ESAIM: PROCEEDINGS AND SURVEYS, February 2025, Vol. 80, p. 17-25
Editors: Pierre Calka, Clément Marteau - Journées MAS 2022, Université de Rouen

CELLULAR AUTOMATA AND SELF-ORGANISATION PHENOMENA

Irène Marcovici[1]

Abstract. Cellular automata are dynamical systems for which time and space are discrete. They are used to model the evolution of a set of components, the *cells*, that interact locally with each other: over time, each cell updates its state according to what it perceives in its neighbourhood. Some cellular automata exhibit a self-organisation behaviour: from an initial disordered state, successive updates of the cells by the local rule lead to the emergence of a macroscopic structure. Conversely, given a desired global behaviour, we can ask ourselves which local rules allow to achieve this collective behavior, in a decentralised way. In this article, we will address several such *inverse problems* (synchronisation, density classification, self-correction of tilings), and study the influence that the introduction of randomness can have on the dynamics.

Résumé. Les automates cellulaires sont des systèmes dynamiques pour lesquels le temps et l'espace sont discrets. Ils permettent de modéliser l'évolution d'un ensemble de composantes, les *cellules*, interagissant entre elles de manière locale : au cours du temps, chacune actualise son état en fonction de ce qu'elle perçoit dans son voisinage. Certains automates cellulaires exhibent des comportements d'auto-organisation : à partir d'un état initial désordonné, les mises à jour successives des cellules par la règle locale conduisent à l'apparition d'une structure macroscopique. À l'inverse, si l'on souhaite parvenir à un certain comportement global, on peut se demander quelles règles locales permettent de l'atteindre de manière décentralisée. Dans cet article, nous présenterons plusieurs *problèmes inverses* de ce type (synchronisation, classification de la densité, auto-correction de pavages), en étudiant l'influence que peut avoir l'introduction d'aléa dans les dynamiques.

Introduction

Self-organisation phenomena are frequent in nature. A striking example is the formation of flocks of birds: despite the absence of a central authority, the birds manage to agree on a common flight direction, by collecting local information on the flight direction of their neighbours. A recent work has also shown that the patterns appearing on some lizard skins are formed through a similar mechanism, in a completely decentralised way [11], and that is just one more of many examples. Cellular automata provide a simple computational model to explore these self-organisation phenomena: we consider a set of entities, the *cells*, arranged on a lattice, and we assume that the states of these cells evolve over time according to a local rule that only depends on the states of a few of their neighbours. To simplify things even further, we suppose that the number of possible cell states is finite, and that all cells are updated simultaneously, in discrete time. We will propose several examples showing that it is possible to reproduce some self-organisation phenomena in this artificial framework. The objective is to extract some elementary mechanisms leading to such behaviours, with the ambition of being able to exploit them in concrete contexts. The general spirit of the problem is that of distributed computing: gathering a global information by exchanging only local information. Indeed, many computer networks operate in a distributed

[1] Univ Rouen Normandie, CNRS, Normandie Univ, LMRS UMR 6085, F-76000 Rouen, France

Article published online by EDP Sciences and available at https://www.esaim-proc.org or https://doi.org/10.1051/proc/202580017

manner, without a central authority, and require the implementation of a form of collective regulation, based on local interactions.

In the present article, we will investigate different *inverse problems*. The approach is as follows: given a certain self-organisation phenomenon, we look for a cellular automaton, as simple as possible, allowing us to achieve this behaviour from a disordered initial state. We will see that introducing some randomness in the local interactions facilitates the implementation of self-organisation phenomena.

In Section 1, we first define cellular automata and introduce some terminology and notations. Then in Section 2, we present two well-known inverse problems defined on finite rings: the synchronisation problem and the density classification. In Section 3, we focus on the self-stabilisation of infinite two-dimensional tilings. The article ends with some remarks and perspectives in Section 4.

1. Definition of cellular automata

Let us start by defining cellular automata in a more formal way. We consider a lattice $\mathbb{L}$, which is usually chosen as the infinite grid $\mathbb{Z}^d$ or as a finite set of the form $\mathbb{Z}_{n_1} \times \cdots \times \mathbb{Z}_{n_d}$, where $\mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z}$. We also fix a finite set S, which can be seen as a finite set of colours. The elements of $S^{\mathbb{L}}$ then correspond to colourings of the *cells* of the grid $\mathbb{L}$ with colours from S, and are called *configurations*.

For a given integer $m \geq 1$ and a family of m elements $n_1, \ldots, n_m \in \mathbb{L}$, let us consider a function $f : S^m \to S$. The *cellular automaton* of *neighbourhoood* $\mathcal{N} = \{n_1, \ldots, n_m\}$ and of *local rule* f is then the function $F : S^{\mathbb{L}} \to S^{\mathbb{L}}$ defined, for any $x \in S^{\mathbb{L}}$ and $k \in \mathbb{L}$, by

$$F(x)_k = f(x_{k+n_1}, \ldots, x_{k+n_m}).$$

In order to determine the new colour $F(x)_k$ of cell k in the configuration $F(x)$, we thus look at the colours of the neighbouring cells $k + n_1, \ldots, k + n_m$ in the configuration x, and then apply the local rule f to the pattern observed.

Probabilistic cellular automata are an extension of cellular automata for which the local function f is no more with values in S but in the set $\mathcal{M}(S)$ of probability distributions on S. For $(a_1, \ldots, a_m) \in S^m$ and $s \in S$, the value $f(a_1, \ldots, a_m)(s)$ then gives the probability to update a cell by the value s if its neighbourhood is in state $(a_1, \ldots, a_m)$. As previously, all the cells are updated in a synchronous way, and conditionally on the configuration, the updates are made independently for different cells. A probabilistic cellular automaton can thus be seen as a Markov chain on $S^{\mathbb{L}}$. We refer to the work of Mairesse and Marcovici for a survey on probabilistic cellular automata [10].

In the next section, we will consider deterministic and probabilistic cellular automata defined on one-dimensional finite lattices with periodic boundary conditions, that is, on $\mathbb{L} = \mathbb{Z}_n$ for some $n \geq 1$, where we recall that $\mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z}$. This amounts to assuming that the cells are arranged along a ring of size n.

2. Inverse problems on finite rings: the power of randomness

In this section, we will introduce and study two *inverse problems*. In both cases, we will assume that we have at our disposal a finite number of cells arranged on a ring $\mathbb{L} = \mathbb{Z}_n$, and that the cells can only take two states: 0 and 1. We denote by $0^{\mathbb{Z}_n}$ the configuration where all cells are in state 0, and by $1^{\mathbb{Z}_n}$ the configuration where all cells are in state 1.

2.1. **The synchronisation problem**

The synchronisation problem consists in designing a binary cellular automaton such that after some time, the cells are alternately all in state 0, then all in state 1, then all in state 0, etc. The aim is thus to give a same local instruction to all the cells, so that they all reach a regime where they alternate in a perfectly synchronous way between state 0 and state 1. The desired behaviour can be interpreted as a digital replica of the synchronisation of metronomes: just as metronomes placed on a board eventually synchronise to beat at exactly the same tempo, we would like the cellular automaton to allow the cells to reach a perfect synchronisation, whatever the size of the ring and the initial configuration.

Formally, we give the following definition.

Definition 2.1. A one-dimensional cellular automaton F on $S = \{0,1\}$ is said to *synchronise* finite configurations if for any ring size $n \geq 1$,

$$\forall x \in S^{\mathbb{Z}_n}, \quad \exists T \geq 1, \quad \forall k \geq 0, \quad \begin{cases} F^{T+2k}(x) = 0^{\mathbb{Z}_n} \\ F^{T+2k+1}(x) = 1^{\mathbb{Z}_n} \end{cases} .$$

In the context of probabilistic cellular automata, we ask the same property to be satisfied, with a time T almost surely finite.

In fact, if we restrict ourselves to a deterministic framework, it is not possible to find a solution, as shown by the following proposition.

Proposition 2.2. *There exist no deterministic cellular automaton that synchronises finite configurations.*

Proof. In order to prove this result, we will use an argument due to Richard [12], showing that for any cellular automaton, there exists a configuration that is rotated indefinitely on the ring under the action the cellular automaton. From such a configuration, one can never reach the alternation between configurations $0^{\mathbb{Z}_n}$ and $1^{\mathbb{Z}_n}$.

Let F be a one-dimensional cellular automaton. We can assume without loss of generality that $\mathcal{N} = \{-r, \dots, r\}$ for some $r \geq 1$, so that F has a local function $f : \{0,1\}^{2r+1} \to \{0,1\}$. Let us introduce successively the values:

$$x_0 = f(0, \dots, 0), \quad x_1 = f(0, \dots, 0, x_0), \quad x_2 = f(0, \dots, 0, x_0, x_1), \quad \text{etc.}$$

Precisely, for $i \leq 2r$, we have $x_i = f(0, \dots, 0, x_0, x_1, \dots, x_{i-1})$, and for $i \geq 2r+1$, $x_i = f(x_{i-2r-1}, \dots, x_{i-1})$. Since the x_i can only take values 0 and 1, there exist integers $m < \ell$ such that $(x_m, \dots, x_{m+2r}) = (x_\ell, \dots, x_{\ell+2r})$. Now, consider the configuration

$$x_m x_{m+1} \dots x_{\ell-1} \in \{0,1\}^{\mathbb{Z}_{\ell-m}}.$$

By definition of the sequence $(x_k)_{k\geq 0}$, its image by F is the sequence

$$x_{m+r+1} x_{m+r+2} \dots x_{\ell+r} = \sigma^{r+1}(x_m x_{m+1} \dots x_{\ell-1}),$$

where σ denotes the shift map on $\{0,1\}^{\mathbb{Z}_{\ell-m}}$. So, on a ring of size $\ell - m$, starting from the configuration $x_m x_{m+1} \dots x_{\ell-1}$, this configuration will be shifted indefinitely by $r+1$ cells and we will never achieve the desired synchronisation behaviour. □

However, if we allow probabilistic updates, a simple solution exists: it is a probabilistic cellular automaton with a neighbourhood of size 2, for which the local rule consists in selecting the value of the cell or of its right-hand neighbour with probability 1/2 each, and in changing the value that has been read. Thus, if the two cells are in the same state i, the new state is $1-i$, while if they are in different states, the new state is drawn uniformly between 0 and 1. The following statement formalises this result.

Proposition 2.3. *The probabilistic cellular automaton of neighbourhood* $\mathcal{N} = \{0,1\}$ *and local rule*

$$f(i,j) = \frac{1}{2}\delta_{1-i} + \frac{1}{2}\delta_{1-j},$$

synchronises finite configurations.

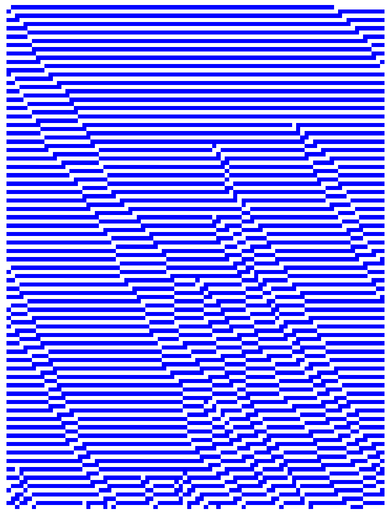

Figure 1. Example of space-time diagram of the synchronising probabilistic cellular automaton of Proposition 2.3. The initial configuration is represented at the bottom and time goes upwards.

Proof. According to the local rule, ranges of 0's (resp. 1's) are deterministically transformed into ranges of 1's (resp. 0's). Therefore, only the interfaces between ranges of 0's and ranges of 1's behave randomly, and it can be seen that when iterating the probabilistic cellular automaton, these interfaces follow independent random walks: precisely, each interface moves one step to the left with probability 1/2, and stays in the same position with probability 1/2, see Figure 1 for an illustration. Since the ring size is finite, the interfaces eventually merge: we have a Markov chain with finite state space $\{0,1\}^{\mathbb{Z}_n}$, and the cycle between the two configurations $0^{\mathbb{Z}_n}$ and $1^{\mathbb{Z}_n}$ is the unique absorbant cycle. □

In fact, any probabilistic cellular automaton such that $f(0,\ldots,0)=\delta_1, f(1,\ldots,1)=\delta_0$, and such that the other transitions assign a positive probability to 0 and to 1, provides a solution to the synchronisation problem. But the advantage of the rule above is that the expected convergence time is quadratic. It is an open question to know whether there exists a probabilistic cellular automaton that converges more rapidly.

2.2. The density classification problem

The density classification problem consists in deciding, in a decentralised way, if an initial configuration contains more 0's or more 1's. More precisely, the goal is to design a probabilistic cellular automaton whose trajectories converge to $0^{\mathbb{Z}_n}$ or to $1^{\mathbb{Z}_n}$, if the initial configuration contains more 0's or more 1's, respectively.

For a configuration $x \in \{0,1\}^{\mathbb{Z}_n}$, we denote by $\rho(x)$ the proportion of 1's in x, that is, $\rho(x)=|x|_1/n$, where $|x|_1=\sum_{i=1}^n \mathbf{1}_{x_i=1}$.

Definition 2.4. A one-dimensional cellular automaton F on $S=\{0,1\}$ is said to classify the density of finite configurations if for any ring size $n\geq 1$,

$$\forall x \in\ S^{\mathbb{Z}_n},\quad \begin{cases}\rho(x)>1/2 \implies \exists T\geq 1,\ \forall k\geq T, \quad F^k(x)=1^{\mathbb{Z}_n}\\ \rho(x)<1/2 \implies \exists T\geq 1,\ \forall k\geq T, \quad F^k(x)=0^{\mathbb{Z}_n}\end{cases}.$$

In the context of probabilistic cellular automata, we ask the same property to be satisfied, with a time T almost surely finite.

The difficulty is twofold: first, it is impossible to centralise the information (cells are indistinguishable); second, it is impossible to use classical counting techniques (cells contain only binary information). Here, one can prove that there is no perfect solution to the density classification problem, even if we allow probabilistic updates.

Proposition 2.5. *There exist no (probabilistic) cellular automaton that classifies the density of finite configurations.*

This proposition was first proved by Land and Belew in the deterministic setting [9], and then simplified and extended by Bušić et al. [1].

Sketch of the proof. We give a sketch of the proof of Bušić et al. [1]. Let us assume that the (probabilistic) cellular automaton F classifies the density. Then, there exists an integer n and a configuration $x \in S^{\mathbb{Z}_n}$ such that $\frac{n}{2} < |x|_1$, and such that the event $|x|_1 < |F(x)|_1$ has positive probability. Now, let us consider a configuration y of the form

$$y = \underset{\leftarrow \quad kn \quad \rightarrow}{\overbrace{x \;\; x \;\; \ldots \;\; x}^{k}} \; \underset{\leftarrow \quad m \quad \rightarrow}{\overbrace{0 \;\; 0 \;\; \ldots \;\; 0}^{m}},$$

for some integers $k, m \geq 1$. We can adjust k and m in order to construct a configuration $y \in S^{\mathbb{Z}_{kn+m}}$ such that $\rho(y) < \frac{1}{2}$, and such that the event $\rho(F(y)) > \frac{1}{2}$ has positive probability. This contradicts the fact that F classifies the density. □

Nevertheless, as shown by Fatès [3] the introduction of randomness expands the possibilities compared to the deterministic framework. Indeed, a probabilistic combination of two deterministic cellular automata, namely the traffic rule and the majority rule, allows us to solve the density classification problem with an arbitrary precision. Let us go into more detail. The traffic rule, known as rule 184 in Wolfram's classification, is a rudimentary model describing the progression of vehicles along a road: 1's represent vehicles, and at each time step, all vehicles with an empty cell in front of them move rightwards. This defines a one-dimensional cellular automaton with radius 1, that is, with neighbourhood $\mathcal{N} = \{-1, 0, 1\}$, whose local rule can be represented as follows:

$\texttt{traf}(x,y,z)$	1	0	1	1	1	0	0	0
$x\,y\,z$	111	110	101	100	011	010	001	000

It is a conservative cellular automaton: the numbers of 0's and 1's are preserved through the evolution. We also consider the majority rule of radius 1, whose local rule `maj` outputs the symbol which is in majority among the three input symbols:

$\texttt{maj}(x,y,z)$	1	1	1	0	1	0	0	0
$x\,y\,z$	111	110	101	100	011	010	001	000

Now, the *majority-traffic* cellular automaton of parameter $\alpha \in (0,1)$ is the probabilistic cellular automaton of local rule:

$$f(x,y,z) = \alpha\, \delta_{\texttt{maj}(x,y,z)} + (1-\alpha)\, \delta_{\texttt{traf}(x,y,z)}.$$

In other words, at each time step, we choose, independently for each cell, to apply the majority rule with probability α and the traffic rule with probability $1-\alpha$.

Proposition 2.6. *Let $n \geq 1$. For any $\varepsilon \in (0,1)$, there exists $\alpha_n \in (0,1)$ such that for any configuration $x \in \{0,1\}^{\mathbb{Z}_n}$, the majority-traffic probabilistic cellular automaton classifies correctly the density of configuration x with probability larger than $1-\varepsilon$.*

Proof. We give a sketch of the proof given by Fatès [3]. It is based on the following property of the traffic cellular automaton: if the initial configuration contains a majority of 1's (resp. 0's), then after less than n iterations, the configuration does not contain anymore two consecutive 0's (resp. 1's). Now, if α is sufficiently close to 0, then with high probability, the first steps of majority-traffic will be the same as if we had iterated

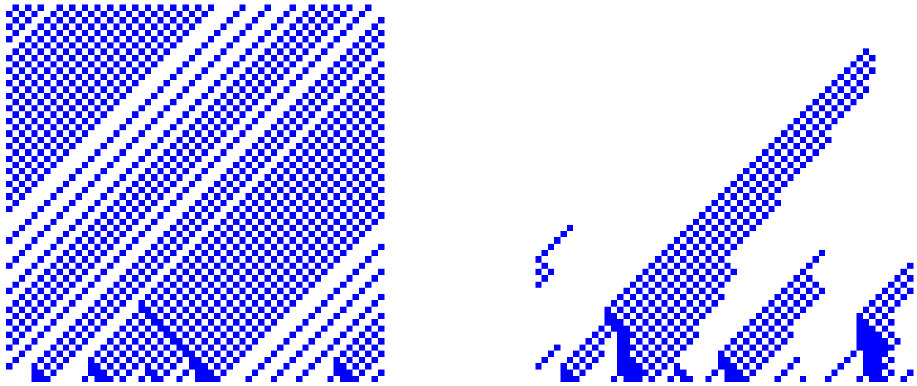

Figure 2. Examples of space-time diagrams of the traffic cellular automaton (left) and of majority-traffic (right), from a same initial configuration represented at the bottom, that contains a majority of 0's (white cells).

the deterministic traffic cellular automaton, so that we will have reached a configuration where there is no two consecutive occurrences of the minority state. To conclude, one just has to observe that such a configuration is well classified with probability 1 by majority-traffic: the applications of the majority rule can only make the configuration evolve to the wanted fixed point. An illustration is given in Figure 2. □

However, note that lowering the value of the parameter α_n improves the quality of the result, but at the cost of increasing the system response time.

3. Self-stabilisation of tilings on the infinite lattice

In this section, we consider another *inverse problem*, now defined on infinite lattices. To simplify the presentation, we will focus on the two-dimensional case, and explore the phenomenon of self-stabilisation in the context of cellular automata which operate on two-dimensional tilings. To illustrate the problem, imagine that an artist has a plan to create a two-dimensional tiling with square tiles having a colour on each side, with the constraint that two adjacent tiles must share the same colour on their adjacent side (they are known as *Wang tiles*). When this tiling is realised, the artist realises that (a) some mistakes have occurred during the tiling process and (b) the original tiling plan has been lost. In this context, is it possible to correct the tiling to respect the constraints of adjacency of colours only by following local rules? In other words, given a finite set of local constraints, we seek a cellular automaton such that, starting from a finite perturbation of a valid configuration, the cellular automaton must eventually fall back into the space of valid configurations where it remains still. Precisely, we require our cellular automaton to have the following form of self-stabilisation.

(1) Starting from a configuration that deviates from a legal configuration only on a finite region, the cellular automaton must evolve back to a legal configuration in a finite number of steps.
(2) Starting from a legal configuration, the cellular automaton must remain unchanged.

Again, the difficulty is that the cells are indistinguishable and the information available to each cell is limited to the state of its close neighbours.

Self-stabilisation is a property omnipresent in biological systems. Indeed, living cells always need to correct their defects in order to keep their behaviour as stable as possible. The present questioning is a modest step aimed at trying to reproduce this phenomenon in a simplified way in an artificial context.

The main reference for this section is our work on self-stabilisation [5], in which we studied this problem from different angles, looking for both deterministic and probabilistic solutions. However, in the context of the present article, we will only give a short presentation of some of the points raised in the article, avoiding technicalities.

3.1. Self-stabilisation in linear time

As a first example, let us assume that only two types of tiles are available, white tiles (denoted by 0) and black ones (denoted by 1). The adjacency constraint thus implies that there are only two allowed configuration: the "all white" configuration $0^{\mathbb{Z}^2}$ and the "all black" configuration $1^{\mathbb{Z}^2}$.

In this specific case, a cellular automaton with the desired behaviour is called an *eroder*, and a well-known example is Toom's North-East-Center majority rule. Let us denote $e_1 = (1, 0)$ and $e_2 = (0, 1)$.

Definition 3.1. Toom's (deterministic) majority cellular automaton is the cellular automaton $\texttt{NEC-Maj}$: $\{0,1\}^{\mathbb{Z}^2} \to \{0,1\}^{\mathbb{Z}^2}$ with neighbourhood $\mathcal{N} = \{0, e_1, e_2\}$ defined, for any $x \in \{0,1\}^{\mathbb{Z}^2}$ and $k \in \mathbb{Z}^2$, by

$$\texttt{NEC-Maj}(x)_k = \texttt{maj}(x_k, x_{k+e_1}, x_{k+e_2}).$$

Proposition 3.2. *Toom's majority cellular automaton stabilises the set $\{0^{\mathbb{Z}^2}, 1^{\mathbb{Z}^2}\}$ from finite perturbations, in linear time with respect to the diameter of the modified area.*

Proof. If the perturbation is included in the triangular area defined by the three points $k, k + re_1, k + re_2$, for some $k \in \mathbb{Z}^2$ and $r \in \mathbb{N}$, then, after one application of Toom's cellular automaton, it is included in the triangular area defined by the points $k, k + (r - 1)e_1, k + (r - 1)e_2$. The result follows. □

In fact, it is known that Toom's cellular automaton also has the property to stabilise the set $\{0^{\mathbb{Z}^2}, 1^{\mathbb{Z}^2}\}$ from *random* perturbations. More than that, a remarkable result is that, starting from a Bernoulli product measure of parameter p, it converges to the configuration $0^{\mathbb{Z}^2}$ if $p < 1/2$, and to $1^{\mathbb{Z}^2}$ if $p > 1/2$. In a way, Toom's cellular automaton thus provides a solution to the density classification problem discussed in the previous section, but here in the context of an infinite grid. We refer to the article of Bušić et al. [1] for further discussions on this topic. This result is specific to this automaton, but the following theorem holds, we refer to the work of Fatès et al. [5] for a formal statement and a proof of this result.

Theorem 3.3. *If a cellular automaton stabilises a tiling set from finite perturbations in linear time, then it also stabilises it from Bernoulli random perturbations with a sufficiently low density of errors.*

The proof is based on the idea of sparseness due to Gács [6–8], and Durand, Romashchenko and Shen [2].

Let us give some classes of tiling sets that can be stabilised in linear time with respect to the diameter of the modified area, and for which the above theorem can therefore be applied:

(1) periodic tilings, or equivalently, tiling sets containing only a finite number of allowed configurations;
(2) single-cell fillable tilings, that is, tiling sets for which, given four tiles located at the bottom, top, left and right of a given cell, there is always at least one admissible way to fill the central cell.

Apart from the example above of the two *monochromatic* configurations $0^{\mathbb{Z}^2}$ and $1^{\mathbb{Z}^2}$, an example of periodic tiling is given by the set containing the two checkerboards configurations. Concerning single-cell fillable tilings, a simple example is given by k-colourings, with $k \geq 5$. As an application of Theorem 3.3, any cellular automaton that stabilises these tilings in linear time also stabilises small random perturbations.

3.2. Other results and open questions

In the previous subsection, Toom's majority cellular automaton provided a directional solution to the stabilisation problem: the cells need to distinguish between the four directions North, South, East, West, even though the constraints defining the tiling set is perfectly symmetrical. In general, finding self-stabilising cellular automata that respect the symmetries of the tiling set appears to be a difficult problem, which does not always have a solution. However, in the specific case of the set $\{0^{\mathbb{Z}^2}, 1^{\mathbb{Z}^2}\}$, the use of randomness in the evolution of the cellular automata allows us to design a simple solution that achieves self-stabilisation with a nearest-neighbour isotropic rule. Namely, a solution is given by the probabilistic cellular automaton whose local rule is the majority on the four nearest neighbours, with a uniform random drawing in case of equality. We can extend the above isotropic probabilistic rule to stabilise any finite tiling set in at most cubic time.

In a similar vein, given a single-cell fillable tiling, it is possible to design a probabilistic cellular automaton that stabilises it from finite perturbations in at most logarithmic time.

In contrast, we have also been able to prove that for some choices of the local constraints, the self-stabilisation problem is inherently hard. More precisely, unless P=NP, there exist tiling sets requiring super-polynomial stabilisation time. An interesting example that is still open is the case of 3-colourings: indeed, the local constraints can produce long-range correlations, so that for the moment, we have no candidate for the self-stabilisation. We refer to [5] for further discussion on these questions.

4. Conclusion and perspectives

We have presented different *inverse problems* consisting of finding local rules allowing to achieve a global organisation behaviour from a disordered initial state.

To get a little closer to a concrete problem affecting computer networks, we have also recently been looking at another inverse problem, the *decentralised diagnosis problem* [4]. The objective is to design a system able to detect the presence of a given number of failures in a distributed network, in a totally decentralised way. More precisely, we ask that when the density crosses a given threshold, all the components are in the alert state, and remain so. Again, we aim to work with a minimal model, and consider only cellular automata with three states: neutral, alert, defect. Our idea is to take advantage from the phase transition phenomena observed in some probabilistic cellular automata where a qualitative change of behaviour occurs when their transition probabilities are continuously varied. Like in the previous section, when working on the two-dimensional lattice, a relatively simple solution can be proposed on Toom's neighbourhood: this neighbourhood breaks the isotropy and installs a direction in which information travels. While this may seem a drawback for concrete applications, it has the advantage of allowing us to derive a more formal analysis of the behaviour of the model. Work is still in progress to propose an isotropic rule, which could be more easily adapted to general networks.

Two main lessons can be drawn from this collection of examples:

(1) the introduction of randomness in the dynamics can be an essential ingredient for self-organisation,
(2) setting up a direction in which information is propagated can also facilitate self-organisation.

But it remains a little frustrating not to be able to give a more general framework to these observations. In fact, when working with complex systems, we expect about any behaviour to be possible, if we allow any type of local rule. But when we impose particular constraints on the system (binary states, neighbourhood of small size, deterministic rule, isotropy, etc.), this restricts the possibilities in ways that are often difficult to identify precisely. However, this seems to be a necessary step in order to be able to go beyond numerical simulations by developing a mathematical analysis of the models involved, and therefore to understand their behaviour theoretically.

References

[1] Bušić, A., Fatès, N., Mairesse, J., Marcovici, I.: Density classification on infinite lattices and trees. Electron. J. Probab. **18**, 22 (2013).
[2] Durand, B., Romashchenko, A., Shen, A.: Fixed-point tile sets and their applications. J. Comput. Syst. Sci. **78**(3), 731–764 (2012).
[3] Fatès, N.: Stochastic cellular automata solutions to the density classification problem. When randomness helps computing. Theory Comput. Syst. **53**(2), 223–242 (2013).
[4] Fatès, N., Marchand, R., Marcovici, I.: A decentralised diagnosis method with probabilistic cellular automata. In: Cellular automata and discrete complex systems. 29th IFIP WG 1.5 international workshop, AUTOMATA 2023, Trieste, Italy, August 30–September 1, 2023. Proceedings. Cham: Springer (2023).
[5] Fatès, N., Marcovici, I., Taati, S.: Self-stabilisation of cellular automata on tilings. Fundam. Inform. **185**(1), 27–82 (2022).
[6] Gács, P.: Reliable computation with cellular automata. J. Comput. Syst. Sci. **32**, 15–78 (1986).
[7] Gács, P.: Reliable cellular automata with self-organization. J. Stat. Phys. **103**(1-2), 45–267 (2001).

[8] Gács, P., Reif, J.: A simple three-dimensional real-time reliable cellular array. J. Comput. Syst. Sci. **36**(2), 125–147 (1988).
[9] Land, M., Belew, R.: No perfect two-state cellular automata for density classification exists. Physical review letters **74**(25), 5148—5150 (1995).
[10] Mairesse, J., Marcovici, I.: Around probabilistic cellular automata. Theor. Comput. Sci. **559**, 42–72 (2014).
[11] Manukyan, L., Montandon, S.A., Fofonjka, A., Smirnov, S., Milinkovitch, M.C.: A living mesoscopic cellular automaton made of skin scales. Nature **544**(7649), 173—179 (2017).
[12] Richard, G.: On the Synchronisation Problem over Cellular Automata. In: Vollmer, H., Vallée, B. (eds.) 34th Symposium on Theoretical Aspects of Computer Science (STACS 2017). Leibniz International Proceedings in Informatics (LIPIcs), vol. 66, pp. 54:1–54:13. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2017).