



HAL
open science

Asynchronous cellular system that solves the parity problem

Nazim A. Fatès

► **To cite this version:**

Nazim A. Fatès. Asynchronous cellular system that solves the parity problem. 30th International Workshop on Cellular Automata and Discrete Complex Systems, Jul 2024, Durham (GB), France. hal-03894581v2

HAL Id: hal-03894581

<https://inria.hal.science/hal-03894581v2>

Submitted on 26 Jun 2024

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.



Distributed under a Creative Commons Attribution 4.0 International License

Asynchronous cellular systems that solve the parity problem

Nazim Fatès

Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

June 26, 2024

Abstract

We present stochastic cellular automata that classify the parity of initial conditions. The model is an interacting particles system where cells are updated by pairs, randomly chosen at each time step. A first rule is proposed, with a symmetry between 0's and 1's. We show that it classifies the parity of the initial configurations thanks to a non-biased random walk of the frontiers between 0's and 1's. We present an analysis of the classification time, as well as numerical simulations, to establish that the classification time scales quadratically with the number of cells. In a second time, breaking the state symmetry, we propose an improvement of this rule with the simultaneous use of two classifying systems. keywords : asynchronous cellular automata ; interacting particle systems ; parity classification

1 Introduction

Cellular automata are an original model of computation where parallelism and locality are primordial. They are formed of elementary computing units, the cells, which are arranged on a grid of arbitrary dimension, and which interact according to a local rule. The cells change their state at a discrete times, either synchronously if a global clock is present, or asynchronously when the updating times are independent. Although they have the same computing power as Turing machines, it is not yet well understood how to program cellular automata in order to obtain a given behaviour, especially when we aim to use a small number of states and only simple interaction rules.

With their parallel and local mode of operation, cellular automata and their derived models form a fertile field of research; they invite us to think differently about computational processes. In this research field, two main families of problems exist: in the direct problems, one analyses the properties of a rule or a set of rules, while in the *inverse problems*, one seeks to find the rules that display a given behaviour, often a computational task. The *parity problem* is an inverse problem: given a finite one-dimensional cellular system with binary

states and spatially periodic conditions (cycles), find a rule which determines the parity of the number of 1's contained in this initial configuration. More precisely, it is a consensus problem: if the parity of the initial configuration is even, all the cells should agree on 0 and remain in this state, while they should all agree on 1 if this number is odd [1].

A deterministic solution to the problem was proposed by Betel and al. [2]. These authors designed a clever way to achieve the classification of the initial conditions by constructing a rule which alternatively tries to attain the all-1 and the all-0 fixed points with internal “signals” that travel back and forth. The difficulty to overcome is that such signals need to be encoded within the system itself, without the use of additional states. The rule they proposed has a radius of four, which means that each cell decides its next state according to the state of the four nearest cells on the left and on the right, plus its own state. In order to be encoded, this rule can be given as a sequence of 512 bits, a 156-digit code, or, more compactly, as a list of 10 sub-rules with wildcards, that is, states than can take any value indifferently¹.

In a different context, Ruivo and de Oliveira proposed to solve the parity problem with a simple rule, namely the Elementary Cellular Automaton 150, with a particular asynchronous updating [3]. With this local rule, each cell simply applies an XOR operation on its own state and the state of its left and right neighbours. The cells are updated according to the parity of their position on the line: all the even cells are updated synchronously and then all the odd cells, which means that a global synchronisation of the transitions is still required. These authors and their coauthors also tackled a more general case where the dynamical systems were defined on various types of graphs [4, 5].

In this work, we revisit this problem with a model which uses *independent random updates* of pairs of cells. This means that this is an *interacting particle system*: at each time step, a pair of adjacent cells is chosen randomly and uniformly and then the state of the two cells of the pair is simultaneously updated. Our aim is to explore to which extent the introduction of randomness in the updating scheme can allow us to simplify the construction of a solution. We will first present a formal framework to define the problem and a first solution and, in a second step, we study the convergence properties of this rule with a worst-case analysis. We also present an improvement of the classification process by using simultaneously two different rules. We then present numerical simulations for the average case and conclude with a few words of perspective.

2 Presentation of the model

We deal with a finite system of n cells arranged with periodic boundary conditions, that is, they form a ring. The set of cells is thus denoted by $\mathcal{L} = (\mathbb{Z}/n\mathbb{Z})$.

¹The solution that was initially published is possibly altered by a mistake, which nevertheless does not compromise the whole construction. [private communications]

We assume that cells have a binary state in $Q = \{0, 1\}$. The state of the system at a given time is called a *configuration*; for a configuration x , the state of cell $i \in \mathcal{L}$ is denoted by x_i . The set of configurations is denoted by $\mathcal{E}_n = Q^{\mathcal{L}}$. We say that the parity $P(x)$ of a configuration x is even or odd if the number of 1's it contains is even or odd, respectively. We will write $P(x) = |x|_1 \pmod 2$, where $|x|_1$ denotes the number of 1's in x . We also call a 1-region (resp. a 0-region) a set of contiguous cells in state 1 (resp. 0) which is maximal (with respect to the inclusion).

We now present our first model for solving the parity problem. Let $(U_t)_{t \in \mathbb{N}}$ be a series of independent identically distributed random variables that draw uniformly a pair of adjacent cells in \mathcal{L} . These random variables represent the *updating method*, here, the pair of cells which is selected for being updated at each time step. For the sake of simplicity we will identify the random variable U_t and its realisation and simply consider that $U_t = \{i, i + 1\}$ represents the fact that the cells i and $i + 1$ are updated at time t .

Starting from a given configuration $x \in \mathcal{E}_n$, the evolution of the system is stochastic; let $(x^t)_{t \in \mathbb{N}}$ be the series of random configurations that will be obtained from x , with the initial condition $x^0 = x$. We now introduce the *balanced rule*: For $\{i, i + 1\} = U(t)$,

$$(x_i^{t+1}, x_{i+1}^{t+1}) = \begin{cases} (1 - x_i^t, 1 - x_{i+1}^t) & \text{if } x_{i-1}^t \neq x_i^t \text{ or } x_{i+1}^t \neq x_{i+2}^t, \\ (x_i^t, x_{i+1}^t) & \text{otherwise;} \end{cases}$$

and $\forall i \in \mathcal{L}, i \notin U_t \implies x_i^{t+1} = x_i^t$.

In words, the cells that are not updated keep their state and the state of the two cells selected for an update will be flipped only if there is a difference of state between the left updated cell and its left neighbour or between the right updated cell and its right neighbour. These patterns, which trigger a flip, are called *active neighbourhood patterns*, the other ones are *passive*. There are four passive neighbourhood patterns: 0000, 0011, 1100, and 1111; the twelve other patterns are active. The model has two fixed points $\mathbf{0} = 0^{\mathcal{L}}$ or $\mathbf{1} = 1^{\mathcal{L}}$; these uniform configurations represent the output of the classification process.

Definition 1. *The classification time $T(x)$ corresponds to the number of time steps needed to attain one of the configurations $\mathbf{0}$ or $\mathbf{1}$ starting from a configuration x , that is,*

$$T(x) = \min\{t \in \mathbb{N} : x^t \in \{\mathbf{0}, \mathbf{1}\}\}.$$

The average classification time of x is denoted by $\mathbb{E}\{T(x)\}$.

Definition 2. *A configuration x is well classified if $T(x)$ is finite and $x^{T(x)} = (P(x))^{\mathcal{L}}$, that is, x^T is equal to $\mathbf{0}$ (resp. $\mathbf{1}$) if the parity of x is even (resp. odd).*

In our model, we update one pair of cells at each time step. However, this introduces an artificial slowing down factor of the order of n . For example if the only possible transformation of the pair would be to replace a 10 by a 01, then

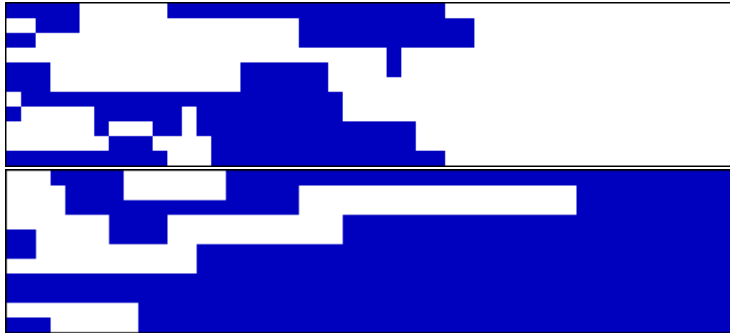


Figure 1: Two space-time diagrams showing the evolution of the system. Cells with state 0 and 1 are depicted in white and blue (or black), respectively. The successive configurations attained by the system are packed from left to right. (top): the initial number of 1's is even (bottom): the initial number of 1's is odd.

a 1 surrounded by 0's would travel to the right at the average speed of $1/n$. In order to allow a fair comparison between the synchronous and asynchronous models, we prefer to handle a *rescaled* time where one time step corresponds to n random updates.

For a given ring size $n \in \mathbb{N}$, we will evaluate a rule with its *rescaled worst expected classification time* $\tau(n)$, that is, the highest value of the average rescaled classification time:

$$\tau(n) = \max_{\{x \in \mathcal{E}_n\}} \mathbb{E}\{T(x)/n\} = \frac{1}{n} \cdot \max_{\{x \in \mathcal{E}_n\}} \mathbb{E}\{T(x)\}.$$

For simplicity, we will drop the adjective *rescaled* as it will be clear from the context if we are dealing with the rescaled time (τ) or with the number of updates (T).

Note that for the sake of coherence, we only consider odd ring sizes. Indeed, if we take an even ring size, a problem might occur with the all-1 configuration: as it can be considered as an initial condition, the system should then converge to all-0, but at the same time it is supposed to be a fixed point by construction because it is an answer for the initial configurations which are classified as odd. The use of odd ring sizes allow us to avoid this paradox.

Also note that the model presented above is somehow minimal: since the system has no external memory, the parity needs to be preserved through time, which means that we cannot do less than updating cells by pairs. Here, the pairs are chosen randomly at each time step without any memory and the neighbourhood of each pair of cell is limited to the left and right neighbouring cells of the cells which are updated.

3 Properties of the balanced rule

Figure 1 depicts two sample evolutions of the system for $n = 11$ cells. These space-time diagrams were selected to display cases of rapid convergence towards a fixed point, however, this convergence takes more time in general. Let us now analyse the convergence properties of this rule and study this behaviour analytically.

Proposition 1. *Every configuration of odd size is almost surely well classified.*

Proof. The proof of this property is quite simple since $(x^t)_{t \in \mathbb{N}}$ is a Markov chain: it is sufficient to examine the reachability relationships in this chain.

First, we can observe that the application of the local rule conserves the parity of a configuration.

Second, we show that the fixed point $\mathbf{0}$ (resp. $\mathbf{1}$) is reachable from every even (resp. odd) configuration. To see why, let us consider an arbitrary configuration x different from $\mathbf{0}$ with an even number of 1's. Let us show that x can reach a configuration y where the number of 1's has decreased by two, that is, $|y|_1 = |x|_1 - 2$.

(a) This is immediate if the x contains a $\mathbf{11}$. Indeed, it necessarily contains the pattern $\mathbf{011}$, as x differs from $\mathbf{1}$, which implies that it contains either a $\mathbf{0110}$ or a $\mathbf{0111}$ and in both cases the flip on the two central 1's apply.

(b) If all the 1's of x are isolated, that is, if x does not contain a pattern $\mathbf{11}$, then we can choose an arbitrary 1 and flip the pairs $\mathbf{01}$ from the right to the left until we reach a configuration which contains a $\mathbf{11}$. Indeed, this is always possible since both $\mathbf{0010}$ and $\mathbf{1010}$ are active neighbourhoods.

By recurrence, we can show that the fixed point $\mathbf{0}$ is reachable from any even configuration. Since this is a fixed point, it implies that all the even configurations different from $\mathbf{0}$ are transient. By exchanging 0's and 1's, the same property can be established for odd configurations and the fixed point $\mathbf{1}$ (we are on a ring of odd size). These two properties imply that all the configurations will be almost surely well classified. \square

Conjecture 1. *The worst expected classification time scales quadratically with the ring size, that is, we have $\tau(n) = \Theta(n^2)$ for $n \in 2\mathbb{N} + 1$.*

The justification of this conjecture is that the density almost follows a non-biased random walk until it hits the fixed point. We say *almost* because since a configuration can reach only one fixed point (which corresponds to its parity), the evolution of the density cannot be strictly non-biased. In particular, for a ring size $n = 2k + 1$, when the number of ones is equal to $2k$, the density can only decrease.

To study more precisely the conjecture, we will proceed in two steps: (a) first we obtain a lower bound by calculating precisely the classification time of configurations from a given set, then (b) we give some evidence why it is not straightforward to establish a quadratic upper bound.

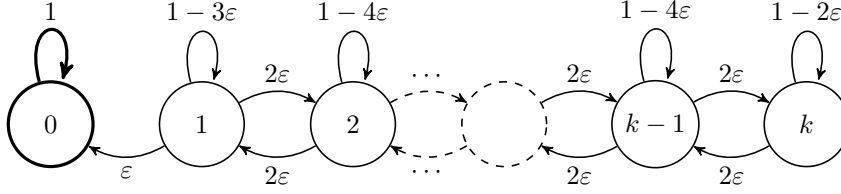


Figure 2: Representation of the Markov chain which describes the evolution of the balanced rule from configuration $1^{2i}0^{n-2i}$. The state of the Markov chain is the number of 11 pairs. The quantity ϵ is equal to $1/n$.

Lemma 1. *The worst expected classification time scales at least quadratically with the ring size, that is, we have $\tau(n) = \Omega(n^2)$, for $n \in 2\mathbb{N} + 1$.*

Proof. For a ring size $n = 2k + 1$ and $k > 3$, let us consider the set of *even* initial configurations $z_i = 1^{2i}0^{n-2i}$ for $i \in \{0, \dots, k\}$. As the rule cannot increase the number of 1-regions, all the configurations reached by the system during its evolution will be equivalent up to shift to an element of $\{z_i\}_i$, up to shifts. Let $(x^t)_{t \in \mathbb{N}}$ denote the stochastic process that describes the evolution of the system and let $X_t = \frac{\lfloor x^t \rfloor}{2}$ denote the number of 11 pairs at a given time t .

Clearly (X_t) is a Markov chain; it has one absorbing state $X_t = 0$, which corresponds to $x^t = \mathbf{0}$, the end of the classification process. Let us evaluate the average time needed to converge to this absorbing state when we start from z_i .

For $i = 1$, we have $x^t = 0000110000 \dots$ (up to shifts) and $X_t = 1$, there are three positions where an update will change the configuration, (a) the two positions with the active neighbourhoods 0001 and 1000 give $X_{t+1} = 2$, and (b) the position with the active neighbourhood 0110, which will transform x^t into $\mathbf{0}$ and the classification ends. The cases (a) and (b) happen with probabilities $2/n$ and $1/n$, respectively.

For $1 < i \leq k - 1$, there are four possibilities to modify a configuration equivalent to z_i : (a) the active neighbourhood is 0001 or 1000, in which case we increase by one $X_{t+1} = X_t + 1$, or (b) the updated neighbourhood is 0111 or 1110, in which case we decrease by one $X_{t+1} = X_t - 1$. The cases (a) and (b) both happen with probability $2/n$.

The special configuration z_k has the form 01^{2k} and in this case the only possible transformation is to turn a 11 into a 00, in which case we reach a configuration equivalent to z_{k-1} . This happens with probability $2/n$.

The transitions of the Markov chain are represented on Figure 2. Let us denote by T_i the classification time starting from a configuration which has $2i$ contiguous cells in state 1's, that is, an equivalent of z_i up to shifts.

For a state that is not a fixed point, we calculate the convergence time of this state as the weighted sum of the convergence time of the states that are reachable from this state, to which we add one, as the time needed to make the transition. This is a classical technique which is sometimes called the step forward method (see e.g. Ref. [6]). We thus find that the sequence (T_i) obeys

the following set of equations: (i) $T_0 = 0$,

$$\begin{aligned} (ii) \quad T_1 &= 1 + \frac{1}{n}T_0 + \frac{n-3}{n}T_1 + \frac{2}{n}T_2, \\ (iii) \quad T_i &= 1 + \frac{2}{n}T_{i-1} + \frac{n-4}{n}T_i + \frac{2}{n}T_{i+1} \text{ for all } i \in \{2, \dots, k-1\}, \text{ and,} \\ (iv) \quad T_k &= 1 + \frac{2}{n}T_{k-1} + \frac{n-2}{n}T_k. \end{aligned}$$

Using the rescaled time $\tau_i = T_i/n$, the system is simplified into: (i) $\tau_0 = 0$,
(ii) $1 + \tau_0 - 3\tau_1 + 2\tau_2 = 0$,
(iii) $1 + 2\tau_{i-1} - 4\tau_i + 2\tau_{i+1} = 0$ for all $i \in \{2, \dots, k-1\}$, and
(iv) $1 + 2\tau_{k-1} - 2\tau_k = 0$.

Solving this system yields $\tau_i = \frac{(i+1)k}{2} - \frac{i(i-1)}{4}$ for $i \in \{1, \dots, k\}$. In particular, for k tending to infinity, we have:

$$\tau_k \sim k^2/4, \text{ and since } k \sim n/2, \text{ this yields } \tau_k = \Theta(n^2),$$

which gives us a lower bound with a quadratic scaling in the ring size n . \square

We have established the classification time for configurations with one region, let us see what happens in the general case.

If we take an even configuration with only two 1-regions, if the length of these regions is even, then it is not difficult to show that the convergence is also quadratic. Indeed, the length of these regions will increase by two or decrease by two and this length will always remain even. So each region will eventually disappear and the system will converge to the all-zero fixed point, as it should. Using martingale techniques, one can show that this will happen in average in a time which is quadratic in the ring size [6]. However, a problem arises with regions of odd size, especially those regions which are separated by an odd number of cells. Consider for instance the configuration $0^k 0011111000000011100^k$, with k large enough. The central region of 0's has a length of 7. The two 1-regions can grow or shrink their size, but, unfortunately, they cannot merge in the central region because the length of this region always remains odd. So when the two regions will be separated by only one 0, they will "bounce" against each other and reduce their size. So the convergence can only happen if the 1-regions merge on the other size of the ring, or if an odd region shrinks to a size of 1 and then moves left or right, which will make the central region have an even size.

We believe that this effect, although slowing down the process, will not modify qualitatively its dynamics, and this why we think that it has a quadratic convergence time. The positive side of the "bouncing phenomenon" is that it lead us to consider another method that avoids this inconvenient. This will be exposed in the next section

4 Joint classifications of even and odd configurations

In the case where we can duplicate the state of the initial configurations that we want to classify, we now propose to use *two separate systems* to determine the class of the input configuration: the first system will converge only if the configuration is even and the second if the configuration is odd. We thus modify the rule above and define, informally, the *even classifier* (resp. the *odd classifier*) as a modification of the balanced rule where the transition $00 \rightarrow 11$ (resp. $11 \rightarrow 00$) are inhibited. In other words, in the even classifier, 1-regions shrink but never expand, and equivalently for 0-regions in the odd classifier. This will make the even and odd classifiers respectively converge only for even and odd configurations, otherwise they will evolve without reaching a fixed point.

Formally, for $q \in \{0, 1\}$, for a configuration $x \in \mathcal{E}_n$, we take the same definitions as above, with U the updating method; we keep:

$$\forall i \in \mathcal{L}, i \notin U_t \implies x_i^{t+1} = x_i^t$$

and, for $\{i, i+1\} = U(t)$, we now define that,

if (a) $x_{i-1}^t \neq x_i^t$ or $x_{i+1}^t \neq x_{i+2}^t$, and (b) $(x_i^t, x_{i+1}^t) \neq (q, q)$ then

$$(x_i^{t+1}, x_{i+1}^{t+1}) = (1 - x_i^t, 1 - x_{i+1}^t),$$

otherwise

$$(x_i^{t+1}, x_{i+1}^{t+1}) = (x_i^t, x_{i+1}^t).$$

The even and odd rules are then defined for $q = 0$ and $q = 1$, respectively ; the even rule makes the number of 1's a non-increasing function of time while the odd rule makes it non-decreasing.

Theorem 1. *For the even (resp. odd) rule, the worst expected classification time of the even (resp. odd) configurations of odd size scales quadratically with the ring size, that is, $\tau(n) = \Theta(n^2)$.*

We prove this claim in three steps: (a) prove the convergence of the configurations, (b) establish a lower bound by calculating precisely the classification time for a given set of configurations, and then (c) give a quadratic upper bound with martingale techniques.

Proof. Without loss of generality, let us show that the even rule correctly classifies the even configurations.

(a) Convergence. The number of 1's can only decrease by pairs and regions of 1's cannot appear, nor can they merge if they are not reduced to isolated 1's. This means that after a finite number of steps, each region of 1's with an even length will disappear, and the region of 1's with an odd length will form an isolated 1. We will thus attain a configuration formed of an even number of isolated 1's. Then, it can be seen that the random movements of these 1's will

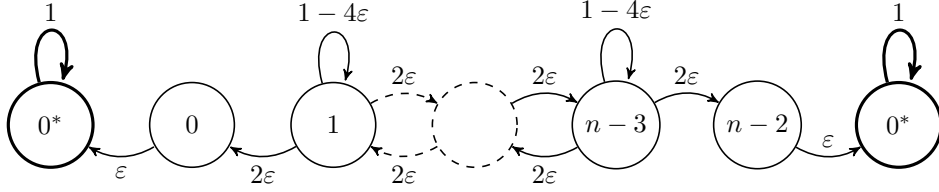


Figure 3: Representation of the Markov chain which describes the evolution of the even rule from a configuration $z_i = 10^i 10^{n-2-i}$. The state of the Markov chain represents the value of i and 0^* is the homogeneous configuration all-zero. This state has been duplicated for the sake of readability. The quantity ϵ is equal to $1/n$.

create pairs of 1's, which will progressively disappear until the systems reaches the fixed point $\mathbf{0}$.

(b) Lower bound. For a ring size $n = 2k + 1$ and $k \geq 2$, consider the configurations $z_i = 10^i 10^{i-1}$ as the initial conditions of the system. The configurations z_i contain two isolated 1's. As long as these 1's are not adjacent, they will move with a non-biased random walk.

To determine their movements, we choose to keep track of their respective position and call them the first and second 1; we then define X_t as the number of cells between the first and second 1. (X_t) is a random variable that takes its values between 0 and $n - 2$; let T_k be the number of steps needed to attain one of the boundaries 0 or $n - 2$ starting from z_k . Note that once the pattern 11 has appeared, nothing can happen except the disappearance of the pair of 1, which happens with a probability $1/n$. It thus takes n updates in average to reach $\mathbf{0}$, that is, we need only one time step with the rescaling time.

The transitions of the Markov chain are represented on Figure 3. Using the step forward method, we obtain a set of equations similar to the ones of asynchronous Elementary Cellular Automata (ECA) 178 [6]: (i) $T_0 = 0$; (ii) $T_{n-2} = 0$ and

$$(iii) \quad T_i = 1 + \frac{2}{n}T_{i-1} + \frac{n-4}{n}T_i + \frac{2}{n}T_{i+1} \text{ for all } i \in \{1, \dots, n-3\};$$

which gives us $T_i = \frac{n}{4}i(n-2-i)$.

This yields a rescaled time $\tau_i = T_i/n = 4i(n-2-i)$, with a maximum time of $\tau_k = 4k(n-2-k) = \Theta(n^2)$. Since the rescaled classification time of z_k is equal to $\tau_k + 1$, it thus scales quadratically with the ring size.

Note that other equivalent possibilities exist to calculate the convergence time of this system: for instance, one may simply model the distance between the two 1's or use a transformation of the configuration (see below).

(c) Upper bound. As a first step, starting from an arbitrary even configuration, we bound the time needed to reach an *archipelago*, that is, a configuration where all the 1's are isolated. Let us denote by τ_i the random variable that it takes for the i -th region to shrink to a single 1. It is easy to

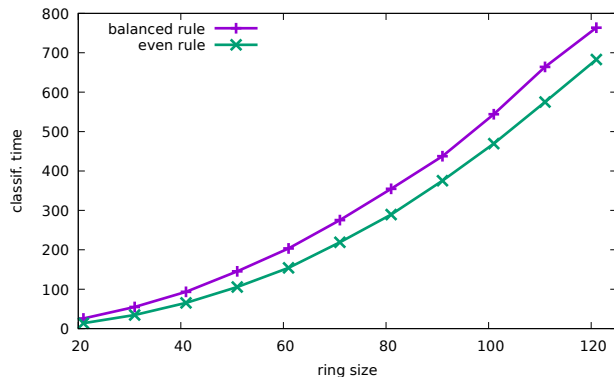


Figure 4: Classification time as function of the ring size. Statistical measures for 100 000 samples.

remark that the expectation of τ_i is linear in the ring size n and that, since $\mathbb{E}\{\max(\tau_i)\} \leq \sum_i \mathbb{E}\{\tau_i\}$, and since the number of regions is linear, the time needed to reach an archipelago is upper-bounded by a quadratic function (in n).

Let us now consider an evolution (x^t) from an archipelago $x = x^0$ formed of $k \in 2\mathbb{N}$ isolated 1's. We can map x^t into a configuration y^t according to the following operation: we build y^t by reading x^t from left to right and consider that the 1's in x^t encode a change of state in y^t , while the 0's encode that the same state is kept. In other words, the state of k -th cell of y^t is the parity of the number of 1's contained in the k first cells of y ; formally: $y_k^t = |\{j, x_j^t = 1\}| \bmod 2$.

It can be checked that the dynamics of (y^t) is as follows: the frontiers of the 1-regions (and the 0-regions) move left or right with probability $1/n$ and the isolated 0's and 1's disappear with probability $1/n$ (this correspond to updating a neighbourhood 0110 in x). This means that it is equivalent to the dynamics of ECA 178 [6]. As a consequence, the second step has an average classification time which scales quadratically with n and the whole process also scales quadratically. \square \square

We thus have a classifying system composed of two sub-systems: one will almost surely answer if the configuration is even, the other one if the configuration is odd. This means that the classification operation should be stopped as soon a consensus is established in one of the two systems. Note that one may also use several copies of the even and odd classifiers and stop the search as soon as a consensus is established in one of the classifiers.

5 Numerical experiments

In order to have a more precise estimation of how the classification time varies with the ring size, we carry out some numerical experiments. Here, we are no longer interested in the worst expected classification time but in the *mean* expected classification time, which means that we measure an average of averages.

We repeat the experiment which consists in drawing 100 000 times an initial condition and measuring classification time. For the classification with the balanced model, the initial configurations are drawn uniformly in the set \mathcal{E}_n . For the two-system experiment with the *even and odd rules*, we simply draw a configuration x uniformly as before, but apply it to the *even rule only* with the condition that if this configuration is odd, we take \bar{x} , the conjugate of x where each cell has its state flipped, as an initial configuration. For reasons of symmetry, since only one of the two systems can converge, this is equivalent to running the two systems on x and taking the classification time of the first system which converges.

Figure 4 displays the mean classification time as function of the ring size. As expected, the classification time is lower for the even rule than for the balanced rule, but the difference is not overwhelming. It was also verified that the measures are in a good agreement with a quadratic scaling: fitting a second-degree polynomial to the data yields a curve whose position is visually almost indistinguishable from the actual measures.

6 Perspectives

This study demonstrated the possibility to solve the cellular automata parity problem with a simple rule where updates occur by pairs at random times and positions, and with a local rule which is expressed in two lines. We showed how to analyse formally the classification time of the models with Markov chains and analytical techniques inspired by the study of Elementary Cellular Automata with fully asynchronous updating [6].

The only deterministic rule known so far has a neighbourhood size of 9 (a radius-4 rule) and requires a perfect synchronisation of the transitions. The Elementary Cellular Automaton 150 used with an alternation of the cells with even and odd positions is also a good deterministic solution, but still requires a global synchronisation of the transitions. In contrast, the proposed stochastic rule makes the updating procedure totally local; however, it has a slower operation mode, as its average time of classification scales quadratically in the ring size. In return, its implementation is potentially simpler as only a local synchronisation of the transitions by pairs is required.

The asynchronous nature of the transitions may facilitate the use of a truly decentralised computing device as it eliminates the need for a clock that sends a signal to trigger the computation in the cells. For example, such systems could be implemented with Brownian cellular automata, where tokens move freely on a line [7]. In such systems, the only requirement would be to guarantee that the

two cells make their transitions together in order to preserve the global parity of the configurations. We leave for future work a more precise analysis of these processes but one can assume that such systems could be made robust to various kind of perturbations, such as miscalculating the next state or misreading the state of the neighbours. It is also worth considering more general cases with interactions embedded on a graph, with or without spatial locality constraints.

More generally, this research goes along the lines that aim to develop robust computing schemes, where noise is considered as an ally rather than an obstacle to overcome. Various authors have demonstrated that stochastic cellular systems can acquire suitable properties for computing in a purely decentralised way. For example, self-correcting discrete lines [8], stochastic solutions to the density classification problem [9, 10, 11] or to the synchronisation problem [12] have been proposed. It is probably too early to say if a new trend is emerging but, for sure, this field of research opens some promising paths out of the classical computing paradigm.

Acknowledgements

The author wishes to express his sincere gratitude to Barbara Wolnik (univ. of Gdansk, Poland) and Irène Marcovici (univ. of Rouen, France) for their precious advice and remarks. The comments of the anonymous reviewers were a great help to improve the manuscript.

References

- [1] Moshe Sipper. “Computing with cellular automata: Three cases for nonuniformity”. In: *Physical Review E* 57 (3 1998), pp. 3589–3592. DOI: 10.1103/PhysRevE.57.3589.
- [2] Heather Betel, Pedro P. B. de Oliveira, and Paola Flocchini. “Solving the parity problem in one-dimensional cellular automata”. In: *Natural Computing* 12.3 (2013), pp. 323–337. DOI: 10.1007/s11047-013-9374-9.
- [3] Eurico L.P. Ruivo and Pedro P.B. de Oliveira. “A perfect solution to the parity problem with elementary cellular automaton 150 under asynchronous update”. In: *Information Sciences* 493 (2019), pp. 138–151. DOI: 10.1016/j.ins.2019.04.045.
- [4] Eurico L.P. Ruivo, Pedro Paulo Balbi, and Kévin Perrot. “An asynchronous solution to the synchronisation problem for binary one-dimensional cellular automata”. In: *Physica D: Nonlinear Phenomena* 413 (2020), p. 132554. DOI: 10.1016/j.physd.2020.132554.
- [5] Pedro Paulo Balbi, Eurico Ruivo, and Fernando Faria. “Synchronous solution of the parity problem on cyclic configurations, with elementary cellular automaton rule 150, over a family of directed, non-circulant, regular graphs”. In: *Information Sciences* 615 (2022), pp. 578–603. DOI: 10.1016/j.ins.2022.10.045.

- [6] Nazim Fatès. “Asynchronous Cellular Automata”. In: *Encyclopedia of Complexity and Systems Science*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018, pp. 1–21. ISBN: 978-3-642-27737-5. DOI: 10.1007/978-3-642-27737-5_671-2.
- [7] Ferdinand Peper, Jia Lee, and Teijiro Isokawa. “Brownian Cellular Automata”. In: *Journal of Cellular Automata* 5.3 (2010), pp. 185–206.
- [8] Damien Regnault and Éric Rémila. “Lost in self-stabilization: A local process that aligns connected cells”. In: *Theoretical Computer Science* 736 (2018), pp. 41–61. DOI: 10.1016/j.tcs.2018.02.015.
- [9] Henryk Fukś. “Nondeterministic density classification with diffusive probabilistic cellular automata”. In: *Physical Review E* 66.6 (2002), p. 066106. DOI: 10.1103/PhysRevE.66.066106.
- [10] Martin Schüle, Thomas Ott, and Ruedi Stoop. “Computing with Probabilistic Cellular Automata”. In: *ICANN '09: Proceedings of the 19th International Conference on Artificial Neural Networks*. Limassol, Cyprus: Springer-Verlag, 2009, pp. 525–533. ISBN: 978-3-642-04276-8.
- [11] Nazim Fatès. “Stochastic Cellular Automata Solutions to the Density Classification Problem - When Randomness Helps Computing”. In: *Theory of Computing Systems* 53.2 (2013), pp. 223–242. DOI: 10.1007/s00224-012-9386-3.
- [12] Nazim Fatès. “Remarks on the cellular automaton global synchronisation problem: deterministic versus stochastic models”. In: *Natural Computing* 18.3 (2019), pp. 429–444. DOI: 10.1007/s11047-018-9683-0.