



# First steps on asynchronous lattice-gas models with an application to a swarming rule

Olivier Bouré, Nazim Fatès, Vincent Chevrier

► **To cite this version:**

Olivier Bouré, Nazim Fatès, Vincent Chevrier. First steps on asynchronous lattice-gas models with an application to a swarming rule. Georgios Ch. Sirakoulis and Stefania Bandini. 10th International Conference on Cellular Automata for Research and Industry, Sep 2012, Fira, Greece. Springer Berlin Heidelberg, 7495, pp.633-642, 2012, Lecture Notes in Computer Science. <10.1007/978-3-642-33350-7\_65>. <hal-00687987v2>

**HAL Id: hal-00687987**

**<https://hal.inria.fr/hal-00687987v2>**

Submitted on 14 Jun 2012

**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.

# First steps on asynchronous lattice-gas models with an application to a swarming rule

Olivier Bouré, Nazim Fatès, Vincent Chevrier  
Nancy Université – INRIA Nancy Grand-Est – LORIA  
{olivier.boure;nazim.fates;vincent.chevrier}@loria.fr

## Abstract

Lattice-gas cellular automata are often considered as a particular case of cellular automata in which additional constraints apply, such as conservation or spatial exclusion of particles. But what about their updating? How to deal with non-perfect synchrony? Novel definitions of asynchronism are proposed that respect the specificity of lattice-gas models. These definitions are then applied to a well-known swarming rule in order to explore the robustness of the model to perturbations of its updating.

keywords: Asynchronous cellular automata, Lattice-Gas cellular automata, Robustness, Swarming behaviour

## 1 Introduction

In the field of discrete dynamical systems, cellular automata (CA) are widely used as a tool for the simulation of natural phenomena and as a model of parallel computing. Their discrete and spatially-extended structure makes them a computationally simple model yet capable of displaying a wide range of complex behaviours. While initial studies only considered a simultaneous updating of all components, recent years have seen an increasing interest in asynchronous CA models, where the perfect synchrony hypothesis is relaxed. We are here particularly interested in *robustness*, which characterises the stability of the behaviour when external disturbances and structural changes are applied. In the case of asynchronism, robustness boils down to exploring how much of the CA behaviour is related to the synchronous scheme, and how much comes intrinsically from the individual rule [10]. Thus, we observed in previous works a variety of behaviours that arise from different updating schemes, including phase transitions [4, 8].

In particular, studies on asynchronous CA have revealed the existence of synchrony-dependent behaviours we call *synchronous singularities*, that is, behaviours which stability depends on the synchronicity of the model. Examples include the periodic patterns of the *Game of Life* [2, 7], of several Elementary Cellular Automata [8], or even the spatially-extended Prisoner's Dilemma [9].

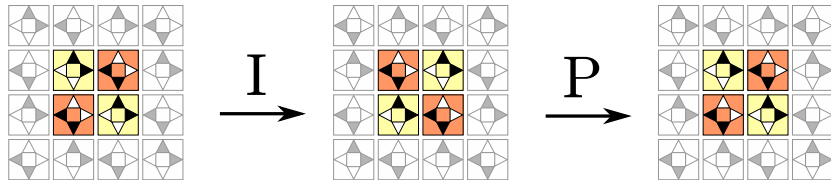


Figure 1: Synchronous singularity in a lattice-gas cellular automaton. A black triangle represents one particle present in a channel, and white an empty channel.

Considering those examples, one may wonder whether such observations can be only made in systems with an ad hoc deterministic transition rule or whether it can occur for models of natural phenomena.

Our aim here is to tackle the robustness question in the scope of lattice-gas models, in which particles travel through the lattice. As an introductory example, consider the following pattern uncovered in a previous work [3]: particles are displaced according to a synchronous updating scheme which consists of the successive application of two rules, the interaction step (I) – which reorganises the particles within a cell to maximise their alignment with the particles of the neighbours – and the propagation step (P) – which displaces particles towards the corresponding neighbours. Paradoxically, in this pattern particles are all opposing the directions of their direct neighbours, in spite of the alignment-favouring local rule (see Fig. 1 and more details in Sec. 3).

A step-by-step look at the evolution of this pattern may let us think that this behaviour requires a deterministic updating rule. However, what is outstanding is that it has been observed to self-organise (a) with a stochastic local rule, (b) from a random initial configuration. Thus the question: if such a pattern can form outside of a deterministic model, how is it related to the perfect synchrony of the system? And how can we question this property in lattice-gas models in order to probe the robustness to asynchronism?

In Sec. 2, we confront the notion of asynchronism to the constraints and hypotheses of lattice-gas cellular automata and propose several definitions of an asynchronous updating scheme. In Sec. 3, we precise further the individual rules considered for our model and compare the observed patterns in the synchronous and asynchronous versions. Finally in Sec. 4, we briefly discuss on the role of asynchronism in discrete dynamical systems, based on our definitions and experimental results.

## 2 Defining asynchronous lattice-gas cellular automata

There exist a wide range of interpretations and definitions given to asynchronism, and even more diverse resulting behaviours (see for example [1, 4] and ref. therein). The problem we are tackling here is that the system we want to

“make asynchronous” is a lattice-gas cellular automaton which, in spite of its compatibility with the classical CA definition, includes additional hypotheses that may not be conserved when changing the updating scheme. Let us now present how we tackle this problem formally.

## 2.1 Definition of lattice-gas cellular automata

The lattice-gas cellular automaton (LGCA) we are interested in is a particular CA, where:

- the *cellular space*  $\mathcal{L} = \{\mathbb{Z}/L\cdot\mathbb{Z}\}^2$  is a 2-dimensional square lattice of length  $L$  with periodic boundary conditions.
- the *neighbourhood*  $(n_i)_{i \in \llbracket 1, 4 \rrbracket} = \{(1, 0); (0, 1); (-1, 0); (0, -1)\}$  associates to a cell  $c \in \mathcal{L}$  the set of its 4 nearest neighbours.

In a LGCA, each cell is connected to its neighbours via *channels* through which particles can travel. Cell configurations  $\vec{x}_c = (x_i(c))_{i \in \llbracket 1, 4 \rrbracket}$  are therefore represented as a vector of the numbers of particles contained by each of its channels  $x_i(c) \in \mathbb{N}$ . The dynamics of LGCA is determined by the successive application of two global transitions:

1. The *interaction step* reorganises particles within cells according to a transition function  $f_I$ , which applies to the configuration of a cell and its neighbourhood:

$$\forall c \in \mathcal{L}, \vec{x}_c^I = f_I(\vec{x}_c, \vec{x}_{c+n_1}, \dots, \vec{x}_{c+n_4}). \quad (1)$$

2. The *propagation step* displaces particles from the channels of a cell to the corresponding neighbours:

$$\forall c \in \mathcal{L}, \forall i \in \llbracket 1, 4 \rrbracket, x_i^P(c) = x_i^I(c - n_i). \quad (2)$$

In addition, we consider two fundamental constraints which we want to keep valid under an asynchronous updating scheme: the *particle exclusion* – channels may contain at most one particle, and are therefore considered as either empty (state 0) or full (1) – and the *particle conservation* – particles must always be conserved when they interact and propagate<sup>1</sup>.

## 2.2 Which asynchronism?

For this work, we will consider  $\alpha$ -*synchronism* [8, 4] which follows the general definition: at any time step, each component updates according to the regular transition function with probability  $\alpha$ , or remains unchanged using the identity function with probability  $1 - \alpha$ . This updating scheme provides us with a useful

<sup>1</sup>Note that our particle-oriented interpretation of the system resembles *Totally Asymmetric Simple Exclusion Processes* [?].

control parameter, the *synchrony rate*  $\alpha$ , allowing for a continuous control of the updating scheme from evolution in perfect synchrony ( $\alpha = 1$ ) to quasi-sequential updates ( $\alpha \rightarrow 0$ ).

The global transition function of LGCA from a time  $t$  to  $t + 1$  is constituted of two steps, interaction and propagation, applied successively to all cells. When we want to apply asynchronism, we need to properly define the meaning of “updating” and “remaining unchanged” in terms of computation. Does asynchronism apply to the sole interaction? The propagation? Shall we consider these transitions as correlated, independent?

To address this issue, we propose as a starting point three possible implementations of asynchronism:

- (1) **Asynchronous interaction** : the interaction is applied with a probability  $\alpha_I$ , called the *interaction rate*. The propagation is always applied.
- (2) **Asynchronous propagation** : the interaction is always applied, but the propagation is applied with a probability  $\alpha_P$ , called the *propagation rate*.
- (3) **Correlated interaction and propagation** : for each cell, interaction and propagation are both applied with a probability  $\alpha_C$ , otherwise none of them is applied.

However, if the asynchronous interaction (1) can be implemented without problem as the interaction rule itself is particle-conserving, desynchronising the propagation (2, 3) is not a straightforward operation and requires further reflexion.

### 2.2.1 Conserving particles during the propagation.

Let us first consider the case where asynchronous propagation is simply determined as the application of the transition rule with probability  $\alpha_P$ , and identity function with probability  $1 - \alpha_P$ . In LGCA models, there is no actual means for a given cell to know whether its neighbours are selected for update or not. This is problematic as we need this type of information to decide whether to propagate the state of a channel or not (see Fig. 2-b and -c).

There exist several model adaptations that can solve this issue: for instance, by adding specific states which keep cells synchronized [11] or by using additional steps to perform validity checks (*e.g.* transactional CA [12]). Similarly, our proposition here consists in modifying the point of view of the asynchronous propagation function by considering no longer the cells but the channels as the base components of the system. We first select randomly the cells which are to be updated, and then determine from non-selected cells which channels are *free*, and thus updated, and which are *blocked* and remaining unchanged. We now specify the formal definitions of asynchronous interaction and propagation using selection functions.

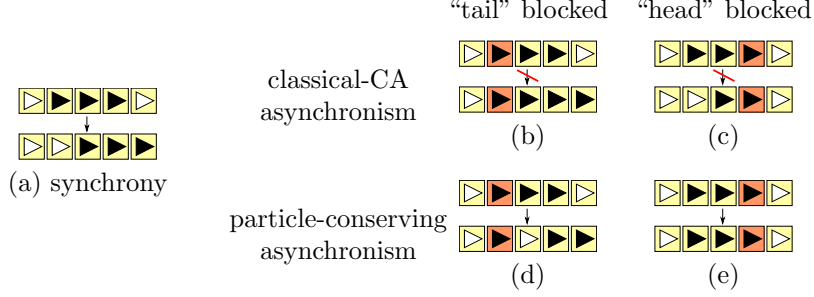


Figure 2: Representation of the propagation step along one channel direction in a sample of five cells, for different updating schemes and situations of particles. Darker cells are not selected for update. (a) is the synchronous classical case, (b,c) the asynchronous case without the particle-conserving changes and (d,e) our proposition for an asynchronous propagation.

## 2.3 Definition of an asynchronous LGCA

### 2.3.1 Interaction step.

We introduce a *selection function*  $\Delta_I : \mathbb{N} \rightarrow \mathcal{P}(\mathcal{L})$  which associates to each time  $t$  the subset of cells to be updated during the interaction step, where each cell has a probability  $\alpha_I$  to be selected. The interaction transition function becomes  $\forall t \in \mathbb{N}, \forall c \in \mathcal{L}, \mathcal{N} = \{n_1, \dots, n_4\}$ :

$$\vec{x}_c^t = f_{\Delta_I}(\vec{x}_c, \vec{x}_{c+n_1}, \dots, \vec{x}_{c+n_4}) = \begin{cases} f_1(\dots) & \text{if } c \in \Delta_I(t) \\ \vec{x}_c & \text{otherwise.} \end{cases} \quad (3)$$

When  $\alpha_I = 0$ , the system is ballistic, *i.e.* particles always go straight.

### 2.3.2 Propagation step.

First, we need to determine which cells update, and which remain unchanged. Similarly to asynchronous interaction, we introduce a *selection function*  $\Delta_P : \mathbb{N} \rightarrow \mathcal{P}(\mathcal{L})$  which returns for time  $t$  the subset of cells to be updated during the propagation step, where each cell has a probability  $\alpha_P$  to be selected. In addition, we define  $\mathcal{B}^t \subset \llbracket 1, 4 \rrbracket \times \mathcal{L}$  as the set of *blocked channels*, which will remain unchanged between times  $t$  and  $t + 1$ . To build this set, we state that a channel  $(i, c)^t$  (channel  $i$  of cell  $c$  at time  $t$ ) is *blocked* if it contains a particle ( $x_i^t(c) > 0$ ), and if one of the two conditions is true: (C1) its containing cell  $c$  is not selected for update, or (C2) its destination channel is blocked. We thus have:

$$(i, c)^t \in \mathcal{B}^t \text{ if } (x_i^t(c) > 0) \text{ and } \begin{cases} c \notin \Delta_P(t) & (C1) \\ \text{or} \\ (i, c + n_i)^t \in \mathcal{B}^t & (C2) \end{cases}$$

The construction of  $\mathcal{B}^t$  describes a general relation between particles. It can be implemented with a recursive algorithm: once a channel is blocked because of condition (C1), the chain of the channels that “point” to this one must be also blocked if they contain a particle because of condition (C2) (see Fig. 2-e). The propagation transition function therefore becomes:

$$x_i^P(c) = \begin{cases} x_i^I(c) & \text{if } (i, c) \in \mathcal{B}^t \\ 0 & \text{if } (i, c) \notin \mathcal{B}^t \text{ and } (i, c - n_i) \in \mathcal{B}^t \\ x_i^I(c - n_i) & \text{otherwise.} \end{cases} \quad (4)$$

When  $\alpha_P = 0$ , the particles are never propagated and thus never leave their cell.

### 3 Observation on a swarming rule

The model of swarming we present here is borrowed from a compilation of biological models by Deutsch *et al.* [6, ch. 8]. It describes a probabilistic swarming interaction rule in which the particles of a cell tend to align themselves with the neighbourhood predominant directions. The probability for a given cell configuration  $\vec{x}_c$  to reorganise into  $\vec{x}_c^I$  is given by:

$$P(\vec{x}_c \rightarrow \vec{x}_c^I) = \frac{1}{Z} \exp \left[ \sigma \cdot \vec{J}_c(\vec{x}^I) \cdot \vec{D}_c(\vec{x}) \right] \cdot \delta(\vec{x}_c, \vec{x}_c^I) \quad (5)$$

where:

- $Z$  is the normalisation factor, so that  $\sum_{\vec{x}_c^I} P(\vec{x}_c \rightarrow \vec{x}_c^I) = 1$ .
- $\sigma \in \mathbb{Z}$  is the alignment sensitivity.
- $\vec{J}_c(\vec{x}^I)$  is the *flux* in cell  $c$ , the resulting vector of particle directions.
- $\vec{D}_c(\vec{x})$  is the *director field*, the sum of cell  $\vec{x}_c$  neighbours’ flux.
- $\delta(\vec{x}_c, \vec{x}_c^I) = 1$  if cell configurations  $\vec{x}_c$  and  $\vec{x}_c^I$  have same number of particles, 0 otherwise. This ensures conservation of particles.

Starting from a random initial configuration of density  $\rho$ , where  $\rho$  is the probability for each channel to contain a particle, we need only to set the parameter  $\sigma$  to determine the local interaction rule. This limits the parametric space to the  $\sigma/\rho$  plane, allowing for a complete exploration of the different qualitative behaviours that the model may display.

#### 3.1 Recapitulation of the synchronous behaviour

The main interest of the swarming model is that, in spite of a simple and stochastic interaction rule, a variety of qualitatively different behaviours can arise. In particular, it has been previously established that, using the two

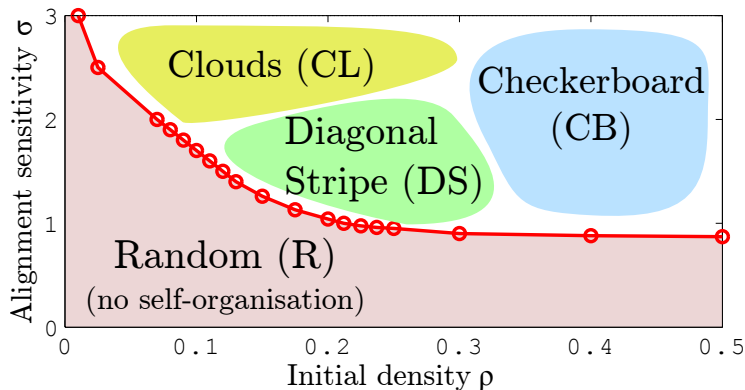


Figure 3: Spatial distribution of the different patterns in the sensitivity-density parametric plane.

control parameters  $\sigma$  and  $\rho$ , two phases may emerge, chaotic and organised [6]. In recent work, we reported that the organised phase can display several stable patterns of global particle organisation [3] (see Fig. 3):

**Checkerboard pattern (CB)** : local groups of anti-aligned particles. Typically occurs for high densities (typically  $\rho \approx 0.5$ ).

**Diagonal stripe pattern (DS)** : a traveling compact diagonal shape of particles in two orthogonal directions, that loops over the lattice boundaries.

**Clouds pattern (CL)** : traveling competitive groups of aligned particles. Typically occurs for a high sensitivity ( $\sigma \approx 3 - 4$ ).

It is interesting to note that these behaviours have been observed for systems of limited size and time, and can therefore be considered as metastable attraction basins. This means that once a behaviour is reached, the system will conserve it until random fluctuations make the system “escape” this behaviour. We now apply the different updating schemes to the system and try to determine whether these patterns are robust to perturbations of the updating. More particularly we are interested in whether their appearance depend on perfect synchrony.

### 3.2 Exploration of the asynchronous behaviour

As a preliminary study, we now propose to consider each of the three patterns independently, by setting the system in adequate initial condition of size ( $L = 20$ ) and iterating the system for a large number of steps ( $t = 5000$ ). We then observe if the expected pattern appears, and compare the results for different values of the different synchrony rates ( $\alpha_I, \alpha_P, \alpha_C$ ).

#### 3.2.1 Asynchronous interaction.

Two observations can be made (see Fig. 4-a):



1. The checkerboard pattern is very sensitive to asynchronism. Indeed, a small but apparently not infinitesimal amount of asynchronism ( $\alpha_I \approx 0.95$ ) is sufficient to switch the behaviour to a diagonal stripe pattern which covers the lattice entirely.
2. For the diagonal stripe and the clouds patterns, as long as the amount of asynchronism remains small, the only effect on the pattern is to introduce additional noise. As soon as a critical value  $\alpha_I$  is reached, the clouds pattern becomes a diagonal stripe, and the diagonal strips becomes random.

### 3.2.2 Asynchronous propagation.

The three tested patterns displayed a sudden change of behaviour for a small amount of asynchronism (see Fig. 4-b).

1. The checkerboard pattern is very unstable and shifts to a global consensus on the directions of propagation. However, for higher the value of  $\alpha_P$ , it is possible for the pattern to “survive” for a few time steps.
2. The diagonal stripe still appears for high  $\alpha_P$  but is soon replaced by “semi-random” patterns, where some order regularly appears in some parts of the lattice and disappears, but never stabilises.
3. The clouds pattern is destabilised and transforms into one or several vertical or horizontal stripes of aligned particles.

These observations suggest a destructive effect of asynchronous propagation on all behaviours, even for a high propagation rate  $\alpha_P \approx 0.95$ . However, this observation does not seem to hold for infinitesimal values of asynchrony ( $\alpha_P = 1^-$ ), which let us think that our patterns have a minimal degree of robustness to asynchronous propagation.

### 3.2.3 Correlated interaction and propagation.

Similar experiments have been applied to the correlated updating scheme. The resulting behaviour is a mix of asynchronous interaction and asynchronous propagation:

1. For higher values of the synchronism rate ( $\alpha_C \approx 0.95$ ), the behaviour follows the sudden changes observed in the asynchronous propagation.
2. For lower values ( $\alpha_C \approx 0.5$ ), the swarming and identity interaction rules become equiprobable and, as particles do not gather in clusters any more, the system enters a random phase.

Considering how differently asynchronous interaction and asynchronous propagation effect on the system dynamics, and the synchrony rates for those changes were observed, this result is somehow “expected”: the behaviour changes correspond more or less to the changes observed for each type of asynchronism.

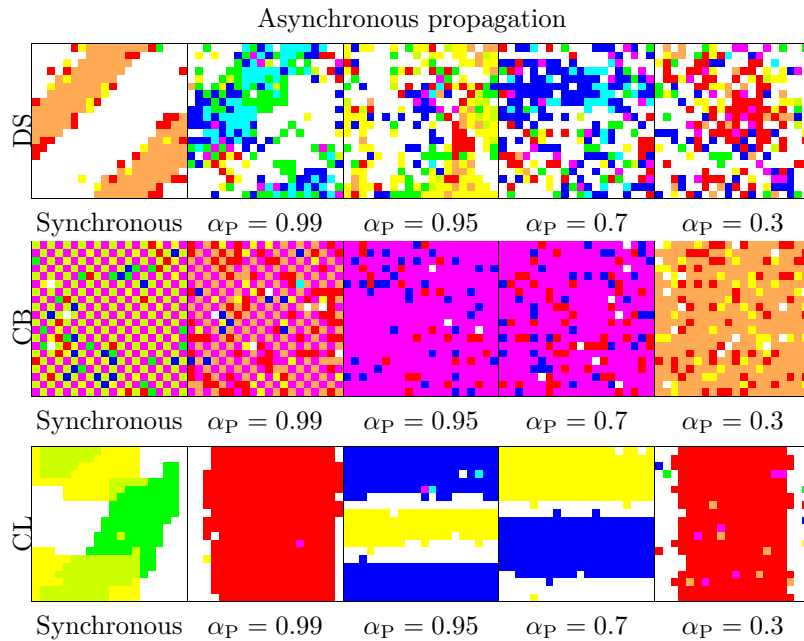
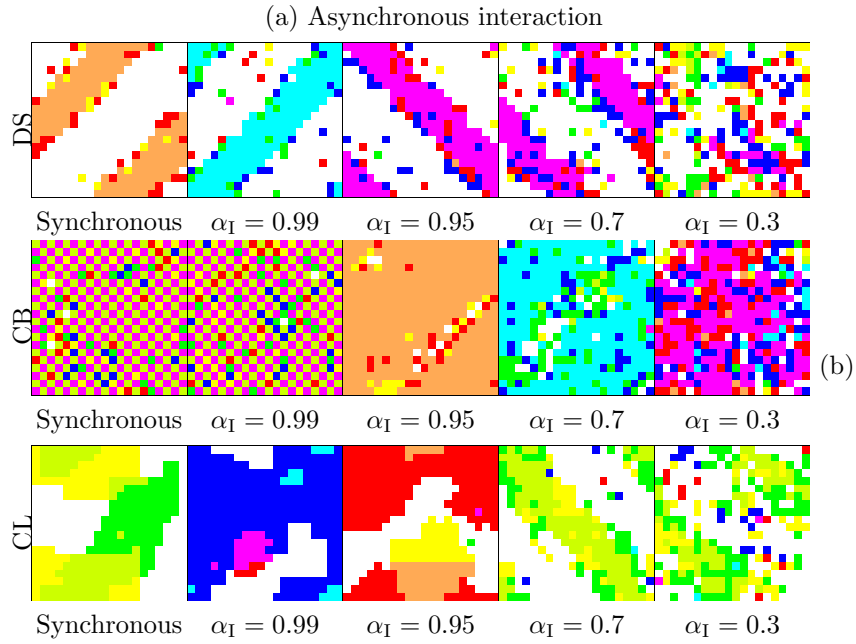


Figure 4: Evolution of the main patterns for different interaction (propagation) rate values  $\alpha_I$  ( $\alpha_P$ , respectively). Configurations are obtained from random initial configurations run for 5000 steps for typical values of  $\sigma$  and  $\rho$  (DS:  $\{\sigma = 1.5; \rho = 0.2\}$ , CB:  $\{\sigma = 1.5; \rho = 0.45\}$ , CL:  $\{\sigma = 4; \rho = 0.2\}$ ). The simulations and visualisations are realised with *FiatLux*, a CA simulator in Java (<http://fiatlux.loria.fr>).

### 3.2.4 Synthesis of the experiments.

Observations on the swarming model, which was thought rather robust in the first place due to the stochastic rule and self-organisation phenomena, highlighted the dependence between the type of asynchronism applied to the system and the stability of each pattern. For instance, we observed that there exist (1) differences between patterns in their robustness to asynchronism (*e.g.* diagonal stripe versus checkerboard under asynchronous interaction) and (2) differences between types of asynchronism in their effects on the system’s behaviour (*e.g.* diagonal stripe under asynchronous interaction and propagation).

Moreover, we ask whether there exists a fundamental change of behaviour between perfect synchrony ( $\alpha = 1$ ) and quasi-synchrony ( $\alpha = 1^-$ ). Indeed, as our system is stochastic, it is difficult to determine experimentally whether patterns such as the checkerboard can be considered as “synchronous singularities”. For now, additional experiments must be conducted in deeper details, as our simulations were limited in space and time, in order to give a proper estimation of the robustness of the behaviour of this model.

## 4 Discussion

This paper presented a first definition of an asynchronous LGCA, which intends to extract from classical CA the idea of questioning the perfect synchrony, and adapt it to the definition and hypotheses of a lattice-gas model of swarming.

Although cellular automata and lattice-gas models share strong resemblances, the introduction of asynchronism managed to reveal an intrinsic difference. Indeed, cellular automata are based on cells that are updated according to the observation of their neighbours’ state, whereas lattice-gas models intend to capture a transport of information between cells.

From this example we learned that a naive expression of asynchronism – for instance reproducing the classical definition from cellular automata – could compromise the constraints of the lattice-gas model: the conservation and spatial exclusion of particles imposed us to change the modelling point of view and consider *channels* as the base components of the updating scheme. However, many issues remain unexplored. For instance, according to our definitions, a single particle can block an entire array of particles, which may contradict the idea of locality of events in spatially-distributed computing.

The next step of this approach consists in building a model where particles are considered as autonomous agents, with their own perception and decision process.

## References

- [1] Stefania Bandini, Andrea Bonomi, and Giuseppe Vizzari, *An analysis of different types and effects of asynchronicity in cellular automata update schemes*, Natural Computing (2012), 1–11.

- [2] Hendrik J. Blok and Birger Bergersen, *Synchronous versus asynchronous updating in the "Game of Life"*, Physical Review E **59** (1999), no. 4, 3876–3879.
- [3] Olivier Bouré, Nazim Fatès, and Vincent Chevrier, *Observation of novel patterns in a stressed lattice-gas model of swarming*, Tech. report, Nancy Université – INRIA Nancy–Grand-Est – LORIA, 2011.
- [4] Olivier Bouré, Nazim Fatès, and Vincent Chevrier, *Robustness of Cellular Automata in the Light of Asynchronous Information Transmission*, Unconventional Computation, Lecture Notes in Computer Science, vol. 6714, 2011, Extended version: <http://hal.inria.fr/hal-00658754/>, pp. 52–63.
- [5] David Cornforth, David G. Green, and David Newth, *Ordered asynchronous processes in multi-agent systems*, Physica D: Nonlinear Phenomena **204** (2005), no. 1-2, 70–82.
- [6] Andreas Deutsch and Sabine Dormann, *Cellular Automaton Modeling of Biological Pattern Formation*, Birkhauser Boston, 2005.
- [7] Nazim Fatès, *Does Life resist asynchrony?*, Game of Life Cellular Automata, Springer, 2010, pp. 257–274.
- [8] Nazim Fatès and Michel Morvan, *An experimental study of robustness to asynchronism for elementary cellular automata*, Complex Systems **16** (2005), 1–27.
- [9] Carlos Grilo and Luís Correia, *Effects of asynchronism on evolutionary games*, Journal of Theoretical Biology **269** (2011), no. 1, 109–122.
- [10] Thomas E. Ingerson and Raymond L. Buvel, *Structure in asynchronous cellular automata*, Physica D: Nonlinear Phenomena **10** (1984), no. 1-2, 59 – 68.
- [11] Ferdinand Peper, Teijiro Isokawa, Yousuke Takada, and Nobuyuki Matsui, *Self-timed cellular automata and their computational ability*, Future Generation Computer Systems **18** (2002), no. 7, 893 – 904.
- [12] Antoine Spicher, Nazim Fatès, and Olivier Simonin, *From reactive multi-agents models to cellular automata - illustration on a diffusion-limited aggregation model*, ICAART, 2009, pp. 422–429.