synchronous and asynchronous updating in cellular automata", Biosystems, vol. 51, Issue 3, pp. 123-143.

Schwefel, H.-P. (1981) "Numerical Optimization of Computer Models", Wiley.

Sipper, M. (1994) "Non-Uniform Cellular Automata: Evolution in Rule Space and Formation of Complex Structures", Artificial Life IV, pp. 394-399, The MIT Press.

Toffoli, T. and Margolus, N. (1987) "Cellular automata machines – A new environment for modeling", The MIT Press.

Tomassini, M. and Perrenoud, M. (2001) "Cryptography with cellular automata", Applied Soft Computing, vol. 1, Issue 2, pp. 151-160.

Tomassini, M., Sipper, M., Zolla, M. and Perrenould, M. (1999) "Generating high-quality random numbers in parallel by cellular automata", Future Generation Computer Systems, vol. 16, pp. 291-305.

Tsalides, P. (1990) "Cellular automata based build-in self test structures for VLSI systems", IEE-Electronics Letters, vol. 26, no. 17, pp. 1350-1352.

Vassilev, V.K., Miller, J.F. and Fogarty, T.C. (1999) "The Evolution of computation in co-evolving demes of non-uniform cellular automata for global synchronization", Proceedings of the 5th European Conference on Artificial Life, Berlin.

Von Neumann, J. (1961) "The General and Logical Theory of Automata", John von Neumann: Collected Works, Vol 5: Design of Computer, Theory of Automata and Numerical Analysis, Taub, A.H. (ed.), Oxford: Pergamon Press.

Wolfram, S. (1994) "Cellular Automata and Complexity – Collected Papers", Addison-Wesley Publishing Company.

Related topics: coupled map lattice, discrete-time cellular neural network, lattice gas.