Results and conjectures on the Sandpile Identity on a lattice

Arnaud Dartois, Clémence Magnien

Discrete Mathematics and Theoretical Computer Science **AB(DMCS)**, 2003, 89–102

Results and conjectures on the Sandpile Identity on a lattice

Arnaud Dartois [1] and Clémence Magnien [1]

[1] *LIX, École Polytechnique, 91128 Palaiseau Cedex, France*

In this paper we study the identity of the Abelian Sandpile Model on a rectangular lattice. This configuration can be computed with the burning algorithm, which, starting from the empty lattice, computes a sequence of configurations, the last of which is the identity. We extend this algorithm to an infinite lattice, which allows us to prove that the first steps of the algorithm on a finite lattice are the same whatever its size. Finally we introduce a new configuration, which shares the intriguing properties of the identity, but is easier to study.

Keywords: Abelian sandpile, Identity, Burning algorithm, Infinite lattice, Toppling

1 Preliminaries

The abelian Sandpile Model was first introduced in (BTW87), and has been widely studied as one of the simplest models for Self-organized criticality (SOC) (KLGP00; Bak97).

This model is defined on a rectangular lattice, in the squares of which are disposed a certain number of *grains* of sand. The evolution rule, called the *toppling* rule, is the following: if a square contains at least four grains, it topples, losing four grains and giving one to each of its neighboring squares. If the toppled square is on the edge of the lattice, grains fall out of the lattice and are lost.

Given a lattice, we call *configuration* any repartition of a number (positive or negative) of grains in the squares of this lattice. Given any configuration u on a $p \times q$ lattice, we denote by $u_{i,j}$ the number of grains in the square indexed by (i, j). A given configuration is said to be positive if each square contains zero or more grains. A positive configuration is said to be *stable* if no toppling is possible in this configuration, *i.e.* if no square contains more than three grains.

We call *avalanche* any sequence of topplings. Since grains can be lost at the bordering squares, any avalanche is of finite length. We call *maximal* avalanche any avalanche that cannot be extended. Any maximal avalanche leads to a stable configuration. It can be checked easily that, given any unstable configuration u, any maximal avalanche leads to the same stable configuration v (Dha99). We then say that v is obtained by the *relaxation* of u, and we denote it by $v = \hat{u}$.

We call *forced toppling* of a square the action of toppling the square whatever the number of grains it contains. Notice that when a square is toppled by force, the number of grains in it may become negative. We call *reverse toppling* of a square the action of adding four grains in the square, and removing one grain from all the neighboring squares. Two configurations, positive or negative, are said to be *equivalent* if they can be reached one from another by a sequence of topplings and reverse topplings. When all squares of the lattice are toppled by force, the configuration evolves in the following fashion: all corner squares lose two grains (they have lost four by their toppling, and regained two by the toppling of their two neighboring squares), all edge squares lose one grain, and the number of grains in the other squares remains the same. We denote by β the configuration obtained from the empty lattice by the reverse toppling of all squares: in β each corner square contains two grains, each edge square contains one, and the other squares are empty. Notice that, if starting from a given configuration u one reverse-topples all squares of the lattice, the obtained configuration is $u+\beta$.

Let us consider the system evolving from the empty lattice in the following two steps process:

1. Addition of grains: choose a square at random, and add a grain into it.
2. Relaxation of the configuration.

Some stable configurations appear infinitely often in this sequel, they are called *recurrent* configurations (DM90).

The recurrent configurations have been extensively studied (Dha90; CR00; DRSV95). It has been shown in (Dha90) that there is one and exactly one recurrent configuration equivalent to any given configuration u. We denote this configuration by $\overline{u}$. Also, if we define the sum $u \oplus v$ of two recurrent configurations as the configuration obtained by the relaxation of $u+v$ (*i.e.* $u \oplus v = \widehat{u+v}$), then the set of the recurrent configurations is a group for addition (Dha90).

The identity I of this group is the configuration such that the configuration obtained from any recurrent configuration u by the relaxation of $u+I$ is equal to u. The identity on the 76×76 and 77×77 lattices are presented in Figure 1.

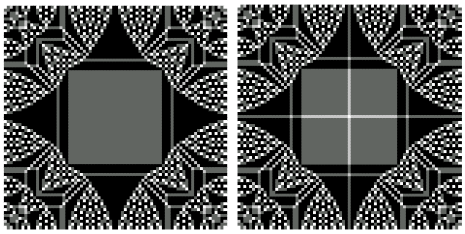

Fig. 1: The identity on the 76×76 and 77×77 lattices

The identity configuration has raised a great amount of interest, in particular for its complicated fractal structures (MN99; Cre96; BR00; DRSV95). We mention here two remarkable observations on this configuration. First, for the identity in the central area of a $2p\times2p$ lattice, there is a whole square in the central area of the lattice where all squares contain exactly two grains (see Figure 1). One other remarkable aspect of this configuration is that the identity configuration $I^{2p+1,2p+1}$ of the $(2p+1)\times(2p+1)$ lattice seems to be related in a very simple way to that on the $2p\times2p$ lattice. Indeed, if we divide the configuration $I^{2p,2p}$ into four equal squares and pull them apart by one lattice spacing so as to leave a cross in the middle, we obtain $I^{2p+1,2p+1}$, provided the number of grains in the cross are properly set. In proper notation, if

$$I^{2p,2p} = \begin{pmatrix} B_1 & B_2 \\ B_3 & B_4 \end{pmatrix} (B_i \text{ are } p\times p \text{ blocks}),$$

where the four blocks B_i are related by the symmetry transformations of the square, then

$$I^{2p+1,2p+1} = \begin{pmatrix} B_1 & R_1 & B_2 \\ R_2 & 0 & R_3 \\ B_3 & R_4 & B_4 \end{pmatrix} (R_i \text{ are } 1 \times p \text{ or } p \times 1 \text{ rows or columns}).$$

There are a number of ways to characterize recurrent configurations. We present some of them now:

Proposition 1.1 *(MD92) A stable configuration u is recurrent if and only if the configuration obtained from u in the following way:*

1. *reverse toppling of all squares;*
2. *relaxation of the configuration*

is equal to u. In other words, u is recurrent if and only if $\widehat{u+\beta} = u$.

Moreover, during the relaxation, each square is toppled exactly once.

If, during the relaxation of the configuration $u+\beta$, some squares of a region X are not toppled, then u is not recurrent. Moreover, any configuration v such that, for all $(i,j) \in X$, $v_{i,j} = u_{i,j}$ is not recurrent (MD92). For instance, a configuration that contains a 2×2 square, each square of which contains at most one grain, is not recurrent.

The identity can be computed by the following algorithm, called *burning* algorithm (Dha90). Starting from the empty lattice, we obtain a sequence of stable configurations $(u^n)_{n \geq 0}$ in the following way: $u^{i+1} = \widehat{u^i+\beta}$. The configuration u^n such that $u^n = u^{n+1}$ is then the identity.

When studying the sequence of configurations given by the burning algorithm, we have noticed that the algorithm goes through two distinct phases. In the first phase, the size of the considered lattice does not seem to have any influence on the obtained configurations (see Figure 2).

During this phase, only the regions located in the corners of the lattice are modified in a significant way. During the progress of the algorithm, the size of these modified regions grows. As long as the regions in two adjacent corners have not met in the middle of the lattice, the configurations in these regions seem to be the same whatever the size of the lattice. Moreover, these configurations present strong similarities with the identity configurations on square lattices. In particular, they present the same fractal structure.

The behavior of the algorithm changes when the modified regions in the corners grow enough for them to meet. In this second phase, the central region keeps on evolving until it becomes the central square, each square of which containing two grains, noticed in every identity configuration in a $2p \times 2p$ lattice. Surprisingly, the region surrounding this central zone remains constant, and does not change configuration until the end of the computation. Figure 3 illustrates the second phase of the algorithm.

2 The infinite model

We attempt here to isolate the first phase of the burning algorithm by introducing an infinite extension of the model. The configurations obtained during this phase presenting many similarities with the identity

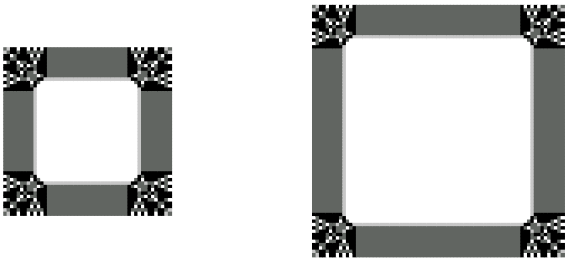

Fig. 2: The 100-th step of the burning algorithm on the lattices of size 50×50 and 75×75

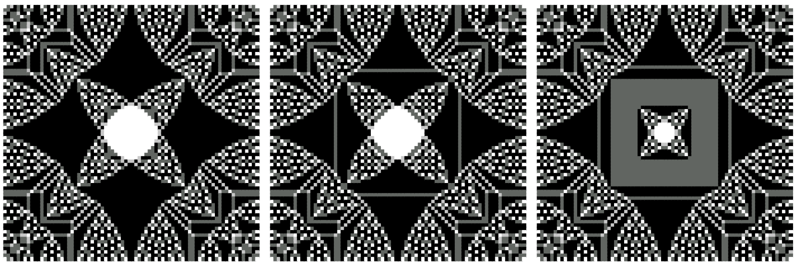

Fig. 3: The 900-th, 910-th and 960-th steps of the burning alogrithm on the 76×76 lattice

configurations on lattices of different sizes, this study enables us to better understand the structure of this configuration.

To study the first phase of the algorithm, the size of the lattice must be large enough so that the modified regions in the corners do not meet. This has led us to consider an infinite lattice, of which we observe the upper-left corner. Each square is indexed by a 2-uple of positive integers. The corner square is indexed by $(1,1)$, and the edge squares are indexed by $(1,l)$ or $(l,1)$. The configuration β obtained by the reverse toppling of all squares in this lattice has two grains on the corner square, and one grain on each edge square. Figure 4 presents this lattice in configuration β.

Starting from the empty lattice, we will study the burning algorithm, and the way the configuration evolves. Notice that, on the contrary of what happens on a finite lattice, the algorithm never ends on the infinite lattice. We will study the (infinite) sequence $(\ddot{t})_{n\geq 0}$ of configurations obtained in the following way:

1. Reverse-topple all squares (add two grains on the corner square, and one grain on each edge square);

2. relax the configuration.

At each step, the configuration is symmetrical with respect to the diagonal of the lattice. Indeed, the empty configuration is symmetrical, and neither the reverse toppling of all vertices nor the relaxation of the configuration break this symmetry. We will therefore consider only the lines of the configurations, and the values of the column can be obtained by symmetry.

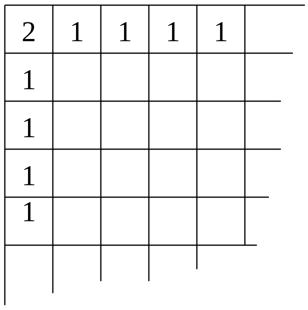

Fig. 4: The infinite lattice in configuration β

We will say that the configuration is *regular* starting from rank k if, for each line, the squares starting from the k-th have the same value.

Proposition 2.1 *For each $n \geq 0$, there exists $K_n \leq n$, such that the configuration u^n obtained from the infinite empty lattice after n steps of the burning algorithm is regular starting from rank K_n,* i.e. *if for all $i, i' > K_n$, j, $u^n_{i,j} = u^n_{i',j}$.*

Proof : We show the result by recurrence on $n \geq 0$.
For $n = 0$, it is obvious.

Let us suppose that the result is true at rank n. The configuration u^{n+1} is obtained by the relaxation of the configuration $u^n + \beta$. During this process each square is toppled at most once (Proposition 1.1).

We will show that, during this process if any square (K_n+1, j) of column K_n+1 can be toppled, then we can topple all squares (i, j) with $i \geq K_n+1$, *i.e.* the whole row j starting from column K_n+1. Let us consider the squares of column K_n+1. When no square of this column has toppled, then the result is true. Suppose that t topplings have taken place in the column K_n+1, and that for each of them we have toppled the whole row of the lattice, starting from column K_n+1. This means that any square of column K_n+1 that did not topple contains the same number of grains or one more than its neighbor of column K_n+2, if this one didn't topple (the square of column K_n+1 contains one grain more if its neighbor of column K_n has toppled). Suppose that a particular square (K_n+1, j) becomes unstable. It contains at least 4 grains. If the square (K_n+2, j) has not yet toppled, it contains at least one grain less than (K_n+1, j), *i.e.* it contains at least 3 grains, so that it is unstable after the toppling of the square (K_n+1, j). It means that if we topple (K_n+1, j) we can topple (K_n+2, j). Applying the same argument with $k(n) = K_n+1$, we can show that we can topple all squares (i, j) with $i \geq K_n+1$.

Hence, the topplings of the regular part can be done row by row. In particular, it implies that after the relaxation, the column starting from K_n+1 are identical, *i.e.* $K_{n+1} \leq K_n+1 \leq n+1$. Hence the result is shown. $\square$

Definition 2.2 *We denote by $k(n)$ the smallest integer such that the configuration u^n is regular starting from rank $k(n)$.*

In the sequel, to distinguish between the different regions of the configuration, we will speak of the *modified corner* to design the non-regular region located in the corner of the lattice, *i.e.* the modified corner contains all squares (i, j) with $i, j \leq k(n)$.

Experimentally, the size of the modified corner increases at the following rate : when its size is $k(n)$, it increases by one after approximatively $k(n)+1$ steps. Therefore we conjecture that the size of the modified corner after n steps of the algorithm is proportional to $O(\sqrt{n})$.

Let us consider the first few steps of the algorithm shown on Figure 5 (the squares with no value contain no grains). The first configuration, not shown, is the empty lattice. The reverse toppling of all vertices is designed by $\xrightarrow{\perp}$, and $\longrightarrow$ designs one step of the relaxation of an unstable configuration.

$$\xrightarrow{\perp} \begin{array}{ccccc} 2 & 1 & 1 & 1 & \cdots \\ 1 & & & & \\ 1 & & & & \\ 1 & & & & \\ \vdots & & & & \end{array}$$

$$\xrightarrow{\perp} \begin{array}{ccccc} 4 & 2 & 2 & 2 & \cdots \\ 2 & & & & \\ 2 & & & & \\ 2 & & & & \\ \vdots & & & & \end{array} \longrightarrow \begin{array}{ccccc} 0 & 3 & 2 & 2 & \cdots \\ 3 & & & & \\ 2 & & & & \\ 2 & & & & \\ \vdots & & & & \end{array}$$

$$\xrightarrow{\perp} \begin{array}{ccccc} 2 & 4 & 3 & 3 & \cdots \\ 4 & & & & \\ 3 & & & & \\ 3 & & & & \\ \vdots & & & & \end{array} \longrightarrow \begin{array}{ccccc} 4 & 0 & 4 & 3 & \cdots \\ 0 & 2 & & & \\ 4 & & & & \\ 3 & & & & \\ \vdots & & & & \end{array} \longrightarrow \begin{array}{ccccc} 4 & 1 & 0 & 4 & \cdots \\ 1 & 2 & 1 & & \\ 0 & 1 & & & \\ 4 & & & & \\ \vdots & & & & \end{array} \longrightarrow \begin{array}{ccccc} 0 & 2 & 1 & 1 & \cdots \\ 2 & 2 & 1 & 1 & \\ 1 & 1 & & & \\ 1 & 1 & & & \\ \vdots & & & & \end{array}$$

Fig. 5: The first three steps of the burning algorithm on the infinite lattice

Before the last reverse toppling of all vertices of this sequence (last configuration on line two), the squares of the first line (starting from the third square) each contain two grains. When we reverse-topple all vertices, one grain is added on each edge square. All squares of the first line (starting from the third square) contain then three grains. The second square on this line is unstable. When it topples, it gives one grain to each of its neighbor squares, including the square to its right. It becomes unstable in turn, and when it topples gives one grain to each of its neighboring squares, including the square to its right, and so on. The lattice being infinite, this sequence of toppling never stops. However, notice that each square of the first line topples after a finite time, and has value 1 after its right neighbor has in turn toppled. This allows us to write that the regular part of the first line is unstable, and takes value 1 after its toppling.

Let us now compare the first steps of the algorithm on the infinite lattice with the first steps in a large enough finite lattice (with side greater than or equal to 4).

We notice that the topplings in the upper left corner happen in exactly the same way in the finite and in the infinite lattice. We can easily check that this remains true, for these first steps, whatever the size of the lattice, as long as it remains greater than four.

This leads to conjecture that this remains true for any number of steps of the algorithm:

Conjecture 2.3 *Let u^n be the configuration obtained on the infinite lattice after n steps of the burning algorithm. Let k be the size of the modified corner in u^n. Then, for any $l, l' \geq 0$, the configuration v^n*

$$\xrightarrow{\perp} \begin{array}{ccccc} 2 & 1 & 1 & 1 & 2 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 2 & 1 & 1 & 1 & 2 \end{array}$$

$$\xrightarrow{\perp} \begin{array}{ccccc} 4 & 2 & 2 & 2 & 4 \\ 2 & 0 & 0 & 0 & 2 \\ 2 & 0 & 0 & 0 & 2 \\ 2 & 0 & 0 & 0 & 2 \\ 4 & 2 & 2 & 2 & 4 \end{array} \longrightarrow \begin{array}{ccccc} 0 & 3 & 2 & 3 & 0 \\ 3 & 0 & 0 & 0 & 3 \\ 2 & 0 & 0 & 0 & 2 \\ 3 & 0 & 0 & 0 & 3 \\ 0 & 3 & 2 & 3 & 0 \end{array}$$

$$\xrightarrow{\perp} \begin{array}{ccccc} 2 & 4 & 3 & 4 & 2 \\ 4 & 0 & 0 & 0 & 4 \\ 3 & 0 & 0 & 0 & 3 \\ 4 & 0 & 0 & 0 & 4 \\ 2 & 4 & 3 & 4 & 2 \end{array} \longrightarrow \begin{array}{ccccc} 4 & 0 & 5 & 0 & 4 \\ 0 & 2 & 0 & 2 & 0 \\ 5 & 0 & 0 & 0 & 5 \\ 0 & 2 & 0 & 2 & 0 \\ 4 & 0 & 5 & 0 & 4 \end{array} \longrightarrow \begin{array}{ccccc} 0 & 2 & 1 & 2 & 0 \\ 2 & 2 & 1 & 2 & 2 \\ 1 & 1 & 0 & 1 & 1 \\ 2 & 2 & 1 & 2 & 2 \\ 0 & 2 & 1 & 2 & 0 \end{array}$$

Fig. 6: The first three steps of the burning algorithm on a square lattice

obtained on the rectangular lattice of size $(2k+l)\times(2k+l')$ *after* n *steps of the burning algorithm is such that* $v^n_{i,j} = u^n_{i,j}$ *for any* $i \leq k+l, j \leq k+l'$.

3 Exploration of the Conjecture

In this section, we demonstrate a slightly simpler version of the Conjecture 2.3. In the sequel, we need the fact that the regular part at time $n-1$ always contains the one at time n. It means that we need k to be an increasing function. Experimentaly it is true, but hard to prove. This is why we introduce $\tilde{k}$:

Definition 3.1 *For a given n, we define* $\tilde{k}(n)$ *as:*

$$\tilde{k}(n) = \begin{cases} 0 & \text{if} \quad n = 0 \\ \max\{\tilde{k}(n-1), k(n)\} & \text{if} \quad n > 0 \end{cases}$$

This implies that, for a given n, if $i > \tilde{k}(n)$ then column i belongs to the regular part of u^{n-1} and u^n.

Moreover, from the proof of Proposition 2.1, we get:

Proposition 3.2 *For a given n, we have:* $\tilde{k}(n) - k(n-1) \leq 1$.

Proof : From the proof of Proposition 2.1, we know that $k(n) \leq k(n-1)+1$. It implies that $\tilde{k}(n) \leq \max(k(n-1)+1, k(n-1))$. Besides $\tilde{k}(n) \geq k(n-1)$. Thus $0 \leq \tilde{k}(n) - k(n-1) \leq 1$. □

The use of $\tilde{k}(n)$ in the sequel might seem like a strong restriction. However, since it has been observed that the size of the modifi ed corner always grows during the burning algorithm, this is not the case. In fact, if k is an increasing function of n, then $k(n) = \tilde{k}(n)$ for all $n \geq 0$.

We denote by t^n_i the toppling vector of a given column i during the step n of the burning algorithm: $t^n_{i,j} = 1$ if the square (j,i) topples during the n-th, and $t^n_{i,j} = 0$ otherwise. Notice that the coeffi cients of t^n_i are equal to 0 or 1. We can deduce easily from Proposition 2.1 that t^n_i is a constant sequence of vectors for $i > k(n)$. The next theorem shows that this sequence is constant from rank $\tilde{k}(n)$:

Theorem 3.3 *For a given n, the sequence* (t^n_i) *is constant for* $i \geq \tilde{k}(n)$.

Proof : By definition of the modified corner, $u^n_{i,j} = u^n_{i',j}$ for all $j > 0$ and $i, i' > \tilde{k}(n) \geq k(n)$. We denote ν^n_j this common value, *i.e.*: $\nu^n_j = u^n_{i,j}$, for all $i > \tilde{k}(n)$. Since $\tilde{k}(n) \geq k(n-1)$, ν^{n-1}_j is also well defined for $i > \tilde{k}(n)$.
For $i > \tilde{k}(n)$ and $j > 1$ we get the equation:

$$\begin{aligned} \nu^n_j &= \nu^{n-1}_j + t^n_{i-1,j} + t^n_{i+1,j} + t^n_{i,j-1} + t^n_{i,j+1} - 4t^n_{i,j}, \\ \text{hence } t^n_{i-1,j} &= 4t^n_{i,j} - t^n_{i,j-1} - t^n_{i,j+1} - t^n_{i+1,j} + \nu^n_j - \nu^{n-1}_j. \end{aligned} \tag{1}$$

This equation is *a priori* not true for $j = 1$, because $t^n_{i,0}$ is not defined. However it remains true if we extend the vector t^n_i by $t^n_{i,0} = 0$.
If we set $V^n_j = \nu^n_j - \nu^{n-1}_j$ for all $j > 0$, and if we define the matrix A such that $A_{i,i} = 4$, $A_{i,i+1} = A_{i+1,i} = -1$, and $A_{i,j} = 0$ elsewhere, then we can rewrite Equation (1) as a vectorial equality:

$$\forall i > \tilde{k}(n), \; t^n_{i-1} = At^n_i - t^n_{i+1} + V^n. \tag{2}$$

Since we know that the sequence (t^n_i) is constant for $i > \tilde{k}(n)$ (cf proof of Proposition 2.1), this equation implies that the vector $t^n_{\tilde{k}(n)}$ verifies the same equality as any other vector t^n_i, for $i > \tilde{k}(n)$. As a conlusion, the sequence (t^n_i) is constant for $i \geq \tilde{k}(n)$. □

Notice that this result is really surprising only if $\tilde{k}(n)$ is really the size of the modified corner, which we strongly conjecture, due to the remark made before. Indeed, in this case, it implies that the toppling vector at the border of the modified corner is also the same as the one in the regular region: a square $(i, k(n))$ at the edge of the modified corner topples during the n-th step of the algorithm if and only if the squares in the regular part of the same line all topple during this step. This means that, although the squares at the limit the modified corner do not have the same values than their neighbors in the regular part of the configuration, they behave the same way. Experimentally, we have noticed the following fact: if the value of a square (i, j) at the limit of the modified corner contains two grains, then all squares $(i, j'), j' > i$ appearing after it (in the regular part of the line) all contain one grain. If the square (i, j) contains three grains, all the squares in the regular part of the line contain two grains. Other values do not appear at the limit of the modified corner.

This means that, in practice, the square at the limit of the modified corner always acts as a trigger for the toppling of the whole regular part of a line. Indeed, when the regular part of a line has value 2, the square preceding the regular part contains three grains. If the value of the line increases by one, the square preceding the regular part becomes unstable, and the regular part of the line can topple. This is similar to what we have observed in the first steps of the algorithm.

In the next lemma, we show that it is always possible to topple first the squares of row index inferior to a certain value $\tilde{k}(n) + l$ and of column index inferior to another value $\tilde{k}(n) + l'$, during the step n of the burning algorithm.

Lemma 3.4 *Let $n \geq 0$ and $l, l' \geq 1$. On the infinite lattice, among all the topplings which occur during the step n of the burning algorithm, it is always possible to begin by the topplings inside the rectangle $(\tilde{k}(n) + l) \times (\tilde{k}(n) + l')$, and then topple the other squares.*

Proof : We denote by $\mathcal{C}$ the set of squares belonging to the upper-left rectangle $(\tilde{k}(n) + l) \times (\tilde{k}(n) + l')$. As $t^n_{i,j} = 0$ for $i, j > \tilde{k}(n)$, and as the lattice is symmetrical along the diagonal, the squares that can topple

and that do not belong to $\mathcal{C}$ are the squares (i,j) such that $i > \tilde{k}(n)+l$ and $j \leq \tilde{k}(n)$, and the squares (j,i) such that $i > \tilde{k}(n)+l'$ and $j \leq \tilde{k}(n)$. We denote by $\overline{\mathcal{C}}$ this set of squares.
We consider the following two steps process:

- We realize iteratively all the possible topplings of $\mathcal{C}$.
- We relax the obtained configuration.

Two cases are possible: either a toppling is possible on the second phase, or not. If no toppling is possible, the result is true.
In the other case, we show that during the second phase, the only possible topplings are the ones of squares of $\overline{\mathcal{C}}$. Let us suppose that it is not the case, and let (i,j) be the first square of $\mathcal{C}$ which topples during the relaxation. (i,j) is necessarily a square at the limit of $\mathcal{C}$ and $\overline{\mathcal{C}}$. By symmetry, we can suppose that $i = \tilde{k}(n)+l$ and $j \leq \tilde{k}(n)$. Then there exists a sequence of topplings of squares of $\overline{\mathcal{C}}$ which made the square $(\tilde{k}(n)+l, j)$ unstable. Among all such sequences of topplings, we choose one such that the number of rows involved is minimal, *i.e.* a sequence $\mathcal{S} = (i_k, j_k)_{k\geq 1}$ such that the greatest index i_k appearing in $\mathcal{S}$ is minimal. Let $\mathcal{S}$ be such a sequence. We denote the squares (i', j') such that i' is maximal by $(i', j_1), \ldots, (i', j_r)$ in apparition order: $\mathcal{S}$ is of the form $\ldots(i', j_1)\ldots(i', j_2)\ldots(i', j_r)\ldots$.

We show by recurrence on s that if $i' > \tilde{k}(n)+l$ there exists a sequence of topplings $\mathcal{S}'$ which made the square $(\tilde{k}(n)+l, j)$ unstable, such that the square $(i'-1, j_s)$ topples before the square (i', j_s) for all $1 \leq s \leq r$.

If $s = 1$, there are two possible cases. If the square $(i'-1, j_s)$ has toppled before the square (i', j_s), then the hypothesis is verified. Else, as any square neither of the row $\tilde{l}+1$ nor of the row i' has toppled before the square (i', j_1), if this square is unstable then $(i'-1, j_1)$ also is. Indeed, since $l > 0$, the squares (i', j_1) and $(i'-1, j_1)$ are in the regular part of the configuration after the step $n-1$: they contain the same number of grains. After addition of β, this property remains verified. We know that the square $(\tilde{l}, j_1)$ did not get any supplementary grain, so that the square $(i'-1, j_1)$ contains more. In particular if the square (i', j_1) is unstable (case where it is near the border of the lattice), the square $(i'-1, j_1)$ also. Then we can construct from S a sequence of topplings $\mathcal{S}'$ where the square $(i'-1, j_1)$ topples before the square (i', j_1).

Let us suppose the result true until rank $s \geq 1$, with $s < r$. Before the toppling of the square (i', j_s), no square of the row $i'+1$ has toppled. Besides, if any square in the same row has toppled, then by recurrence hypothesis, there exists a sequence $\mathcal{S}'$ which made the square $(\tilde{k}(n)+l, j)$ unstable such that the square of the same column and of row $i'-1$ toppled before it. Then if the square $(i'-1, j_s)$ has not topppled before the square (i', j_s) in $\mathcal{S}$, it is also unstable in $\mathcal{S}'$, and we can topple it before (i', j_s) in $\mathcal{S}'$. Hence the result.

By definition of $\mathcal{S}$, there is no toppling sequence at the beginning of the second phase which enables the square $(\tilde{k}(n)+l, j)$ to topple, and whose maximal row index is strictly less than i'. But we just constructed from $\mathcal{S}$ a sequence $\mathcal{S}'$ which enables the square $(\tilde{k}(n)+l, j)$ to topple in such a way that the toppling of any square of row i' occurs after its neighbor square on row $i'-1$. This means that the topplings of the row i' are not necessary in the sequence $\mathcal{S}'$, and that we can construct a sequence $\mathcal{S}''$ which made the square $(\tilde{k}(n)+l, j)$ unstable and such that no square of index strictly greater than i' topples. This is a contradiction. Hence the result. □

With this result, we are at last able to prove a slightly simpler version of Conjecture 2.3:

Theorem 3.5 *Let u^n be the configuration obtained on the infinite lattice after n steps of the burning algorithm. We recall that $\tilde{k}(n)$ is defined as the maximum value between $\tilde{k}(n-1)$ and $k(n)$ (Definition*

3.1). Then, for all $l, l' \geq 1$, the configuration v^n obtained after n steps of the burning algorithm on the rectangular lattice of size $(2\tilde{k}(n)+l) \times (2\tilde{k}(n)+l')$, is such that $v^n_{i,j} = u^n_{i,j}$ for any $i \leq \tilde{k}(n)+l$, $j \leq \tilde{k}(n)+l'$.

Proof : We show the result by recurrence on $n \geq 0$. For $n = 0$, it is obvious.
Let us suppose that the result is true at rank $n-1$. Let L be a rectangular lattice of size $(2\tilde{k}(n)+l) \times (2\tilde{k}(n)+l')$, and v^{n-1} be the configuration on L obtained after $n-1$ steps of the burning algorithm. By recurrence hypothesis, $v^{n-1}_{i,j} = u^{n-1}_{i,j}$ for all $i \leq \tilde{k}(n)+l$, $j \leq \tilde{k}(n)+l'$.
We split L into the following four rectangles:

- $[(1,1),(\tilde{k}(n)+l,\tilde{k}(n)+l')]$;
- $[(\tilde{k}(n)+l,\tilde{k}(n)+l'+1),(1,2\tilde{k}(n)+l')]$;
- $[(\tilde{k}(n)+l+1,1),(2\tilde{k}(n)+l,\tilde{k}(n)+l')]$;
- $[(2\tilde{k}(n)+l,\tilde{k}(n)+l'+1),(\tilde{k}(n)+l+1,2\tilde{k}(n)+l')]$.

Among all the topplings which occur during the step n of the burning algorithm, it is always possible to begin by the topplings inside the rectangle $[(1,1),(\tilde{k}(n)+l,\tilde{k}(n)+l')]$ of the upper left corner and then topple the other squares (Lemma 3.4).

Then as the $\tilde{k}(n)+l$ first squares of column $\tilde{k}(n)+l'$ that topple during the global relaxation of $v^{n-1}+\beta$ on L, have been toppled, we can apply the same argument (Lemma 3.4) to the second rectangle $[(\tilde{k}(n)+l,\tilde{k}(n)+l'+1),(1,2\tilde{k}(n)+l')]$, and then to the third one and the fourth one.

Eventually we have done only valid topplings and the configuration w on L obtained at the end of the process is symmetrical and verifies: $w_{i,j} = u^n_{i,j}$ for all $i \leq \tilde{k}(n)+l$, $j \leq \tilde{k}(n)+l'$. In particular it is stable. Hence $w = v^n$, which proves the result. □

4 Discussion

The result presented in the previous section gives an insight on the structure of the identity configuration. Indeed, we have shown that, up to a certain number of steps, the computation of the identiy on a lattice of a given size is exactly the same as the computation of the identity on a bigger lattice. This explains the similarities between the identity configurations on lattices of all sizes.

Moreover, this result allows us to compute the identity on a square lattice in a faster way. First, the intermidiate steps of the computing of the identity can be stored, and be used later to compute the identity on a lattice of a greater size. Second, it is obvious by the symmetry of the $2p \times 2p$ square that one needs only a quarter $p \times p$ of the lattice to compute the identity configuration. We can therefore apply the burning algorithm to a $p \times p$ lattice, considered as the upper-left corner of a $2p \times 2p$ lattice. We have shown in Theorem 3.5 that, as long as the size of the modified corner is less than a given $k < p$, then the computation can be made on a $k \times k$ lattice. Therefore, we can begin the computation on a smaller lattice, and then increase the size of the lattice at the same rate as the size of the modified corner increases. This saves the time of updating the squares in the regular part that are outside the $k \times k$ lattice during the computation. We have seen that the burning algorithm has complexity $O(n^2)$, where n is the number of squares in the lattice. The process we have described saves an amount of time proportional to n^2, therefore it does not decrease the time of the computation much, but the multiplicative constant of the n^2 term is lessened.

We have seen that, on the infinite lattice, the lines in the regular part all have value 0, 1, or 2. Squares with value 0 appear only in the part that has not been modified by the computation. Moreover, we have noticed that, if the regular part of a line is 1, then the square preceding the regular part has value 2. If the regular part of a line is 2, the preceding square has value 3.

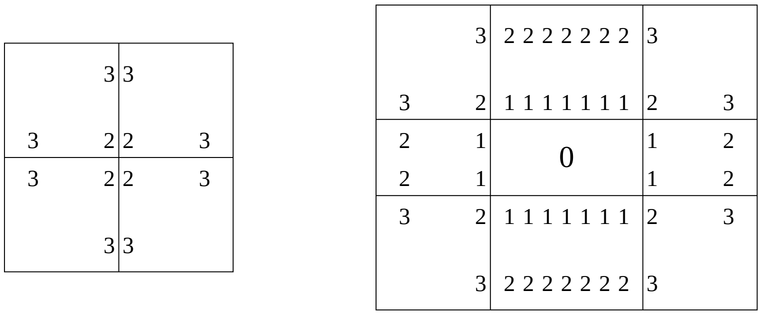

Fig. 7: A configuration equivalent to the identity

This is very similar to what has been observed on the identity configuration on a $(2p+1)\times(2p+1)$ lattice (see figure 1) : the value of a square $(i,p+1)$ if the middle column of the $(2p+1)\times(2p+1)$ lattice is 2 if the square (i,p) contains three grains, and its value is 1 if the neighboring square contains two grains. In fact, we conjecture that this observation can be extended to a larger lattice :

Conjecture 4.1 *Let I be the identity configuration on the $2p\times 2p$ lattice. If u is the configuration on the $(2p+l)\times(2p+l')$ lattice, obtained from I in the following way:*

- $u_{i,j} = u_{2p+l-i+1,j} = u_{i,2p+l'-i+1} = u_{2p+l-i+1,1p+l'-j+1} = I_{i,j}$, *for all* $i,j<p$ *;*
- $u_{i,j} = u_{2p+l-i+1,j} = I_{i,p}-1$ *for all* $i<p$, $p<j<p+l$ *;*
- $u_{i,j} = u_{i,2p+l'-i+1} = I_{p,j}-1$ *for all* $p<j<p+l'+1$, $j<p$ *;*
- $u_{i,j} = 0$ *for all* $p<i<p+l+1$, $p<j<p+l'+1$,

then u is equivalent to the identity configuration on the $(2p+l)\times(2p+l')$ lattice.

See Figure 7 for an example of this construction. Notice that any configuration u on the $(2p+l)\times(2p+l')$ lattice, with $l,l'>1$, obtained as described in this conjecture, is not recurrent: any such configuration contains two or more adjacent squares containing zero grain, which, as we have seen, is impossible in a recurrent configuration.

Experimentally, the only recurrent configurations obtained in this manner are those on the $(2p+l)\times(2p+l')$ lattice, with $l,l'\leq 1$. In this case, the configuration u obtained is exactly the identity, which confirms what we have stated on the identity on the $(2p+1)\times(2p+1)$ lattice in section 1.

It seems therefore that, if we can prove that the lines in the regular part of the configuration have value 2 if the neighboring square contains three grains, and value 1 if the neighboring square contains two grains (as we have observed), then, using Theorem 3.5 we can prove Conjecture 4.1. However, there still lacks one step to prove this. Indeed, the shape of the modified corner is in general not a perfect square (see Figure 2 for instance): one corner is truncated. Therefore, if the size of the modified corner after the n-th step of the algorithm is k, the configuration in the modified corner is not the same as in the identity on the $2k \times 2k$ lattice (although it presents strong similarities with it): the second phase of the algorithm modifies the configuration.

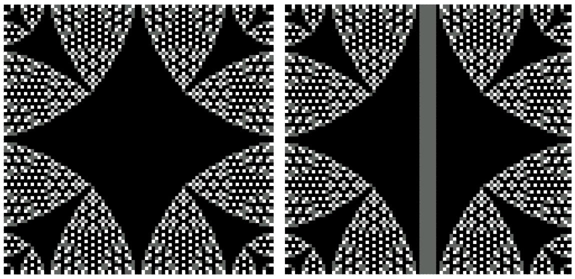

Fig. 8: The configurations $\overline{1}$ on the 80×80 and 80×85 lattices

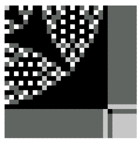

Fig. 9: The 253-th step of the burning algorithm on the infinite lattice, starting from the configuration where all squares contain one grain

We now introduce another configuration, $\overline{1}$, which is the recurrent configuration equivalent to the configuration of the lattice where all squares contain one grain. See figure 8 for an example of this configuration.

This configuration shares many of the identity configuration's intriguing properties. In particular, it presents a very similar complicated fractal structure. Moreover, the configuration $\overline{1}$ on the $(2p+1) \times (2p+1)$ lattice is related to that on the $2p \times 2p$ lattice in the same way that the identity configurations on the same two lattices.

This configuration can also be computed by the burning algorithm, but starting in this case from the configuration where all squares contain exactly one grain. During the computation of this configuration, we have noticed that the algorithm goes through only one phase, similar to the first phase of the computation of the identity configuration. For the computation of $\overline{1}$, there is no second phase: when the two modified corners meet the configuration is recurrent, and the algorithm stops.

This is due to the fact that the shape of the modified corner in $\overline{1}$ is exactly a square, while this is not the case for the identity (compare Figures 2 and 9).

Starting the burning algorithm on the infinite lattice from this configuration, we notice the same facts than when we start from the empty lattice: if the value of a square (i, j) at the limit of the modified corner contains two grains, then all squares $(i, j'), j' > i$ appearing after it (in the regular part of the line) all contain one grain. If the square (i, j) contains three grains, all the squares in the regular part of the line contain two grains. Other values do not appear at the limit of the modified corner.

The arguments we have used for the study of the algorithm starting from the empty lattice also hold in this case. In particular, Theorem 3.5 is still valid, and a very similar version of Conjecture 4.1 can be presented:

Conjecture 4.2 *Let $\overline{1}$ be the recurrent configuration equivalent to the configuration in which all squares contain exactly one grain on the $2p \times 2p$ lattice. If u is the configuration on the $(2p+l) \times (2p+l')$ lattice, obtained from $\overline{1}$ in the following way:*

- $u_{i,j} = u_{2p+l-i+1,j} = u_{i,2p+l'-i+1} = u_{2p+l-i+1,2p+l'-j+1} = \overline{1}_{i,j}$, *for all* $i, j < p$ *;*
- $u_{i,j} = u_{2p+l-i+1,j} = \overline{1}_{i,p} - 1$ *for all* $i < p$, $p < j < p+l$ *;*
- $u_{i,j} = u_{i,2p+l'-i+1} = \overline{1}_{p,j} - 1$ *for all* $p < j < p+l'+1$, $j < p$ *;*
- $u_{i,j} = 1$ *for all* $p < i < p+l+1$, $p < j < p+l'+1$,

then u is equivalent to the configuration $\overline{1}$ on the $(2p+l) \times (2p+l')$ lattice.

To conclude, if we can prove that the lines in the regular part of the configuration have value 2 if the neighboring square contains three grains, and value 1 if the neighboring square contains two grains, then we need only to show that the modified corner grows as a perfect square to prove Conjecture 4.2 from Theorem 3.5. Therefore it takes one step less to prove Conjecture 4.2 than Conjecture 4.1, which proves that the configuration $\overline{1}$, although it is very similar to the identity, is simpler to study.

5 Conclusion

We have studied here the burning algorithm, used to compute the identity, and other recurrent configurations, in the abelian sandpile model. Observing that this algorithm goes through two phases, we have focused on the study of the first phase, by introducing an infinite extension of the algorithm. This has led us to an interesting result about the unfolding of this algorithm: the first configurations it computes are the same, whatever the size of the considered lattice. Most of all, this study gives an insight into new directions to study the structure of the identity configuration.

Finally, we have introduced a new configuration, $\overline{1}$, which is the recurrent configuration equivalent to the configuration in which all squares contain exactly one grain. This configuration shares many of the identity interesting properties, but presents stronger regularities, and has never been studied to the extent of our knowledge.

References

[Bak97] P. Bak. *How nature works - The science of SOC*. Oxford university press, 1997.

[BR00] Y. Le Borgne and D. Rossin. Identité du groupe du tas de sable sur des grilles rectangulaires. In LACIM, 2000.

[BTW87] P. Bak, C. Tang, and K. Wiesenfeld. Self-organized criticality: an explanation of the $1/f$ noise. *Physics Review Letters*, 59(4):381–384, 1987.

[CR00] R. Cori and D. Rossin. On the sandpile group of a graph. *European Journal of Combinatorics*, 21(4):447–459, 2000.

[Cre96] M. Creutz. Cellular automata and self organized criticality, 1996. Int. conf. on multi-scale phenomena and their simulation," 30/Sept-4/Oct/96, Bielefeld, Germany. Available at `http://xxx.lanl.gov/abs/hep-lat/9611017`.

[Dha90] D. Dhar. Self-organized critical state of sandpile automaton models. *Phys. Rev. Lett.*, 64, 1990.

[Dha99] D. Dhar. The abelian sandpile model and related models. *Physica A*, 263, 1999. Preprint available on Arxiv, at `http://arxiv.org/abs/cond-mat/9808047`

[DM90] D. Dhar and S.N. Majumdar. Abelian sandpile model on the Bethe lattice. *Journal of Physics*, A 23:4333–4350, 1990.

[DRSV95] D. Dhar, P. Ruelle, S. Sen, and D. Verma. Algebraic aspects of sandpile models. *Journal of Physics*, A 28:805–831, 1995.

[KLGP00] D. V. Ktitarev, S. Lübeck, P. Grassberger, and V. B. Priezzhev. Scaling of waves in the Bak-Tang-Wiesenfeld sandpile model. *Physical Review E*, 61:81–92, 2000.

[MD92] S.N. Majumdar and D. Dhar. Equivalence between the abelian sandpile model and the $q \longrightarrow 0$ limit of the potts model. *Physica A*, 185:129–145, 1992.

[MN99] C. Moore and M. Nilsson. The computational complexity of sandpiles. *Journal of Statistical Physics*, 96:205–224, 1999.