



HAL
open science

Probing robustness of cellular automata through variations of asynchronous updating

Olivier Bouré, Nazim A. Fatès, Vincent Chevrier

► **To cite this version:**

Olivier Bouré, Nazim A. Fatès, Vincent Chevrier. Probing robustness of cellular automata through variations of asynchronous updating. [Research Report] 2012, pp.21. hal-00658754v1

HAL Id: hal-00658754

<https://inria.hal.science/hal-00658754v1>

Submitted on 11 Jan 2012 (v1), last revised 14 Jun 2012 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Probing robustness of cellular automata through variations of asynchronous updating

Olivier Bouré, Nazim Fatès, Vincent Chevrier

Nancy Université – INRIA Nancy–Grand-Est – LORIA
{olivier.boure,nazim.fates,vincent.chevrier}@loria.fr

Abstract

Typically viewed as a deterministic model of spatial computing, cellular automata are here considered as a collective system subject to the noise and lack of constraints of natural computing. The updating scheme, classically synchronous, is replaced by stochastic versions which either randomly update cells, or disrupt the cell-to-cell transmission of information. Applying our panel of updating schemes to Elementary Cellular Automata, we observe a wide variety of behaviours, which we study in the scope of macroscopic statistical phenomena and microscopic analysis of the internal mechanisms. Finally we discuss on the role of updating scheme in the emergent behaviours, and the interest in using them to probe models in order to reveal robust properties.

Keywords: Asynchronous cellular automata, robustness, discrete dynamical systems, phase transitions, directed percolation.

1 Introduction

Cellular automata (CA) are known as a model of computation based on the parallel evolution of cells according to their neighbourhood state. Originating from self-replicating systems as introduced by von Neumann (1966), they are considered as an alternative to the sequential computing models. Indeed, their spatially-extended and yet simple structure make them a suitable model for parallel computing and simulations of natural phenomena.

In their original form they are deterministic and synchronously evolved, that is, all components of the system are updated simultaneously. However, as natural computing supposes the use of non-classical methods involving a number of interacting components, the determinism of CA is somewhat incompatible with the presence of noise. This raises the question to its robustness, that is, the stability of the behaviour despite external disturbances, such as an asynchronous update or non-perfect information. This question was initially discussed when Ingerson and Buvel (1984) “wanted to estimate how much of the behaviour of

cellular automata comes from synchronous modeling and how much is intrinsic to the iteration process”. For that purpose, they questioned the synchronicity by randomising the transition function of Elementary Cellular Automata, and “probed” their robustness, looking for behavioural changes. Their observations concluded that the updating scheme plays a fundamental role in the behaviour of the system.

Since then authors tackled the question of whether CA are robust to non-synchronous updating, either by keeping the same individual rule (*e.g.* Fatès (2009; 2010)) or adding adapted constructs (*e.g.* Peper et al. (2002)). For instance, (Grilo and Correia 2011) investigated the change of behaviour for asynchronous symmetric 2-player evolutionary games, such as the Prisoner’s Dilemma. Similarly, several models of CA have been studied under asynchronous updating, such as Elementary Cellular Automata (Fatès 2009, Regnault 2008) or the *Game of Life* (Blok and Bergersen 1999, Fatès 2010) and revealed the existence of qualitatively different behaviours.

The definitions of asynchronous CA used so far have mainly focused on either randomising the transition function or updating cells sequentially (Bandini et al. 2010a). But as asynchronism is first defined as a negation of synchronicity, several interpretations remain possible. Our approach, emerging from the fields of autonomous agents (Bandini et al. 2010b) and deployment environments (Ackley and Williams 2011, Gouaïch et al. 2005), intends to consider CA no more only as a model of computation but as a collective system, where cells are autonomous components and their interactions are subject to perturbations. Instead of questioning whether cells are updated at each time step, we represent materially the perception by the cells of their neighbourhood, and simulate failures in the cell-to-cell transmission of information.

Questioning the perfect synchrony hypothesis. Let us first consider an illustrative example: two planes separated by a fixed distance are flying in opposing directions and must decide to go *up* or *down* in order to avoid collision. The decision function, the same for both planes, is deterministic: if the opposing plane goes down, then go up and vice versa. Here three “types” of updating schemes are considered:

- (a) The perfect synchrony assumes that planes update their state simultaneously at each time step. An initial configuration with both planes going in the same direction inevitably results in a periodic cycle and a crash.
- (b) However, when planes update their state with a given probability, the system finds a stable situation after a few time steps. This updating scheme is called *asynchronous update*.
- (c) This time, planes do update their state at the same time, but the updated state of one is not always correctly transmitted to the other. In that case, when one plane fails to update its “image”, the system stabilises. We call this novel type of updating scheme *asynchronous information transmission*.

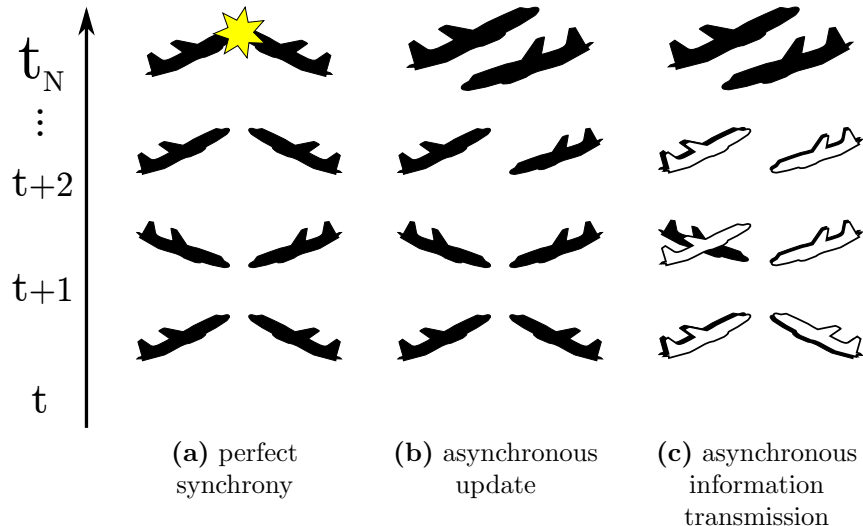


Figure 1: Different updating schemes applied to a rudimentary collision avoidance system. Black planes represent actual states, and in the case (c), white planes denotes how they are perceived by the opposite plane.

As one can deduce from Fig. 1, both asynchronous updating schemes alter the long-term behaviour of the system, as they directly prevent the stability of cyclic points such as the one occurring for the synchronous case. However, their intrinsic difference also suggests that there may exist models where a change of updating scheme induce a qualitative change of behaviour. We intend to observe and investigate these differences in the scope of cellular automata.

Asynchronous information transmission in cellular automata. To study the effects of asynchronous information transmission on cellular automata, we propose a novel updating scheme of cellular automata, extending two features of the classical definition: (1) we extend the state space to take into account the internal state of the cell as well as how it is perceived by the neighbourhood and (2) we describe the updating process of the system in the frame of a cellular cycle, that is, the sequential application of two steps, (a) the local computation of the new state, and (b) the transmission of the updated state to the cells of the neighbourhood.

Extending an earlier work (Bouré et al. 2011a), we present a more general definition of asynchronous information transmission and a more complete study of Elementary Cellular Automata. In Sec. 2 we give a general formal definition of asynchronous information transmission for cellular automata. In Sec. 3, we give an overview of our observations on asynchronous Elementary Cellular Automata and in Sec. 4, we present the results of our analysis of phase transitions. Section 5.1 presents two case studies of a local approach to the study

of the effects on two models, which present different reaction depending on the asynchronism. Finally, we discuss our results in Sec. 6.

2 Asynchronous Cellular Automata

2.1 Definition of Cellular Automata

A cellular automaton is a discrete dynamical system defined by $A = \{\mathcal{L}, Q, \mathcal{N}, f\}$ where :

- $\mathcal{L} \subset \mathbb{Z}^d$ the array of the cellular space, where $c \in \mathcal{L}$ represents a *cell*.
- Q is a finite set of states.
- $\mathcal{N} \subset \mathcal{L}$ is a finite set of vectors called the *neighbourhood*, which associates to a cell the set of its neighbouring cells. \mathcal{N} and \mathcal{L} are such that for all $c \in \mathcal{L}$ and for all $n \in \mathcal{N}$, the *neighbour* $c + n$ is in \mathcal{L} .
- f is the *local transition rule*, which defines the next state of a cell according to the state of this cell and the ones of its neighbours.

A *configuration* x^t represents the state of the automaton at time t ; it is defined as a function $x^t : \mathcal{L} \rightarrow Q$ which maps each cell to a state. Classically, cellular automata are synchronously updated, meaning that at each time the local transition rule is applied simultaneously to all cells. The *global transition function* is therefore defined as $x^{t+1} = F(x^t)$, so that, for $\mathcal{N} = \{n_1, \dots, n_k\}$:

$$\forall c \in \mathcal{L}, x^{t+1}(c) = f(x^t(c), x^t(c + n_1), \dots, x^t(c + n_k)).$$

Without loss of generality, we assume that the neighbourhood \mathcal{N} does *not* contain the cell itself. This hypothesis is necessary to explicitly represent the flow of information between a cell and its neighbours. Note that this does not restrict the expressiveness of f since the current state of a cell $x^t(c)$ is always a parameter of f , possibly not taken into account in the transition calculus.

2.2 Asynchronism as a disruption of cell activity.

To give a proper definition of asynchronous information transmission requires us to extend the classical framework of cellular automata.

1. The cell update is represented by a two-step *cell cycle* (see Fig. 2):
 - the *state update* step, where a cell changes its state according to the local transition function and its perception of the neighbourhood.
 - the *information transmission* step, where the cell transmits the updated state to its neighbours.
2. The state space Q must be extended to distinguish:

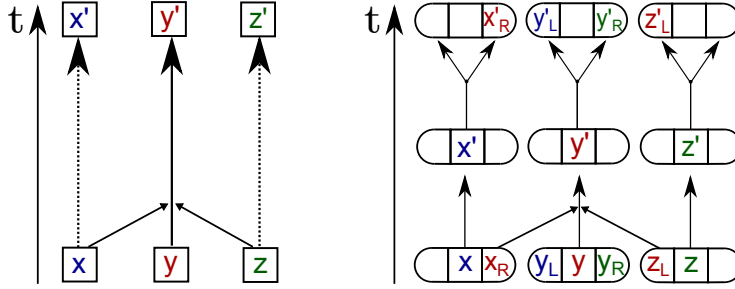


Figure 2: Left: schematic representation of the classical update. Right: representation of the update as a cellular cycle, for a cell with two neighbours. A triplet $\{y_L, y, y_R\}$ represents a cell with eigenstate y , and its observable states, for each of its neighbours (y_L for left cell and y_R for right). The classical transition corresponds to $y' \leftarrow f(x, y, z)$. However, for the cellular cycle, the transition is divided into the update step $y' \leftarrow f(x_R, y, z_L)$ and the transmission step, which performs $y'_L \leftarrow y'$ and $y'_R \leftarrow y'$.

- the *eigenstate*, that is, the actual, internal state of a cell in Q ,
- a vector of *observation states*, that is, elements of Q that represent how a cell is perceived by its neighbours.

In the frame of this extension of cellular automata, we define three asynchronous updating schemes (see Fig. 3):

α -synchronism: at each time step, each cell is updated with a fixed probability α , or else left unchanged.

β -synchronism: each cell is always updated but the transmission of the new state to the neighbourhood globally is realized with a fixed probability β .

γ -synchronism: each cell is always updated but the transmission to each neighbour is determined independently with a probability γ .

As a result, perturbations consist in applying one of the steps of the cell cycle with a probability defined as the *synchrony rate*. Please note that for the sake of simplicity, α , β and γ denote both the type of asynchronism and the associated synchrony rates.

α -synchronism. We introduce a *selection function* $\Delta_\alpha : \mathbb{N} \rightarrow \mathcal{P}(\mathcal{L})$ which returns for time t the subset of cells to be updated, where each cell has a probability α to be selected. Note that when $\alpha = 1$ the updating is fully synchronous and the system is deterministic. The global transition function becomes $\forall t \in \mathbb{N}, \forall c \in \mathcal{L}, \mathcal{N} = \{n_1, \dots, n_k\}$:

$$F_{\Delta}(x^t(c)) = \begin{cases} f(x^t(c), x^t(c+n_1), \dots, x^t(c+n_k)) & \text{if } c \in \Delta(t) \\ x^t(c) & \text{otherwise.} \end{cases}$$

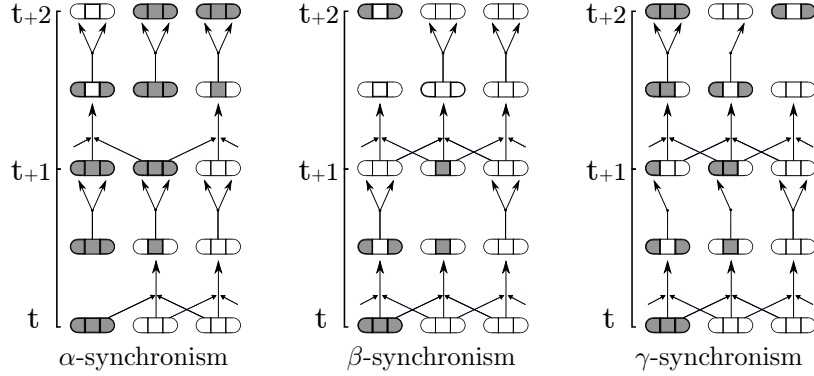


Figure 3: Example of the time cycle for a 3-cell sample for three perturbations of the updating scheme: α -synchronous (left), β -synchronous (middle) and γ -synchronous (right) (0 is denoted by a white space, 1 by gray). The applied rule is ECA **50** (see Sec. 3).

Asynchronous information transmission. By contrast with asynchronous updating, the asynchronous information transmission aims at disrupting the exchange of information about states between cells. Here we give a unified framework for this new type of asynchronism, extending the classical definition of cellular automata with a cellular cycle. We later give definitions of β - and γ -synchronisms in this context by stating how each of them handles the observability of cells.

1. the state space $Q' = Q \times Q^m$ in order to differentiate, for a cell configuration $x^t(c)$, its *eigenstate* $e^t(c) \in Q$ and a vector of m *observation states* $\mathbf{o}^t(c) \in Q^m$,
2. the transition function f' , which breaks up into two functions applied sequentially, the local updating function f_u and the transmission function f_{tr} .

We thus define $f' = f_{tr} \circ f_u$, such that:

- $f_u : Q'^{k+1} \rightarrow Q'$ is the *update* function and computes the new eigenstate of a cell based on its eigenstate and the observation state of the neighbours, leaving the observation states unchanged:

$$f_u \left(\begin{pmatrix} e \\ \mathbf{o} \end{pmatrix}, \begin{pmatrix} \cdot \\ \mathbf{o}_1 \end{pmatrix}, \dots, \begin{pmatrix} \cdot \\ \mathbf{o}_k \end{pmatrix} \right) = \begin{pmatrix} \phi(e, \mathbf{o}_1, \dots, \mathbf{o}_k) \\ \mathbf{o} \end{pmatrix}. \quad (1)$$

where $\phi : Q'^{k+1} \rightarrow Q$ returns the result of the local transition function f with the corresponding observation states for each cell of the neighbourhood.

- $f_{\text{tr}} : Q' \rightarrow Q'$ is the *transmission* function, as it carries out the “propagation” of the eigenstate $e^t(c)$ to the neighbourhood: this operation consists of replacing the values of observation states $\mathbf{o}^t(c)$ of the cell by the eigenstate. To do so, we introduce a set of random variables $(\theta_i^{t,c})_{t \in \mathbb{N}, c \in \mathcal{L}, i \in \llbracket 1, m \rrbracket}$ which represent the atomic transmissions of the eigenstate $e^t(c)$ to the i -th element of $\mathbf{o}^t(c)$, such that:

$$\forall t \in \mathbb{N}, \forall c \in \mathcal{L}, \forall i \in \llbracket 1, m \rrbracket, \theta_i^{t,c}(e, o) = \begin{cases} e & \text{with a probability } P \\ o & \text{otherwise.} \end{cases} \quad (2)$$

where P is the synchrony rate corresponding to the asynchronism.

The transmission function f_{tr} is thus defined, for the configuration (e, \mathbf{o}) of a cell c at time t , as a stochastic function such as:

$$f_{\text{tr}}^{t,c} \left(\begin{matrix} e \\ \mathbf{o} \end{matrix} \right) = \begin{pmatrix} e \\ \theta_1^{t,c}(e, o_1) \\ \vdots \\ \theta_m^{t,c}(e, o_m) \end{pmatrix}. \quad (3)$$

β -synchronism. This asynchronism consists in considering the perturbation as a disruption of the emission of the eigenstate by a cell. It was previously defined in similar terms by (Bouré et al. 2011a). However, for the sake of a unified definition, we represent the observation state by a single element of Q which is perceived by all neighbouring cells ($m = 1$):

- the state of a cell, for a given configuration x^t , is given by $x^t(c) = (e^t(c), \mathbf{o}^t(c))$, where $\mathbf{o}^t(c) \in Q$.
- $\phi(e, o_1, \dots, o_k) = f(e, o_1, \dots, o_k)$.

γ -synchronism. This asynchronism consists in considering the transmissions of a cell’s eigenstate to each neighbour independently. As a consequence, the vector of observation states associates an element of Q for each element of the neighbourhood ($m = k$):

- the state of a cell, for a given configuration x^t , is given by $x^t(c) = (e^t(c), \mathbf{o}^t(c))$, where $\mathbf{o}^t(c) \in Q^k$.
- The transition function ϕ is given by:

$$\begin{aligned} \phi(e, \mathbf{o}_1, \dots, \mathbf{o}_k) &= \phi \left(e, \begin{pmatrix} o_{1,1} \\ \vdots \\ o_{1,k} \end{pmatrix}, \dots, \begin{pmatrix} o_{k,1} \\ \vdots \\ o_{k,k} \end{pmatrix} \right) \\ &= f(e, o_{1,1}, \dots, o_{k,k}). \end{aligned}$$

As we present them, β - and γ -synchronism differ only in the representation of the observation states. For this reason, their effects on cellular automata behaviours are expected to be similar in most cases and γ -synchronism will only be mentioned when a difference of behaviour with β -synchronism is observed.

3 Qualitative observations

We now aim at observing qualitatively the effect of β - and γ -synchronism, and particularly whether the effects on the ECA behaviour will differ from the α -synchronism.

Elementary Cellular Automata (ECA). An *ECA* is a one-dimensional binary cellular automaton with nearest-cell neighbourhood. It is denoted by $A_E = \{\mathcal{L}, \mathcal{N}, \{0, 1\}, f\}$ where:

- $\mathcal{L} = \{\mathbb{Z}/L\mathbb{Z}\}$ is a 1-dimensional ring of length L ,
- $\mathcal{N} = \{-1, +1\}$, *i.e.* the 2-cell neighbourhood,
- the local transition function $f : Q^3 \rightarrow Q$.

It is common for ECA to use the rule of their transition function as a designation for the model, using the code $W = f(0, 0, 0) \cdot 2^0 + f(0, 0, 1) \cdot 2^1 + \dots + f(1, 1, 1) \cdot 2^7$. Through the use of left/right reflexion and 0/1 complementarity, it is possible to determine 88 classes of ECA rules to study, from which we extract the representative with the smallest rule number.

In the following, initial conditions are constructed by setting the value of the eigenstates randomly and uniformly in $\{0, 1\}$. As we consider the disruption of information transmission as a perturbation, we set the values of the observable states equal to the eigenstate, thus assuming that no error exists at initial time.

General observations on asynchronous ECA. To estimate the behaviour of ECA, we use two approaches:

1. We visually compare space-time diagrams, obtained by displaying the values of eigenstates over a few time steps. Although it provides a good impression on changes of behaviour, this approach cannot hold as a formal classification criterion.
2. In order to quantify the behaviour, we use a simple parameter, the density, defined as the ratio of cells with eigenstate 1. We give an estimation of the asymptotic density, by averaging its value over a sampling time, after a transient time and for a large number of cells.

It is well known that the classification of ECA rules is a difficult problem, due to the richness and the complexity of the encountered behaviours (*e.g.* Wolfram classes in Culik II and Yu (1988)). Similarly, studies on α -synchronism have suggested that there is no correlation between Wolfram classes and their robustness

to α -synchronism (*e.g.* ECA **50** in Fatès (2009)). As β - and γ -synchronism define new updating mechanisms, they raise the question of whether they have the same effects than α -synchronism and we are required to search the entire ECA rule space looking for behavioural changes¹.

For the vast majority of rules, neither the space-time diagram nor the density profile allow to qualify any change of behaviour between α -synchronism and any other type of asynchronism. However, some rules show remarkable changes, as illustrated by the examples of Figure 4:

- A notable difference occurs for ECA **51** (**NOT** rule): unlike α -synchronism, the β -synchronous version is perfectly robust to asynchrony. This can be explained trivially by pointing out the fact that the NOT rule does not actually take the neighbourhood into account in the transition function.
- Although the density parameter in ECA **51** and ECA **57** follow an identical behaviour, a quick comparison of their space-time diagrams reveals that the behaviour is radically different.
- For rules with a behaviour similar to ECA **45**, the density shows a slight different value for each asynchronism, but looking at the space-time diagrams suggests that this change has only a limited effect on the behaviour of the system.
- Finally, some rules show a discontinuity in the plot profile of the density, such as for ECA **6**, **50**, **58**, suggesting that a qualitative change of behaviour occurs for a critical value of the synchrony rate.

Table 1 presents a classification of the changes of behaviour as measured by the density parameter. We are particularly interested in the last class of rules, for which an abrupt jump of density occurs for a small change of the synchrony rate. As the presence of a discontinuity marks a change of behaviour that is both qualitative and measurable, we are confronted to explicit examples of non-robustness in asynchronous cellular automata. To clarify further how updating schemes effects on cellular automata, we now propose to study in the context of statistical physics.

4 Directed percolation in asynchronous ECA

The occurrence of phase transitions is probably one of the most remarkable phenomena that arises in asynchronous cellular automata. In this phenomenon, there exists a non-trivial value of the synchrony rate, the critical threshold, which separates two distinct “phases”, that is, qualitatively different behaviours of the system.

For the α -synchronous case, Fatès (2009) identified second-order phase transitions which were grouped in three classes:

¹The complete results can be found at: <http://www.loria.fr/~boure/eca>.

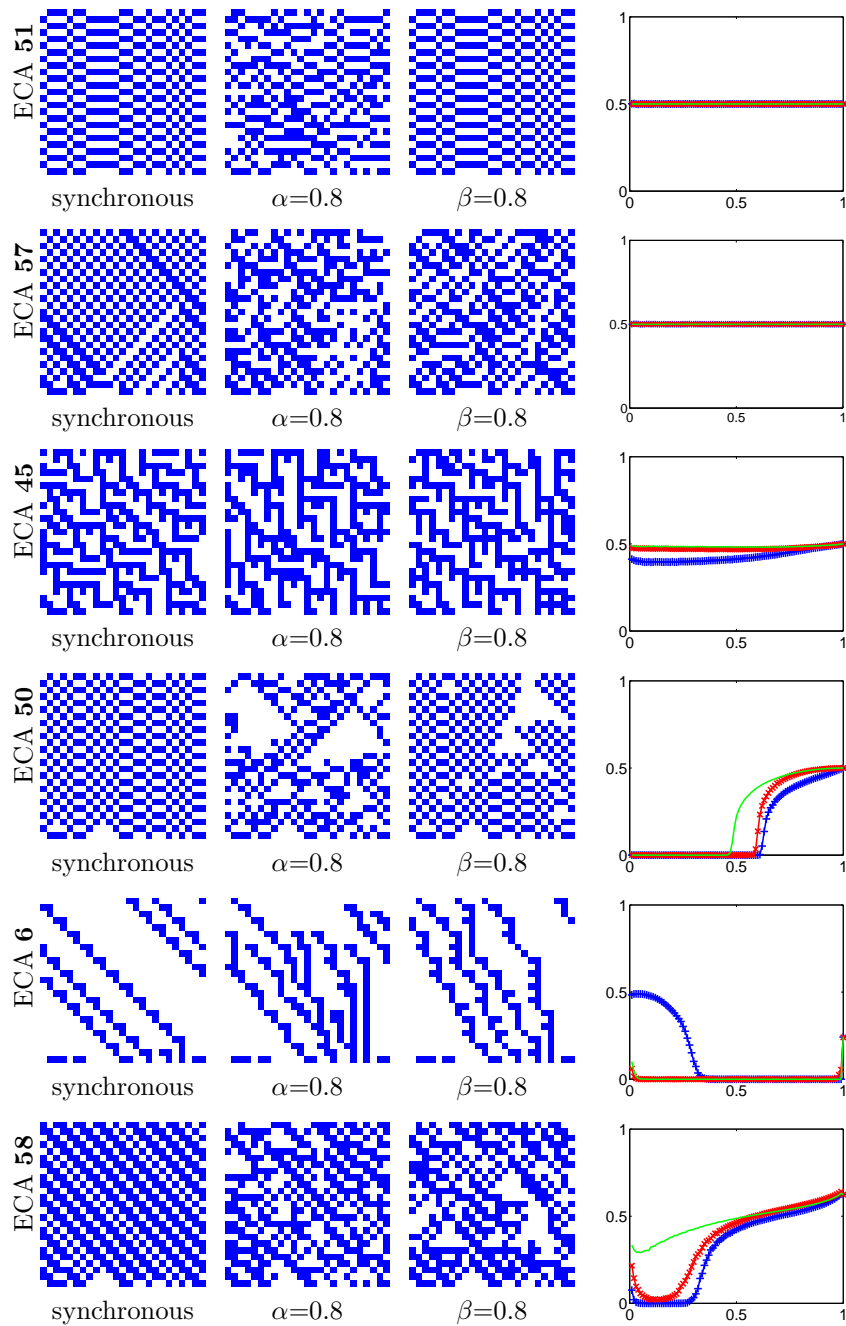


Figure 4: A sample of the observations made on ECA rules on asynchronous updating schemes. The plot show the profile of the density parameter as a function of the synchrony rate (blue + for α , red \times for β , and green line for γ). All diagrams start from the same initial configuration. Time goes from bottom to top.

Table 1: Classification of the minimal ECA per observed difference of behaviour, using the density parameter.

| | |
|------------------------------|--|
| No observable difference | 0, 1, 2, 5, 7, 8, 10, 11, 13, 14, 15, 23, 24, 25, 27, 28, 29, 30, 32, 34, 37, 40, 41, 42, 43, 51, 56, 57, 60, 73, 74, 77, 78, 90, 94, 105, 126, 128, 130, 136, 138, 140, 142, 150, 154, 156, 160, 162, 168, 170, 178, 184, 200, 204, 232 |
| Difference of level or slope | 3, 4, 9, 12, 19, 22, 33, 35, 36, 44, 45, 46, 54, 62, 72, 76, 104, 108, 110, 122, 132, 152, 164, 172 |
| Discontinuity | 6, 18, 26, 38, 50, 58, 106, 134, 146 |

- the DP_{hi} class comprises all ECA with a “positive” phase transition: for high value of the synchrony rate, the density parameter converges to non-zero values, whereas for lower rates, it becomes zero after a sufficient simulation time. This class includes ECA **18, 26, 50, 58, 106, 146**.
- the DP_{low} class gathers ECA with a “negative” phase transition: active and passive phases are reversed in order. It includes ECA **6, 38, 134**.
- the DP_2 class, which comprises only ECA **178**, shows a positive phase transition but it is observed for a different order parameter².

Applying our novel updating scheme to the same rules, we observed that the phase transitions were not always reproduced. In particular:

- For β -synchronism, only the positive phase transitions (DP_{hi} and DP_2 classes) are observed, whereas rules of the DP_{low} class do not show any (Bouré et al. 2011a).
- For γ -synchronism, observations are similar to β -synchronism, with the exception of ECA **58** for which the all-0 phase does not exist.

A major observation lies in the identification of the universality class of the phase transitions, that is, a class of models which follow the same power law at criticality.

Estimation of the critical behaviour. Fatès (2009) described a protocol and showed that all phase transitions observed for α -synchronism relate to the directed percolation universality class. We propose to reuse the same experimental protocol and apply it to the transitions observed for β - and γ -synchronism:

1. We fix the synchrony rate, start from a random initial configuration and let it evolve for a given number of steps.

²For the sake of conciseness, we decide to put this class aside and concentrate on observations based on the density parameter.

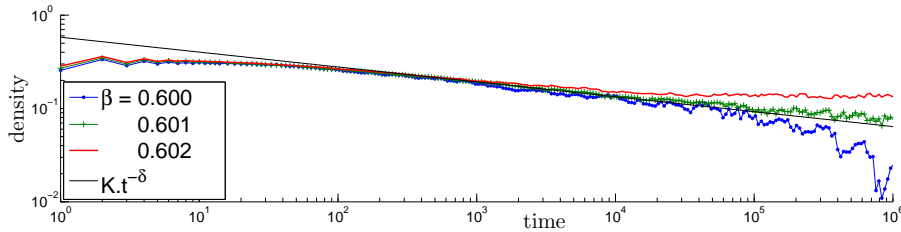


Figure 5: Phase transition analysis for β -synchronous ECA **50**. The straight line follows a power law $f(t) = K \times t^{-\delta}$ with the critical exponent $\delta = 0.1595$. Measures are averaged over 32 automata of size 10000.

2. We monitor the evolution of the order parameter for a long simulation time until we observe a sub-critical or super-critical behavior. In a log-log plot, a concave curve occurs for the passive phase (density converges to zero) and a convex curve for the active phase (density converges to a non-zero value). As we expect the order parameter to follow a power law $K.t^{-\delta}$ near criticality, the curve at critical value should appear as a straight line of slope $-\delta$.
3. We repeat the experiment with a value closer to the critical point until a satisfactory precision is reached (here 10^{-3}).

Figure 5 presents the results of this experiment for β -synchronous ECA **50**.

For the particular example of ECA **50**, the critical threshold measured for each type of asynchronism are $\alpha_c = 0.628$, $\beta_c = 0.601$ and $\gamma_c = 0.487$, and the evolution of the density parameter for such values is in good agreement with a power law $K.t^{-\delta}$ with $\delta \sim 0.1595$, which is the expected critical exponent for the directed percolation universality class in $1 + 1$ dimensions (Hinrichsen 2000). Table 2 lists the values of the critical synchrony rates, for which a phase transitions was observed. All show good evidence of a directed percolation phenomenon.

We uncovered here a major difference in the effects of updating schemes on cellular automata. From the study of phase transitions in ECA, we could show that the robustness of simple rules may depend on the type of updating scheme, asynchronous update or asynchronous information transmission. In order to understand further how these may affect the global behaviour, we propose to analyse these models at a lower level.

5 Microscopic approach

For this section, we will focus on the study of the local evolution of cells under α - and β -synchronous updating schemes, using two examples that display different global behaviour: ECA **50** for the DP_{hi} class and ECA **6** for the DP_{low} class.

Table 2: Measurements of the critical synchrony rates for different rules and asynchronisms.

| Class | ECA | α_c | β_c | γ_c |
|-------------------|-----|------------|-----------|------------|
| DP _{hi} | 18 | 0.714 | 0.749 | 0.655 |
| | 26 | 0.475 | 0.526 | 0.406 |
| | 50 | 0.628 | 0.601 | 0.487 |
| | 58 | 0.340 | 0.289 | – |
| | 106 | 0.815 | 0.848 | 0.813 |
| | 146 | 0.675 | 0.732 | 0.635 |
| DP _{low} | 6 | 0.283 | – | – |
| | 38 | 0.041 | – | – |
| | 134 | 0.082 | – | – |

5.1 First case study: ECA 50

ECA **50**, also referred to as the **FLIP-IF-NOT-ALL-0** rule, is defined by the following transition table:

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |

A remarkable phenomenon can be observed for β -asynchronism: although the study of the evolution of the density parameter suggests a directed percolation phenomenon similarly to α -synchronism, a quick inspection of the space-time diagrams over a few time steps reveals a radical difference in the local behaviours (see Fig. 6). In particular, one may observe the conservation over time of some patterns of the automaton configuration, and explain these by listing the possible local transitions.

Observing the behaviour. When displaying the evolution of ECA **50** under different updating schemes, we observe a regularity in the space-time diagrams of the automaton for β -synchronism. Contrary to α -synchronism, it seems to conserve some properties of the synchronous diagram (see Fig. 6). In particular, the following patterns were identified:

- an *anomaly* is a group of 3 or more consecutive 0s that may appear locally in a configuration. As the synchrony rate goes up, the size of anomalies increases, until they cover the whole lattice.
- a *checkerboard pattern* is defined as a group of spatially alternated cells with eigenstate 0 and 1 which, through a synchronous update, will normally shift every time step, drawing a “checkerboard” on the space-time diagram. It is interesting to note that a β -synchronous update will shift all eigenstates 1 to 0, but not necessarily the other way, hence creating the previously described anomalies.

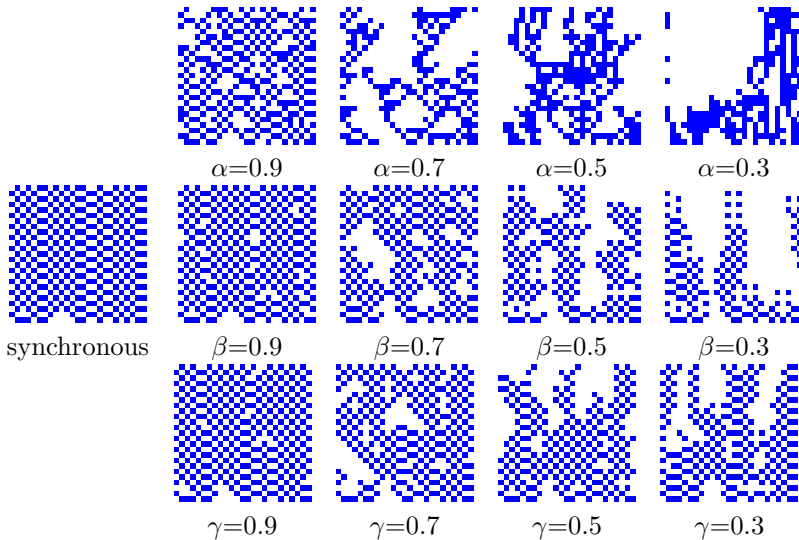


Figure 6: Samples of space-time diagrams of ECA **50** for different synchrony rates and different types of asynchronism, starting from the same initial configuration.

- the *particles* are pairs of cells with identical eigenstate (*i.e.* 00 and 11) separating regions of checkerboard pattern. The occurrence of anomalies in the vicinity of these cells may provoke a spatial non-biased random walk or the annihilation with another nearby particle, thus allowing checkerboard regions to join. Once destroyed, checkerboard pattern seems unable to recreate pairs.

Configurations in β -synchronism can therefore be interpreted as a partition of the space in checkerboard regions separated by anomalies and moving particles. With sufficient time and if the synchrony rate β is high enough, particles will meet and annihilate, thus “joining” checkerboard regions. As a consequence, if the size of the lattice is even, the system tends to organise in the long run into a unique checkerboard region covering the entire lattice. This behaviour derived from the stability of checkerboard patterns under β -synchronism is a novel property, exclusively observed in β -synchronism, and suggests that the behaviour is indeed qualitatively different in α - and β -synchronism. We now intend to explain this stability by analysing the possible transitions within ECA **50**.

Analysing the behaviour. Let us consider the case where the initial configuration is a unique region of checkerboard pattern. From a local observation of the automaton evolution, we can explain why the checkerboard pattern is seemingly so stable: by mapping 1-cells to nodes of an oriented graph, we show that all possible transitions from a checkerboard pattern result in another, shifted checkerboard pattern. We derive the following properties, as illustrated

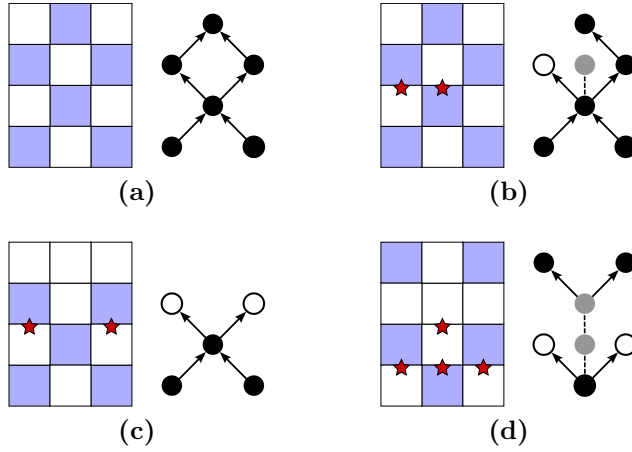


Figure 7: The effects of β -synchronism on local interactions in ECA **50**. For each subfigure, left is a configuration sample and right the corresponding graph. The configuration visualisation show the cell eigenstates, with red stars denoting when a cell fails to update its observable state. The color code for graph representation is the following: no node denote state $(0, 0)$, black node $(1, 1)$, white node $(1, 0)$ and gray node $(0, 1)$, where (e, o) denotes e the eigenstate and o the observable state.

on Fig. 7:

- (a) The 1-cells in synchronous ECA **50** can be exactly mapped to an oriented graph where all neighbouring “nodes” share a child node.
- (b) In general, if an isolated cell does not transmit its state to its neighbourhood, this does not have any particular effect on the regularity of the system.
- (c) For local configurations 010, if the two side 0-cells both fail to transmit their new state 1, then the middle cell will conserve its eigenstate set to 0 and create a “hole” in the pattern. This phenomenon is at the origin of the apparition of the anomalies as described before.
- (d) A last specific event may occur: in the same situation than before, if the central cell switching from 1 to 0 fails to transmit its state twice in a row, then the anomalies do not appear.

These properties are sufficient to describe the behaviour of β -synchronous ECA **50**, for all values of the synchrony rate β . By introducing an oriented graph to describe the system, we also display informally that it is closely related to *site-percolation*, a graph-based model used as a reference of percolation phenomena. Similarly, the relationship between ECA **50** and directed percolation theory has been analytically explained for α -synchronism by Regnault (2008). As a conclusion, the rule ECA **50** displays paradoxal response to α - and β -synchronism.

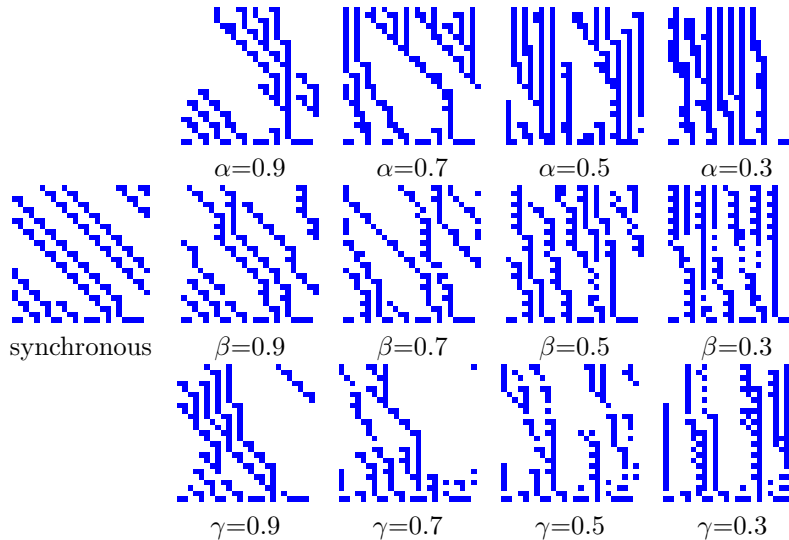


Figure 8: Samples of space-time diagrams of ECA **6** for different synchrony rates and different types of asynchronism, starting from the same initial configuration.

On the one hand, a directed percolation phenomenon occurs for both types of asynchronism, which suggests similar effects. But on the other hand, we identified specific properties of β -synchronous ECA **50** which suggests that the local behaviour differs from one asynchronism to the other.

5.2 Second case study: ECA **6**

ECA **6** shows peculiar behaviours under different asynchronisms. If a directed percolation phenomenon occurs for the α -synchronous version, the phase transition disappears for β -synchronism. This qualitative change of behaviour suggests that in this particular case, different types of asynchronism result in different behavioural response (see Fig. 8).

In order to understand the mechanism involved in asynchronous ECA **6**, one must consider the evolution of cells states at a microscopic level. We propose as a simplified view of configurations to consider *branches* as contiguous groups of 1-cells surrounded by 0. We now try to determine their governing rules, based on the transition table of the rule:

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

Percolation in α -synchronism. We are particularly interested in two “events”: the destruction of the branch or the creation of a new one. In order to describe the qualitative behaviour of α -synchronous ECA **6**, we must determine which

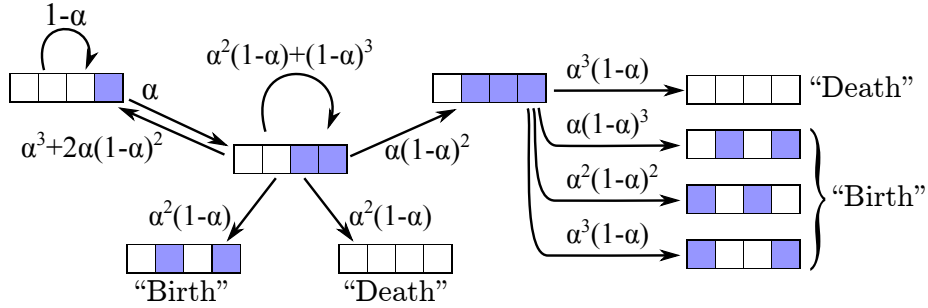


Figure 9: A sample of transition graph which illustrates the possible evolutions from a single branch in α -synchronous ECA **6**. By considering only groups of 1 smaller than 4, we find that the probability of “Birth” events is higher than “Death” events.

is more likely to occur depending on the value of the synchrony rate α . To determine this, two effects are to be analysed:

1. We first consider the case of a single branch and start to build the graph of the possible evolutions of the configuration, with their associated probabilities (see Fig. 9). We see that for all values of the synchrony rate α , we are more likely to observe the creation of a new branch (“Birth”) than the destruction of the first (“Death”).
2. For a synchronous updating scheme, when two branches are separated by a critical distance, the right branch is immediately destroyed due to a lack of space (see the synchronous diagram on Fig. 10). For asynchronism, the distance between branches is subject to vary due to random events, and it becomes possible to destruct branches through this process.

For values of the synchrony rate α close to 1, the probability to create or destroy branches is almost equal. On the other hand, the chance to destruct branches through their interactions is high as, for high α , this phenomenon happens almost surely as soon as the distance between branches becomes critical. However, for low values of α , the opposite behaviour occurs: as bigger clusters of continuous 1s appear more often, it becomes more probable to create new branches than to destroy them, thus overcoming the effect of interactions branches. This behaviour is confirmed by experiments, for which the behaviour remains active and the density parameter converges to a fixed value (see Fig. 4).

Although we are still unable to predict the critical value α_c as well as to identify analytically the directed percolation phenomenon, studying the local dynamics of ECA **6** allows us to understand better why two different phases may occur for different values of α . We now intend to consider the β -synchronous case in order to investigate why the phase transition does not exist.

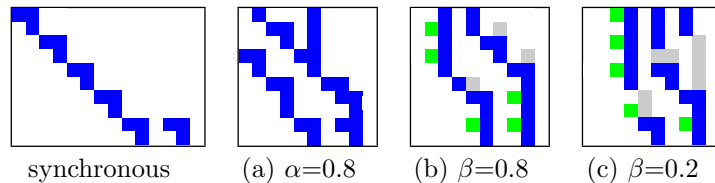


Figure 10: Branching phenomenon in ECA **6** under α - and β -synchronism. Light gray cells represent state $(0, 1)$ and green/darker gray state $(1, 0)$.

Novel mechanisms under β -synchronism. This same approach cannot be used to study the dynamics of β -synchronous, because of the extension of the state space. However, it is possible to give an insight on why the active phase does not exist for β -synchronism, by comparing the mechanism involved behind the creation of branches.

Let us consider the examples given in Fig. 10, where creation of branches are visualized under different updating scheme. In the α -synchronous case, as soon as an isolated 1 is created, it remains as stable as a normal branch (case (a)). However, in the β -synchronous case, branches are created when a cell changes its eigenstate to 0 but keep its observable as 1 (light gray cells). These branches have a short life expectancy, as (1) they cannot activate neighbouring cells until some space is created between branches and (2) they disappear as soon as they finally manage to update their observable state to 0 (case (b)). The only means to create branches in β -synchronous ECA **6** occurs for low values of the synchrony rate β (case (c)): in that case, the branch will survive sufficiently long for some space to be created and then it will survive like a normal branch.

Although the arguments above do not formally constitute a proof, this difference in the local evolution of cell states may provide a possible explanation of why a phase transition only occurs for α -synchronism. In this case, the change of updating scheme induces a qualitative behavioural change locally and globally.

6 Discussion

This paper presented a general formalism for a new type of asynchronism in cellular automata, called asynchronous information transmission. We then compared this novel type of perturbation to asynchronous updating using the density parameter and observed a wide variety of behaviours.

In particular, we considered examples of non-robustness from Elementary Cellular Automata, for which directed percolation phenomena occur for the density parameter in the spectral range of the synchrony rate. For instance:

- For ECA **50**, the macroscopic behaviour is conserved for all types of asynchronism, but a microscopic analysis of the dynamics revealed novel properties for β -synchronism, such as the conservation of patterns like particles and checkerboard over time.

- With ECA **6**, we have shown that the change of updating scheme may induce modifications in the local dynamics, thus modifying the global behaviour qualitatively.
- Finally for ECA **58**, in spite of the proximity of the definitions of β - and γ -synchronism, the observed behaviour are qualitatively different.

There is an open question on the occurrence of phase transitions for some ECA rules under α -synchronism. Our observation completes this issue, by establishing correlations in the non-robustness of certain models, and bringing new problems where differences appear between the different updating schemes.

The main conclusion one can draw from this study is that changing the definition of the updating schemes in collective models such as cellular automata may subject the global behaviour to drastic changes, as it effects directly on the mechanisms of interactions between cells. Therefore, the appearance, conservation or disparition of properties in the model's evolution constitutes a major observation which provides useful information about how updating scheme effects the behaviour of the system.

The updating schemes as a component of CA. In this paper, we introduced two novel ways to evolve a discrete-time system which effected differently on chosen models. Similarly, there must exist a great variety of updating schemes, and as many behavioural responses. As a consequence, for models where robustness of the emergent behaviour is a desired property, the updating scheme should hold a modular position, that is, it should be interchangeable, in order to allow a wider search of the updating scheme space orthogonally to searches of the individual rule space. Considering updating schemes as a component will raise new questions and extend the field of possible applications of perturbations in discrete dynamical systems, such as identifying an asynchronism based on the responses of a set of individual rules, or searching to which updating schemes a given rule is robust.

Probing a model to reveal its robustness. The study presented in this paper is part of a broader approach, which aims to subject models to different kinds of updating scheme and, by observing the effects on the local and global behaviour, to distinguish which properties are relevant to the individual rules itself, and which are specifically dependent on the updating scheme. Although it is yet far too early to draw any conclusion on what can be said of such observations, we reckon that probing a model may allow us to specify which properties are robust to a change in features of the model, and thus intrinsically related to the auto-organisation of individual rules.

In addition, the importance of updating scheme has been revealed not only for game theory and cellular automata, but also for bio-inspired models. For instance, we observed in (Bouré et al. 2011b) that for a model of swarming in lattice-gas cellular automata, the emergent behaviour could be related to discretisation effects and geometrical properties of the system. In higher-level

contexts, this type of approach could be used (1) to study further the internal mechanisms and (2) to gain knowledge on the required hypotheses on a model's definition to ensure the robustness of the desired behaviour.

References

- Dave H. Ackley and Lance R. Williams. Homeostatic architecture for robust spatial computing. In *Proceedings of the 5th IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, 2011.
- Stefania Bandini, Andrea Bonomi, and Giuseppe Vizzari. What do we mean by asynchronous CA? A reflection on types and effects of asynchronicity. In *Proceedings of ACRI 2010*, pages 385–394, 2010a.
- Stefania Bandini, Andrea Bonomi, and Giuseppe Vizzari. Modeling and programming asynchronous automata networks: The MOCA approach. In *Proceedings of ACRI 2010*, pages 345–355, 2010b.
- Hendrik J. Blok and Birger Bergersen. Synchronous versus asynchronous updating in the "Game of Life". *Physical Review E*, 59(4):3876–3879, Apr 1999.
- Olivier Bouré, Nazim Fatès, and Vincent Chevrier. Robustness of Cellular Automata in the Light of Asynchronous Information Transmission. In *Unconventional Computation*, volume 6714 of *Lecture Notes in Computer Science*, pages 52–63, 2011a.
- Olivier Bouré, Nazim Fatès, and Vincent Chevrier. Observation of novel patterns in a stressed lattice-gas model of swarming. Technical report, Nancy Université – INRIA Nancy–Grand-Est – LORIA, 2011b.
- Karel Culik II and Sheng Yu. Undecidability of CA classification schemes. *Complex Systems*, 2:177–190, 1988.
- Nazim Fatès. Asynchronism induces second order phase transitions in elementary cellular automata. *Journal of Cellular Automata*, 4(1):21–38, 2009.
- Nazim Fatès. Does *Life* resist asynchrony? In *Game of Life Cellular Automata*, pages 257–274. Springer, 2010.
- Abdelkader Gouaïch, Fabien Michel, and Yves Guiraud. MIC*: a deployment environment for autonomous agents. In *Post-Proceedings of the 1st International Workshop in Environments for Multiagent Systems*, pages 109–126. Springer-Verlag, 2005.
- Carlos Grilo and Luís Correia. Effects of asynchronism on evolutionary games. *Journal of Theoretical Biology*, 269(1):109–122, 2011.
- Haye Hinrichsen. Nonequilibrium critical phenomena and phase transitions into absorbing states. *Advances in Physics*, 49:815–958, 2000.

- Thomas E. Ingerson and Raymond L. Buvel. Structure in asynchronous cellular automata. *Physica D: Nonlinear Phenomena*, 10(1-2):59 – 68, 1984.
- Ferdinand Peper, Teijiro Isokawa, Yousuke Takada, and Nobuyuki Matsui. Self-timed cellular automata and their computational ability. *Future Generation Computer Systems*, 18(7):893 – 904, 2002.
- Damien Regnault. Directed percolation arising in stochastic cellular automata analysis. In *Mathematical Foundations of Computer Science*, volume 5162 of *Lecture Notes in Computer Science*, pages 563–574. Springer-Verlag, 2008.
- John von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, Champaign, IL, USA, 1966.