On the Addition of Recurrent Configurations of the Sandpile-Model

Matthias Schulz

University of Karlsruhe, Department for Computer Sciences
Am Fasanengarten 5, 76128 Karlsruhe, Germany
schulz@ira.uka.de

Abstract. The sandpile model, introduced by Bak, Tang and Wiesenfeld in 1987, is the standard example for a dynamic model showing Self-Organized Criticality (SOC). Also, it has many nice algebraic properties; for example, there is a set of configurations which is a group with a certain naturally defined addition.

We look at elements c, d of this group and try to find out how long it takes to naively compute the sum $c \oplus d$. While we can easily give an upper bound, it is harder to find a lower bound. We prove some facts about the number of topplings (elementary operations) that have to be performed during the addition of two elements of the group and give a heuristic for quickly finding local minima.

1 Introduction

The sandpile model was introduced by Bak, Tang and Wiesenfeld in 1987 [1] as a model to explain $\frac{1}{f}$ noise. Grains of sand fall onto a grid, and if four or more grains are lying upon a site in the grid, one grain of sand falls to the left, right, top and bottom respectively. It has been found that the model displays Self-Organized Criticality (SOC), which means that from some time on a critical state is maintained if grains keep on falling randomly onto the grid.

Dhar and others [3] found many interesting algebraic properties of the sandpile model, especially the set of recurrent configurations (configurations which can occur infinitely often in the process described above). One of the most interesting findings was the fact that the set of recurrent configurations, together with an addition, is an Abelian group.

In this paper, we consider the question how many topplings there will be at least if we add two recurrent configurations. While it is hard to find the global optimum, we introduce a probabilistic algorithm which gives us at least local minima whose quality depends on the strategy used at one point.

2 Basics

Consider the grid $Z = \{0, \ldots n-1\} \times \{0, \ldots n-1\}$. For each $z \in Z$, let $N(z)$ be the set of sites in the von-Neumann-neighborhood of z.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 236–243, 2008.

From a configuration $c : Z \to \mathbb{N}_0$ and a site $z \in Z$ which satisfies $c(z) > 3$, we get the successive configuration c_z defined as

$c_z = c - 4e_z + \sum_{z' \in N(z)} e_{z'}$,

where $\forall z \in Z : e_z(z') = \begin{cases} 1 & \text{if } z' = z \\ 0 & \text{otherwise} \end{cases}$.

Figuratively, if a site contains 4 or more grains of sand, 4 grains fall off this site and onto the sites in the neighborhood or off the grid, when the site was at the border of the grid. We say that the site *toppled* and a *toppling* occurred.

If all sites containing at least 4 grains topple at the same time and we bound the number of grains initially in c, we get the rule for a cellular automaton.

It has been shown that the process of letting sites with at least 4 grains topple eventually ends and leads to a configuration where each site contains at most 3 grains of sand (cf. for example [2]). Also, the order in which the sites topple is irrelevant, as all possible sequences lead to the same configuration, and the number of times a site z topples during the process is also independent of the order of the topplings. This process is called a *relaxation* and the resulting configuration when starting from configuration c will be denoted by c_{rel}.

Let c be a configuration on Z. For all $z \in Z$, let $f_c(z)$ be the number of topplings of z during the relaxation of c. Further, let $B \in \mathbb{Z}^{Z \times Z}$ be the matrix with

$$B(i,j) = \begin{cases} 4 & \text{if } i = j \\ -1 & \text{if } i \in N(j) \\ 0 & \text{otherwise.} \end{cases}$$

Then $c_{rel} = c - B \cdot f_c$.

Further, B is invertible; both is shown in [2].

Let $\mathcal{C}$ be the set of configurations $c : \mathbb{N}_0 \to Z$ with $c_{rel} = c$. (This means: $\forall z \in Z : 0 \le c(z) \le 3$.) The elements of $\mathcal{C}$ are called *stable configurations*.

We define the operation $\oplus$ on $\mathcal{C}$ by $c \oplus d = (c + d)_{rel}$. This operation is associative and commutative, as shown in [3].

We define the maximal stable configuration m satisfying $\forall z \in Z : m(z) = 3$ and the set of configurations $\mathcal{R} = m \oplus \mathcal{C}$. The elements of $\mathcal{R}$ are called *recurrent configurations*.

Note that for each configuration $c \in \mathcal{C}$ and for each configuration $d \in \mathcal{R}$, there exists a configuration $e \in \mathcal{C}$ such that $c \oplus e = d$:

From the definition of $\mathcal{R}$, we know that there is a configuration $e' \in \mathcal{C}$ such that $m \oplus e' = d$, and therefore $c \oplus ((m - c) \oplus e') = (c \oplus (m - c)) \oplus e' = m \oplus e' = d$.

Also, if c is a recurrent configuration, then for all $d \in \mathcal{C}$ $c \oplus d \in \mathcal{R}$, too.

Therefore, if we have a Markov chain $(\mathcal{C}, P)$ with the states being stable configurations and transition probability matrix satisfying $P(c, d) = 0$ if there exists no $z \in Z$ such that $d = c \oplus e_z$, $\mathcal{R}$ is the set of recurrent configurations of this Markov chain.

A very interesting fact about $\mathcal{R}$ is that $(\mathcal{R}, \oplus)$ is an Abelian group, which is proven in [3].

By $id \in \mathcal{R}$ we denote the identity element of the group $(\mathcal{R}, \oplus)$.

Let $b \in \mathcal{C}$ be the configuration which satisfies

$\forall z \in Z : b(z) = 4 - |N(z)|$.

(So, $b(z) = 2$ if z is a site in corner of Z, $b(z) = 1$ if z is a site on the border of Z, and $b(z) = 0$ otherwise.)

It has been shown in [4] that

$c \in \mathcal{R} \iff c \oplus b = c$ and

$c \in \mathcal{R} \iff f_{c+b} = \mathbf{1}$,

where $\forall z \in Z : \mathbf{1}(z) = 1$.

3 The Problem

Consider two recurrent configurations c and d. A naive way to compute $c \oplus d$ would be to compute the sum $c + d$ and to relax this configuration. This means $\mathbf{1}^\top f_{c+d}$ topplings have to be done.

It is quite easy to see that the worst case, i.e. the case for which $|f_{c+d}| = \mathbf{1}^\top f_{c+d}$ is maximal, is when $c = d = m$; it can be shown that $|f_{m+m}| \in O(n^4)$ holds.

On the other hand, it is much harder to find a tight lower bound for $|f_{c+d}|$ when $c, d \in \mathcal{R}$, or to find configurations $c, d \in \mathcal{R}$ for which $|f_{c+d}|$ is minimal. While we can find a configuration c such that $m - c \in \mathcal{R}$ for $n \leq 4$, there is a very obvious reason why there cannot be such a configuration c for $n > 4$: In this case the configuration $c + d$ contains more than $3n^2$ grains of sand and the surplus grains have to fall off the grid, which means that sites on the border of Z must topple.

In the following sections, we will show that each recurrent configuration contains at least $2n^2 - 2n$ grains of sand, which means that during the relaxation of the sum $c + d$ of two recurrent configurations c, d at least $n^2 - 4n$ grains of sand must fall off the grid.

For $n = 4$ we will give recurrent configurations $c, d \in \mathcal{R}$, such that $c + d = m$.

We will show how to find for $c \in \mathcal{R}$ a configuration $\bar{c} \in \mathcal{R}$ such that $|f_{c+\bar{c}}|$ is minimal.

We will show how we can get from a recurrent configuration c to a recurrent configuration c' such that $f_{c'+(\bar{c'})} \leq f_{c+\bar{c}}$ holds. ($\leq$ here means component-wise less or equal.) By repeating this process, we reach a local minimum.

4 Number of Grains in Recurrent Configurations

Let c be recurrent configuration. All grains that remain on a site z after it toppled during the relaxation of $c + b$ can be taken away without the resulting configuration c' becoming non-recurrent, since the same sequence of topplings during the relaxation of $c + b$ is a possible toppling sequence for $c' + b$: If a site z contained more than four grains of sand at the moment it toppled, then the "surplus" grains are the ones that were taken away to get c'. Therefore, during the relaxation of $c' + b$ each site contains exactly four grains of sand at the moment it topples if we use the same sequence of topplings as for $c + b$.

We call a recurrent configuration c *minimal recurrent*, if $\forall z \in Z : c - e_z \notin \mathcal{R}$. This means that no grain of sand can be taken from c without getting a non-recurrent configuration; it follows that each site contains exactly four grains of sand at the moment it fires.

By counting the grains of sand each site contains just before it fires during the relaxation of $c + b$, we obviously get $4n^2$ grains. Since we counted each grain of sand that still is in $(c + b)_{rel}$ twice and each grain of sand that got lost at the edge once, we get

$4n^2 = 2|c| + |b|$

and therefore, since $|b| = 4n$,

$|c| = 2n^2 - 2n$.

This means that each recurrent configuration contains at least $2n^2 - 2n$ grains of sand and that the sum of two recurrent configurations c and d contains at least $4n^2 - 4n$ grains of sand. From this we get $|f_{c+d}| \geq \frac{n^2-4n}{2}$, since a toppling can lead to the loss of no more than 2 grains of sand (except for $n = 1$, in which case $\frac{n^2-4n}{2} < 0$ anyways).

5 Examples

It is easy to see that the highest n for which we can hope to find $c, d \in \mathcal{R}$ such that $|f_{c+d}| = 0$ (and therefore obviously minimal) is 4. We here give two configurations $c, d \in \mathcal{R}$ which satisfy $c + d \leq m$.

$$c = \begin{pmatrix} 2 & 1 & 3 & 1 \\ 1 & 2 & 2 & 1 \\ 2 & 1 & 1 & 2 \\ 2 & 0 & 2 & 1 \end{pmatrix}, d = \begin{pmatrix} 1 & 2 & 0 & 2 \\ 2 & 1 & 1 & 2 \\ 1 & 2 & 2 & 1 \\ 1 & 3 & 1 & 2 \end{pmatrix}$$

Note that $c + d = m$ and we get d by the point reflection in the center of the grid. (Therefore $c \in \mathcal{R} \iff d \in \mathcal{R}$). It is easy to verify that $c \oplus b = c$ and $d \oplus b = d$.

However, for $n = 5$, we don't find configurations c, d with $f_{c+d} = 3$, which is the lower bound we get from the inequality $|f_{c+d}| \geq \frac{n^2-4n}{2}$, although the proof is rather inelegant.

6 Searching for Local Minima

Now, we take a look at larger values for n and describe a search strategy for recurrent configurations c, d such that $|f_{c+d}|$ becomes small.

First, we will show how to construct $\bar{c} \in \mathcal{R}$ such that for all $d \in \mathcal{R}$ $f_{c+\bar{c}} \leq f_{c+d}$ holds.

Then we will describe how to optimize c.

6.1 Minimizing f_{c+d} for Fixed c

Let c be a recurrent configuration. Then for all $d \in \mathcal{R}$ the following inequality holds:

$f_{c+d} \geq f_{c+((m-c)\oplus id)}$:

There are no topplings during the relaxation of $(c \oplus d) + (m - (c \oplus d))$, and therefore

$c + d - B \cdot f_{c+d} + (m - (c \oplus d)) =$
$c + d + (m - (c \oplus d)) - B \cdot f_{c+d} =$
$c \oplus (d \oplus (m - (c \oplus d))) =$
$c \oplus (d + (m - (c \oplus d)) - B \cdot f_{d+(m-(c\oplus d))})$.

We define $\bar{c} = d \oplus (m - (c \oplus d))$ and get

$c + d + (m - (c \oplus d)) - B \cdot f_{c+d} = c + \bar{c} - B \cdot f_{c\oplus\bar{c}}$
$\Rightarrow c+d+(m-(c\oplus d))-B \cdot f_{c+d} = c+d+(m-(c\oplus d))-B \cdot f_{d+(m-(c\oplus d))} - B \cdot f_{c+\bar{c}}$
$\Rightarrow f_{c+d} = f_{c+\bar{c}} + f_{d+(m-(c\oplus d))}$,

since B is invertible.

We see that $f_{c+\bar{c}} \leq f_{c+d}$.

We know that $d \oplus (m - (c \oplus d)) \in \mathcal{R}$ since $d \in \mathcal{R}$, and we know that $c \oplus (d \oplus (m - (c \oplus d))) = m$.

On the other hand, $id \oplus (m - c) \in \mathcal{R}$ and $c \oplus (id \oplus (m - c)) = m$, and it follows that for all $d \in \mathcal{R}$ the equation $id \oplus (m - c) = d \oplus (m - (c \oplus d))$ holds.

Therefore for all $d \in \mathcal{R}$ the inequality $f_{c+((m-c)\oplus id)} \leq f_{c+d}$ holds.

From now on, we call the configuration $(m - c) \oplus id$ the *minimizing configuration* of c, denoted as $\bar{c}$.

The task to minimize the topplings during the relaxation of the sum of two recurrent configurations is now reduced to finding a recurrent configuration c such that $|f_{c+\bar{c}}|$ becomes minimal.

6.2 The Cutting Algorithm

Let c be a recurrent configuration, $\bar{c}$ the minimizing configuration of c and e a configuration in $\mathcal{C}$ which is component-wise less or equal to c.

It can be easily shown that $f_{c+\bar{c}} = f_{(c-e)+(\bar{c}\oplus e)} + f_{\bar{c}+e}$.

This means that there are fewer topplings during the relaxation of $(c - e) + (\bar{c} \oplus e)$ if there were topplings during the relaxation of $\bar{c} + e$. If we make sure that $c - e$ is still a recurrent configuration, we find a pair of recurrent configurations which induces fewer topplings during the relaxation of their sum.

(Also, if $c - e$ is recurrent, it is easy to see that $\bar{c} \oplus e$ is the minimizing configuration of $c - e$.)

The Cutting Algorithm gives for the recurrent configuration c (and its minimizing configuration $\bar{c}$) a recurrent configuration d (and its minimizing configuration $\bar{d}$) such that $d \oplus \bar{d} = c \oplus \bar{c}$ and $f_{d+\bar{d}} \leq f_{c+\bar{c}}$ hold.

We start with $c + b$, work along a possible toppling sequence for $c + b$ (i.e. in each time step, we choose a site that can topple in the changed configuration) and transfer grains of sand that are left on a site z after z has toppled from c to $\bar{c}$.

As in the beginning of section 4, the chosen toppling sequence for $c + b$ is also a toppling sequence for the configuration we get after transferring grains from c to $\bar{c}$.

After each grain transfer we let the changed configuration $\bar{c}$ relax; in the end we get the configurations $d = c - e \in \mathcal{R}$ and $\bar{c} \oplus e$, which is the minimizing configuration $\bar{d}$ of d. (The configuration e here is the configuration which assigns each site the number of grains that were transferred on this site from c to $\bar{c}$.)

Now, $(c-e) \oplus (\bar{c} \oplus e) = m$ and $f_{(c-e)+(\bar{c}\oplus e)} = f_{c+\bar{c}} - f_{\bar{c}+e} \leq f_{c+\bar{c}}$.

(Note that e depends on the toppling sequence and the number of grains which are taken transferred from each site.)

We will denote the results $(c-e, \bar{c}\oplus e)$ of the Cutting Algorithm for arguments $c, \bar{c}$ as $cut(c, \bar{c})$.

The nice thing about the Cutting Algorithm is the fact that you can repeat it and thereby get better results:

After computing $(d, \bar{d}) = cut(c, \bar{c})$, it is often possible to find a configuration $e \in \mathcal{C}, e \leq d$ such that $d - e \in \mathcal{R}$ and $f_{\bar{d}+e} \neq 0$.

In pseudo code this process could be described as follows:

1: DO
2: $\quad (c, \bar{c}) \leftarrow cut(c, \bar{c})$
3: $\quad (\bar{c}, c) \leftarrow cut(\bar{c}, c)$
4: WHILE(!*exit condition*)

A local minimum is reached in $(c, \bar{c})$ if neither $cut(c, \bar{c})$ nor $cut(\bar{c}, c)$ can be a "better" pair than $(c, \bar{c})$.

The exit condition in the pseudo code program should be chosen in a way that makes it very likely that a local minimum has been reached. (A simple possibility would be to count how many times in a row the vector $f_{c+\bar{c}}$ has not changed and exit if this number is higher than a chosen threshold.)

6.3 The Local Minimum Condition

In this subsection we will show how to determine whether a pair $(c, \bar{c})$ is a local minimum. We will also be able to find a "better" pair $(d, \bar{d})$.

A pair $(c, \bar{c})$ is a local minimum if there exists no $e \in \mathcal{C}$ such that either $c - e \in \mathcal{R} \wedge f_{\bar{c}+e} \neq 0$ or $\bar{c} - e \in \mathcal{R} \wedge f_{c+e} \neq 0$ holds, since we could get a "better" pair $(c - e, \bar{c} \oplus e)$ respectively $(c \oplus e, \bar{c} - e)$ otherwise.

If $(c, \bar{c})$ is not a local minimum, then there exists a configuration $e \in \mathcal{C}$ such that $c-e \in \mathcal{R} \wedge f_{\bar{c}+e} \neq 0$ or $\bar{c}-e \in \mathcal{R} \wedge f_{c+e} \neq 0$ holds. Without loss of generality, we assume the former. Then there exists a site $z \in Z$ and a number $k \in \mathbb{N}$ such that $(\bar{c} + ke_z)(z) \geq 4$ and $c - ke_z \in \mathcal{R}$; surely $k \geq 4 - \bar{c}(z)$ holds.

It follows that $f_{\bar{c}+(4-\bar{c}(z))e_z} \neq 0$ and $c - (4 - \bar{c}(z))e_z \in \mathcal{R}$ hold, and we can formulate a local minimum condition as follows:

If $\exists e \in \mathcal{C} : \bar{c} - e \in \mathcal{R} \wedge f_{c+e} \neq 0$ holds, we can find a site $z \in Z$ such that $f_{c+(4-c(z))e_z} \neq 0$ and $\bar{c} - (4 - c(z))e_z \in \mathcal{R}$ hold.

So we can determine whether a given pair $(c, \bar{c})$ is a local minimum by looking at each site $z \in Z$ and checking for z if any of the two configurations $c - (4 - \bar{c}(z))e_z$ and $\bar{c} - (4 - c(z))e_z$ is recurrent; this has a time complexity in $O(n^4)$ and is therefore rather slow.

(One case where it is easy to verify that a local minimum has been reached: If both c and $\bar{c}$ are minimal recurrent configurations, we know that we have reached a local minimum.)

We define the function *check*$(c, \bar{c})$ which checks for all sites $z \in Z$ if $c - (4 - \bar{c}(z))e_z \in \mathcal{R}$ holds. In this case, c is set to $c - (4 - \bar{c}(z))e_z$ and $\bar{c}$ to $(\bar{c} + (4 - \bar{c}(z))e_z)_{rel}$, and at the end of the procedure the value 0 is returned. If no site z satisfies this condition, the value 1 is returned. (Note that $|f_{c+\bar{c}}|$ decreases if 0 is returned.)

6.4 Outline of the Algorithm

For a fixed number k, we set $(c, \bar{c})$ by turns to *cut*$(c, \bar{c})$ and *cut*$(\bar{c}, c)$, each time checking whether $f_{c+\bar{c}}$ changes.

After $f_{c+\bar{c}}$ has not changed for k runs of the loop, we use *check*$(c, \bar{c})$ and *check*$(\bar{c}, c)$ to see whether a local minimum has been reached or setting $(c, \bar{c})$ to a pair $(c', \bar{c'})$ with $f_{c'+\bar{c'}} \leq f_{c+\bar{c}}$ and $f_{c'+\bar{c'}} \neq f_{c+\bar{c}}$.

If a local minimum has not been reached, we again use the Cutting Algorithm until $f_{c+\bar{c}}$ has not changed for k times and use the *check* algorithm again.

This is repeated until a local minimum is reached, which eventually must happen, since $|f_{c+\bar{c}}|$ always decreases when the *check* algorithm finds that no local minimum has been reached.

7 Analysis

There are two functions used to decrease $|f_{c+\bar{c}}|$: *cut*$(c, \bar{c})$ and *check*$(c, \bar{c})$.

While *cut*$(c, \bar{c})$ is generally much faster to compute than *check*$(c, \bar{c})$, *cut*$(c, \bar{c})$ is a probabilistic function and may not reduce $|f_{c+\bar{c}}|$ although it would be possible; on the other hand, *check*$(c, \bar{c})$ reduces $|f_{c+\bar{c}}|$ if $(c, \bar{c})$ is not a local minimum, but generally needs much more time to do so than a call of *cut*$(c, \bar{c})$ does.

Therefore, we recommend using *cut*$(c, \bar{c})$ most of the time and *check*$(c, \bar{c})$ only after no improvements have been made for some time.

7.1 Strategies for Choosing the Toppling Sequence

During *cut*$(c, \bar{c})$, we have to choose a sequence for the sites of Z to topple during the relaxation of $c + b$, and for each site that toppled we have to choose how many grains we transfer to $\bar{c}$.

Several strategies have been tried, in every strategy we used rectangular distribution:

1. *Random/Biased*: If we choose a *random* strategy, we randomly choose a site z which contains at least 4 grains at each step and let it topple.

   If we choose a *biased* strategy, we choose a number k and pick up to k times a site z with at least 4 grains on it and choose the first one for which the sum of grains in both configurations is at least 8 and let it topple. If no such site is picked, we choose the last of the k sites and let it topple.
2. *Generous/Sparing*: A *generous* strategy always transfers all grains on z after it toppled from one configuration to the other.

A *sparse* strategy only transfers grains on z from one configuration to the other if this leads to topplings, and in this case only as many as necessary for a toppling to occur.

Experiments suggest that a *generous* strategy yields the better results than a *sparing* strategy, considering running time as well as $|f_{c+\bar{c}}|$.

For example, on a 200×200 grid with $k = 10$, the *random, generous* strategy yielded a result after 188 iterations of the inner loop and 2 *check* calls (which confirmed the local optimum) a configuration c for which $|f_{c+\bar{c}}| = 56865720$.

On the same grid, the *biased, generous* strategy with three tries needed 183 iterations and just two *check* calls to verify the local minimum and yielded a configuration c for which $|f_{c+\bar{c}}| = 56847147$.

The *random, sparing* strategy needed 839 iterations of the inner loop and 12 *check* calls to yield a a configuration c for which $|f_{c+\bar{c}}| = 57027080$.

8 Conclusion

We have introduced the problem of finding two recurrent configurations such that the relaxation of their sum needs as few topplings as possible.

While it is easy to find a second recurrent configuration which minimizes the number of topplings if the first configuration is given, it is non-trivial to find the global minimum among all pairs of recurrent configurations.

We have given a probabilistic algorithm which finds local minima. The performance of the algorithm depends on the strategy to choose the sequence of sites $z \in Z$ during one routine; the best strategy found is to choose the next site randomly, although further experiments concerning the quality of the various strategies should be done.

We conjecture that for each local minimum $(c, \bar{c})$ the vector $f_{c+\bar{c}}$ is minimal, meaning there are no recurrent configurations $d, \bar{d}$ such that $f_{d+\bar{d}} \leq f_{c+\bar{c}}$ and $f_{d+\bar{d}} \neq f_{c+\bar{c}}$ hold.

Other future work includes the searching for better strategies for the choosing of a toppling sequence for $c + b$ in the Cutting Algorithm and the question how hard it is to decide whether a given pair $(c, \bar{c})$ is the global optimum.

References

1. Bak, P., Tang, C., Wiesenfeld, K.: Self-organized criticality: An explanation of the 1/f noise. Phys. Rev. Lett. 59, 381–384 (1987)
2. Chung, F., Ellis, R.: A chip-firing game and dirichlet eigenvalues. Discrete Mathematics 257, 341–355 (2002)
3. Dhar, D., Ruelle, P., Sen, S., Verma, D.N.: Algebraic aspects of abelian sandpile models. J.PHYS.A 28, 805 (1995)
4. Majumdar, S.N., Dhar, D.: Equivalence between the abelian sandpile model and the $q \longrightarrow 0$ limit of the potts model. Physica A: Statistical and Theoretical Physics 185, 129–145 (1992)