A note on the Density Classification Problem in Two Dimensions

Nazim Fatès

A note on the Density Classification Problem in Two Dimensions

Nazim Fatès
Inria, Villers-lès-Nancy, F-54 600, France
nazim.fates@loria.fr

The density classification problem is explored experimentally in the case of two-dimensional grids. We compare the performance of deterministic and stochastic CA, as well as interacting particle systems. The question of how to design a rule that would attain an arbitrary precision is examined and we show that it seems more difficult to solve than in the one-dimensional case.

1 Introduction

The *density classification problem* is a simple inverse problem which consists in deciding in a distributed way if an initial configuration contains more 0s or more 1s, that is, if the initial density of 1s is smaller than or greater than 1/2. The difficulty of this problem – also called the majority problem – comes from the need to reach a consensus : all the components of the system should agree on a state 0 or 1, depending on whether the initial configuration contained more 0s or more 1s, respectively.

This problem was first formulated by Packard in 1988 [9], after he considered the problem of self-correcting noisy cellular automata, as studied by Gács *et al.* [7]. Since then, the problem has attracted a large amount of research efforts in order to search for the best solutions. After noticing that finding good rules was a difficult task, Land and Belew issued a key result: no perfect density classifier exists [8]. This nevertheless did not stop the quest for good rules as it was not known how well a rule could perform.

We recently showed that for one-dimensional systems, if we allow stochastic transitions, it is then possible to solve the density classification problem with an arbitrary precision [5]. This means that for a given ring size, a probabilistic cellular automaton can be designed such that the probability that it misclassifies a randomly drawn configuration can be made arbitrarily small. However, our construction does not guarantee that the same level of precision will be kept for all ring sizes. It should also be noted that an increase of the quality of classification comes at the expense of an increase in the convergence time. It is to date an open problem to find the stochastic rules which present the best compromise between the classification quality and the classification time.

On the positive side, it was shown that the result is not merely a "theoretical" construction but it also leads to the effective construction of rules that outperform the best rules known so far. However, the question as to whether such constructions can be found for different topologies was raised. The purpose of this note is to examine if a similar result can be obtained for two-dimensional cellular systems.

It may be thought at first sight that it is possible to generalise the solution to the two-dimensional case by "slicing" a toroidal grid into rings. Unfortunately, this solution does not work as different "slices" may give different "answers" while the density classification problem requires that the system reaches a *global* consensus. It is thus necessary to establish some kind of "communication" between the two dimensions of the cellular automaton in order to obtain the property of a classification with an arbitrary precision. Our intuition was that this operation was rather easy but our experiments revealed that the situation was more complicated than it first seemed. Note that it is well-known that deciding properties of cellular

Submitted to:
Automata 2012

automata is generally more difficult in two dimensions than in one-dimensions (see e.g. [12, 3, 6]), but the situation here is different since we want to solve an inverse problem, that is, to exhibit *one* particular rule that would solve the problem (totally or partially).

This note exposes the experimental results which were obtained in a first attempt to answer this question. In Sec. 2, we define formally the problem and then study three cellular rules in Sec. 3. We introduce an IPS rule in Sec. 4 and show that this rule exhibits a good performance. Finally we briefly discuss the results and give some further motivations to study this problem in two dimensions.

2 Definitions

A *cellular automaton* (CA) is constituted of a collection of cells placed on a discrete space $\mathscr{L}$ that is usually a subset of $\mathbb{Z}^d$, where d is called the *dimension* of the lattice. We here focus on *finite* two-dimensional square lattices, that is, we take $\mathscr{L} = (\mathbb{Z}/L.\mathbb{Z})^2$, where L is the *lattice size*.

A *configuration* represents the state of the automaton; each cell can have a binary state, the configuration space is $E = \{0,1\}^{\mathscr{L}}$. For the sake of simplicity, we consider that configurations are functions from $\mathscr{L}$ to $\{0,1\}$ and denote by $x^t(c)$ the state of a cell $c \in \mathscr{L}$ at time t. Each cell is linked to a set of k cells, defined with local translations obtained with k vectors that form a set $\mathscr{N} = (n_1, \ldots, n_k)$, called *the neighbourhood*.

The local rule of the cellular automaton f is a function from $\{0,1\}^k$ to $\{0,1\}$ that defines how cells update their state according to the states of their neighbours. When the cells are updated synchronously, the evolution of the systems maps a configuration x^t to a configuration x^{t+1} such that for all $c \in \mathscr{L}$:

$$x^{t+1}(c) = f(x^t(c+n_1), \ldots, x^t(c+n_k)).$$

The density classification problem consists in designing a cellular system that would converge to the uniform configurations $\mathbf{0} = 0^{\mathscr{L}}$ or $\mathbf{1} = 1^{\mathscr{L}}$, depending on whether the density of ones in the initial condition is lower or greater than 1/2, respectively. Formally, for a given initial configuration $x \in E$, the probability of good classification $\mathsf{PGC}(x)$ is the probability that there exists a time T such that $x^T = \mathbf{0}$ and $d(x) < 1/2$ or such that $x^T = \mathbf{1}$ and $d(x) > 1/2$ where $d(x) = |\{c, x(c) = 1\}|/n$.

We assume that the initial configurations obey the binomial distribution, that is, they are drawn uniformly in the configuration space. For a given classifier $\mathscr{C}$ and a given lattice size L, the *quality* Q is the average success rate. It is calculated as: $Q = \sum_{x \in E} \mathsf{PGC}(x) \,/\, 2^n$. Note that the presence of fixed points other than $\mathbf{0}$ or $\mathbf{1}$ represents a potential loss in quality: if, starting from an initial condition x, the system enters in one such fixed point, the configuration is considered "non-classified" and $\mathsf{PGC}(x)$ will decrease accordingly.

Before examining various rules, we introduce the majority function $\mathtt{maj}$, which associates to a t-uple in $\{0,1\}^{2m+1}$ the value $\mathtt{maj}(q_1, \ldots, q_{2m+1}) = 0$ if $\sum_{i=1}^{2m+1} q_i < m$ and $\mathtt{maj}(q_1, \ldots, q_{2m+1}) = 1$ otherwise.

3 Three cellular automata rules

3.1 Toom's rule

Toom's rule consists in applying the majority rule on the neighbourhood constituted of the cell itself and in North and East neighbours [4]. Formally, the local rule is defined as the application of $\mathtt{maj}$ on the neighbourhood $\mathscr{N}_3 = \{(0,0), (0,1), (1,0)\}$.

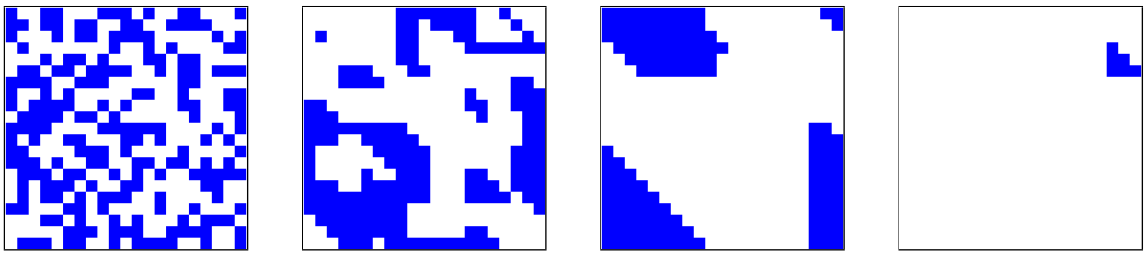

Figure 1: Example of an evolution of Toom's rule with $L = 21$. Blue/dark and white cells represent cells in state `0` and `1`, respectively.

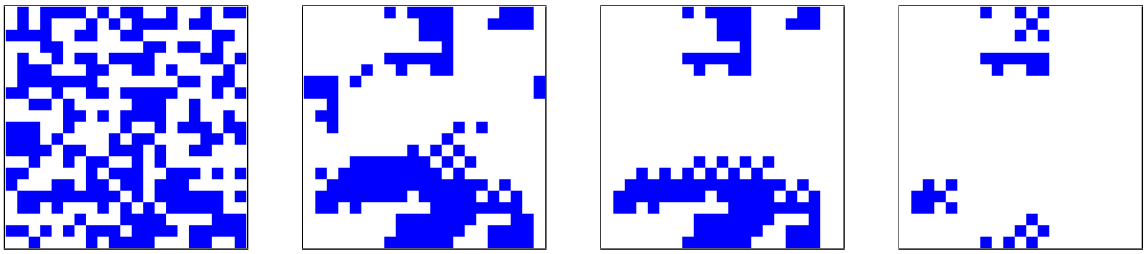

Figure 2: Example of an evolution of Reynaga's rule with $L = 21$.

Marcovici *et al.* showed that this rule is a perfect density classification on infinite lattice-size lattices [2] (the system is initialised with a Bernoulli law of parameter d with $d \neq 1/2$). The proof relies on arguments from percolation theory and on the "eroder" property of the model, that is, the fact that a set of cells in one state surrounded by a set of states in another state will fatally disappear (as if it were an island that was progressively eroded by the see).

Figure 1 shows the evolution of the rule with an initial random condition of density $d \sim 0.47$. This configuration is well-classified in 33 steps, which is rather rapid as compared to the classification times observed for one-dimensional systems.

Unfortunately, the rule gives poor results as a classifier, even for small lattices. For instance, for $L = 9$, using 10^5 random samples, we measured a quality of only $Q = 0.536$, which shows that rules offers a very little gain over a pure random guess! In fact only 0.7% of the configurations are misclassified: the bad quality of the classification is mainly due to the existence of non-converging configurations. For larger lattices, the result only worsens and we fall below $1/2$ for $L = 21$.

The reason why this rule does not obtain good results is the presence of numerous of non-classified configurations. For instance, the configuration with contiguous horizontal lines of `1`-cells is a fixed point. The configuration where only the cells of one diagonal are in state `1` will evolve into a cycle of length L, obtained by a South-West translation of the `1`-cells.

3.2 Reynaga's rule

The next rule we examine was introduced by Reynaga in the context of pattern processing [10]. It was designed as a means of generalising the famous Gács-Kurdiumov-Levin (GKL) rule [7]. The rule is initially defined on von Neumann neighbourhood $\mathcal{N}_5 = \{(0,0),(0,1),(1,0),(0,-1),(-1,0)\}$.

Informally, it consists in dividing the neighbourhood into two parts and in applying the majority rule on one or the other part depending on whether the central cell is in state 0 or 1 (see Fig. 2).

Using the letters C,N,E,S,W to symbolise the respective states of the central cell, of the North, East, South and West cells of the von Neumann neighbourhood, the local function that defines Reynaga's rule reads:

$$\texttt{Rey}(C,N,E,S,W) = \begin{cases} \texttt{maj}(C,N,E) & \text{if } C = 1 \\ \texttt{maj}(C,S,W) & \text{otherwise.} \end{cases}$$

A statistical sampling made out of 10^5 random initial conditions gives for $L = 21$, $Q = 0.61$. This is better than the previous rule, but remains quite low compared to other known rules in one or two dimensions. For this rule too, the proportion of non-converging configurations remains quite high, with a rate of approximately one fourth. The number of fixed points is indeed numerous: a configuration made of horizontal or vertical lines of 1-cells is a fixed point. It also very easy to "cheat" this automaton: a simple "cross" of 1-cells makes the system always converge to the **1** configuration while the initial density of the pattern approaches zero when the lattice size increases.

Surprisingly enough, the extension of this rule to the Moore neighbourhood only worsens the result. Indeed, we found that the quality drops to $Q = 0.55$ and the proportion of non-converging configurations increases to one third.

3.3 A two-dimensional traffic-majority rule

Our goal now is to generalise the one-dimensional construction proposed in order to solve the density classification problem with an arbitrary precision [5]. This solution was constructed as a stochastic "blend" of the traffic rule (ECA 184), which is number-conservative, and the majority rule. If we want to obtain a similar result in two dimensions, we thus need to blend the majority rule with a number-conservative rule that would, like the traffic rule, separate similar cells as much as possible, as it is the case in one dimension. For instance, in the case of a grid with an even size and an equal initial number of 0s and 1s, we would like to obtain a "checkerboard pattern" where no two 0s or 1s are adjacent. Unfortunately, we are not aware of the existence of any such rule.

We now introduce a first attempt to generalise the traffic rule. Before we express it in the CA formalism, it may be easier to describe it in terms of particle movements: To take the metaphor of traffic, the local effect of the rule consists in moving cars (state 1) with a priority to East, and, with the direction North as a second choice. Conversely, the empty spaces (state 0) will attempt to move South if possible and to move West otherwise.

Although the formulation of the rule locally uses only the four adjacent neighbours, its formulation in terms of CA necessitates the use of the Moore neighbourhood: $\mathcal{N}_9 = \{(c_x, c_y), |c_x - c'_x| \leq 1\}$. (We invite readers to verify this statement). To define this local rule, denoted by `Traffic2D`, we take $C,N,NE,\ldots$ as the letters that represent the state of the central cell, the North, North-East cells, etc. ; we then define `Traffic2D` depending on the state of the central cell C. For $C = 1$ we have:

$$\texttt{Traffic2D}(1,N,NE,\ldots,NW) = \begin{cases} 0 & \text{if } E = 0 \text{ or } (N = 0 \text{ and } NE = 0) \\ 1 & \text{otherwise,} \end{cases}$$

and for $C = 0$, the law gives a symmetric behaviour when we permute $0/1$ and the two spatial directions:

$$\texttt{Traffic2D}(0,N,NE,\ldots,NW) = \begin{cases} 1 & \text{if } W = 1 \text{ or } (S = 1 \text{ and } SW = 1) \\ 0 & \text{otherwise.} \end{cases}$$

Figure 3 shows an example of evolution of this rule from a random initial configuration. We observe that the configuration quickly stabilises into a configuration where all 1-cells translate to the right and

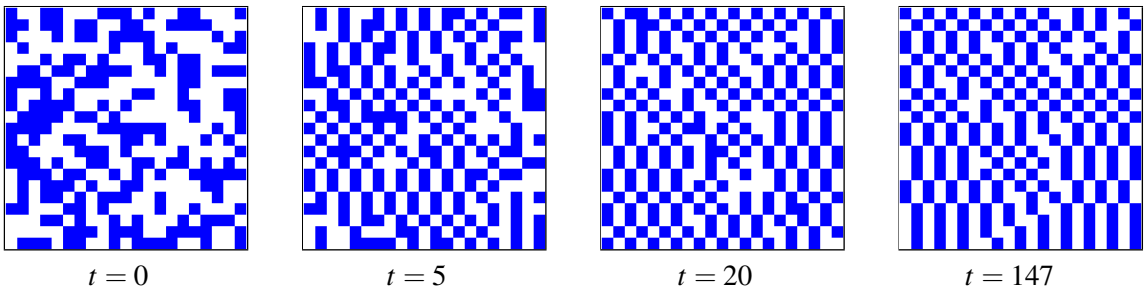

Figure 3: Example of an evolution of the `Traffic2D` rule with $L = 21$.

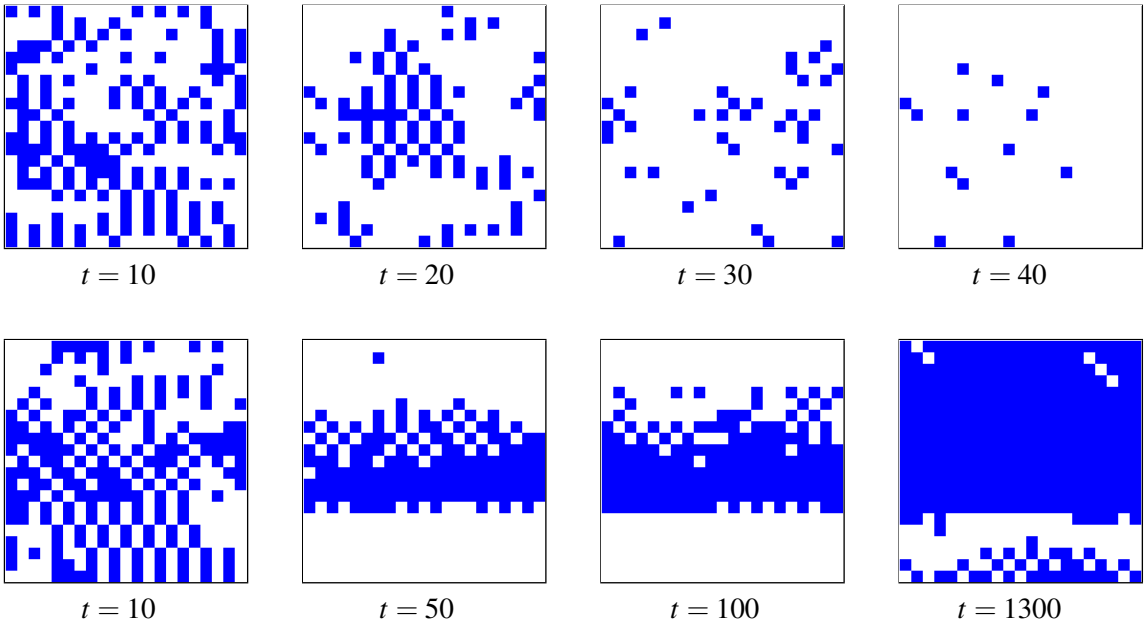

Figure 4: Example of an evolution of the `TraMaj2D` rule with $L = 21$: (top) evolution to a good classification, (bottom) evolution towards a bad classification (rare event); in both cases, the initial condition is identical to the one of Fig. 3.

where no two `1`-cells are found next to another in the horizontal direction.

The `TraMaj2D` rule is then defined as a stochastic rule which consists in applying the majority rule with probability ε and the `Traffic2D` rule with probability $1-\varepsilon$. We analysed the behaviour of this rule for 10^3 random samples and a maximum simulation time of 5.10^4. The results for $L = 21$ and $\varepsilon = 0.1$ give a quality of $Q = 0.70$. For $\varepsilon = 0.01$, the quality improves to 0.81 but the average classification time increases dramatically. It was not possible to improve the quality further as a decrease in ε only slightly improves the results while the average time of classification increases.

Though much better than the two previous rules, the overall results are rather disappointing when compared to the one-dimensional case. It seems that we are facing a paradox: although one could think that a two-dimensional model would obtain better results, the additional dimensions seems more a burden than an aid. In our view, this strange phenomenon is due to the absence of a "good" number-conserving rule in 2D. To design such a rule, we would need a strong connection between the two directions of propagation. If we give a priority of one direction over the other, as in the `Traffic2D` rule, we can not

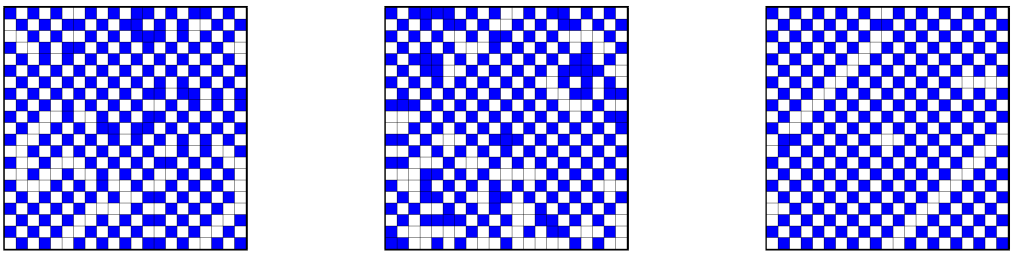

Figure 5: Three configurations reached with different settings of the checkerboard rule with $L = 21$: (left) $\beta = 0$; (middle) $\beta = 1$; (right) $\beta = 0.1$. The three systems were intialized with the same initial configuration of density ~ 0.47.

hope a perfect spreading of the particles as in the one dimensional case. On the other hand, as we are in a *synchronous* update, it is difficult to find a solution that would both conserve the number of particles and at the same time spread these particles correctly in the lattice. There may be of course such rules but we are currently unaware of any, and we leave the question of their existence as an open question.

4 An Interacting Particle System

In order to cope with the difficulties provided by the synchronous CA models that we studied previously, we now turn our attention to interacting particle systems, in which cells are updated *asynchronously* by pairs. This updating by pairs provides a decisive advantage over classical CA: it is easy to model an exchange of positions of two particles, or in an equivalent way, a movement of particles (that is, an exchange with an empty position).

We now propose to use a rule that was designed heuristically, that is, by a succession of trials and observations. It consists in using two rules in a stochastic fashion: (a) the "checkerboard rule" drives the system into a pattern that resembles a checkerboard, (b) the majority rule makes the system drift towards one of the two homogeneous fixed point. Rule (b) is applied with probability ε and rule (a) with probability $1-\varepsilon$.

Let us now define this IPS rule formally. An *interacting particle system* (IPS) can be seen as a fully asynchronous cellular automaton where cells are updated by pairs. Let $\mathscr{I} \subset \mathscr{L} \times \mathscr{L}$ be the *interaction graph* of the IPS, which represents all the cell pairs that can be updated. In this work, we use interactions with the 8-nearest cells, which we denote by: $\mathscr{I}_8 = \{(c,c') \in \mathscr{L}^2, |c_x - c'_x| = 1 \text{ or } |c_y - c'_y| = 1\}$, with the convention $c = (c_x, c_y), c' = (c'_x, c'_y)$.

The updating of an IPS I is made through a selection function $\Delta : \mathbb{N} \to \mathscr{I}$ which chooses at each time step a pair of cells to update. The local transition function ϕ of an IPS maps the neighbourhood state of two cells (here c and c') to their new respective state. For a neighbourhood $\mathscr{N}$ of size k, it thus defined from $Q^k \times Q^k$ to Q^2. We these notations, we write: for each time t, with $(c,c') = \Delta(t)$, $(x^{t+1}(c), x^{t+1}(c')) = \phi \begin{pmatrix} x^t(c+n_1), \ldots, x^t(c+n_k) \\ x^t(c'+n_1), \ldots, x^t(c'+n_k) \end{pmatrix}$ and $x^{t+1}(c) = x^t(c)$ if c is not selected by Δ at time t.

The *checkerboard rule* is defined as follows: for a pair of cells c_a and c_b selected for update, let n_I^a (resp. n_I^b) be the number of cells in the *von Neumann* neighbourhood of c_a (resp. c_b) whose state is equal to the state of c_a (resp. c_b). For a real number $\beta \in [0,1]$, the rule consists in always inverting the states of c_a and c_b if $n_I^a + n_I^b > 3$ and inverting these states with probability β if $n_I^a + n_I^b = 3$.

In order to set the value of β, we empirically observed the behaviour of the checkerboard rule on an odd lattice size for different values of this parameter. For $\beta = 0$, the system quickly stabilises on a pattern that is formed by large checkerboard regions, separated by some non-checkerboard "fracture lines" (see Fig. 5-left). For $\beta = 1$, we observe that checkerboard regions form in some parts of the lattice while other parts remain unstable (see Fig. 5-middle). A trade-off between the two behaviours was obtained for $\beta = 0.1$: for this setting, one observes a relatively quick stabilisation on a pattern where the non-checkerboard regions are rather reduced (see Fig. 5-right).

The second component of our rule is the majority rule, which is applied on the Moore neighbourhood with probability ε in order to drive the system to a homogeneous fixed point. A good compromise between a good quality of classification and a reasonable computation time was obtained for $\varepsilon = 0.001$.

For these settings, using 10^4 random samples, we measured quality of classification close to $Q = 0.90$ for a size $L = 21$. This result represents a small improvements over the best two-dimensional rules that are known by the author (Alonso-Sanz and Bull report a quality of $Q = 0.87$ for a two-dimensional block CA enhanced with 4 bits of memory [1]). However, the question as to how we can further improve this performance is still open.

5 Discussion

This note presented an exploration of the density classification problem in two dimensions with deterministic and stochastic cellular automata and interacting particle systems. The results are surprising as they highlight a kind of paradox: On the one hand, for *infinite* grids, analytical results are obtained rather easily, but the one-dimensional-case is very difficult to tackle [2]. On the other hand, if we take *finite* grids, the situation seems reversed: in one dimension, a stochastic CA was found to classify the density with an arbitrary precision, but obtaining a similar result in two dimensions is still an open question. A possibility to move forward would be to search for other types of symmetry breaking, for instance producing stripes instead of checkerboards.

Before we close this note, we would like to say a few words on the motivations to examine this problem in two dimensions. In *The chemical basis of morphogenesis*, Turing raised the question as to how an organism made of identical cells may form a shape that would be obtained by an autonomous differentiation of its cells [13]. The difficulty for answering this question is deeply linked to the symmetries of the problem: if all cells obey the same laws and if the initial condition does not contain any information, how can a structured shape appear in the cellular system? The idea of Turing is that the process that drives the cells' differentiation should result in what we now call a *symmetry breaking*: from a situation where the system looks perfectly symmetric *from a macroscopic point of view*, the system should reach a global state that has *less* symmetries, that is, a state that contains *more information*. For instance, the first step of the morphogenesis process in living organisms consists in transforming a sphered egg where all cells are equivalent into an ellipsoidal egg, where North and South poles are differentiated.

Our interest is to link the density classification problem with this question of symmetry breaking. Indeed, if we adopt a statistical point of view, the problem itself can be reformulated as a symmetry breaking process : it amounts to starting from a uniform distribution (all cells can be in state 0 or 1 with equal probability) and reaching a distribution reduced to the two homogeneous fixed points (cells have reached a consensus state **0** or **1**). We thus call for further explorations to determine whether the regularities seen in cellular automata are an advantage or a drawback to achieve a symmetry breaking. Our systems display a kind of a "maximal simplicity": a discrete regular space, cells have a binary state which is updated at discrete time steps. The method chosen for updating cells is thus crucial : in

cellular automata, the synchronous updating may not help breaking the symmetry. Interacting particle systems, which involve a fully asynchronous updating, may give more freedom as seen in the present note. More generally, we would like to know whether the presence of randomness can be an aid to perform computations... As noted by Suzudo [11], the problem of creating checkerboard patterns from random initial conditions is already interesting per se.

References

[1] Ramón Alonso-Sanz & Larry Bull (2009): *A very effective density classifier two-dimensional cellular automaton with memory*. Journal of Physics A 42, p. 485101.

[2] Ana Bušić, Nazim Fatès, Jean Mairesse & Irène Marcovici (2012): *Density Classification on Infinite Lattices and Trees*. In David Fernndez-Baca, editor: Proceedings of LATIN 2012: Theoretical Informatics, Lecture Notes in Computer Science 7256, Springer, pp. 109–120, doi:10.1007/978-3-642-29344-3_10.

[3] Alberto Dennunzio & Enrico Formenti (2008): *Decidable Properties of 2D Cellular Automata*. In Masami Ito & Masafumi Toyama, editors: Developments in Language Theory, Lecture Notes in Computer Science 5257, Springer Berlin / Heidelberg, pp. 264–275, doi:10.1007/978-3-540-85780-8_21.

[4] R. L. Dobrushin, V. I. Kriukov & A. L. Toom (1991): *Stochastic Cellular Systems : Ergodicity, Memory, Morphogenesis*. Manchester University Press.

[5] Nazim Fatès (2012): *Stochastic Cellular Automata Solutions to the Density Classification Problem – When Randomness Helps Computing*. Theory of Computing Systems, doi:10.1007/s00224-012-9386-3. In Press - Short version appeared in the proceedings of Stacs'11, see `http://dx.doi.org/10.4230/LIPIcs.STACS.2011.284`.

[6] Nazim Fatès & Lucas Gerin (2009): *Examples of Fast and Slow Convergence of 2D Asynchronous Cellular Systems*. Journal of Cellular Automata 4(4), pp. 323–337.

[7] Peter Gács, Georgii L. Kurdiumov & Leonid A. Levin (1987): *One-Dimensional Homogeneous Media Dissolving Finite Islands*. Problemy Peredachi Informatsii 14, pp. 92–98.

[8] Mark Land & Richard K. Belew (1995): *No Perfect Two-State Cellular Automata for Density Classification Exists*. Physical Review Letters 74(25), pp. 5148–5150, doi:10.1103/PhysRevLett.74.5148.

[9] Norman H. Packard (1988): *Dynamic Patterns in Complex Systems*, chapter Adaptation toward the edge of chaos, pp. 293 – 301. World Scientific, Singapore.

[10] Rene Reynaga & Eligio Amthauer (2003): *Two-dimensional cellular automata of radius one for density classification task p=1/2*. Pattern Recognition Letters 24(15), pp. 2849–2856, doi:10.1016/S0167-8655(03)00143-0.

[11] Tomoaki Suzudo (2004): *Searching for Pattern-Forming Asynchronous Cellular Automata - An Evolutionary Approach*. In Peter M. A. Sloot, Bastien Chopard & Alfons G. Hoekstra, editors: Proceedings of ACRI'04, Lecture Notes in Computer Science 3305, Springer, pp. 151–160, doi:10.1007/978-3-540-30479-1_16.

[12] Véronique Terrier (2006): *Low complexity classes of multidimensional cellular automata*. Theor. Comput. Sci. 369(1-3), pp. 142–156, doi:10.1016/j.tcs.2006.07.061.

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