A guided tour of asynchronous cellular automata

NAZIM FATÈS*

Inria Nancy Grand-Est, LORIA UMR 7503
F-54 600, Villers-lès-Nancy, France

Received September 29, 2014; In final ???

Research on asynchronous cellular automata has received a great amount of attention these last years and has turned to a thriving field. We present a state of the art that covers the various approaches that deal with asynchronism in cellular automata and closely related models.

Key words: asynchronous cellular automata, survey, discrete dynamical systems, complex systems

Foreword: This article is an extended version of the invited paper that appeared in the proceedings of Automata'13, 19th International Workshop on Cellular Automata and Discrete Complex Systems, LNCS 8155, 2013, p. 15-30. Updated versions of this article should be available on the preprint servers Arxiv and HAL (see: arxiv 1406.0792 and hal 00908373) or by writing to the author.

* email: `nazim.fates@loria.fr`

1 INTRODUCTION

By their very simplicity, cellular automata are mathematical objects that occupy a privileged situation in the study of complex systems. They are formed of a regular arrangement of simple automata, the *cells*, which can hold a finite number of states. Cellular automata are as mosaics with tiles that autonomously change their colour: the cells are updated at discrete time steps and their new state is calculated according to only a local information, usually limited to the states of the neighbouring cells. These local laws of interaction may generate amazing behaviours at the global scale, even when they are simply expressed.

Cellular automata were initially studied by von Neumann and Ulam to study the properties of self-reproduction of living organisms with a simple "mechanical" tool [133]. Since then, they have been employed in various scientific domains. Their study can be divided into three main axes: (1) They are dynamical systems where time, space and states are discrete. Their regular structure simplifies the mathematical definitions of the system but the exact or partial prediction of the trajectories of the system is often a highly challenging task. (2) They represent a model of spatially-extended, distributed and homogeneous computing systems. As such, they represent an alternative to the classical computing frameworks that use sequential algorithms, variables, functions, etc. (3) They are employed to model the numerous complex systems seen in Nature. Researchers have been particularly interested in the properties of self-organisation or robustness they can display.

An important feature in the definition of cellular automata regards their updating: in their original definition, they are updated *synchronously*, that is, all the cells change their state at the same (discrete) time step. This global update implies a *strong* simultaneity: cells need to gather simultaneously the state of their neighbours, they need to process this information simultaneously, the transitions have to occur in a single time step.

Making this hypothesis of perfect synchrony has many advantages, first of all to simplify the description of the system. With a synchronous update, it is for instance easy to build a Turing-universal system, to "program" the system to obtain a given behaviour, to show that a given property is undecidable or to study under which restrictions this property becomes decidable, etc. (see the survey by Kari [60] for more details). Synchronous updates are also a convenient tool for modelling natural or artificial phenomena: there is no need to take into account complex updating procedures as all the cells share the same time.

In spite of these manifest advantages, there are reasons why the hypothesis of *perfect synchrony* needs to be questioned:

(a) In the context of dynamical systems, the problem is to study how cellular automata "react" to perturbations of their updating. How can we interpret the potential sensitivity of a system to changes of its definition? On the contrary, what can be said if the system "resists" to a change of its updating scheme?

(b) In the context of parallel computing, we ask how to design a computing device that does not require a central clock. Various advantages can be expected from the removal of a pace maker: increase of the speed of computations, economy of energy, simplicity of design, etc. Beyond these potential gains, developing asynchronous massively parallel algorithms represents a research challenge by itself.

(c) When cellular automata represent a model of a natural phenomenon, the question is to know what triggers the transitions of the cells' state. How do we represent this source of activity in the model? Answering is far from being simple and the argument that "there is no global clock in Nature to synchronise the transitions" is somewhat incomplete. Indeed, it can be objected that a model is a simplified representation of a phenomenon and does not need to faithfully account for all the details of "reality".

All these questions raise rich problems and they are discussed in the works that we present in this survey. The field of asynchronous cellular automata has attracted the interest of numerous authors and has evolved from a "marginal" to a "respected" topic during the last decade. The scientific production has now reached a level which makes it difficult to follow all the contributions that appear. The purpose of this survey is thus to introduce the readers to this quite diversified "landscape", trying as much as possible to cover the various "sites" that it contains. As a "guided tour", it does not claim to be an objective description of the field: a guided tour is by definition a circuit that takes visitors from place to place according to the arbitrary choices of the guide. It is therefore important to bear in mind that the descriptions that will follow will be as brief as possible and should by no means prevent us from reading the texts mentioned *themselves*. Our hope is that readers that are unfamiliar with cellular automata will find landmarks for their orientation and those which are interested in a particular topic will find useful references.

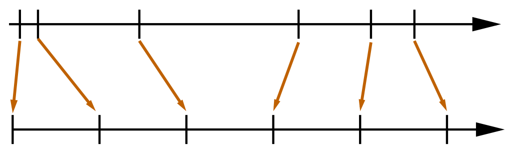

Figure 1
Mapping from a continuous to a discrete time scale.

2 DEFINING ASYNCHRONISM

Our visit begins by considering the definitions of asynchronism. The etymology is clear: α-συν-χρόνος (*a-sun-chronos*) means *not-same-time* in Greek. The word thus merely indicates that there are parts of the system that do not share the same time. As an illustration, we may figure out a choreography where each dancer has its own pace and its own sequence of movements: the choreography may be chaotic but the dancers may also succeed in forming a coherent performance if some coordination is maintained between them.

The *privative* nature of the definition of a-synchronism suggests that there are many interpretations of the word. In fact, we are allowed to speak of asynchronism as soon as we break the framework of perfect updating. To date, there is no agreement on how this word should be defined. Moreover, it is frequent that different terms are used for naming the *same* updating scheme. The definitions that we present below are thus by no means “official”: we simply make the choice to use in priority the terms that we have employed in our own research.

2.1 Full vs. partial asynchronism

In general, asynchronism is seen as an external and uncontrolled phenomenon, it is thus most often modelled as a stochastic process. The two main stochastic updating schemes that have been employed are:

- *fully asynchronous updating:* At each time step, the local rule is applied to only one cell, chosen uniformly at random among the set of cells.
- *α-asynchronous updating:* At each time step, each cell has a given probability α to apply the rule and a probability $1 - \alpha$ to stay in the same state. The parameter α is called *the synchrony rate** .

* Note that the terms α-**a**synchronism and α-synchronism have been used and are both relevant: α can denote the name of the scheme and the synchrony rate. We use here the term

Many authors consider that the fully asynchronous updating is the most "natural" updating method. The argument that justifies this choice is a syllogism that can be decomposed as follows: (a) "real" time is continuous, (b) transitions occur at random moments on this continuous timeline, (c) since there is no chance of a simultaneous updating, or only a negligible chance, the only thing that matters is *the order* in which cells are updated, (d) this order can be obtained by a sequential stochastic sampling on the set of cells (see Fig. 1 and e.g. Ref. [113] for a similar presentation).

It can be remarked that this argument is physically relevant if the transitions of the cellular automaton are "infinitely" short, that is, if the time to go from one state to another can be neglected. This is surely a valid hypothesis for some particular contexts (e.g. a radioactive disintegration) but this cannot be considered as *the* asynchronous updating model.

In many cases, especially in biological systems, some synchrony between cells needs to be assumed. As this degree of synchrony is difficult to measure, the problem is not so much about choosing the "right" model of updating but rather to estimate the *robustness* of model, that is, if it will totally or partially resist the perturbation of its updating scheme. In this context, the α-asynchronous method defines a system with a continuous variation from a perfect synchronism ($\alpha = 1$) to the limiting case of full asynchronism ($\alpha \to 0$). Note however that when looking at the *asymptotic* behaviour of a system, a discontinuity may exist between the case $\alpha \to 0$ and the fully asynchronous case. Indeed, the possibility that two neighbouring cells simultaneously update their state, be it as small as wanted, may radically change the trajectory of a system. As an example, consider the minority rule in 2D with a von Neumann neighbourhood: with a fully asynchronous updating, the two uniform fixed points are not reachable from a non-uniform configuration [40] but this is not longer true if we allow a small degree of synchrony.

It should also be noted that fully asynchronous updating is defined with a finite set of cells. The passage to the limit for an infinite set of cells needs to be done with a model that has a *continuous time* and the mathematical model that accurately describes a (stochastic) fully asynchronous updating is called an *interacting particle system*. (See e.g. Ref. [20] for examples where such systems are used for solving the density classification problem in two dimensions.)

α-**a**synchronism, as it is the form that was first proposed and which has been adopted by various authors such as Regnault, Correia, Worsch, Fukś, etc.

2.2 How to describe asynchronism?

The other question that is generally asked when defining asynchronism is to know if the timing of the transitions should be defined with the use of a global clock or with a clock that is proper to each cell. Schönfisch and de Roos call the former *step-driven* methods and the latter *time-driven* methods [113]. It may be thought at first that "time-driven" methods are more adequate for making "realistic" simulations than "step-driven" methods. Indeed, it seems better to give to the cells an explicit representation of time and to avoid to artificially share a transition signal between all the cells. However, this idea needs to be examined more closely. As remarked by various authors, this distinction is somewhat artificial as it is in general possible to build a "step-driven" method that emulates a "time-driven" method, and vice versa. For example, as discussed before, the random updates of a fully asynchronous scheme and the updates obtained by independent clocks that use a *continuous* time are quasi-equivalent, up to a rescaling. The α-asynchronous updating can also be defined from the point of view of the cells by separating two updates by a random time which obeys a geometric law. (In other words, the probability that k time steps separate two updates of a cell is equal to: $\alpha(1-\alpha)^{k-1}$.)

There are of course many other types of updating schemes where randomness is involved. For instance, one may consider *random sweeps* where cells are updated sequentially by following random permutations of the updating orders (this scheme is also called *random order*) or *fixed sweeps* where the permutation order is drawn at the beginning and kept fixed during the whole evolution of the system [113]. We will also present below how to define an asynchronism which results from an imperfect transmission of the state from the neighbours (see Sec. 3.4).

Non-random updating schemes can also be considered: for instance, in the *sequential ordered* scheme, cells are updated sequentially following an order that results from their spatial arrangement (for example from left to right and from top to bottom); cells can also be updated depending on their parity at even or odd time steps (see e.g. [96]). We refer to the work of Schönfisch and de Roos [113], Cornforth et al. [26], Bandini et al. [9] for the presentation of a collection of various deterministic or stochastic updating schemes.

It is also necessary to distinguish the *non-deterministic* schemes from the stochastic ones. As in classical automata theory, non-determinism means that a given subset of cells may be updated and all the possibilities are considered, regardless of their "likeliness to appear". The evolution of the system is thus represented by a set of configurations; this set evolves according to the outcomes of each transition that can be applied.

The problem of the definition of asynchronism is thus completely open and we end this section with the following questions:

Questions 1 *What taxonomy of the updating schemes can be issued? What are the guidelines that can drive modellers for choosing a particular updating scheme? Under which restrictions (states, neighbourhoods, class of rules, etc.) can we establish equivalences between updating schemes?*

3 EXPERIMENTAL APPROACHES

Classifying "classical" cellular automata has been a central theme of research and is far from being a closed question (for recent references, see e.g. the work by Schüle [115, 114] and the survey by Martínez [77]). It can then be thought at first that classifying *asynchronous* rules is a daunting task because of the additional complexity that is induced by the asynchronous updating. In fact, this is only partially true as in many cases the asynchrony may "break" the complexity of a rule and render it more simple to study. In this section, we discuss the contributions that qualitatively or quantitatively estimate the effects of asynchronism with numerical simulations.

3.1 General classifications

In 1984, Ingerson and Buvel carried out a pioneering work where they could show that the behaviour of simple rules could be totally disrupted by simple modifications of the updating [19]. Most importantly, they questioned to which extent was the behaviour of a rule the consequence of the local rule and to which extent it was due to the updating scheme.

This question was re-examined by Bersini and Detours, who explored the difference between the Game of Life and closely related asynchronous variants [14]. Their main observation was the existence of a "stabilising effect" of asynchronous updating. The experiments were made on small-size grids, no larger than 20 by 20 cells. With such lattice sizes, they were able to observe that the fully asynchronous Game of Life may "freeze" on some fixed-point patterns with a labyrinth-like aspect. However, more recent work has demonstrated that it was not possible to infer the large-size behaviour from these experiments and that the stabilising effect was intimately linked to finite-size effects of the numerical experiments [15, 36].

Schönfisch and de Roos gave a decisive impulse to the research on asynchronism by comparing various updating schemes and by exhibiting clear examples where the schemes alter significantly the behaviour of a rule [113]. They gave a detailed analysis of the statistical properties of the schemes but their experiments were limited to some specific rules. The question thus remained open to know how these observations could be generalised to a larger class of rules.

On this basis, Fatès and Morvan examined how the 256 Elementary cellular automata (ECA) reacted to α-asynchronism [41]. To estimate the changes of behaviour of the system quantitatively, the authors used an approximation of the asymptotic density, that is, the value of the density that would be reached by an infinite-size system with an infinite simulation time. This parameter was considered as a first means to detect changes in the behaviour: a strong variation of the asymptotic density indicates that the system has undergone a transformation while an absence of variation does not necessarily imply that the system remained stable.

The 256 rules were then classified into four qualitative sets according to their responses to the variation of the synchrony rate α: (a) continuous variation of the behaviour (e.g. ECA 232), (b) discontinuity around $\alpha = 1$ (e.g. ECA 2 or 110), (c) phase transition for a critical value $\alpha_c < 1$ (e.g. ECA 50), and (d) non-regular behaviour (e.g. ECA 184). One of the surprising results of this study was that no direct correspondence could be drawn between these new classes of robustness and the previously known classes of synchronous behaviour (e.g., the informal Wolfram classes).

Similar observations were made by Bandini et al., who tested the effects of numerous asynchronous schemes on one-dimensional binary rules where the local function depends only on two neighbours (also called "radius-1/2" rules) [9].

3.2 Phase transitions

Blok and Bergersen were the first authors to identify the change that occurs in the Game of Life when cells are updated with a given probability [15]. They used α-asynchronism to show the existence of a *qualitative* transition from a "static" behaviour, where the system would settle on fixed points, to a "living" behaviour, where the system evolves by forming labyrinth-like patterns that do not fixate. The change of behaviour is a second-order phase transition, that is, there exist two qualitatively different phases which obey some well-known laws from statistical physics. In this case, the phase transition was shown to belong to the directed percolation universality class, which means that at the critical point, the evolution of the order parameters (e.g. the density) obeys the same power laws as an oriented percolation process that serves as a reference.

Fatès identified that similar phenomena occurred in Elementary Cellular Automata and that no less than *nine* rules also displayed phase transitions. It was shown that the density follows a power-law decay for the critical synchrony rate, in good agreement with the behaviour expected from the directed percolation universality class [34]. A unique rule, namely ECA 178, was shown to belong to another universality class, a fact that is explained by the symmetric role that is played by 0s and 1s in the transition rule.

The phase transition occurring in the Game of life was also re-examined by studying how this phenomenon was affected by perturbations of the topology [35, 36]. The main finding was that the critical value of the phase transition strongly depends on the regularity of the grid and that the qualitative change of behaviour becomes more difficult to observe as links between cells are removed.

Concerning other two-dimensional rules, Regnault et al. carried out a pioneering work by explaining in detail how the asynchronous minority rule displayed various types of behaviour depending on the topology on which it is applied [97, 99, 105]. A simple puzzling observation is that the minority rule will settle out on a checkerboard or on a stripe-like pattern depending on whether the rule is defined with the von Neumann or the Moore neighbourhood. To our knowledge, there is no mathematical explanation of this empirical observation.

Remark that two different complementary views exist on phase transitions: the most common way of describing a phase transition is to establish that for an *infinite* system, a qualitative difference of behaviour occurs for an infinitesimal variation of the control parameter. An alternative approach was adopted by Regnault who could prove that for a particular rule and a *finite* system, the transition corresponds to a variation of the convergence time from a linear to a polynomial function of the system's size [98].

3.3 Coalescence

A curious phenomenon was remarked when comparing the evolutions of two different initial conditions that were updated with the *same* local rule and the *same* sequence of updates: a rapid "coalescence" may occur, that is, the two systems take the same state and then evolve with the same trajectory (as the same sites are updated).

This phenomenon is in some cases easy to understand, as when the coalescence occurs on an attractive fixed point, but it was also observed for a non-fixed-point region of the state space (as for ECA 46 [41]). From a more pragmatic point of view, the following interpretation can be given: there are asynchronous systems whose evolution rapidly becomes governed by the random number generator that dictates the updates, regardless of the initial condition.

Rouquier and Morvan studied systematically the coalescence phenomenon for the 256 ECA [102, 104]. Their study revealed that it was possible to observe that some ECA would always coalesce, while others would never coalesce, and that there existed some rules which displayed a phase transition between a coalescing and non-coalescing behaviour. It is an open problem to explain analytically the non-trivial cases of rapid coalescence. It is also interesting to compare these results with those obtained by other methods of coupling (see e.g. [109, 103]).

3.4 Asynchronous information transmission

While the approaches of asynchronism studied so far are based on the dichotomy updated versus not updated, Bouré et al. defined a model of asynchrony which considers imperfect communications between neighbours [17, 16]. This approach is declined in two versions, called β- and γ-asynchronism, which respectively consider stochastic failures of the communication of a state to the whole neighbourhood or to each neighbour independently.

Among the various observations made with these two types of asynchronism, the most intriguing phenomenon is the disappearance of some, but not all, of the phase transitions that were obtained with the α-asynchronism. More precisely, ECA 6, 38 and 134 do not show any transition for β- and γ-synchronism. ECA 58 gives an even more puzzling case as it does show a phase transition for α- and β-asynchronism but *not* for γ-asynchronism. It is an open problem to understand the origin of such radical differences of responses to the rate of transmission failures.

Experiments also displayed that in some cases, quantities can be conserved when using only a particular model of asynchronism (e.g., ECA 50 has some parity conservation with β-asynchronism but not with α-asynchronism). This underlines the necessity to continue to "invent" various perturbations of the classical updating in order to gain insight on how cellular automata are dependent on their updating schemes.

3.5 Other variants

An interesting development on the work of asynchronism concerns how it mixes with traditional noise, that is, on randomness imposed on the *state* of the cells that compose the automaton. An early reference that addresses this question is given by Gharavi and Anantharam, who revisited a well-known result of Toom and who considered delays in the cells' communications [49]. We refer to the work of Kanada [59] on the 256 ECA rules, and to the work of Mamei et al. [75] for additional insights into this problem of mixing noise and asynchronism.

More recently, Silva and Correia gave a detailed account on how some ECA can react to asynchronism combined with noise [118]. Interestingly, they propose to evaluate the robustness according to the difference patterns. This brings them to introduce a sampling compensation in order to cope with less frequent updates.

The case of asynchronous models simulated on a non-regular topology was tackled by Baetens et al., who examined an asynchronous updating with a non-regular topology generated with a Voronoi tessellation [8].

To conclude this section, it seems that only a small part of the universe of asynchronous cellular automata has been explored so far. This brings us to put an emphasis on the following questions:

Questions 2 *What is a good protocol to numerically estimate the changes of behaviour induced by asynchrony? What are the relevant order parameters to quantify these changes? How common is it to observe discontinuities of behaviour induced by a* continuous *change of the updating scheme?*

4 ANALYTICAL APPROACHES

We now turn our attention to the mathematical analysis of asynchronous cellular automata. It is important to remark that although this part is presented separated from the previous one, there is a *joint* movement of going from simulations to analysis and back. (This co-development is not necessarily done by the same authors of course.)

4.1 Markov chain analysis and classifications

Agapie et al. conducted one of the first analytical studies of asynchronous rules using Markov chain theory [5]. They focused on several models of finite cellular automata with fully asynchronous updating. However, as far as we could understand, their analysis was limited to a specific case where the local rule was stochastic, totalistic, symmetric with respect to an exchange of `0`s and `1`s, and with positive rates (the probability to reach each state is strictly positive). It is worth noting that the number of borders of a configuration is a central parameter in their analysis and that this parameter is also found in various other approaches.

One of the first analytical results of classification were given by Fatès et al. who analysed the doubly-quiescent ECA [42]. In this study, the 64 rules considered are classified according to their worst expected convergence time to a fixed point. This time falls in the following classes: logarithmic time, linear time, quadratic time, exponential time and non-converging rules [†]. The visual inspection of the space-time diagrams of the rules of each class shows a good correspondence between the visual "behaviour" and the class. In other words, the time of convergence to a fixed point is not an ad hoc parameter but does capture a part of the "behaviour" of the stochastic rules.

These results were later partially extended to the more difficult case of α-asynchronous updating by Regnault et al. [43], while Chassaing and Gerin examined the continuous limit of the processes when the grid was made infinite [23].

Fatès and Gerin also examined how to classify the two-dimensional totalistic rules with fully asynchronous updating [40]. They proposed a partial classification of 64 rules and an analysis of the convergence of some well-known rules. Among the interesting phenomena remarked, they exhibited a list of rules which showed an "erratic" behaviour: the question was to determine if these rules were exhibiting a non-converging behaviour or a "metastable" behaviour, that is, if a (long) random sequence of updates could drive the system to a fixed point. By adapting techniques from automatic planning, Hoffmann et al. could solve this problem for a specific rule and showed that it converged to a fixed point in (at most) exponential time [55].

Readers interested in the classification of rules with regard to their convergence time can refer to a recent synthesis note [38] and a recent work on the fast convergence of the ECA rules [39].

4.2 Detailed analysis of the asymptotic densities

As mentioned above, for the α-asynchronous systems, the study of the asymptotic density was mainly made with numerical simulations. By focusing their efforts on eight simple ECA rules, Fukś and Skelton succeeded to give an exact calculation of this density [47]. They considered infinite systems where the initial condition was generated by a Bernoulli measure and determined how the asymptotic density varies as a function of the initial density (that is, the parameter of the Bernoulli measure). Such exact results are generally

[†] The classes are here given with a "rescaled time scale" where one step corresponds to as many random updates as there are cells in the finite ring.

rather difficult to obtain and it is an open problem to extend them to a wider class of rules.

Following this direction of research, Fukś and Fatès considered a development of Gutowitz's "local structure theory": contrary to a classical mean-field approach where the state between neighbouring cells is assumed to be uncorrelated, correlations of order 2 or larger are taken into account to try to predict the asymptotic density of the system [46]. It was shown that this approach does detect the occurrence of phase transitions. The limit is that the position of the critical synchrony rate remains difficult to find: for some rules, even approximations with nine cells cannot predict precisely the position of the critical threshold that separates the active and inactive phases.

4.3 Reversibility

As mentioned above, the asynchronous updating of a system does not perturb its fixed points. However, when the updating is stochastic cycles no longer exist and one needs to re-examine the meaning of reversibility. One such interpretation was proposed by Das et al., who define reversibility as a possibility to return to the initial condition in the case where the updating sequence (or "update pattern") could be set freely. They studied which are the Elementary Cellular Automata with null or periodic boundary conditions that generate such a form of "cycles" [110, 27].

Another point of view considered the case of fully asynchronous updating: as the evolution of the system is adequately described by a Markov chain, reversibility is identified with the property of recurrence of this chain [116]. A classification of the ECA rules into three classes was then proposed based on this tool: (a) The *recurrent* rules are those which make the system always return to its initial condition. (b) The *irreversible* rules are those which contain initial conditions which are never returned to after a (random) time. Among this class, (c) the *strongly irreversible* rules are those which contain a state that is never returned to as soon as it is updated. It is an open problem to determine how to extend these results to a wider class of systems, in particular to deal with the case of infinite-size systems.

Wacker and Worsch also examined the question of reversibility of asynchronous cellular automata [134]. In their work, a rule is said to be reversible if there is another rule whose state-transition graph is the "inverse" of the original. The novelty with respect to the synchronous case is that the out-degree of the nodes is no longer equal to one as a single configuration can lead to many others. Interestingly, the results presented on ECA are not far from those found in Ref. [116] and it is an open question to determine which are the conditions that make the two points of view equivalent.

4.4 m-asynchronous models and their topological properties

The study of the dynamical properties of cellular automata, such as injectivity, surjectivity, permutivity, etc., has been a central topic in the theoretical considerations of the field (see e.g. [60]). Manzoni examined how these properties could be re-defined and studied in the asynchronous updating context [76].

This work was taken a step further by Dennunzio et al., who developed the notion of *m-asynchronous* cellular automata in order to generalise the various updating methods used so far [30]. They provided a formal framework to describe the updating probabilities on each cell, even in the case where the size of the system is infinite, and produced various theorems that allow one to deal with the non-deterministic nature of the updating.

For more details on this line of research, we refer to the recent survey by Formenti where more details and examples can be found [45].

To synthesise, the contributions met in this section show the necessity to adapt the tools to the stochastic process theory for the specific case of cellular automata. This brings us to ask:

Questions 3 *What is the position of asynchronous cellular automata with respect to stochastic cellular automata? (a mere subset?) What are the analytical tools that can ease the analysis of the Markovian systems obtained with random updates?*

5 COMPUTING WITH ASYNCHRONOUS CELLULAR AUTOMATA

We now consider the contributions related to the computing abilities of asynchronous models and briefly describe the techniques that have been proposed to construct such (virtual or real) computing objects.

5.1 Simulation of (a)synchronous models by (a)synchronous models

Nakamura was among the first authors to investigate how to compute with an asynchronous cellular automaton [81, 82]. He described several techniques to construct a universal rule and showed how to simulate a given q-state deterministic rule with an asynchronous rule that has the same neighbourhood and whose state space is extended to $3q^2$ states (see also Lipton et al. [70], Toffoli [126] and Nehaniv [84] for similar constructions). The construction relies on the idea that when a cell is updated, it then waits the neighbouring cells to "catch up" and makes the next transition only when all its neighbours are up to date. Additionally, it keeps its old state available for the neighbouring cells in order for them to perform the "right" transitions. This construction was later improved by the use of only $q^2 + 2q$ states by Lee, Peper et al. [67, 89].

Peper et al. also proposed to consider the case where a cell can "activate" their neighbouring cells and showed that the cost in the number of states for the simulation of q-state rule could be reduced to $\mathcal{O}(q\sqrt{q})$ states [90].

Other discussions on the universality of asynchronous rules are found in the study by Takada et al., in which many important arguments and useful references can be found [124]. In particular, the authors present a result showing the existence of an asynchronous, rotation-symmetric rule with 15 states and von Neumann neighbourhood that has the property of universal construction and computation.

An alternative point of view was given by Golze who simulates an n-dimensional synchronous rule with an asynchronous rule defined on a space with $n + 1$ dimensions [50]. This solution simplifies the problem as there is no longer the need to save the previous and the current state in order to achieve correct computations. Another advantage of having an additional dimension is to read one state of the synchronous simulated system (guest) on the asynchronous simulating system (host): it simply corresponds to reading a line (or a hyperplane) of the host. This technique, called "global synchronisation", is presented as a means to solve various problems, such as the Firing Squad Synchronisation Problem, which would not be solvable without this requirement. However, it can be noted that this technique can be interpreted as the "deployment" of Nakamura's technique on an additional dimension. Reciprocally, one can also see Nakamura's technique as the "compressed" version of Golze's solution, where only the necessary information is retained.

The case where asynchronous computations have to be made with stochastic *and* asynchronous components was tackled by Wang [135]. Unfortunately, this author does not position his work with regard to the previous contributions (Nakamura, Golze) and it is difficult to see if this proposition significantly differs from the previous achievements.

An original way to simulate a universal Turing machine with a fully asynchronous updating has also been proposed by Dennunzio et al. [29]. The authors introduce the notion of "scattered strict simulation" in which they tolerate that only a subset of cells is used to perform the simulation. They find that asynchrony induces a quadratic slowdown compared to the speed of the simulated Turing machine.

5.2 Computations and order-independence

A key observation in the theory of asynchronous systems relies in the property of what we could call "non-overlapping influences": if two cells c and c' are such that the neighbourhood of c and c' do not overlap (that is, have no cell in common), it does not matter whether c is updated before c', or c' before c, or both of them are updated at the same time. The study of this property has given birth to various works that we now examine.

Gács was one of the first authors to determine if the evolution of an asynchronous system could be independent of the order of updating [48]. He showed that although this property was undecidable, there exists a sufficient condition to verify this independence.

This question was later re-examined by Mortveit, Macauley et al., who studied in which cases repetitions of sequential updates on Elementary Cellular Automata (ECA) could produce a set of periodic points that would be independent of the updating order [73, 74, 72]. This conducted the authors to present a list of 104 ECA which display such an update independence. Their work also uses an original representation of ECA that differs from the classical Wolfram code and that could prove useful for future analysis of asynchronous systems. (Another notation is presented in Ref. [33, 42]).

Order-independence was also a key point considered by Worsch, who examined how to simulate an arbitrary rule by a universal asynchronous simulator [136]. He extended Golze's results by tackling a large scope of updating policies: *purely asynchronous* (no restriction on the set of cells to update), α-asynchronous, *N-independent* (where two neighbouring cells are never updated at the same time), and non-deterministic fully asynchronous. He showed that for each such policy, there is a universal rule (the host) that can "simulate", in a particular sense, any other guest. Worsch's work raises many questions, in particular as to how to properly define the notion of simulation of an asynchronous rule by another. (See Ref. [7] for some reflexions made in the context of stochastic cellular automata.)

We also point out that Vielhaber has designed a formal framework in which the computations of functions on finite binary rings ($\mathbb{Z}/n\mathbb{Z}$) are made not by changing the local rule but by a proper use of the order of updating on a *fixed* rule [132]. In particular, he showed that ECA 57 with periodic boundary conditions was a rule especially adapted for such a purpose. Interestingly, this technique could be generalised to make this particular rule Turing-universal in the sense that the computation of an algorithm could be done only by setting up the proper sequence of updates.

5.3 Models of concurrency

Among the early references that can be found on asynchronous cellular automata, Priese wrote a note where he considers (two-dimensional) cellular automata as a particular case of asynchronous rewriting systems (called Thue-systems) and widens the scope by considering also the case where more than one cell may be re-written at a time (the overlapping problem) [93]. He uses his construction to show how to build asynchronous circuits which are equivalent to asynchronous concurrent Petri nets.

Following this path, Zielonka examined how asynchronous rules could be used to describe the situations of concurrency that arise in distributed systems [25]. Pighizzini clarified the computing abilities of Zielonska's models [92] and the problem of how to turn non-deterministic Büchi asynchronous cellular automata into deterministic models was solved by Muscholl [80]. Droste generalised to partially ordered multisets (pomsets) the original notion of Zielonska's asynchronous mappings [31]; these questions were later re-investigated by Kuske [32, 61, 62].

With similar preoccupations, Hagiya et al. used formal methods from logic to verify the properties of some rewriting systems, showing the links between their approach and (a)synchronous systems [53].

5.4 Asynchronous circuits

Another major field of research on asynchronous cellular automata was developed by Peper, Lee and their collaborators. In their constructions, asynchronous computations are realised with particles that follow Brownian movements and which interact through special "gates" [4, 3, 66, 91]. These constructions result in delay-insensitive circuits that are Turing-universal (see e.g. [64, 69] and references therein).

Recently, Schneider and Worsch presented a 3-state rule that uses Moore neighbourhood which can simulate any delay-insensitive circuit [112] and Lee et al. presented a generalisation of their work in the context of number-conserving cellular automata [68].

To end this section, we propose to put an emphasis on the following questions:

Questions 4 *What is a good definition of the simulation of an (a)synchronous system by another (a)synchronous system? What are the techniques to simulate various asynchronous systems by other asynchronous systems? (e.g.: When can an α-asynchronous system with a given synchrony rate simulate another system with a different synchrony rate?)*

6 MODELLING WITH ASYNCHRONOUS CELLULAR AUTOMATA

Asynchronous rules have been designed for specific goals such as finding new algorithms, developing new types of computing devices, modelling various natural or artificial complex systems, etc. In fact, giving a representative view of these contributions would necessitate a whole independent survey. The task is all the more difficult as often authors use asynchronous updating without even mentioning it. For the sake of brevity, we will thus only give a few entry points, concentrating on the papers where the question of the updating is explicitly discussed.

6.1 Game theory and Ecology

As mentioned earlier, the hypothesis of perfect synchrony poses the problem of the *realism* of a model: How to interpret the behaviours that are only due to the updating and not to the rule that governs the cells? Huberman and Glance gave evidence of the existence of such "artifacts" and challenged the validity of the simulations of spatially-extended models of the Prisoner's dilemma: a change in the updating models brings out new conclusions, drastically opposed to what was known with the classical models [58]. This question was re-examined by Newth and Cornforth who showed that asynchronism could also lead to the observation of new cooperative phenomena not seen in the synchronous setting [86]. (See also Ref. [85] for a non-spatially-extended version of the problem.)

Grilo and Correia also considered this problem but instead of restricting their study to the fully asynchronous scheme, they employed α-asynchronous updating to explore a wide range of degrees of synchrony [51]. Their study revealed that the changes induced by smooth variations of the synchrony rate may brutally affect the level of cooperation in the system, a behaviour that is strongly reminiscent of the second-order phase transition seen in binary systems (see Sec. 3.2). Saif and Gade also investigated this issue and found that there was a first order transition between a regime with an all-defector state to a mixed state [107]. All these works share in common the conclusion that some previously observed equilibrium states are artifacts of a synchronous updating on a regular lattice.

Ruxton and Saravia have discussed the importance of the ordering in the context of ecological modelling, studying a stochastic model of colonisation of an environment by a species [106]. They argue in favour of adapting the updating scheme to the physical reality of the system that is modelled. The authors also emphasise the need to describe precisely the updating scheme that is used to facilitate the reproducibility of the experiments. These arguments come to strengthen the need for studying in detail the "emergence phenomena" that are seen in Ecology and question whether the predictions of the models can be observed in "real-life" systems [100, 22].

6.2 Synchronisation in physical and biological systems

In an approach close to the work of Turing on morphogenesis [130], Gunji used asynchronous cellular automata to analyse the pattern formation mechanisms that occur in molluscs [52]. Another interesting biological example is given by Messinger et al., who investigated the link between emergence of synchrony and the simultaneous opening and closing stomatal arrays in plants [78].

In Physics, we mention the work of Le Caër [63] and Radicchi et al. [94], who studied how numerical simulations of cellular systems would be dependent on the updating. In the latter work, the local rule is itself stochastic; the authors emphasise the fact that neither totally synchronous nor totally asynchronous updating is fully relevant for modelling natural systems.

6.3 Problem solving

Tomassini and Venzi [127], Capcarrere [21] and Nehaniv [83] have studied how asynchronous rules solve the density classification problem and the global synchronisation problem. Readers interested in this issue are referred to a study by Vanneschi and Mauri, in which an enlightening discussion on these various contributions is found and where the authors present findings of robust and generic rules [131].

Suzudo examined the use of genetic algorithms to find mass-conservative (also called number-conserving) asynchronous models that would generate non-trivial patterns [122, 123]. He classified these patterns into three categories: checkerboards, stripes and sand-like. In this work asynchronism is mainly used to ensure that number of particles remain constant, but it is also a useful technique for generating regular patterns out of randomness: this task is known to be difficult in the synchronous setting (see e.g. [37]).

Beigy and Meybodi investigated how asynchronous systems perform learning tasks and presented applications of their work for pattern generation and control of cellular mobile networks [12].

It is also worth mentioning that Lee et al. [65] and Huang et al. [57] designed models of self-reproduction that use asynchronous models (also called self-timed cellular automata).

6.4 Other problems

On the simulation side, Overeinder and Sloot were among the first to examine how to deal with the simulation of asynchronous automata on distributed systems [88]. Bandman and other authors studied how to simulate chemical systems with asynchronous cellular automata [11, 117]. Hoseini et al. made an implementation of asynchronous rules with FPGAs [56]. They propose a particular design of the FPGA in order to construct a "conformal computer", that is, a computer made of physical cells "arrayed on large thin flexible substrates or sheets. Sheets may be cut, joined, bent, and stacked to conform to the physical and computational needs of an application".

Original applications were considered by Bandini et al., who used asynchronous rules with memory for the design of an illumination facility [10] and by Minoofam et al., where asynchronism produce calligraphic patterns in the Arabic Kufic style [79]. (Unfortunately, this paper lacks precision on the model that is used).

As we have seen in this section, there is a broad range of domains where asynchronous models have been employed and those which we cited above are only a small part.

Questions 5 *How can we develop a unified simulation environment to facilitate the comparison of various updating schemes? Is there a method for identifying the artifacts that are due to a perfect synchronous updating? Are such effects avoidable?*

7 ASYNCHRONISM IN OTHER DISCRETE MODELS OF COMPLEX SYSTEMS

We end this guided tour on an opening on the use of asynchrony in the systems whose structure is close to cellular automata. Again, this is such a wide topic that we will indicate only a few entry points to the literature.

7.1 Links with multi-agent systems

One first proposition to link the updating in multi-agent systems and cellular automata was made by Cornforth et al., but the models they studied are in fact standard asynchronous cellular automata [26]. Spicher et al. considered the question of how to "translate" a multi-agent system with sequential updating into a synchronous cellular automaton [119]. So-called *transactional cellular automata* were defined to model the movements of particles between neighbouring cells. One positive effect of using a synchronous cellular automaton is to remove the spurious effects that could be linked to a particular updating order. (The authors give the example of diffusion-limited aggregation.)

The link between large-scale multi-agent systems and asynchronous cellular automata was also examined by Tošić [129]. This author argues that the structure of cellular automata needs to be modified in several aspects, among which it should be made asynchronous, in order to serve as a basis for modelling large groups of interacting agents.

An alternative approach to model (discrete) multi-agent systems was proposed by Chevrier and Fatès, who studied the dynamics of a simple multi-turmite systems, also known as multiple Langton's ants. Their formalism, inspired by cellular automata, captures the possibility to have *synchronous* interacting agents [24]. The difficulty relies in describing how to solve conflicts that occur when two or more agents simultaneously want to modify the environment. The solution relies on a framework invented by Ferber and Müller called *influence-reaction* [44]. Belgacem and Fatès later extended this work by considering a wider range of updating procedures and discovered some phenomena (e.g., gliders) that resisted variations in the updating choices [13].

Interesting observations were also made by Şamiloğlu et al. who analysed the clustering effects in a group of self-propelled particles [108]. They model asynchronism with the introduction of delays in the updating and observe that the coherence of the groups are strongly diminished as the bounds on the delays are increased.

7.2 Lattice-gas cellular automata

Lattice-Gas Cellular Automata (LGCA) can be seen as a "bridge" between cell-based updating and agent-based updating. Applying asynchrony in this context is not a straightforward operation and a first proposition of an asynchronous LGCA was made by Bouré et al. [18]. In their model, movements of particles are defined explicitly, like in multi-agents, but the updating is made cell by cell, like in classical cellular automata. Various responses to asynchrony are observed depending on the patterns on which the system stabilises. In particular, strange patterns such as checkerboards are shown to disappear where randomness in the updating is added. It is an open problem to know if an infinitesimal amount of asynchrony is sufficient to destroy this pattern.

These first results show the need to explore various possibilities to define an asynchronous LGCA. In particular, it is interesting to look at a way to update particles independently.

7.3 Automata networks, neural networks and other models

The effect of asynchronous updating in genetic regulatory networks has also been investigated by many authors. Aracena et al. introduce a labelled directed graph that allows one to determine to which extent deterministic update schedules are equivalent [6]. Demongeot et al. [28], and Noual [87] examine the robustness of the system under the variation of updating schemes and this perturbation is coupled with various topological modifications of the network such as adding or removing links in the graph or changing boundary conditions.

The question of the effect of the updating in neural networks has been discussed by Scherrer [111], Taouali et al. [125]. In particular, the latter authors introduce an interesting distinction between the use of (a)synchronous updating at the modelling level and at the implementation level.

In the context of "amorphous computing", Stark discussed the computing abilities of a computing medium formed out of non-regularly placed cells which obey asynchronous updating [121, 120]. This author suggests that asynchrony plays an enhancing role for the computing abilities of such systems.

We mention that the differences between synchronous and asynchronous updating were also investigated in coupled map lattices [71, 101, 1]. Similarly, the effects of the updating in the Asymmetric Exclusion Process (ASEP) have been studied by Rajewsky et al. [95]. Tomita et al. studied asynchronous graph-rewriting systems and showed how to make such systems simulate their synchronous counterparts [128].

To end this section, we wish to highlight the following questions:

Questions 6 *What light can be shed by asynchronous cellular automata on other closely related models and vice versa? Can we transfer the techniques used to analyse the simple asynchronous cellular systems to more complex models? What is the interplay between the regular topology of cellular automata and the regularity of their updating?*

8 CLOSING WORDS

This guided tour allowed us to consider the various contributions that deal with the question of asynchronism in cellular automata and closely related models. As we have seen, asynchrony is a *privative* property that does not in itself specify a system: there are plenty of ways to construct an asynchronous system and all of them are a priori valid.

One of the main current challenges is to continue to explore this question with a *joint work* of mathematical analysis and numerical simulations. As we have seen, analytical results have been more difficult to obtain than numerical ones, but the situation is progressively changing as more techniques from the probability theory are being developed for the specific case of cellular automata. We find it rather amazing that it is still an open question to determine the convergence time of some simple binary rules [38, 39].

It is also important to clarify the position of asynchronous cellular automata into the wider field of stochastic cellular automata. Indeed, asynchrony is not a mere type of noise: recall for example that the addition of asynchrony to a deterministic model does not change its fixed points. However, many phenomena such as the existence of singularities or phase transitions can certainly find their explanations using the stochastic process theory and statistical physics.

As far as modelling is concerned, the main challenge would be to carry out an experimental work to validate some models of asynchronism or to dismiss some others for *specific situations*. As we mentioned earlier, the no-global-clock argument — "Nature does not possess a clock to synchronise the transitions." — cannot be received directly and be taken alone as a valid objection to the use of synchronous models. Instead, we consider that studying a *single* updating scheme is not sufficient and one should instead compare various possibilities to model a "natural computing" system.

The principal observation from this guided tour is the existence of a great variety of approaches to asynchronism. This raises the question of *what time is* in the context of computer science and numerical simulations. The positive sciences define time as an object – identified with $\mathbb{R}$, with $\mathbb{Z}$, a collection of coordinates, etc. – but it may well be that time is not some "thing" that can be studied "objectively".

Does this mean that time is subjective and that our models should reflect this subjectivity? Such considerations would lead us out of the scientific method and would therefore be dismissed as non rational. Can we then escape the dilemma of "objective versus subjective time"? No simple answer can be given and for sure, time is one of the central problems of philosophy. It is certainly not a coincidence if one the most important philosophical contributions of the past century bears as title: *Sein und Zeit* (*Being and Time*).

ACKNOWLEDGEMENTS

May all the persons that have contributed to the elaboration of this text receive our sincere expression of gratefulness. The author knows how much he owes to them, including the organisers of Automata'13 and their precious support, and the anonymous referees, whose constructive propositions were greatly appreciated. In echo to Heraclitus' word that "the invisible harmony is greater than the apparent one" (DK B54), he hopes that they will pardon him for not being named here. The author is aware of the *limits* of this text and he will be grateful to all the corrections, indications, remarks, and suggestions that will be given to him.

REFERENCES

[1] Guillermo Abramson and Damián H. Zanette. (1998). Globally coupled maps with asynchronous updating. *Physical Review E*, 58:4454–4460.

[2] Svetlana M. Achasova. (1995). Synchronous-asynchronous cellular computations. In Victor Malyshkin, editor, *Parallel Computing Technologies*, volume 964 of *Lecture Notes in Computer Science*, pages 1–6. Springer Berlin Heidelberg.

[3] Susumu Adachi, Ferdinand Peper, and Jia Lee. (2004). Computation by asynchronously updating cellular automata. *Journal of Statistical Physics*, 114(1-2):261–289.

[4] Susumu Adachi, Ferdinand Peper, and Jia Lee. (2004). Universality of hexagonal asynchronous totalistic cellular automata. In Peter Sloot, Bastien Chopard, and Alfons Hoekstra, editors, *Proceedings of ACRI'04*, volume 3305 of *Lecture Notes in Computer Science*, pages 91–100. Springer Berlin Heidelberg.

[5] Alexandru Agapie, Robins Höns, and Heinz Mühlenbein. (2004). Markov chain analysis for one-dimensional asynchronous cellular automata. *Methodology And Computing In Applied Probability*, 6(21):181–201.

[6] Julio Aracena, Eric Goles, Andrés Moreira, and Lilian Salinas. (2009). On the robustness of update schedules in boolean networks. *Biosystems*, 97(1):1–8.

[7] Pablo Arrighi, Nicolas Schabanel, and Guillaume Theyssier. (2013). Stochastic cellular automata: Correlations, decidability and simulations. *Fundamenta Informaticae*, 126(2-3):121–156.

[8] Jan M. Baetens, Pieter Van der Weeën, and Bernard De Baets. (2012). Effect of asynchronous updating on the stability of cellular automata. *Chaos, Solitons & Fractals*, 45(4):383–394.

[9] Stefania Bandini, Andrea Bonomi, and Giuseppe Vizzari. (2012). An analysis of different types and effects of asynchronicity in cellular automata update schemes. *Natural Computing*, 11(2):277–287.

[10] Stefania Bandini, Andrea Bonomi, Giuseppe Vizzari, and Vito Acconci. (2009). An asynchronous cellular automata-based adaptive illumination facility. In Roberto Serra and Rita Cucchiara, editors, *Proceedings of AI*IA 2009: Emergent Perspectives in Artificial Intelligence*, volume 5883 of *Lecture Notes in Computer Science*, pages 405–415. Springer Berlin Heidelberg.

[11] Olga L. Bandman. (2010). Parallel composition of asynchronous cellular automata simulating reaction diffusion processes. In Stefania Bandini, Sara Manzoni, Hiroshi Umeo, and Giuseppe Vizzari, editors, *Proceedings of ACRI'10*, volume 6350 of *Lecture Notes in Computer Science*, pages 395–398. Springer.

[12] Hamid Beigy and M.R. Meybodi. (2008). Asynchronous cellular learning automata. *Automatica*, 44(5):1350–1357.

[13] Selma Belgacem and Nazim Fatès. (2012). Robustness of multi-agent models: the example of collaboration between turmites with synchronous and asynchronous updating. *Complex Systems*, 21(3):165–182.

[14] Hugues Bersini and Vincent Detours. (1994). Asynchrony induces stability in cellular automata based models. In Rodney A. Brooks and Pattie Maes, editors, *Proceedings of the 4th International Workshop on the Synthesis and Simulation of Living Systems – Artificial Life IV*, pages 382–387. MIT Press.

[15] Hendrik J. Blok and Birger Bergersen. (1999). Synchronous versus asynchronous updating in the "game of life". *Physical Review E*, 59:3876–9.

[16] Olivier Bouré. (2013). *" Le simple est-il robuste ? " : une étude de la robustesse des systèmes complexes par les automates cellulaires*. PhD thesis, Université de Lorraine.

[17] Olivier Bouré, Nazim Fatès, and Vincent Chevrier. (2012). Probing robustness of cellular automata through variations of asynchronous updating. *Natural Computing*, 11:553–564.

[18] Olivier Bouré, Nazim Fatès, and Vincent Chevrier. (2013). First steps on asynchronous lattice-gas models with an application to a swarming rule. *Natural Computing*, 12(4):551–560.

[19] Buvel, R.L. and Ingerson, Thomas E. (1984). Structure in asynchronous cellular automata. *Physica D*, 1:59–68.

[20] Ana Bušić, Nazim Fatès, Jean Mairesse, and Irène Marcovici. (2013). Density classification on infinite lattices and trees. *Electronic Journal of Probability*, 18(51):1–22.

[21] Mathieu Capcarrere. (2002). Evolution of asynchronous cellular automata. In Juan Julián Merelo Guervós, Panagiotis Adamidis, Hans-Georg Beyer, Hans-Paul Schwefel, and José-Luis Fernández-Villacañas, editors, *Proceedings of PPSN VII*, volume 2439 of *Lecture Notes in Computer Science*, pages 903–912. Springer.

[22] Geoffrey Caron-Lormier, Roger W. Humphry, David A. Bohan, Cathy Hawes, and Pernille Thorbek. (2008). Asynchronous and synchronous updating in individual-based models. *Ecological Modelling*, 212(3–4):522–527.

[23] Philippe Chassaing and Lucas Gerin. (2007). Asynchronous cellular automata and brownian motion. In *DMTCS Proceedings of AofA'07*, volume AH, pages 385–402.

[24] Vincent Chevrier and Nazim Fatès. (2010). How important are updating schemes in multi-agent systems? An illustration on a multi-turmite model. In *Proceedings of AAMAS '10*, pages 533–540, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

[25] Robert Cori, Yves Métivier, and Wieslaw Zielonka. (1993). Asynchronous mappings and asynchronous cellular automata. *Information and Computation*, 106(2):159–202.

[26] David Cornforth, David G. Green, and David Newth. (2005). Ordered asynchronous processes in multi-agent systems. *Physica D*, 204(1–2):70 – 82.

[27] Sukanta Das, Anindita Sarkar, and Biplab K. Sikdar. (2012). Synthesis of reversible asynchronous cellular automata for pattern generation with specific hamming distance. In Georgios Ch. Sirakoulis and Stefania Bandini, editors, *Proceedings of ACRI'12*, volume 7495 of *Lecture Notes in Computer Science*, pages 643–652. Springer.

[28] Jacques Demongeot, Eric Goles, Michel Morvan, Mathilde Noual, and Sylvain Sené. (2010). Attraction basins as gauges of robustness against boundary conditions in biological complex systems. *PLoS One*, 5(8):e11793.

[29] Alberto Dennunzio, Enrico Formenti, and Luca Manzoni. (2012). Computing issues of asynchronous CA. *Fundamenta Informaticae*, 120(2):165–180.

[30] Alberto Dennunzio, Enrico Formenti, Luca Manzoni, and Giancarlo Mauri. (2013). m-asynchronous cellular automata: from fairness to quasi-fairness. *Natural Computing*, 12(4):561–572.

[31] Manfred Droste and Paul Gastin. (1996). Asynchronous cellular automata for pomsets without auto-concurrency. In Ugo Montanari and Vladimiro Sassone, editors, *Proceedings of CONCUR'96*, volume 1119 of *Lecture Notes in Computer Science*, pages 627–638. Springer Berlin Heidelberg.

[32] Manfred Droste, Paul Gastin, and Dietrich Kuske. (2000). Asynchronous cellular automata for pomsets. *Theoretical Computer Science*, 247(1-2):1–38.

[33] Nazim Fatès. (2004). *Robustesse de la dynamique des systèmes discrets : le cas de l'asynchronisme dans les automates cellulaires*. PhD thesis, École normale supérieure de Lyon.

[34] Nazim Fatès. (2009). Asynchronism induces second order phase transitions in elementary cellular automata. *Journal of Cellular Automata*, 4(1):21–38.

[35] Nazim Fatès. (2010). Critical phenomena in cellular automata: perturbing the update, the transitions, the topology. *Acta Physica Polonica B - Proceedings Supplement*, 3(2):315–325.

[36] Nazim Fatès. (2010). Does *life* resist asynchrony? In Andrew Adamatzky, editor, *Game of Life Cellular Automata*, pages 257–274. Springer London.

[37] Nazim Fatès. (2012). A note on the density classification problem in two dimensions. In *Exploratory papers presented at Automata 2012*, La Marana, Corse, France.

[38] Nazim Fatès. (2013). A note on the classification of the most simple asynchronous cellular automata. In Jarkko Kari, Martin Kutrib, and Andreas Malcher, editors, *Proceedings of Automata'13*, volume 8155 of *Lecture Notes in Computer Science*, pages 31–45. Springer.

[39] Nazim Fatès. (2014). Quick convergence to a fixed point: A note on asynchronous elementary cellular automata. In Jarosław Wąs, Georgios Ch. Sirakoulis, and Stefania Bandini, editors, *Proceedings of ACRI'14*, volume 8751 of *Lecture Notes in Computer Science*, pages 586–595. Springer.

[40] Nazim Fatès and Lucas Gerin. (2009). Examples of fast and slow convergence of 2D asynchronous cellular systems. *Journal of Cellular Automata*, 4(4):323–337.

[41] Nazim Fatès and Michel Morvan. (2005). An experimental study of robustness to asynchronism for elementary cellular automata. *Complex Systems*, 16:1–27.

[42] Nazim Fatès, Michel Morvan, Nicolas Schabanel, and Eric Thierry. (2006). Fully asynchronous behavior of double-quiescent elementary cellular automata. *Theoretical Computer Science*, 362:1–16.

[43] Nazim Fatès, Damien Regnault, Nicolas Schabanel, and Eric Thierry. (2006). Asynchronous behavior of double-quiescent elementary cellular automata. In José R. Correa, Alejandro Hevia, and Marcos A. Kiwi, editors, *Proceedings of LATIN 2006*, volume 3887 of *Lecture Notes in Computer Science*, pages 455–466. Springer.

[44] Jacques Ferber and Jean-Pierre Müller. (1996). Influences and reaction: A model of situated multiagent systems. In *Proceedings of the 2nd International Conference on Multi-agent Systems*, pages 72–79.

[45] Enrico Formenti. (2013). A survey on m-asynchronous cellular automata. In Jarkko Kari, Martin Kutrib, and Andreas Malcher, editors, *Proceedings of Automata'13*, volume 8155 of *Lecture Notes in Computer Science*, pages 46–66. Springer.

[46] Henryk Fukś and Nazim Fatès. (2014). Bifurcations of local structure maps as predictors of phase transitions in asynchronous cellular automata. In Jarosław Wąs, Georgios Ch. Sirakoulis, and Stefania Bandini, editors, *Proceedings of ACRI'14*, volume 8751 of *Lecture Notes in Computer Science*, pages 556–560. Springer.

[47] Henryk Fukś and Andrew Skelton. (2011). Orbits of the bernoulli measure in single-transition asynchronous cellular automata. In Nazim Fatès, Eric Goles, Alejandro Maass, and Ivan Rapaport, editors, *Proceedings of Automata'11*, Discrete Mathematics and Theoretical Computer Science Proceedings, pages 95–112. DMTCS.

[48] Péter Gács. (2001). Deterministic computations whose history is independent of the order of asynchronous updating. *Arxiv CoRR*, cs.DC/0101026.

[49] Reza Gharavi and Venkat Anantharam. (1992). Effect of noise on long-term memory in cellular automata with asynchronous delays between the processors. *Complex Systems*, 6(3):287–300.

[50] Ulrich Golze. (1978). (A-)synchronous (non-)deterministic cell spaces simulating each other. *Journal of Computer and System Sciences*, 17(2):176–193.

[51] Carlos Grilo and Luís Correia. (2011). Effects of asynchronism on evolutionary games. *Journal of Theoretical Biology*, 269(1):109–122.

[52] Yukio Gunji. (1990). Pigment color patterns of molluscs as an autonomous process generated by asynchronous automata. *Biosystems*, 23(4):317–334.

[53] Masami Hagiya, Koichi Takahashi, Mitsuharu Yamamoto, and Takahiro Sato. (2004). Analysis of synchronous and asynchronous cellular automata using abstraction by temporal logic. In Yukiyoshi Kameyama and Peter Stuckey, editors, *Proccedings of FLOPS'04 : Functional and Logic Programming*, volume 2998 of *Lecture Notes in Computer Science*, pages 7–21. Springer Berlin Heidelberg.

[54] Armin Hemmerling. (1982). On the computational equivalence of synchronous and asynchronous cellular spaces. *Elektronische Informationsverarbeitung und Kybernetik*, 18(7/8):423–434.

[55] Jörg Hoffmann, Nazim Fatès, and Héctor Palacios. (2010). Brothers in arms? On AI planning and cellular automata. In Helder Coelho, Rudi Studer, and Michael Wooldridge, editors, *Proceedings of ECAI 2010*, volume 215 of *Frontiers in Artificial Intelligence and Applications*, pages 223–228. IOS Press.

[56] Mariam Hoseini, Zhou Tan, Chao You, and Mark Pavicic. (2010). Design of a reconfigurable pulsed quad-cell for cellular-automata-based conformal computing. *International Journal of Reconfigurable Computing*, 7907:Article ID 352428.

[57] Xin Huang, Jia Lee, Tian-Hao Sun, and Ferdinand Peper. (2013). Self-adaptive self-reproductions in cellular automata. *Physica D*, 263:11 – 20.

[58] Bernardo A. Huberman and Natalie Glance. (1993). Evolutionary games and computer simulations. *Proceedings of the National Academy of Sciences, USA*, 90:7716–7718.

[59] Yasusi Kanada. (1994). The effects of randomness in asynchronous 1D cellular automata (poster). *Artificial Life IV*.

[60] Jarkko Kari. (2005). Theory of cellular automata: A survey. *Theoretical Computer Science*, 334(1–3):3 – 33.

[61] Dietrich Kuske. (2000). Emptiness is decidable for asynchronous cellular machines. In Catuscia Palamidessi, editor, *Proceedings of CONCUR 2000 – Concurrency Theory*, volume 1877 of *Lecture Notes in Computer Science*, pages 536–551. Springer Berlin Heidelberg.

[62] Dietrich Kuske. (2007). Weighted asynchronous cellular automata. *Theoretical Computer Science*, 374(1-3):127–148.

[63] Gérard Le Caër. (1989). Comparison between simultaneous and sequential updating in $2^{n+1} - 1$ cellular automata. *Physica A*, 157(2):669 – 687.

[64] Jia Lee. (2011). A simple model of asynchronous cellular automata exploiting fluctuation. *Journal of Cellular Automata*, 6(4-5):341–352.

[65] Jia Lee, Susumu Adachi, and Ferdinand Peper. (2007). Reliable self-replicating machines in asynchronous cellular automata. *Artificial Life*, 13(4):397–413.

[66] Jia Lee, Susumu Adachi, Ferdinand Peper, and Shinro Mashiko. (2005). Delay-insensitive computation in asynchronous cellular automata. *Journal of Computer and System Sciences*, 70(2):201–220.

[67] Jia Lee, Susumu Adachi, Ferdinand Peper, and Kenichi Morita. (2004). Asynchronous game of life. *Physica D*, 194(3–4):369–384.

[68] Jia Lee, Katsunobu Imai, and Qing sheng Zhu. (2012). Fluctuation-driven computing on number-conserving cellular automata. *Information Sciences*, 187:266–276.

[69] Jia Lee and Qing-Sheng Zhu. (2012). A direct proof of turing universality of delay-insensitive circuits. *International Journal of Unconventional Computing*, 8(2):107–118.

[70] R.J. Lipton, R.E. Miller, and L. Snyder. (1977). Synchronization and computing capabilities of linear asynchronous structures. *Journal of Computer and System Sciences*, 14(1):49–72.

[71] Erik D. Lumer and Grégoire Nicolis. (1994). Synchronous versus asynchronous dynamics in spatially distributed systems. *Physica D: Nonlinear Phenomena*, 71(4):440–452.

[72] Matthew Macauley, Jon McCammond, and Henning Mortveit. (2011). Dynamics groups of asynchronous cellular automata. *Journal of Algebraic Combinatorics*, 33(1):11–35.

[73] Matthew Macauley, Jon McCammond, and Henning S. Mortveit. (2008). Order independence in asynchronous cellular automata. *Journal of Cellular Automata*, 3(1):37–56.

[74] Matthew Macauley and Henning S. Mortveit. (2010). Coxeter groups and asynchronous cellular automata. In Stefania Bandini, Sara Manzoni, Hiroshi Umeo, and Giuseppe Vizzari, editors, *Proceedings of ACRI'10*, volume 6350 of *Lecture Notes in Computer Science*, pages 409–418. Springer.

[75] Marco Mamei, Andrea Roli, and Franco Zambonelli. (2005). Emergence and control of macro-spatial structures in perturbed cellular automata, and implications for pervasive computing systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 35(3):337–348.

[76] Luca Manzoni. (2012). Asynchronous cellular automata and dynamical properties. *Natural Computing*, 11(2):269–276.

[77] Genaro Martínez. (2013). A note on elementary cellular automata classification. *Journal of Cellular Automata*, 8(34-):233–259. preprint on arXiv 1306.5577.

[78] Susanna M. Messinger, Keith A. Mott, and David Peak. (2007). Task-performing dynamics in irregular, biomimetic networks: Research articles. *Complexity*, 12(6):14–21.

[79] Seyyed Amir Hadi Minoofam and Azam Bastanfard. (2010). Square kufic pattern formation by asynchronous cellular automata. In Stefania Bandini, Sara Manzoni, Hiroshi Umeo, and Giuseppe Vizzari, editors, *Proceedings of ACRI'10*, volume 6350 of *Lecture Notes in Computer Science*, pages 79–82. Springer Berlin Heidelberg.

[80] Anca Muscholl. (1994). On the complementation of Büchi asynchronous cellular automata. In Serge Abiteboul and Eli Shamir, editors, *Automata, Languages and Programming*, volume 820 of *Lecture Notes in Computer Science*, pages 142–153. Springer Berlin Heidelberg.

[81] Katsuhiko Nakamura. (1974). Asynchronous cellular automata and their computational ability. *Systems, Computers, Controls*, 5(5):58–66.

[82] Katsuhiko Nakamura. (1981). Synchronous to asynchronous transformation of polyautomata. *Journal of Computer and System Sciences*, 23(1):22 – 37.

[83] Chrystopher L. Nehaniv. (2003). Evolution in asynchronous cellular automata. In *Proceedings of the eighth international conference on Artificial life*, pages 65–73. MIT Press.

[84] Chrystopher L. Nehaniv. (2004). Asynchronous automata networks can emulate any synchronous automata network. *International Journal of Algebra and Computation*, 14(5-6):719–739.

[85] David Newth. (2009). Asynchronous iterated prisoner's dilemma. *Adaptive Behavior*, 17(2):175–183.

[86] David Newth and David Cornforth. (2009). Asynchronous spatial evolutionary games. *Biosystems*, 95(2):120–129.

[87] Mathilde Noual. (2011). Synchronism vs asynchronism in boolean networks. *CoRR*, abs/1104.4039.

[88] Benno J. Overeinder and Peter M. A. Sloot. (1993). Application of time warp to parallel simulations with asynchronous cellular automata. In *Proceedings of the 1993 European Simulation Symposium*, pages 397–402.

[89] Ferdinand Peper, Susumu Adachi, and Jia Lee. (2010). Variations on the game of life. In Andrew Adamatzky, editor, *Game of Life Cellular Automata*, pages 235–255. Springer London.

[90] Ferdinand Peper, Teijiro Isokawa, Noriaki Kouda, and Nobuyuki Matsui. (2002). Self-timed cellular automata and their computational ability. *Future Generation Computer Systems*, 18(7):893–904.

[91] Ferdinand Peper, Jia Lee, and Teijiro Isokawa. (2010). Brownian cellular automata. *Journal of Cellular Automata*, 5(3):185–206.

[92] Giovanni Pighizzini. (1994). Asynchronous automata versus asynchronous cellular automata. *Theoretical Computer Science*, 132(2):179–207.

[93] Lutz Priese. (1978). A note on asynchronous cellular automata. *Journal of Computer and System Sciences*, 17(2):237–252.

[94] Filippo Radicchi, Daniele Vilone, and Hildegard Meyer-Ortmanns. (2007). Phase transition between synchronous and asynchronous updating algorithms. *Journal of Statistical Physics*, 129(3):593–603.

[95] Nikolaus Rajewsky, Ludger Santen, Andreas Schadschneider, and Michael Schreckenberg. (1998). The asymmetric exclusion process: Comparison of update procedures. *Journal of Statistical Physics*, 92(1-2):151–194.

[96] Nikolaus Rajewsky and Michael Schreckenberg. (1997). Exact results for one-dimensional cellular automata with different types of updates. *Physica A*, 245(1–2):139 – 144.

[97] Damien Regnault. (2008). *Sur les automates cellulaires probabilistes comportements asynchrones*. PhD thesis, École Normale Supérieure de Lyon.

[98] Damien Regnault. (2013). Proof of a phase transition in probabilistic cellular automata. In Marie-Pierre Béal and Olivier Carton, editors, *Proceedings of Developments in Language Theory*, volume 7907 of *Lecture Notes in Computer Science*, pages 433–444. Springer.

[99] Damien Regnault, Nicolas Schabanel, and Eric Thierry. (2009). Progresses in the analysis of stochastic 2D cellular automata: A study of asynchronous 2D minority. *Theoretical Computer Science*, 410(47-49):4844–4855.

[100] Pejman Rohani, Timothy J. Lewis, Daniel Grünbaum, and Graeme D. Ruxton. (1997). Spatial self-organisation in ecology: pretty patterns or robust reality? *Trends in Ecology & Evolution*, 12(2):70–74.

[101] Juri Rolf, Tomas Bohr, and Mogens H. Jensen. (1998). Directed percolation universality in asynchronous evolution of spatiotemporal intermittency. *Physical Review E.*, 57(3):R2503–R2506.

[102] Jean-Baptiste Rouquier. (2008). *Robustesse et émergence dans les systèmes complexes : le modèle des automates cellulaires*. PhD thesis, École Normale Supérieure de Lyon.

[103] Jean-Baptiste Rouquier. (2009). An exhaustive experimental study of synchronization by forcing on elementary cellular automata. In *Proceedings of the First Symposium on Cellular Automata "Journées Automates Cellulaires" (JAC 2008)*, pages 250–261. MCCME Publishing House, Moscow.

[104] Jean-Baptiste Rouquier and Michel Morvan. (2009). Coalescing cellular automata: Synchronization by common random source for asynchronous updating. *Journal of Cellular Automata*, 4(1):55–78.

[105] Jean-Baptiste Rouquier, Damien Regnault, and Éric Thierry. (2011). Stochastic minority on graphs. *Theoretical Computer Science*, 412(30):3947–3963.

[106] Graeme Ruxton and Leonardo Saravia. (1998). The need for biological realism in the updating of cellular automata models. *Ecological Modelling*, 107(2):105–112.

[107] M. Ali Saif and Prashant M. Gade. (2009). The prisoner's dilemma with semi-synchronous updates: evidence for a first-order phase transition. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(7):P07023.

[108] Andaç T Şamiloğlu, Veysel Gazi, and A Buğra Koku. (2006). Effects of asynchronism and neighborhood size on clustering in self-propelled particle systems. In Albert Levi, Erkay Savaş, Hüsnü Yenigün, Selim Balcisoy, and Yücel SaygÄśn, editors, *Proceedings of ISCIS 2006*, volume 4263 of *Lecture Notes in Computer Science*, pages 665–676. Springer Berlin Heidelberg.

[109] J. R. Sánchez and R. López-Ruiz. (2006). Self-synchronization of cellular automata: An attempt to control patterns. In Vassil N. Alexandrov, G. Dick van Albada, Peter M. A. Sloot, and Jack Dongarra, editors, *Proceedings of the 6th International Conference on Computational Science - Part III*, volume 3993 of *Lecture Notes in Computer Science*, pages 353–359. Springer.

[110] Anindita Sarkar, Anindita Mukherjee, and Sukanta Das. (2012). Reversibility in asynchronous cellular automata. *Complex Systems*, 21(1):71.

[111] Bruno Scherrer. (2005). Asynchronous neurocomputing for optimal control and reinforcement learning with large state spaces. *Neurocomputing*, 63:229 – 251.

[112] Oliver Schneider and Thomas Worsch. (2012). A 3-state asynchronous CA for the simulation of delay-insensitive circuits. In Georgios Ch. Sirakoulis and Stefania Bandini, editors, *Proceedings of ACRI'12*, volume 7495 of *Lecture Notes in Computer Science*, pages 565–574. Springer.

[113] Birgitt Schönfisch and André de Roos. (1999). Synchronous and asynchronous updating in cellular automata. *BioSystems*, 51:123–143.

[114] Martin Schüle. (2012). *Natural computation*. PhD thesis, Eidgenössische Technische Hochschule (ETH) Zürich.

[115] Martin Schüle and Ruedi Stoop. (2012). A full computation-relevant topological dynamics classification of elementary cellular automata. *Chaos*, 22(4):043143.

[116] Biswanath Sethi, Nazim Fatès, and Sukanta Das. (2014). Reversibility of elementary cellular automata under fully asynchronous update. In T.V. Gopal, Manindra Agrawal, Angsheng Li, and S.Barry Cooper, editors, *Proceedings of TAMC'14*, volume 8402 of *Lecture Notes in Computer Science*, pages 39–49. Springer.

[117] Anastasia Sharifulina and Vladimir Elokhin. (2011). Simulation of heterogeneous catalytic reaction by asynchronous cellular automata on multicomputer. In *Proceedings of the 11th international conference on Parallel computing technologies*, PaCT'11, pages 204–209, Berlin, Heidelberg. Springer-Verlag.

[118] Fernando Silva and Luís Correia. (2013). An experimental study of noise and asynchrony in elementary cellular automata with sampling compensation. *Natural Computing*, 12(4):573–588.

[119] Antoine Spicher, Nazim Fatès, and Olivier Simonin. (2010). Translating discrete multi-agents systems into cellular automata: Application to diffusion-limited aggregation. In Joaquim Filipe, Ana Fred, and Bernadette Sharp, editors, *Agents and Artificial Intelligence*, volume 67 of *Communications in Computer and Information Science*, pages 270–282. Springer Berlin Heidelberg.

[120] W. Richard Stark. (2013). Amorphous computing: examples, mathematics and theory. *Natural Computing*, 12(3):377–392.

[121] W. Richard Stark and William H. Hughes. (2000). Asynchronous, irregular automata nets: the path not taken. *BioSystems*, 55:107–117.

[122] Tomoaki Suzudo. (2004). Searching for pattern-forming asynchronous cellular automata - an evolutionary approach. In Peter M. A. Sloot, Bastien Chopard, and Alfons G. Hoekstra, editors, *Proceedings of ACRI'04*, volume 3305 of *Lecture Notes in Computer Science*, pages 151–160. Springer.

[123] Tomoaki Suzudo. (2004). Spatial pattern formation in asynchronous cellular automata with mass conservation. *Physica A: Statistical Mechanics and its Applications*, 343:185 – 200.

[124] Yousuke Takada, Teijiro Isokawa, Ferdinand Peper, and Nobuyuki Matsui. (2006). Construction universality in purely asynchronous cellular automata. *Journal of Computer and System Sciences*, 72(8):1368–1385.

[125] Wahiba Taouali, Thierry Viéville, Nicolas P. Rougier, and Frédéric Alexandre. (2011). No clock to rule them all. *Journal of Physiology - Paris*, 105(1–3):83–90.

[126] Tommaso Toffoli. (1978). Integration of the phase-difference relations in asynchronous sequential networks. In Giorgio Ausiello and Corrado Böhm, editors, *Proceedings of the Fifth Colloquium on Automata, Languages and Programming*, volume 62 of *Lecture Notes in Computer Science*, pages 457–463. Springer Berlin Heidelberg.

[127] Marco Tomassini and Mattias Venzi. (2002). Artificially evolved asynchronous cellular automata for the density task. In Stefania Bandini, Bastien Chopard, and Marco Tomassini, editors, *Proceedings of ACRI 2002*, volume 2493 of *Lecture Notes in Computer Science*, pages 44–55. Springer.

[128] Kohji Tomita, Satoshi Murata, and Haruhisa Kurokawa. (2007). Asynchronous graph-rewriting automata and simulation of synchronous execution. In Fernando Almeida e Costa, Luis Mateus Rocha, Ernesto Costa, Inman Harvey, and António Coutinho, editors, *Advances in Artificial Life*, volume 4648 of *Lecture Notes in Computer Science*, pages 865–875. Springer Berlin Heidelberg.

[129] Predrag T. Tošić. (2011). On modeling large-scale multi-agent systems with parallel, sequential and genuinely asynchronous cellular automata. *Acta Physica Polonica B - Proceedings Supplement*, 4(2):217–235.

[130] Alan Turing. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society (London)*, 237:5–72.

[131] Leonardo Vanneschi and Giancarlo Mauri. (2012). A study on learning robustness using asynchronous 1D cellular automata rules. *Natural Computing*, 11(2):289–302.

[132] Michael Vielhaber. (2013). Computation of functions on n bits by asynchronous clocking of cellular automata. *Natural Computing*, 12(3):307–322.

[133] John von Neumann. (1966). *Theory of self-reproducing automata*. University of Illinois press Urbana. A. Burks (editor).

[134] Simon Wacker and Thomas Worsch. (2012). Phase space invertible asynchronous cellular automata. In Enrico Formenti, editor, *Proceedings of AUTOMATA & JAC*, volume 90 of *EPTCS*, pages 236–254.

[135] Weiguo Wang. (1991). An asynchronous two-dimensional self-correcting cellular automaton. In *Proceedings of FOCS'91 (Foundations of Computer Science)*, pages 278–285. IEEE Computer Society.

[136] Thomas Worsch. (2013). Towards intrinsically universal asynchronous CA. *Natural Computing*, 12(4):539–550.