Automata 2009

On The Complexity of Sandpile Prediction Problems

J. Andrés Montoya [1,2]

Escuela de Matemáticas
Universidad Industrial de Santander
Bucaramanga, Santander, Colombia

Carolina Mejía[3]

Escuela de Matemáticas
Universidad Industrial de Santander
Bucaramanga, Santander, Colombia

Abstract

In this work we study the complexity of Sandpile prediction problems on several classes of directed graphs. We focus our research on low-dimensional directed lattices. We prove some upper and lower bounds for those problems. Our approach is based on the analysis of some reachability problems related to sandpiles.

Keywords: Graph Automata, Complexity classes, Prediction Tasks, Discrete Dynamical Systems, Abelian Sandpile Model, Cellular Automata.

1 Introduction

The abelian sandpile model has been intensively studied in the physics milieu since its introduction by Bak et al [5]. This model allows us to simulate dissipative dynamical process such as forest fires, earth quakes, extinction events, the dynamic of the stock market, or avalanches [4].

One can look at The Abelian Sandpile Model as an special class of graph automata. If one assume this point of view there are several algorithmic issues that one can (has to) take into account. In [3] Babai defines *The Sandpile Prediction Problem* which we study in this paper. Babai asks for a characterization of its complexity. It is known that the problem is in P when restricted to undirected graphs, so few is known about the complexity of the problem in the directed case.

[1] Thanks to VIE-UIS and thanks to COLCIENCIAS research proyect 111548925292

[2] Email: juamonto@uis.edu.co

[3] Email: cmejiam@uis.edu.co

This paper is electronically published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

Relations to previous work and contributions. In [3] Babai defines some computational problems related to The Abelian Sandpile Model, he shows that most of those problems can be reduced to The Sandpile Prediction Problem. It follows from the work of Tardos [13] that The Sandpile Prediction Problem, and each one of the problems considered by Babai, is polynomial time solvable when restricted to undirected graphs. Babai asks for lower bounds concerning the complexity of those problems in the directed case. This problem, Babai's lower bounds problem, is still open and we have not solved it in this paper. Besides of Babai's work there exists some previous work concerning the complexity of prediction problems related to sandpiles. First at all, it is important to remark that The Abelian Sandpile Model is close related to the *Chip-firing game* of Bjorner et al [7]. In the Chip-firing game framework one can consider graphs without *sink*, and it implies that infinite long avalanches can occur. Bjorner and Lovasz studied the Chip-firing game on directed graphs [6], they obtained a fine classification of Chip-firing's possible dynamics, and as a corollary they could establish upper bounds on the algorithmic hardness of the *Chip-firing Game Prediction Problem.*

If we consider the restriction of The Sandpile Prediction Problem to undirected graphs, this problem belongs to P. Questions concerning lower bounds come from *parallel complexity*; circuit complexity and applications [4], [12]. Moore and Nilsson proved in [12], that the problem is already P-complete when restricted to lattices (grids) of dimension bigger than 2. The complexity of one-dimensional lattices is well understood thanks to Miltersen's work [11] and our contributions: we prove TC^0-hardness for *The Sandpile Reachability Problem* on one-dimensional undirected lattices, and we prove that the same problem belongs to AC^0 when restricted to one-dimensional directed lattices.

Dimension two is highly non trivial. We show that, on two-dimensional directed lattices, exponential long avalanches can occur. It implies that we can not simulate the associated sandpile automata in polynomial time. We don't know if the prediction problem on directed lattices belongs to P. In the two-dimensional case the gap between upper and lower bounds is very high. The best lower bound is L-hardness. The best upper bound comes from the best upper bound for the general case: we know that the prediction problem on directed graphs belongs to UP (*Unambiguous Polynomial Time)* [10].

We have established lower bounds for some restrictions of The Sandpile Prediction Problem: we prove L-hardness for the restrictions to planar digraphs and directed two-dimensional lattices, we prove P-hardness and $\sqrt{n}$-strict P-completeness for the restriction to acyclic digraphs.

Organization of the work. This work is organized into five sections including the introduction. In section two we introduce The Abelian Sandpile Model on directed graphs. In section three we define The Sandpile Prediction Problem and The Sandpile Reachability Problem, and we study the restrictions of these two problems to undirected graphs, planar graphs and directed acyclic graphs. In section four we study the sandpile prediction and the sandpile reachability problems when restricted to directed and undirected lattices of dimensions one and two. In section five we state some important open problems.

2 Sandpiles: The Directed Case

In this section we introduce the basic definitions and some of the basic results concerning The Abelian Sandpile Model.

Definition 2.1 A sandpile graph is a pair (G, S), where G is a finite digraph and $S \subseteq V(G)$ is a nonempty set of nodes satisfying the following two conditions.

(i) For any $v \in S$ we have that $\deg_{+}(v) = 0$, where $\deg_{+}(v)$ denotes the outdegree of G.

(ii) Given $w \in V(G) - S$, there exists a path from w to some element of S.

Given (G, S) a sandpile graph, the set S will be called the *sink*. Most of the time we will say that G is a sandpile graph and that S is the sink of G. We will use the symbol $V(G)^{*}$ to denote the set $V(G) - S$, that is: $V(G)^{*}$ denotes the set of nodes of G out of the sink. A *configuration* on G is a function $g : V(G)^{*} \rightarrow \mathbb{N}$. Given g a configuration on G and given $v \in V(G)^{*}$ we will say that v is *g-stable* if and only if $g(v) \lneqq \deg_{+}(v)$, and we will say that g is an *stable configuration* if and only if for all $v \in V(G)^{*}$, we have that v is g-stable.

Definition 2.2 Given (G, S) a sandpile graph, the sandpile automaton on G is the graph automaton $SP(G)$ defined by

(i) The set of cells of $SP(G)$ is the set $V(G)^{*}$.

(ii) The set of configurations of $SP(G)$ is the set $\{g : g \text{ is a configuration on } G\}$.

(iii) Given g a configuration of $SP(G)$ and given v a cell, the state of v under g is equal to $g(v)$.

(iv) Given g a configuration, the set of possible transitions from g is given by the following transition rule:

Given $v \in V(G)^{*}$, if $g(v) \geq \deg_{+}(v)$, then we have that $g \rightarrow g_v$ is a possible transition, where g_v is the configuration on G defined by

$$g_v(w) := \begin{cases} g(v) - \deg_{+}(v), \text{ if } w = v \\ g(w) + 1, \text{ if } v \text{ is an ancestor of } w \\ g(w) \text{ if } v \text{ is not an ancestor of } w \end{cases}$$

Any transition of $SP(G)$ is called a *firing* or a *toppling.* So, given g a configuration, the transition $g \rightarrow g_v$ is a firing, and if such a transition occurs, we say that the node v was fired (toppled) or we say that a firing (toppling) at v has occurred.

Remark 2.3 We will only consider graphs without loops and without multiple edges.

Given (G, S) a sandpile graph and given g an initial configuration, we can choose an unstable node, fire it and obtain a new configuration. Note that we can choose any unstable node to produce a firing, in this sense sandpile automata are nondeterministic. A sequence of firings $g_1 \rightarrow g_2 \rightarrow \ldots \rightarrow g_n$ is called an *avalanche* of length n with initial configuration g, and we say that it is an avalanche from g to

g_n. If g_n is stable we say that g_n is a *stabilization* or a *relaxation* of g. If we fix a configuration g on V, we can consider the following three sets: $Aval\,(G, g)$, the set of avalanches whose initial configuration is g; $Aval_M\,(G, g)$ the set of maximal avalanches beginning in g (A is maximal if and only if A can not be extended, that is: A is maximal if and only if its final configuration is stable); $st\,(G, g)$ the set of relaxations of g.

Furthermore, given G, g and

$$A = g \to g_1 \to ... \to g_n$$

an avalanche, the *score vector* of A, which we denote SC_A, is equal to $(t_v)_{v \in V(G)^*}$, where for any $v \in V\,(G)^*$ the entry t_v is equal to the number of times node v was fired during the occurrence of A.

Theorem 2.4 *(The fundamental theorem of sandpiles)*

Let (G, S) be a sandpile graph and let g be a configuration, we have:

(i) *Any avalanche beginning in g is finite.*

(ii) $|st\,(G, g)| = 1$.

(iii) *Given A,$B \in Aval_M\,(G, g)$, we have that $SC_A = SC_B$.*

A proof of this theorem can be found in [14]. Theorem 2.4 says many things about sandpile automata. Item 1 says that sandpile automata are *terminating*. Item 2 says that sandpile automata are *confluent*, i.e. the input (the initial configuration) determines an unique output (its stabilization). Item 3 says that, though there are many computation paths, sandpile automata are strongly deterministic, since given $SP\,(G)$ a sandpile automaton and given two computation paths of $SP\,(G)$ on input g, the second path is simply a permutation of the first, and as a consequence they have the same length.

3 The Sandpile Prediction Problem

Given $C\,(G) = \mathbb{N}^{V(G)^*}$ the set of all the configurations on G and given $st\,(G)$ the set of all the stable configurations on G, we can define two functions $st_G : C\,(G) \to st\,(G)$ and $SC_G : C\,(G) \to C\,(G)$ in the following way:

(i) $st_G\,(g) :=$ the stabilization of g.

(ii) $SC_G\,(g) := SC_A$, where A is any element of $Aval_M\,(G, g)$.

Note that, for any sandpile graph (G, S), the functions st_G and SC_G are computable, since the avalanches are always finite. Given g a configuration on G, in order to compute either $st_G\,(g)$ or $SC_G\,(g)$, we only have to simulate the automaton $SP\,(G)$ on input g.

Notation 3.1 Given $g \in C\,(G)$ we will use the symbol SC_g to denote the vector $SC_G\,(g)$.

We can obtain as an easy consequence of the invariance of the score vector the following theorem

Theorem 3.2 *Given (G,S) a sandpile graph and given f_1, f_2 and f_3 three configurations, we have that*

(i) $st_G(f_1+f_2+f_3) = st_G(st_G(f_1+f_2)+f_3)$.

(ii) $st_G(f_1+f_2) = st_G(f_2+f_1) = st_G(st_G(f_2)+st_G(f_1))$.

We are ready to define The Sandpile Prediction Problem, with its definition we finish this subsection

Problem 3.3 *(SPP, sandpile prediction)*

- *Input: A sandpile graph (G,S) and an initial configuration $g \in C(G)$.*
- *Problem: Compute $st_G(g)$.*

3.1 Sandpile Reachability

Let us introduce a second computational problem related to sandpile prediction.

Problem 3.4 *(SPR, sandpile reachability)*

- *Input: $((G,S), g, v)$, where (G,S) is a sandpile graph, g is a configuration on G and $v \in V(G)^*$.*
- *Problem: Decide if $SC_g(v) \geq 1$.*

Theorem 3.5 *SPR is reducible to SPP*

Proof. Given (G,S) a sandpile graph on $\{1,...,n,n+1,...,n+k\}$, whose sink S is equal to $\{n+1,...,n+k\}$, the *laplacian* of (G,S) is the matrix $L^S(G) = [a_{ij}]_{i,j\leq n}$ defined by

(i) Given $i \leq n$ we have that $a_{ii} = -\deg_+(i)$.

(ii) Given $i \neq j$ we have that $a_{ij} = 1$ if $(i,j) \in E(G)$, otherwise $a_{ij} = 0$.

Suppose that g is a configuration on G, we can think of g as an element of $\mathbb{N}^n$. If node v fires, we obtain a new configuration g_v. Note that $g_v = g + L_v^S(G)$, where $L_v^S(G)$ is the *v-th* row of $L^S(G)$. Thus, we have that for any configuration g

$$st_G(g) = g + \left(L^S(G)\right)^T (SC_g)$$

where $\left(L^S(G)\right)^T$ is the transposition of $L^S(G)$. We call last equation the *motion equation* of sandpiles. This equation has many important consequences, it allows us to compute in polynomial time the vector $st_G(g)$ if oracle access to SC_g is provided. Interesting enough the reciprocal is true, that is: given $st_G(g)$ we can compute in polynomial time the vector SC_g. Suppose we can solve the problem SPP in polynomial time. First at all we compute (in polynomial time) the vector $st_G(g)$. Now we note that $L^S(G)$ is a non singular matrix, (according to *Kirchhoff's matrix theorem*, see reference [14], we have that $\left|\det\left(L^S(G)\right)\right|$ is equal to the number of spanning forests of G directed to S. Note that this quantity is not zero, since S is accessible from any node $v \in V(G)^*$). Then, we have that the linear system

$$st_G(g) - g = \left(L^S(G)\right)^T X$$

has a unique solution which is equal to SC_g. Hence, we can solve the linear system and compute SC_g. Therefore, we have that we can solve SPR in polynomial time, (if oracle access to SPP is provided). By the way, we have proven that these two vectors associated to the pair (G, g) are polynomial time equivalent (the computations of these two vectors are Turing equivalent). □

3.2 Restrictions of SPP

Given $\mathcal{C}$ a class of sandpile graphs, $SPP[\mathcal{C}]$ will denote the restriction of SPP to the class $\mathcal{C}$, (we define $SPR[\mathcal{C}]$ accordingly). Along this paper we will consider several such restrictions.

3.2.1 ***Undirected Graphs***

Let $\mathcal{U}$ be the class of undirected sandpiles. Note that, we can think of an undirected graph as if it were a symmetric digraph. Furthermore, we can characterize the class of undirected sandpile graphs as the class of symmetric sandpile digraphs: an undirected sandpile graph is a sandpile digraph (G, S) such that $G - S$ is symmetric.

Theorem 3.6 *$SPP[\mathcal{U}]$ is P-complete.*

Proof. Given $((G, S), g)$ an input of $SPP[\mathcal{U}]$, we can compute $st_G(g)$ by running the sandpile automaton $SP(G)$ on input g. This computation is polynomial time bounded, since Tardos' bound [13] implies that the size of any avalanche beginning in g is polynomial bounded, and the simulation of any toppling can be carried out in linear time (constant time on bounded degree graphs, constant time on a parallel computer with $\deg(G)$ processors, constant time on the sandpile automaton $SP(G)$). Thus, we have that $SPP[\mathcal{U}] \in P$.

Moore and Nilsson [12] proved that SPP is P-hard when restricted to undirected grids of dimension 3. It implies that $SPP[\mathcal{U}]$ is P-hard. □

3.2.2 ***Acyclic Graphs***

Let $\mathcal{A}$ be the class of acyclic sandpiles. Because of the confluence of avalanches we can fix, in advance, an evaluation scheme.

Definition 3.7 (order evaluation scheme)

Let (G, S) be a sandpile graph, if $\preceq$ is an ordering of $V(G)^*$, we define the order evaluation scheme of $SP(G)$, given the ordering $\preceq$, in the following way:

At any stage of the computation we look for the $\preceq$-lowest unstable vertex, and we fire it.

Theorem 3.8 $SPP[\mathcal{A}] \in P$.

Proof. Let M be the algorithm defined by:

Given (G, S) an acyclic sandpile graph and given $g \in C(G)$, the algorithm M works on input $((G, S), g)$ in the following way.

(i) M computes $\preceq$, a topological order of G.

(ii) Until all nodes in $V(G)^*$ become stable, the algorithm M computes the $\preceq$-lowest unstable node, and M fires it.

Suppose $\|g\| = \sum_{v \in V(G)^*} g(v)$. It is easy to verify that the sandpile triple $((G,S), g)$ becomes stable after at most $|V(G)^*| \, \|g\|$ firings. Hence, M is a polynomial time algorithm (the size of $((G,S), g)$ is equal to $|V(G)| \, \|g\|$). □

Remark 3.9 Suppose that M is the algorithm used in the proof of theorem 3.5. Note that, given $((G,S), g, v)$ an instance of SPR, the algorithm M, on input $((G,S), g, v)$, makes a single query to the oracle, this single query is equal to: what is $st_G(g)$? It implies that for any class $\mathcal{C}$ of sandpile graphs the problem $SPR[\mathcal{C}]$ is Turing reducible to $SPP[\mathcal{C}]$.

Theorem 3.10 *$SPR[\mathcal{A}]$ is P-complete*

Proof. We show that the evaluation of monotone boolean circuits is $\log space$ reducible to $SPR[\mathcal{A}]$. The reduction is very similar to the reduction used by Moore and Nilsson in [12]. Moore and Nilsson defined a reduction from the *monotone circuit value* problem, ($MCVP$ for short), to $SPP[\mathcal{U}]$. Our reduction is a little bit more elementary, since directed edges allow us to avoid the construction of *diodes*, which are the gadgets used by Moore and Nilsson to force *irreversibility on chips' movement*. Furthermore, we have used the orientations of our edges to achieve a reduction of $MCVP$ into $SPR[\mathcal{A}]$, which seems to be a problem easier than $SPP[\mathcal{A}]$.

Let (C, f) be an instance of $MCVP$. We can suppose, without loss of generality, that C is stratified and that for any inner gate of C, say v, we have that $\deg_-(v) = 2$ and $\deg_+(v) \neq 1$. We have to design some gadgets which allow us to simulate: and gates, or gates, the output gate and the input gates. The simulation is based on the following analogy

After its evaluation, gate v sends a true-signal to gate w, (along the wire connecting v to *w), if and only if node v fires and sends a chip to node w, (along the edge connecting v to w)*

So, we define a sandpile graph (G,S) whose nodes are the gates of C plus an additional node s. Furthermore, the directed edges of G correspond to the directed wires of C plus the edge (o,s), where o is the output gate of C. Finally we set

$$S = \{s\} \cup \left\{v : \deg_+(v) = 0\right\}$$

To simulate the boolean type(role) of each one of the gates we play with the values of g.

(i) If c is a $\wedge$-gate whose outdegree is equal to $n \neq 0$, we place $n-2$ chips on c, that is we set $g(c) = n - 2 = \deg_+(c) - 2$.

(ii) If c is an $\vee$-gate whose outdegree is equal to $m \neq 0$, we place $m-1$ chips on c, that is we define $g(c) = m - 1 = \deg_+(c) - 1$.

(iii) If c is an input gate whose outdegree is equal to m. Then, if $f(c) = 1$ we place m chips on c , otherwise we do not place chips on c, that is we set $g(c) = f(c)\deg_+(c)$.

Given (C, f), the algorithm M can compute the tuple $((G,S), g, o)$ using logarithmic space. It is easy to check that f satisfies C if and only if $SC_g(o) \geq 1$. □

Remark 3.11 The same proof can be used to prove that $SPR[\mathcal{A}]$ is strictly $\sqrt{n}$-hard for P (more information can be found in [8]).

3.2.3 ***Planar Sandpiles***

Let $\mathcal{P}$ be the class of planar sandpile digraphs. We can use the same reduction of theorem 3.10 to prove that the planar restriction of $MCVP$ ($MPCVP$ for short) is NC^1-reducible to $SPR[\mathcal{P}]$. It is known (see reference [9]) that $MPCVP$ is L-hard under quantifier free reductions, (hence L-hard under NC^1-reductions).

We will prove that *The Directed Planar Reachability Problem* ($REACH(planar)$ for short) is NC^1-reducible to $SPR[\mathcal{P}]$.

Problem 3.12 *($REACH(planar)$, directed planar reachability)*

- *Input: (G, s, t), where G is a planar digraph and $s, t \in V(G)$.*
- *Problem: Decides if t is accessible from s.*

Remark 3.13 The problem $REACH(planar)$ is L-hard under uniform projections [1], and it is a problem supposed to be harder than $MPCVP$.

Given G a planar graph, a *combinatorial embedding* of G is an algorithm c_G, which on input $v \in V(G)$ outputs a cyclic ordering of the edges that are incident with v. We will suppose that, given (G, s, t) an instance of $REACH(planar)$, a combinatorial embedding c_G of G is attached to the code of (G, s, t).

Remark 3.14 Combinatorial embeddings of planar graphs can be computed using logarithmic space (see reference [1]).

Let $REACH(\mathcal{P})$ be the directed planar reachability problem restricted to $\mathcal{P}$.

Theorem 3.15 *$REACH(planar)$ is NC^1-reducible to $REACH(\mathcal{P})$.*

Proof. Let (G, s, t) be an instance of $REACH(planar)$. Let G_u be the undirected planar graph obtained from G by forgetting the orientations of its edges. First at all we add a center node to any face of G. We can use c_{G_u} to recognize the faces of G_u. To this end we can use the following procedure.

Given v a node and given e an edge incident with v, we go from v to u, the other end of e. After entering u we choose the next edge (next to e in the cyclic ordering $c_{G_u}(u)$). We continue our walk till node v is reached again, along the walk we obey the following rule:

each time we enter a node, we leave it using the next edge

The nodes which we visit, and the edges that we use along a single walk constitute a single face. So, given v and e we can compute one of the two faces determined by v and e. If we use last procedure on any pair (v, e) we can compute all the faces of G.

Given F a face of G, we add a new node v_F, and given u a node which lies on the border of F we add the edge (u, v_F). Let H_G be the output graph of our procedure.

Note that H_G was obtained from G by adding a center-sink node to any face of G, and by connecting each one of those center-sink nodes to the nodes located on the borders of their corresponding faces (the edges being directed from the border to the center). Furthermore, given $s, t \in V(G)$, we have that $(G, s, t) \in REACH\,(planar)$ if and only if $(H_G, s, t) \in REACH\,(\mathcal{P})$. What have we gained with the construction of H_G? Note that H_G is a planar sandpile graph, since each one of the centers is a sink node, and any node is either a sink or can reach at least one center-sink in one step.

We can suppose, without loss of generality, that any face of G has $O(\log(|V(G)|))$ edges. Thus, we have proven that $REACH\,(planar)$ is NC^1-reducible to $REACH\,(\mathcal{P})$. □

Let (G, s, t) be an instance of $REACH$ (the reachability problem on general digraphs). We use the symbol $N(t)$ to denote the set of ancestors of t, and we use the symbol $\varphi(s, t, m)$ to denote the sentence $\exists v \in N(t)\,(SC_{me_s}(v) \geq 1)$. We define the *sandpile distance* between s and t in the following way.

$$\delta_G(s,t) = \begin{cases} \text{If } d_G(s,t) \lneqq \infty. \text{ Then } \delta_G(s,t) = \min_m \{\varphi(s,t,m)\} \\ \qquad\qquad\qquad\qquad \infty \text{ else} \end{cases}$$

Note that $\delta_G(s, t)$ is equal to the minimum number of chips one have to place on s in order to fire at least one neighbor of t.

Lemma 3.16 *$(G, s, t) \in REACH\,(\mathcal{P})$ if and only if $\delta_G(s, t) \leq n^n$.*

Proof. Suppose that $(G, s, t) \in REACH\,(\mathcal{P})$. Let $s, v_1, v_2, ..., v_m, t$ be a simple path from s to t and let D be an upper bound on the outdegrees of the nodes $s, v_1, v_2, ..., v_m$. Note that $m, D \leq n - 1$. Suppose that there are n^n chips placed on s. We fire s until we place at least n^{n-1} chips on v_1. Then, we fire v_1 until we place at least n^{n-2} chips on v_2 and so on. We can show, using an inductive argument, that we can place at least $n^{n-m} \geq n$ chips on v_m. Hence, we can fire v_m, since $\deg_+(v_m) \leq n - 1$ and $n - m \geq 1$. It is clear that if $(G, s, t) \notin REACH\,(\mathcal{P})$, then $\delta_G(s, t) = \infty$. Thus, we have proven that $(G, s, t) \in REACH\,(\mathcal{P})$ if and only if $\delta_G(s, t) \leq n^n$. □

Corollary 3.17 *$REACH\,(\mathcal{P})$ is NC^1- reducible to $SPR\,[\mathcal{P}]$.*

4 Sandpile Lattices

In this section we consider a very special class of sandpile graphs, the class of sandpile lattices. We focus our research on sandpile reachability problems.

A directed lattice of dimension d is an orientation of a finite hypercubic lattice (grid) of dimension d (square lattice if $d = 2$, cubic lattice if $d = 3$). A sandpile lattice is a sandpile graph $H = (G, S)$ such that G is a directed lattice

Let $SPR\,[d]$ denotes the problem SPR restricted to d-dimensional sandpile lattices, we want to analyze the complexity of $SPR\,[d]$.

Theorem 4.1 *If $d \geq 3$, we have that $SPR\,[d]$ is P-hard.*

Proof. To prove the theorem we can use the same reduction of theorem 3.10. It is sufficient to note that given an acyclic digraph G, there exists a, suitable, log *space* computable embedding of G into a sandpile lattice of dimension 3 (see reference [12]). □

From now on, we will focus our analysis on low-dimensional lattices

4.1 One-dimensional lattices

The behavior of directed sandpiles seems to be strictly more complex than the behavior of undirected sandpiles. This is true if we consider either general graphs or d-dimensional sandpile lattices with $d \geq 2$. It is an interesting fact that in the one-dimensional case we have just the opposite fact. Let $SPR_u[1]$ be the sandpile reachability problem restricted to undirected one-dimensional lattices. We will show that $SPR_u[1]$ is hard for TC^0 with respect to constant depth reductions. Hence, $SPR_u[1]$ is not in $AC^{1-\epsilon}$ for any $\epsilon \gneqq 0$. On the other hand, we will show that $SPR[1]$ belongs to AC^0. Therefore, we will have that $SPR[1]$ is strictly easier than $SPR_u[1]$.

4.1.1 The lower bound: TC^0-hardness of $SPR_u[1]$

Let (G_n, S_n) be the one-dimensional undirected sandpile lattice on $\{0, 1, ..., 3n+1\}$, (we have that $S_n = \{0, 3n+1\}$). Suppose that g is a configuration on G_n which satisfies the following three conditions.

(i) If $i \leq n$, then we have $g(i) = 0$.

(ii) If $i \in \{n+1, ..., 2n\}$, then we have $g(i) \in \{1, 2\}$.

(iii) If $i \geq 2n+1$, then we have $g(i) = 0$.

Let $\|g\| = \sum_i g(i) \leq 2n$

Theorem 4.2 *There exist numbers $i, j \in \{0, 1, ..., 3n+1\}$ such that*

(i) *$i \lneqq j$ and $j - i \in \{\|g\|, \|g\| - 1\}$*

(ii) *If $k \notin \{i, i+1, ..., j\}$, then $st_{G_n}(g)(k) = 0$.*

(iii) *If $k \in \{i, i+1, ..., j\}$, then $st_{G_n}(g)(k) \in \{0, 1\}$; and there exists at most one k such that $i \lneqq k \lneqq j$ and $st_{G_n}(g)(k) = 0$.*

Proof. Let $i_1 \leq i_2 \leq ... \leq i_k$ be the positions where the value of g is equal to 2. Let g_0 be the configuration which takes the value 1 on the set $\{n+1, ..., 2n\}$ and the value 0 on its complement. Note that

$$g = g_0 + e_{i_1} + ... + e_{i_k}$$

The abelian property of the *abelian* sandpile model (3.2) implies that

$$st_G(g) = st_G(g_{k-1} + e_{i_k})$$

where $g_1 = st_G(g_0 + e_{i_1})$ and given g_{i_r} we have that $g_{r+1} = st_G\left(g_r + e_{i_{r+1}}\right)$. First at all we try to compute g_1. It is easy to check that g_1 is a configuration constituted by a zero floating in a connected sea of ones, and that the position of the isolated zero is the mass center of the configuration $g_0 + e_{i_1}$, that is: we have a position $j_1 \in \{n+1, ..., 2n\}$ such that $g_1(j) = 0$. Furthermore, we have that if $j \in \{n+1, ..., 2n\} - \{j_1\}$, then $g_1(j) = 1$; and either $g_1(n) = 1$ or $g_1(2n+1) = 1$.

Now, we try to compute g_2. If $j_1 = i_2$, then there exists an interval $I_1 \supseteq \{n+1, ..., 2n\}$ such that $g_1 + e_{i_2}$ takes the value 1 on I_1 and the value 0 out of I_1. In this case $g_1 + e_{i_2}$ is already a stable configuration and is equal to g_2. If $j_1 \neq i_2$, then there exists an interval $I_1 \supseteq \{n+1, ..., 2n\}$ such that $g_1 + e_{i_2}$ takes the value 0 out of I_1; $(g_1 + e_{i_2})(j_1) = 0$; $(g_1 + e_{i_2})(i_2) = 2$; and $g_1 + e_{i_2}$ takes the value 1 on any other point of I_1. So, the configuration $g_1 + e_{i_2}$ looks like a zero and a two floating in a connected and isolated sea of ones. It is easy to check that $g_2 = st_G(g_1 + e_{i_2})$ is a stable configuration of one of the following two types:

(i) (Type 1) There exists an interval $I_2 \supseteq \{n+1, ..., 2n\}$ such that $st_G(g_1 + e_{i_2})$ takes the value 0 out of I_2 and the value 1 on I_2. Furthermore, the length of I_2 is $n+2$.

(ii) (Type 2) There exists an interval $I_2 \supseteq \{n+1, ..., 2n\}$ such that $st_G(g_1 + e_{i_2})$ takes the value 0 out of I_2 and the value 1 on $I_2 - \{x\}$, where $x \in I_2$ and $st_G(g_1 + e_{i_2})(x) = 0$. Furthermore, the length of I_2 is $n+3$.

At this point, it should be clear that we can use an inductive argument to prove that for any $j \leq k$, the configuration g_j is a configuration of one of the following two types:

(i) (Type 1) There exists an interval $I_j \supseteq \{n+1, ..., 2n\}$ such that g_j takes the value 0 out of I_j and the value 1 on I_j. Furthermore, the length of I_j is $n+j$.

(ii) (Type 2) There exists an interval $I_j \supseteq \{n+1, ..., 2n\}$ such that g_j takes the value 0 out of I_j and the value 1 on $I_j - \{x\}$, where $x \in I_j$ and $g_j(x) = 0$. Furthermore, the length of I_j is $n+j+1$.

If we take $j = k$ we obtain the theorem. □

We are ready to prove the main theorem of this subsection. We will prove that the computation of the majority function is constant depth reducible to $SPR_u[1]$.

Given $(x_1, ..., x_n)$, we have that $Maj(x_1, ..., x_n) = 1$ if and only if $\sum x_i \geq \lfloor \frac{n}{2} \rfloor + 1$. Note that $Maj(x_1, ..., x_n, x_{n+1} = x_1, ..., x_{2n} = x_n) = 1$ if and only if $\sum x_i \geq n+1$ if and only if $\sum x_i \gneqq n$.

Theorem 4.3 *$SPR_u[1]$ is TC^0-hard.*

Proof. We show that the computation of the majority function is constant depth reducible to $SPR_u[1]$. Suppose that $x = (x_1, ..., x_n)$ is a boolean vector. Let $m = 2n$ and let $(y_1, ..., y_m) = (x_1, ..., x_n, x_1, ..., x_n)$. We define a configuration g_x on $\{0, 1, ..., 3m+1\}$ as follows

$$g_x(i) = \begin{cases} y_j + 1 \text{ if } i = m + j \text{ and } j \in \{1, ..., m\} \\ 0 \text{ else} \end{cases}$$

Note that g_x satisfies the conditions in the statement of theorem 4.2. Let us call the *shadow* of g_x to the area out of $\{m+1, ..., 2m\}$ that will be filled with chips after the relaxation process. If $Maj\,(x) = 1$, then the shadow of g_x will be large, it will fill at least $n+2$ positions. On the other side, if $Maj\,(x) = 0$, then the shadow of g_x will be small, it will fill at most $n+1$ positions. Let A_m be equal to the set

$$\{(i,j) : 0 \leq i \leq m \ \&\ j \gneqq 2m \ \&\ j - i \geq m + n + 1\}$$

Note that $Maj\,(x_1, ..., x_n) = 1$ if and only if

$$\bigvee_{(i,j)\in A_m} (((G_m, g, i+1) \in SPR_u\,[1]) \wedge ((G_m, g, j-1) \in SPR_u\,[1]))$$

Thus, we have proven that we can compute the majority function using a $D\log time$ uniform family of depth-three circuits, with an or gate on the top; a second layer composed by and-gates and a first layer composed by $SPR_u\,[1]$ oracle gates. Therefore, we have that $SPR_u\,[1]$ is TC^0-hard. □

Remark 4.4 (upper bounds) It follows from the work of Moore and Nilsson that $SPR_u\,[1]$ belongs to NC^3.

4.1.2 The upper bound

In this subsection we prove an upper bound for $SPR\,[1]$, specifically we prove that $SPR\,[1]$ belongs to AC^0.

Theorem 4.5 *$SPR\,[1]$ belongs to AC^0.*

Proof. Let G be a directed one dimensional lattice, we suppose that the universe of G is equal to $\{1, ..., n\}$. Note that, the nodes of G can be classified into three groups as follows:

(i) *Source nodes*, which are nodes whose outdegree is 2.

(ii) *Dummy nodes*, which are nodes whose outdegree is 1.

(iii) *Sink nodes*, which are nodes whose outdegree is 0.

We will suppose that any node is an inner node, i.e. in order to avoid some technicalities we will impose *periodic conditions* on our one-dimensional lattices. Additionally we will suppose that

$$S\,(G) = \{v : v \text{ is a sink node}\}$$

is a nonempty set. Let g be a configuration on G and let i be a node, note that

(i) If i is a source node, then $SC_g\,(i) \geq 1$ if and only if $g\,(i) \geq 2$.

(ii) If i is a dummy node, there exists a pair $j, k \in \{1, ..., n\}$ such that $j \lneqq i \lneqq k$; for any l such that $j \lneqq l \lneqq k$ we have that l is a dummy node; and one of the two nodes j, k is a source, while the other one is a sink. Furthermore, we have that if $x \in \{j, k\}$ is a source, then $SC_g\,(i) \geq 1$ if and only if either $g\,(x) \geq 2$ or there exists j in the interval between x and i (the interval $(x, i]$, which includes i and excludes x) such that $g\,(j) \geq 1$.

(iii) If i is a sink node, then we have that $SC_g(i) = 0$.

Given $((G,S),g)$, we codify $((G,S),g)$ as a vector $t_{G,g} = (t_i)_{i \leq n}$, where for any $i \leq n$ we have that t_i is equal to the boolean triple (l_i, r_i, g_i) defined by

(i) $l_i = 1$ if and only if the edge on the left of i is an outgoing edge.

(ii) $r_i = 1$ if and only if the edge on the right of i is an outgoing edge.

(iii) $g_i = 1^{g(i)}$.

Given $k, i, h \leq n$ (we suppose that $1 \lneqq i \lneqq n$), the boolean formula α_{kih} is defined in the following way:

(i) If $k \lneqq i \lneqq h$ we have that α_{kih} is equal to $\beta_{kih} \vee \gamma_{kih}$, where β_{kih} is the formula

$$(l_k \wedge r_k) \wedge (\sim l_i \wedge r_i) \wedge (\sim l_h \wedge \sim r_h) \wedge \left((g_k \geq 2) \vee \left(\bigvee_{k \lneqq j \leq i} g_j \geq 1 \right) \right)$$

$$\wedge$$

$$\left(\bigwedge_{k \lneqq j \lneqq i} (\sim l_j \wedge r_j) \right) \wedge \left(\bigwedge_{i \lneqq j \lneqq h} (\sim l_j \wedge r_j) \right)$$

The formula γ_{kih} is defined accordingly (interchanging the role of k and h).

(ii) If $k = i = h$, then $\alpha_{iii} = (l_i \wedge r_i \wedge g_i \geq 2)$

Finally we define a formula ψ_i in the following way

$$\psi_i = \alpha_{iii} \vee \left(\bigvee_{k \lneqq i \lneqq h} \alpha_{kih} \right)$$

It is easy to check that $\psi_i(t_{G,g}) = 1$ if and only if $SC_g(i) \geq 1$. Note that the formula ψ_i has a depth which is upperbounded by 6. Thus, we have proven that we can solve the problem $SPR[1]$ using a $D \log time$ uniform family of circuits of bounded depth and defined on the logical basis $\{\wedge, \vee, \sim\}$, i.e. we have proven that $SPR[1]$ belongs to AC^0. □

4.2 Two-dimensional lattices

In this section we will analyze the problems $SPR[2]$ and $SPP[2]$. We will prove that:

(i) On two-dimensional sandpile directed lattices exponential long avalanches can occur.

(ii) $SPR[2]$ is L-hard under NC^1 reductions.

4.2.1 Long Avalanches

In this section we prove that there are not polynomial bound on the size of the avalanches, when we consider the class of two-dimensional sandpile lattices. This theorem rules out the possibility of using the naive *sandpile automata simulation algorithm* to solve, in polynomial time, the problems $SPP[2]$ and $SPR[2]$.

Let (G,S) be a sandpile graph such that $S=\{s\}$ and there exists a path $v_0,v_1,v_2,...,v_n,s$ with the following three properties:

(i) $\deg_+(v_0)=1$.

(ii) For any i, if $1\leq i\leq n$, then $\deg_+(v_i)\geq 2$.

(iii) For any $i\geq 0$ we have that $\deg_-(v_i)=1$.

Lemma 4.6 *Given $g=\left(|G|^2+1\right)e_{v_0}$, the length of any maximal avalanche triggered by g is lowerbounded by 2^n.*

Proof. First we note that, in order to stabilize the sandpile it is necessary to throw at least one chip trough the sink. It implies that $SC_g(v_n)\geq 1$. Note that in order to place one chip on v_n we have to fire v_{n-1} at least one time. Hence, one toppling at node v_n forces at least two topplings at node v_{n-1}. Two topplings at node v_{n-1} forces at least four topplings at node v_{n-2}, and so on. We can show, using an inductive argument, that one toppling at node v_n forces at least 2^n topplings at node v_0. Thus, we have proven that the length of any maximal avalanche with initial configuration $\left(|G|^2+1\right)e_{v_0}$ is bigger than 2^n. □

Theorem 4.7 *There is not polynomial bound on the size of the avalanches for The Abelian Sandpile Model on two-dimensional sandpile directed lattices.*

Proof. First at all we define $((G_n,S_n))_{n\geq 1}$ a sequence of two-dimensional sandpile directed lattices. Given $n\geq 1$ we define (G_n,S_n) in the following way:

(i) $V(G_n)=\{(m,i):m\leq n+1 \text{ and } i\in\{0,1\}\}$.

(ii) $E(G_n)=A_1\cup A_2\cup A_3$, where

$$\begin{aligned} A_1&=\{((m,0),(m+1,0)):m\leq n\}\\ A_2&=\{((m,1),(m-1,1)):1\leq m\leq n+1\}\\ A_3&=\{((m,0),(m,1)):1\leq m\leq n\}\\ &\qquad\cup\{((0,1),(0,0)),((n+1,1),(n+1,0))\}. \end{aligned}$$

(iii) $S_n=\{(n+1,0)\}$.

Note that the path $(0,0),(1,0),...,(n+1,0)$ satisfies the conditions in the statement of lemma 4.6, and note that $|G_n|=2n+4$. From lemma 4.6 we have that the length of any maximal avalanche beginning in $g_n=\left((2n+4)^2+1\right)e_{(0,0)}$ is lowerbounded by 2^n. □

Remark 4.8 Note that if we define g_n as $(n+1)e_{(0,0)}$ we obtain the same lower bound on the length of the maximal avalanches triggered by g_n.

Last theorem rules out the possibility of solving in polynomial time the problem $SPP[2]$ by means of the naive sandpile automata simulation algorithm. It does not imply that we can not solve $SPP[2]$ in polynomial time, note that, with some effort we could compute a closed-form formula (of low arithmetical complexity) for the function $h : \mathbb{N} \to \mathbb{N}^{V(G_n)^*}$ defined by $h(n) = st_{G_n}(g_n)$ [4]

So, we can predict, (even better than in polynomial time), the final states of our exponential long avalanches. Can we always predict? At the moment we do not know if $SPP[2]$ belongs to P, this problem could be intractable, but we conjecture that $SPP[2] \in P$.

Remark 4.9 In this paper we have chosen a sequential updating protocol (we choose one unstable node v and we fire v). There are other updating protocols, like for example the *parallel updating protocol* (we fire all unstable nodes at once). Those others protocols yield only polynomial speed ups. Hence, exponential long avalanches remains exponential long and *exponential problematic.*

4.2.2 Lower bounds for $SPR[2]$

In this section we will prove that $SPR[2]$ is L-hard under NC^1 reductions. Consider the following problem

Problem 4.10 *($REACH(grids)$, directed grid reachability)*

- *Input: (G, s, t), where G is a grid digraph and $s, t \in V(G)$.*
- *Problem: Decides if t is accessible from s.*

In this subsection we prove that this problem is NC^1 reducible to $SPR[2]$. It is known that the problem $REACH(grids)$ is L-hard under quantifier free reductions (see reference [2]).

Theorem 4.11 *$REACH(grids)$ is NC^1 reducible to $SPR[2]$.*

Proof. We can work along the lines of the proofs of theorem 3.15 and corollary 3.17. It is important to remark that in the case of grid graphs we don't have to compute the faces, which is the most consuming part of the reductions used in theorem 3.15 and corollary 3.17. □

Corollary 4.12 *$SPR[2]$ is $L-hard$ under NC^1 reductions.*

5 Open problems

Let us finish this work stating some interesting open problems.

(i) We know that $SPR[2]$ is L-hard, but the best upper bound is UP-computability. Improve the lower and(or) upper bounds for $SPR[2]$. We conjecture that $SPR[2]$ belongs to P.

(ii) We know that $SPR[\mathcal{P}]$ is L-hard, but the best upper bound is UP-computability. Improve the lower and(or) upper bounds for $SPR[\mathcal{P}]$.

(iii) We know that if $d \geq 3$, SPR and $SPR[d]$ are equivalent and P-hard, but we don't know if SPR belongs to P. Prove that SPR belongs to P.

[4] Note that given $n, m \geq 1$ and given $g_{n,m} = me_{(0,0)} \in C(G_n)$, the relaxation of g_n is the configuration g_n^* defined by

$$g_n^*((k,i)) = \begin{cases} 1, & \text{if } 1 \leq k \leq (m \mod (n+1)), \\ 0, & \text{in otherwise} \end{cases}$$

References

[1] E. Allender, S. Datta, S. Roy. The Directed Planar Reachability Problem. *Proceedings FST & TCS, LNCS 3821:* 238-249, 2005.

[2] E. Allender, D. Barrintong, T. Chakraborty, S. Datta, S. Roy. Grid Graph Reachability Problem. In *Electronic Colloquium on Computational Complexity.* Report No. 149 (2005).

[3] L. Babai. The abelian sandpile model. Manuscript, available at http://people.cs.uchicago.edu/~laci/REU05/.

[4] P. Bak. *How nature works: The science of self-organized criticality.* Copernicus, NY, 1996.

[5] P. Bak, C. Tang, K. Wiesenfeld. Self Organized Criticality. *Physical Review A,* 38:364-374 (1988).

[6] A. Bjorner, L. Lovasz. Chip-firing Games on Directed Graphs. *J. Algebraic Combinatorics.* 1:305-328, 1992.

[7] A. Bjorner, L. Lovasz, P. Shor. Chip-firing Games on Graphs. *Europ. J. Comb.* 12:283-291, 1991.

[8] A. Condon. A Theory of Strict P-completeness. STACS proceedings:33-44, 1992.

[9] T. Chakraborty, S. Datta. One-input-face MPCVP is hard for L, but in $LogDCFL$. In *Electronic Colloquium on Computational Complexity.* Report No. 130 (2006).

[10] C. Mejía. El modelo de pilas de arena, el caso dirigido. Revista Integración-UIS, 24(2):101-116, 2006 (in spanish).

[11] P. Miltersen. The computational complexity of one-dimensional sandpiles. *Theory of computing systems.* 41(1):119-125, 2007.

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

[13] G. Tardos. Polynomial bound for a chip firing game on graphs. *SIAM J. Discrete Mathematics.* 1:397-398, 1988.

[14] E. Toumpakari. *On the abelian sandpile model.* Ph.D. Thesis, University of Chicago, 2005.