An asynchronous cellular system that solves the parity problem

Nazim A. Fatès

An asynchronous cellular system that solves the parity problem

Nazim Fatès

December 12, 2022

Abstract

We present stochastic solutions to the cellular automata parity problem. The model is an interacting particles system where cells are updated by pairs, randomly chosen at each time step. We analyse the convergence properties of two rules and show that they possess the required properties to classify the parity of the initial configurations. We present a formal analysis of the classification time, as well as numerical simulations, to establish that the classification time scales quadratically with the number of cells.

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 as a grid of arbitrary dimension, and which interact according to a local rule. The locality is imposed as the cells change their state ignoring the whole state of the system and knowing only what happens at a given distance. 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 at using 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 about computational processes in a different way from the traditional sequential deterministic machines. While a great part of this research has been devoted to understanding the different properties of models or classes of models, one can also be interested in synthesising a given behaviour in order to solve a problem, that is, to solve an *inverse problem*.

The parity problem is an inverse problem where one is asked to determine the parity of the number of `1`'s contained in an initial configuration of a one-dimensional cellular system with binary states. More precisely, it is a consensus problem: if the parity of the initial configuration is even, then the system should

converge to a stable homogeneous configuration where all the cells agree on 0 while the convergence point should be formed of all 1's if this number is odd [1].

A deterministic solution of the problem was found 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 inside the system itself, and without the use of additional states. The rule they proposed has radius of interaction of range four, which means that each cells 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 156-digit code or, more compactly, as a sequence of 10 sub-rules with wildcards, that is, states than can take indifferently any value.

In a second step, Ruivo and de Oliveira proposed to solve the parity problem with a simple rule, namely the Elementary Cellular Automaton 150, with an 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.

In this work, we revisit this problem with a model which uses *independent random updates* of pairs of cells. More precisely, it falls in the category of *interacting particle systems*: 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 and then, in a second step, we will analyse formally the convergence properties of this rule. We also present an improvement of the classification process by using simultaneously two different rules. We then present some numerical simulations 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 and the set of cells is denoted by $\mathcal{L} = (\mathbb{Z}/n\mathbb{Z})$.

A *configuration* is the state of the system at a given time; the state of each cell is an element of $Q = \{0, 1\}$ and the set of configurations is denoted by $\mathcal{E}_n = Q^{\mathcal{L}}$. For a configuration $x \in \mathcal{E}_n$, the state of cell $i \in \mathcal{L}$ is denoted by x_i. 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 $P(x) = |x|_1 \mod 2$, 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 inclusion.

We now present our 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; the initial term is such that $x^0 = x$. We now define the *balanced rule* as:

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

In words, the cells that are not updated keep their state, and, for $\{i, i+1\} = U(t)$,

if $x_{i-1}^t \neq x_i^t$ or $x_{i+1}^t \neq x_{i+2}^t$ then: $\quad (x_i^{t+1}, x_{i+1}^{t+1}) = (1 - x_i^t, 1 - x_{i+1}^t)$,

otherwise: $\quad (x_i^{t+1}, x_{i+1}^{t+1}) = (x_i^t, x_{i+1}^t)$.

In words, the state of the two cells selected for an update will be flipped only if the state of the left cell updated and its left neighbour differ or if the state of the right cell updated and its right neighbour differ. These patterns of four states which trigger a flip are called *active neighbourhoods*. The pair is not modified if the states of the four cells form one the following four neighbourhood patterns: `0000`, `0011`, `1100`, or `1111`.

Definition 1. *The classification time $T(x)$ is the random variable which corresponds to the time needed to attain one of the two fixed points $\mathbf{0} = \mathtt{0}^{\mathcal{L}}$ or $\mathbf{1} = \mathtt{1}^{\mathcal{L}}$ 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 $x \in \mathcal{E}_n$ is the expectation of $T(x)$; it 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 some 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 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:

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

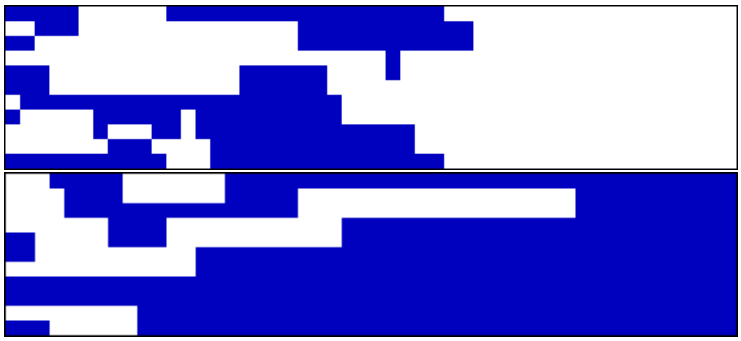

Figure 1: Two space-time diagrams showing the evolution of the system. Cells with state `0` and `1` are depicted in white and blue, respectively. The successive configurations attained by the system are packed from left to right. (top): the initial number of `1`'s is even (bottom): the initial number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

For the sake of simplicity, we will simply 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 show a case of rapid convergence towards a fixed point, however, most of the times, this convergence takes more time. 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. Indeed, the evolution of the system $(x^t)_{t\in\mathbb{N}}$ is described by a Markov chain and it is sufficient to examine the reachability relationships of this chain.

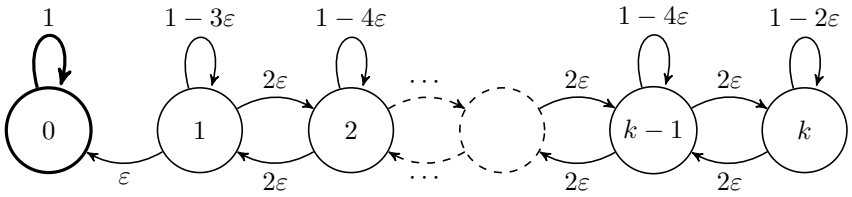

Figure 2: Representation of the Markov chain which describes the evolution of the balanced rule from configuration $\mathtt{1}^{2i}\mathtt{0}^{n-2i}$. The state of the Markov chain is the number of `11` pairs. The quantity ϵ is equal to $1/n$.

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: $|x|_{\mathtt{1}} = 2k, k > 0$. Let us show that this configuration can reach a configuration where two `1`'s have been transformed in `0`'s. This is immediate if the x contains a `11`. Indeed, it necessarily contains the pattern `011`, as x differs from $\mathbf{1}$, which implies that it contains either a `0110` or a `0111` and in both cases the flip on the two central `1`'s apply. If all the `1`'s of x are isolated, that is, if x does not contain a pattern `11`, then we can choose an arbitrary `1` and flip the pairs `01` from the right to the left until we reach a configuration which contains a `11`. Indeed, this is always possible since both `0010` and `1010` are active neighbourhoods.

In summary, the fixed point $\mathbf{0}$ is reachable from any even configuration and all the other even configurations, 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. □

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 as 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, we take the case of an even initial configuration, for a ring size $n = 2k + 1$, when the number of ones is equal to $2k$, the density can only decrease. This special case is however not a problem, as we are going to see in the following.

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.

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 $z_i = \mathtt{1}^{2i}\mathtt{0}^{n-2i}$ for $i \in \{0, \ldots, k\}$. As the rule cannot increase the number of `1`-regions, all the configurations reached by the system during its evolution will be equivalent to some z_i, up to shifts. Let $(x^t)_{t\in\mathbb{N}}$ denote the stochastic process that describes the evolution of the system and let $X_t = |x^t|_1/2$ denote the number of `11` pairs at a given time t.

Clearly (X_t) is 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 by using the step forward method.

For $i = 1$, we have $x^t = \mathtt{0000110000}\ldots$ (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 $\mathtt{01}^{2k}$ 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. Using the step forward method (see e.g. Ref. [4]), we find that the sequence (T_i) obeys the following set of equations: (i) $T_0 = 0$ and

$$(ii)\ T_1 = 1 + \frac{1}{n}T_0 + \frac{n-3}{n}T_1 + \frac{2}{n}T_2,$$

$$(iii)\ T_i = 1 + \frac{2}{n}T_{i-1} + \frac{n-4}{n}T_i + \frac{2}{n}T_{i+1} \text{ for all } i \in \{2, \ldots, k-1\}, \text{and,}$$

$$(iv)\ T_k = 1 + \frac{2}{n}T_{k-1} + \frac{n-2}{n}T_k.$$

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 $i \in \{2, \ldots, k-1\}$, 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 $i \in \{1, \ldots, k\}$. 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. □

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 [4]. 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 $\mathtt{0}^k\mathtt{001111100000011100}^k$, 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 remain odd. So when the two regions will be separated by only one 0, they will "bounce" against 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, though 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 suggested us to consider a slightly different way to tackle the problem that avoids this inconvenient.

4 Separate 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 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 $\mathtt{00} \to \mathtt{11}$ (resp. $\mathtt{11} \to \mathtt{00}$) are inhibited. In other words, in the even and odd classifier, 1-regions and 0-regions are respectively shrunk but never extended. 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 $q \in \{\mathtt{0}, \mathtt{1}\}$, 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 $q = \mathtt{0}$ and $q = \mathtt{1}$, 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, we have* $\tau(n) = \Theta(n^2)$.

We will prove these assertion 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. **(a) Convergence.** Without loss of generality, let us show that the even rule correctly classifies the even configurations. The number of 1's can only decrease by pairs and regions of 1's cannot merge (nor appear). 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 en 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 create pairs of 1's, which will progressively disappear until we reach the fixed point $\mathbf{0}$.

(b) Lower bound. For a ring size $n = 2k + 1$ and $k \geq 2$, consider the configurations $z_i = \mathtt{10}^i\mathtt{10}^{i-1}$ 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_i. 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 [4]: (i) $T_0 = 0; (ii)$ $T_{n-2} = 0$ and

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

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

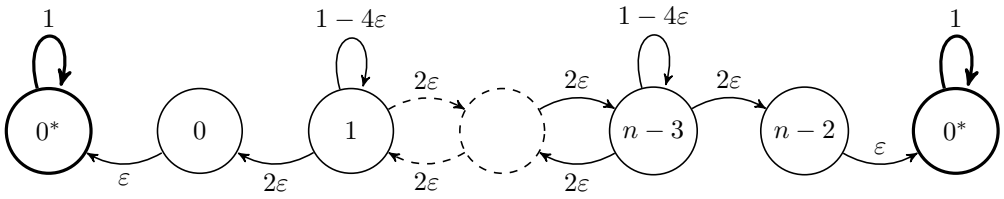

Figure 3: Representation of the Markov chain which describes the evolution of the even rule from a configuration $z_i = \mathtt{1}\mathtt{0}^i\mathtt{1}\mathtt{0}^{n-2-i}$. The state of the Markov chain represents the value of i and 0^* is the homogeneous configuration all-zero. This state has been duplicated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

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. Without loss of generality, let us consider a configuration $x = x^0$ which is even and which is formed of $k \in 2\mathbb{N}$ isolated 1's in x. 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 will 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: $y^t_k = |\{j, x^t_j = \mathtt{1}\}| \mod 2$.

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 [4]. As a consequence, the second step has an average classification time which scales quadratically with n and the whole process also scales quadratically. □

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 our search as soon as a consensus is established in one of the classifiers.

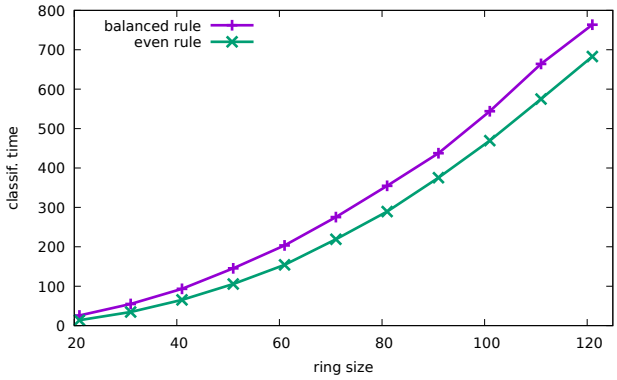

Figure 4: Classification time as function of the ring size. Statistical measures for 100 000 samples.

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 time of classification but in the mean expected time of classification, 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 the time of classification. 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 if this configuration is odd, we take $\bar{x}$, the conjugate of x where each cell has its state flipped, as an initial configuration, and apply it to the *even rule only*. For reasons of symmetry, one can easily be convinced that this is (statistically) equivalent to running the two systems on x and taking the classification time of the first system which shows a convergence.

Fig. 4 displays the mean classification time as function of the ring size. As expected, the classification time if 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: by fitting a second-degree polynomial to the data, one obtains a theoretical curve whose position is visually almost indistinguishable from the actual measures.

6 Perspectives

This note demonstrated the possibility to solve the cellular automata parity problem with a rule that is simple in the sense that the updates occur by pairs

at random times and random positions 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 martingales, by using techniques which are quite similar to those used for studying Elementary cellular automata with fully asynchronous updating [4].

The only deterministic rule known so far has a neighbourhood size of 9 (it is 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 stochastic rule we proposed makes the updating procedure totally local ; however, it has slower operation mode, as its average time of classification scales quadratically in the ring size. In return, its implementation can be 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 there is no longer the need to have a clock which distributes a signal that triggers the computation in the cells. For example, such systems may be applied by implementing Brownian cellular automata, where tokens freely move on a line [5]. 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.

More generally, this research goes along the lines which try 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 a in purely decentralised way. For example, self-correcting discrete lines [6], stochastic solutions to the density classification problem [7, 8, 9] or to the synchronisation problem [10] 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 excursions out of the classical deterministic sequential computing paradigm.

Acknowledgements The authors wishes to express his sincere gratitude to Barbara Wolnik and Irène Marcovici for their precious remarks.

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. ISSN: 0020-0255. DOI: 10.1016/j.ins.2019.04.045.

[4] 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.

[5] Ferdinand Peper, Jia Lee, and Teijiro Isokawa. "Brownian Cellular Automata". In: *Journal of Cellular Automata* 5.3 (2010), pp. 185–206.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] Nazim Fatès. "Remarks on the cellular automaton global synchronisation problem: deterministic versus stochastic models". In: *Natural Computing* 18.3 (2019), pp. 429–444.