An efficient algorithm for generating symmetric ice piles*

Roberto Mantaci[1], Paolo Massazza[2], and Jean-Baptiste Yunès[1]

[1] LIAFA, Université Paris 7 Denis Diderot, Case 7014, 75205 Paris Cedex 13, France
Roberto.Mantaci@liafa.univ-paris-diderot.fr,
Jean-Baptiste.Yunes@univ-paris-diderot.fr

[2] Università degli Studi dell'Insubria, Dipartimento di Scienze Teoriche e Applicate - Sezione Informatica, Via Mazzini 5, 21100 Varese, Italy
paolo.massazza@uninsubria.it

Abstract. We define the *Symmetric Ice Pile Model* $\mathrm{SIPM}_k(n)$, a generalization of the *Ice Pile Model* $\mathrm{IPM}_k(n)$, and we show an efficient algorithm for generating the symmetric ice piles with n grains. More precisely, we show how to exploit an existing algorithm for generating $\mathrm{IPM}_k(n)$ in order to generate $\mathrm{SIPM}_k(n)$ in amortized time $O(1)$ and in space $O(\sqrt{kn})$.

1 Introduction

In this paper, we consider the problem of generating particular unimodal sequences that describe the reachable states of the symmetric version of the well-known *Ice Pile Model* $\mathrm{IPM}_k(n)$, a discrete dynamical system introduced by Goles Morvan and Phan [7] as a restriction of the discrete dynamical model proposed in 1973 by Brylawski [2] in order to study linear partitions of an integer n.

$\mathrm{IPM}_k(n)$ admits a description in terms of a simple game that simulates the movements of ice grains organized into adjacent columns of decreasing heights. If there are two adjacent columns, say i and $i+1$, with heights differing by at least 2, a grain can fall down from column i to column $i+1$ (the game starts with n grains stacked in column 0). Moreover, if a column i of height p is separated from a column j of height $p-2$ by a plateau of at most $k-1$ columns of height $p-1$, then a grain can slide from i to j crossing the plateau. $\mathrm{IPM}_k(n)$ is an extension of the *Sand Pile Model* $\mathrm{SPM}(n)$, in which only the fall rule is permitted. $\mathrm{SPM}(n)$ has been widely studied in combinatorics, poset theory, physics, and in the theory of cellular automata to represent granular objects, see [1], [6], [7], [8].

A natural extension of $\mathrm{SPM}(n)$ has been introduced [5], [12] in order to fix the lack of symmetry (grains either stay or move to the right). Thus, a grain possibly falls down nondeterministically from column i either to column $i-1$ or to column $i+1$. This new model, called *Symmetric Sand Pile Model* and denoted by $\mathrm{SSPM}(n)$, consists of all the integer sequences describing the reachable states

* Partially supported by Project M.I.U.R. PRIN 2010-2011: Automi e linguaggi formali: aspetti matematici e applicativi

of a system starting with n grains in column 0 (the index of a column can be negative).

Despite the simplicity of this rule, the underlying structure drastically changes. While SPM(n) turns out to be a distributive lattice with exactly one fixed point (the bottom, easily characterized), SSPM(n) is neither a lattice nor admits a unique fixed point. Nevertheless, a characterization of fixed points exists [5, Section 3.2], [12, Th. 2] and their number is known [5, Lemmas 15,16,17].

In this paper we introduce the symmetric version of $\mathrm{IPM}_k(n)$, denoted by $\mathrm{SIPM}_k(n)$, where left and right slide moves on plateaux of length smaller than or equal to $k-1$ are also permitted, in addition to the left and right fall rules. In particular, we show that $\mathrm{SIPM}_k(n)$ can be generated by means of a CAT (Constant Amortized Time) algorithm. We recall that CAT algorithms for generating sand and ice piles have been presented in [9] and [10], and that a CAT algorithm for generating symmetric sand piles has been proposed in [11], where a decomposition property of symmetric sand piles in terms of product of sand piles is exploited. We prove that a similar decomposition property holds for symmetric ice piles: this lets us design a CAT algorithm that sequentially generates the elements of $\mathrm{SIPM}_k(n)$ using $O(\sqrt{kn})$ space.
In Section 2 we lay down preliminaries such as definitions as well as properties and characteristics of unimodal sequences and generalized unimodal sequences, the combinatorial objects used for modeling symmetric ice piles. We also recall some properties of sand and ice piles, as well as the basics of the CAT algorithm used to generate $\mathrm{IPM}_k(n)$. In Section 3 we characterize unimodal sequences that can be forms of symmetric ice piles and generalized unimodal sequences that can be configurations of $\mathrm{SIPM}_k(n)$. These properties are the key for proving that our algorithm is correct and computing its complexity. The algorithm is outlined in Section 4 whereas its complexity is analysed in Section 5.

2 Preliminaries

A linear partition of n is a non-increasing sequence of strictly positive integers, $s = (s_0, \ldots, s_l)$, such that $\sum_{i=0}^{l} s_i = n$; the *height*, the *length* and the *weight* of s are $h(s) = s_0$, $l(s) = l+1$ and $w(s) = n$, respectively. We consider the set LP(n) of linear partitions of n equipped with the negative lexicographic or *neglex* ordering, $<_{\mathrm{nlex}}$, defined as follows: $(s_0, \ldots, s_l) <_{\mathrm{nlex}} (t_0, \ldots, t_m)$ if and only if there exists i, $0 \le i \le \min(l, m)$ such that $s_i > t_i$ and $s_j = t_j$ for $j < i$. A *unimodal sequence* of n is a sequence of strictly positive integers, $a = (a_0, \ldots, a_l)$, such that $\sum_{i=0}^{l} a_i = n$ and $a_0 \le a_1 \le \ldots \le a_j \ge a_{j+i} \ge \ldots \ge a_l$ for some j. The smallest index q such that $a_{q-1} \le a_q \ge a_{q+1} \ge \cdots \ge a_l$ is called the *center* of a, noted by $c(a)$. Obviously, $h(a) = a_{c(a)}$ and, if a is a constant sequence then $c(a) = 0$. From here on, for $i \ge 0$ we let $a_{<i} = (a_0, a_1 \ldots, a_{i-1})$ and $a_{\ge i} = (a_i, a_{i+1}, \ldots, a_l)$. Given $h > 1$, we indicate by $p^{[h]}$ the sequence $(\underbrace{p, p, \ldots, p}_{h})$, interpreted as a *plateau* of h columns of height p. The *catenation* product of sequences is denoted by " $\cdot$ ", $(a_0, \ldots, a_l) \cdot (b_0, \ldots, b_p) = (a_0, \ldots, a_l, b_0, \ldots, b_p)$, and the reversal of a by $\overline{a} = (a_l, \dots, a_0)$. If $a = b \cdot c$ we say that b and c are a *prefix* and a *suffix* of a, respectively.

We equip the set $\mathrm{US}(n)$ of unimodal sequences of n with a particular linear order " $<$ ". Thus, let $a, b \in \mathrm{US}(n)$, $x_1 = h(a)$, $x_2 = h(b)$ and consider the (unique) factorizations $a = c \cdot x_1^{[p_1]} \cdot d$, $\quad b = e \cdot x_2^{[p_2]} \cdot f$ with $h(c), h(d) < x_1$ and $h(e), h(f) < x_2$. We say that $a < b$ if and only if either $(x_1 > x_2)$ or $(x_1 = x_2, p_1 > p_2)$ or $(x_1 = x_2, p_1 = p_2, w(d) > w(f))$ or $(x_1 = x_2, p_1 = p_2, w(d) = w(f), d <_{\mathrm{nlex}} f)$ or $(x_1 = x_2, p_1 = p_2, d = f, c <_{\mathrm{nlex}} e)$.

By possibly considering negative indices, we say that $(a_j, a_{j+1} \dots, a_l)$ is a *generalized unimodal sequence* of *form* $b = (b_0, \dots, b_{l-j})$ if and only if b is a unimodal sequence and $b_i = a_{i+j}$, $0 \leq i \leq l - j$. Index j is the *position* of a. In other words, a generalized unimodal sequence is obtained by a unimodal sequence by (right- or left-) shifting all its entries by j places. We identify a generalized unimodal sequence by means of a pair (a, j) where a is a unimodal sequence (the form) and j an integer (the position). We also write $\boldsymbol{a}$ to indicate a generalized unimodal sequence whenever the value j does not matter.

We extend the linear order $<$ to the set $\mathrm{GUS}(n)$ of generalized unimodal sequences of n by setting $(a, i) < (b, j)$ if and only if either $a < b$ or $a = b$ and $i < j$. Trivially, one has $\mathrm{LP}(n) \subset \mathrm{US}(n) \subset \mathrm{GUS}(n)$. If $\boldsymbol{a} = (a, j) \in \mathrm{GUS}(n)$ then for any i with $j \leq i \leq j + l(a) - 1$ we set $\boldsymbol{a}_{<i} = (a_j, \dots, a_{i-1})$ and $\boldsymbol{a}_{\geq i} = (a_i, \dots, a_{j+1(a)-1})$. The *right height difference* of $\boldsymbol{a} = (a, j)$ at i is defined as $\delta_r(\boldsymbol{a}, i) = a_i - a_{i+1}$ (assume $a_i = 0$ for $i > j + l(a) - 1$ or $i < j$). Analogously, the *left height difference* of $\boldsymbol{a}$ at i is $\delta_l(\boldsymbol{a}, i) = a_i - a_{i-1}$.

We interpret the elements of $\mathrm{GUS}(n)$ as blocks of n grains disposed into adjacent columns and define four (partial) functions called *moves*. Let $\boldsymbol{a} = (a, j) \in \mathrm{GUS}(n)$, then the right fall of a grain in column i with $j \leq i \leq j + l(a) - 1$ is

$$\mathrm{RFall}(\boldsymbol{a}, i) = \begin{cases} (a_j, \dots, a_{i-1}, a_i - 1, a_{i+1} + 1, \dots, a_{j+l(a)-1}) & \text{if } \delta_r(\boldsymbol{a}, i) > 1, \\ \perp & \text{otherwise} \end{cases}$$

The function RSlide_k allows the crossing of a plateau of length at most $k - 1$:

$$\mathrm{RSlide}_k(\boldsymbol{a}, i) = (a_j, \dots, a_{i-1}, \underbrace{p, p, \dots, p}_{k'+2}, a_{i+k'+2}, \dots, a_{j+l(a)-1})$$

if there is $k' < k$ such that

$$\boldsymbol{a} = (a_j, \dots, a_{i-1}, p + 1, \underbrace{p, p, \dots, p}_{k'}, p - 1, a_{i+k'+2}, \dots, a_{j+l(a)-1})$$

otherwise $\mathrm{RSlide}_k(\boldsymbol{a}, i) = \perp$. The functions $\mathrm{LFall}(\boldsymbol{a}, i)$, and $\mathrm{LSlide}_k(\boldsymbol{a}, i)$ are defined symmetrically.

Whenever the value k is obvious, we simply write $\boldsymbol{a} \stackrel{i}{\Rightarrow} \boldsymbol{b}$ if either $\boldsymbol{b} = \mathrm{LFall}(\boldsymbol{a}, i)$ or $\boldsymbol{b} = \mathrm{RFall}(\boldsymbol{a}, i)$ or $\boldsymbol{b} = \mathrm{LSlide}_k(\boldsymbol{a}, i)$ or $\boldsymbol{b} = \mathrm{RSlide}_k(\boldsymbol{a}, i)$. In general, we write $\boldsymbol{a} \stackrel{\star}{\Rightarrow} \boldsymbol{b}$ if there is a sequence of moves leading from $\boldsymbol{a}$ to $\boldsymbol{b}$.

2.1 The Ice Pile Model

$\mathrm{IPM}_k(n)$ consists of the closure of $\{(n)\}$ under RFall and RSlide_k. Linear partitions of $\mathrm{IPM}_k(n)$ have been characterized combinatorially in [7, Th. 3].

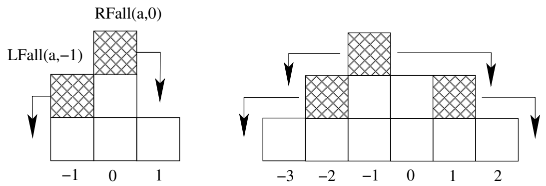

Fig. 1. Moves in GUS(n): RFall and LFall, RSlide_3 and LSlide_3.

Such theorem implies two upper bounds that are important for our purposes :

- For any ice pile a with height h one has $w(a) \leq h + k\frac{(h+1)h}{2}$.
- For any ice pile $a \in \mathrm{IPM}_k(n)$ one has $l(a) \leq O(\sqrt{kn})$.

Given $a, b \in \mathrm{IPM}_k(n)$ we say that b *dominates* a, noted by $a \prec b$, if and only if for all $e \geq 0$ one has $\sum_{i=0}^{e} b_i \geq \sum_{i=0}^{e} a_i$. Note that $a \prec b$ implies $b <_{\mathrm{nlex}} a$. The *dominance* order $\prec$ is related to the order induced by $\Rightarrow$. Indeed, if $a \prec b$ then $b \overset{\star}{\Rightarrow} a$ (see [7]).

An ice pile a is called a *staircase of height* h if and only if either $a = h^{[k_0]} \cdot (h - j_0)$, with $0 < k_0 < k$ and $0 \leq j_0 \leq h$, or

$$a = h^{[k]} \cdot (h-1)^{[k]} \cdot (h-2)^{[k]} \cdots (h-i)^{[k]} \cdot (h-i-1)^{[k_1]} \cdot (h-i-1-j)$$

with $i \geq 0$, $0 \leq k_1 < k$ and $j > 0$. We indicate by $\mathrm{IPM}_{k,h}(n)$ the set of ice piles of height at most h with n grains, $\mathrm{IPM}_{k,h}(n) = \{a \in \mathrm{IPM}_k(n) \mid h(a) \leq h\}$. Note that for a suitable n with $(k+1)h < n \leq h + k\frac{(h+1)h}{2}$, the ice pile $\min_{<_{\mathrm{nlex}}}(\mathrm{IPM}_{k,h}(n))$ is

$$h^{[k+1]} \cdot (h-1)^{[k]} \cdot (h-2)^{[k]} \cdots (h-i)^{[k]} \cdot (h-i-1)^{[k_1]} \cdot (h-i-1-j).$$

Definition 1. *For any $h > 0$, we call h-*critical *an ice pile $a \in IPM_{k,h-1}(n))$ such that $h^{[k+1]} \cdot a$ is not an ice pile.*

An ice pile $a \in \mathrm{IPM}_{k,h}(n))$ is $(h+1)$-*critical* if and only if it has a prefix in the set of ice piles

$$\Pi_h = \{a \in \mathrm{IPM}_k(r) | r > 0 \wedge \exists e, 0 \leq e < h, a = \prod_{i=0}^{e} (h-i)^{[k]}(h-e)\}.$$

The *set of moves* of a is $\mathrm{M}(a) = \{i | 0 \leq i \leq l(a),\ \mathrm{RFall}(a,i) \neq \bot \text{ or } \mathrm{RSlide}_k(a,i) \neq \bot\}$. If a is a staircase or $a = \min_{<_{\mathrm{nlex}}}(\mathrm{IPM}_{k,h}(n))$ then $|\mathrm{M}(a)| \leq 3$, indeed $\mathrm{M}(a) \subseteq \{l(a) - j, l(a) - 2, l(a) - 1\}$ for a suitable $j \leq k$. If $\mathrm{M}(a) = \emptyset$ then a is a *fixed point*. We denote by $a^{(e)}$ the eth ice pile of $\mathrm{IPM}_k(n)$ (w.r.t. $<_{\mathrm{nlex}}$) and by $G(a)$ the set of ice piles generated from a, $G(a) = \{b | a \overset{\star}{\Rightarrow} b\}$.

Lemma 1. *Let $a = \min_{<_{nlex}}(IPM_{k,h}(n))$. Then $G(a) = IPM_{k,h}(n)$.*

All ice piles of height h which are smaller than a staircase of height h are $(h+1)$-critical:

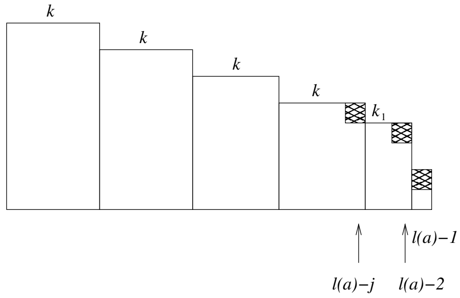

Fig. 2. Possible moves for a staircase.

Lemma 2. *Let $a, b \in IPM_{k,h}(n)$ and suppose that a is a staircase of height h. Then $(b <_{nlex} a) \iff b$ is $(h+1)$-critical.*

Corollary 1. *Let $a \in IPM_{k,h}(n)$ be a staircase of height h. Then,*

$$G(a) = \{b \in IPM_{k,h}(n) | b \text{ is not } (h+1)\text{-critical}\}.$$

For complexity evaluation purposes, we give a bound for the number of ice piles generated by a staircase or by the smallest ice pile of given height. (see fig. 3).

Lemma 3. *Let a be either $\min_{<_{nlex}}(IPM_{k,h}(n))$ or a staircase in $IPM_{k,h}(n)$. If a is not a fixed point then $|G(a)| = \Omega(\frac{l(a)}{k})$.*

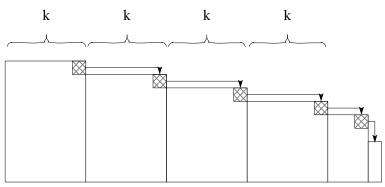

Fig. 3. A sequence of $\Omega(\frac{l(a)}{k})$ moves in a staircase.

We recall here a result [10, Lemma 2.16] on which the CAT generation of $\text{IPM}_k(n)$ is based.

Lemma 4. *For any $e > 0$, let $i_e = \max(M(a^{(e)}))$. Then we have*

$$A(a^{(e)}) \overset{i_e}{\Rightarrow} a^{(e+1)}$$

where $A(a^{(e)}) = \min_{<_{nlex}}\{a \in IPM_k(n) \mid a_{<i_e+1} = a^{(e)}_{<i_e+1}, i_e \in M(a)\}$.

We consider a function Next : $\mathrm{IPM}_k(n) \mapsto \mathrm{IPM}_k(n)$ such that for $e > 0$, $\mathrm{Next}(a^{(e)}) = a^{(e+1)}$ ($\mathrm{Next}(a^{(e)}) = \perp$ if $a^{(e)}$ is a fixed point). In [10] it is shown how this function can be implemented so that if $a = (n)$ then, for any fixed integer k, the iteration of the instruction $a{:=}\textsc{Next}(a)$ (until $\mathrm{M}(a) \neq \emptyset$) generates $\mathrm{IPM}_k(n) = G((n))$ in time $O(|G((n))|)$ (and so is a CAT algorithm) using $O(\sqrt{n})$ space. More generally, for any $a \in \mathrm{IPM}_k(n)$ one can generate $G(a)$ in time $O(|G(a)|)$.

3 The Symmetric Ice Pile Model

In [5] and [12] the Symmetric Sand Pile Model SSPM(n) (obtained by closing $\{(n)\}$ with respect to RFall and LFall) has been studied and its relation with SPM(n) analysed. Here we define the *Symmetric Ice Pile Model* $\mathrm{SIPM}_k(n)$ as the set of integer sequences (called symmetric ice piles) obtained by closing $\{(n)\}$ w. r. t. RFall, RSlide_k, LFall and LSlide_k. Note that if $(a, j) = (a_j, \ldots, a_{l(a)+j-1}) \in \mathrm{GUS}(n)$ is reached from the initial configuration (n) (all the grains in column 0) then one has $j \leq 0 \leq l(a) + j - 1$, since the evolution rules never empty a column completely (positions j of sequences in $\mathrm{SIPM}_k(n)$ are nonpositive).

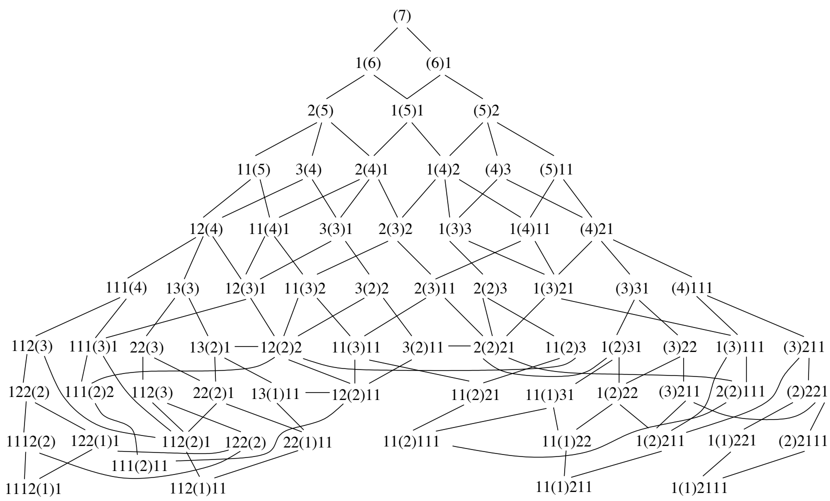

Fig. 4. The poset $\mathrm{SIPM}_2(7)$ (w.r.t. the order relation induced by $\Rightarrow$).

Figure 4 shows several elements having the same form but with different positions (parentheses are used to indicate column 0). For example $(111211, -2)$, $(111211, -3)$ and $(111211, -4)$ are all in $\mathrm{SIPM}_2(7)$.

As a matter of fact, symmetric ice piles are closely related to ice piles.

Definition 2. *Let $a = (a_0, \ldots, a_l)$ be a unimodal sequence. Then a is decomposable at i if and only if $a_i = h(a)$ and $\overline{a_{<i}}, a_{\geq i}$ are ice piles.*

The elements of $\mathrm{SIPM}_k(n)$ have a form which can be decomposed into the concatenation of a reversed ice pile with an ice pile. More precisely, we extend to $\mathrm{SIPM}_k(n)$ a decomposition property proved for $\mathrm{SSPM}(n)$ in [12, Lemma 3].

Lemma 5. *Let $a \in US(n)$, then the two following conditions are equivalent:*

1. *there is an integer i such that a is decomposable at i, with $\overline{a_{<i}} \in IPM_k(r)$, $a_{\geq i} \in IPM_k(s)$ and $r + s = n$;*
2. *there is an integer j such that $(a, j) \in SIPM_k(n)$.*

Definition 3. *For $a \in US(n)$ (resp. $\boldsymbol{a} \in GUS(n)$) we denote by $D(a)$ (resp. $D(\boldsymbol{a})$) the set of all indices i such that a (resp. $\boldsymbol{a}$) admits a decomposition at i (or equivalently, is decomposable at i).*

Remark 1. Obviously, if $\boldsymbol{a} = (a, j) \in \mathrm{GUS}(n)$, one has $D(\boldsymbol{a}) = D(a) + j = \{i + j \mid i \in D(a)\}$. Note also that $D(\boldsymbol{a})$ is always an integer interval. Indeed, the set $\{e \mid \boldsymbol{a}_e = h(\boldsymbol{a})\}$ is an integer interval $[e_1, e_2]$ (with $e_1 = c(\boldsymbol{a})$) and then $D(\boldsymbol{a}) = [i_1, i_2]$ where :

$$i_1 = \begin{cases} max(e_1, e_2 - k + 1), & \text{if } \boldsymbol{a}_{\geq e_2+1} \text{ is } h(\boldsymbol{a})\text{-critical} \\ max(e_1, e_2 - (k+1) + 1) = max(e_1, e_2 - k), & \text{otherwise} \end{cases}$$

$$i_2 = \begin{cases} min(e_2, e_1 + k - 1), & \text{if } \overline{\boldsymbol{a}_{<e_1}} \text{ is } h(\boldsymbol{a})\text{-critical} \\ min(e_2, e_1 + (k+1) - 1) = min(e_2, e_1 + k), & \text{otherwise} \end{cases}$$

Now, we want to determine when a *generalized* unimodal sequence belongs to $\mathrm{SIPM}_k(n)$. Following [12], we start defining a unilateral (possibly reversed) ice pile $f(\boldsymbol{a}, i)$ associated with a pair $\boldsymbol{a} \in \mathrm{GUS}(n)$ and $i \in D(\boldsymbol{a})$.

Definition 4. *Let $\boldsymbol{a} \in GUS(n)$ and $i \in D(\boldsymbol{a})$. If $i \geq 0$, we define $f(\boldsymbol{a}, i)$, called the* completion of $\boldsymbol{a}$ at i, *as the ice pile $s = (s_0, s_1, \ldots)$ of minimal weight such that $s_{\geq i} = \boldsymbol{a}_{\geq i}$. In this case we will call* complement of $\boldsymbol{a}$ at i *the ice pile $c(\boldsymbol{a}, i) = (s_0, s_1, \ldots, s_{i-1})$.*
Similarly, if $i < 0$, we define $f(\boldsymbol{a}, i)$ as the generalized unimodal sequence $\boldsymbol{s} = (\ldots, s_{-2}, s_{-1}, s_0)$ such that s is the reversed ice pile of minimal weight with $\boldsymbol{s}_{<i} = \boldsymbol{a}_{<i}$. In this case the complement of $\boldsymbol{a}$ at i *is the reversed ice pile $c(\boldsymbol{a}, i) = (s_i, s_{i+1}, \ldots, s_{-1}, s_0)$. In either case, $w(\boldsymbol{a}, i)$ denotes the weight of $f(\boldsymbol{a}, i)$.*

Example 1. Let $n = 21$, $k = 2$, $a = (1, 2, 2, 3, 4, 4, 4, 1)$ and $\boldsymbol{a} = (a, -1)$. Then $c(\boldsymbol{a}, 3) = (6, 5, 5)$ and hence $f(\boldsymbol{a}, 3) = (6, 5, 5, 4, 4, 4, 1)$, while $c(\boldsymbol{a}, 5) = (6, 5, 5, 4, 4)$ but still the same completion $f(\boldsymbol{a}, 5) = (6, 5, 5, 4, 4, 4, 1) = f(\boldsymbol{a}, 3)$. To study a case with $i < 0$, let $b = (1, 2, 2, 3, 4, 4, 4, 1)$ and $\boldsymbol{b} = (b, -7)$. Then $f(\boldsymbol{b}, -2) = ((1, 2, 2, 3, 4, 4, 4, 5), -7) = f(\boldsymbol{b}, -1) = f(\boldsymbol{b}, -3)$.

For any symmetric ice pile $\boldsymbol{a}$ which is decomposable at i, the weight of $c(\boldsymbol{a}, i)$ is uniquely determined by either $\boldsymbol{a}_{<i}$ ($i < 0$) or $\boldsymbol{a}_{\geq i}$ ($i \geq 0$) and easily computed.

Lemma 6. *Let $\boldsymbol{a} = (a, j) \in GUS(n)$ be decomposable at i. Then $w(c(\boldsymbol{a}, i))$ can be computed in time $O(1)$.*

The following Lemma is used for proving Lemma 8, which provides a characterization for the configurations of $\text{SIPM}_k(n)$.

Lemma 7. *Let $\boldsymbol{t} \in SIPM_k(n)$ such that $\boldsymbol{t} = (t, 0)$ and let $\boldsymbol{a} = (a, j)$ be another generalized unimodal sequence such that $j < 0$, $l(a) = |j| + l(t)$, $\overline{a} \in IPM_k(n)$ and such that for all i with $0 \leq i < l(t)$ one has $a_i \leq t_i$. Then $\boldsymbol{a}$ is reachable from $\boldsymbol{t}$.*

Lemma 8. *A sequence $\boldsymbol{a} = (a, j) \in GUS(n)$ belongs to $SIPM_k(n)$ if and only if $\exists\ i$ with $j \leq i \leq l(a) - 1 + j$ such that $\boldsymbol{a}$ is decomposable at i and $w(\boldsymbol{a}, i) \leq n$.*

The condition of the previous lemma characterizing generalized unimodal sequences belonging to $\text{SIPM}_k(n)$, can be replaced by an equivalent (although apparently stronger) condition.

Lemma 9. *Let $\boldsymbol{a} = (a, j) \in GUS(n)$. Then, there exists an integer $i_0 \in D(\boldsymbol{a})$ such that $w(\boldsymbol{a}, i_0) \leq n$ if and only if for all $i \in D(\boldsymbol{a})$, one has $w(\boldsymbol{a}, i) \leq n$.*

Corollary 2. *Let $\boldsymbol{a} \in GUS(n)$. If $0 \in D(\boldsymbol{a})$ then $\boldsymbol{a} \in SIPM_k(n)$.*

Lemma 9 also allows to deduce that for all unimodal sequence a, the set $S(a)$ of integers j such that (a, j) is in $\text{SIPM}_k(n)$ is an integer interval.

Corollary 3. *Given $a \in US(n)$, let j_1, j_2 be two integers such that $(a, j_1), (a, j_2) \in SIPM_k(n)$. Then for all j with $j_1 \leq j \leq j_2$ one has $(a, j) \in SIPM_k(n)$.*

Our algorithm generates all possible unimodal sequences a that are form of some symmetric ice pile, and, for each form, all integers j such that $(a, j) \in \text{SIPM}_k(n)$. We only need to determine the bounds of $S(a)$.

Corollary 4. *Let $a \in US(n)$ Then $(a, j) \in SIPM_k(n)$ if and only if $j \in [j_{\min}, j_{\max}]$ with $j_{\min} = \min\{j | (a, j) \in SIPM_k(n)\}$ and $j_{\max} = \max\{j | (a, j) \in SIPM_k(n)\}$.*

For the computation of the two integers $j_{\min}$, $j_{\max}$ we have:

Corollary 5. *Let $a \in US(n)$ Then $j_{\min} = \min\{j | (a, j) \in SIPM_k(n)\}$ and $j_{\max} = \max\{j | (a, j) \in SIPM_k(n)\}$ are computed in time $O(1)$.*

Example 2. Let $n = 21$, $k = 2$ and $a = (1, 2, 2, 3, 4, 4, 4, 1)$ with $D(a) = [i_1, i_2] = [4, 6]$. Clearly $[j_{\min}, j_{\max}] \supseteq [-i_2, -i_1] = [-6, -4]$ because $(a, -6)$ and $(a, -4)$ are decomposable at 0. However the largest $\delta > 0$ such that $(a, -4 + \delta) \in \text{SIPM}_2(21)$ is 1. Indeed, $(a, -3)$ is decomposable at 1 and $w((a, -3), 1) = 18 < 21$, whereas $(a, -2)$ is decomposable only at j, $2 \leq j \leq 4$, but with weight $w((a, -2), 2) = 23 > 21$. On the other hand, $(a, -7) \notin \text{SIPM}_2(21)$ since it is decomposable only at j, $-3 \leq j \leq -1$ with $w((a, -7), -1) = 25 > 21$. Therefore $(a, j) \in \text{SIPM}_2(21)$ if and only if $j \in [-6, -3]$.

We conclude this section with a slightly different characterization of forms of symmetric ice piles which is more suitable for our generation algorithm.

Lemma 10. *A unimodal sequence a is the form of an element in $SIPM_k(n)$ if and only if $a = c \cdot x^{[p]} \cdot d$ with $\bar{c} \in IPM_k(t_1), d \in IPM_k(t_2)$, $h(\bar{c}), h(d) < x$, $t_1+t_2+xp = n$ and either $p < 2k+1$ (1) or $p = 2k+2$ and $\bar{c}, d$ are not x-critical (2) or $p = 2k+1$ and at least one of $\bar{c}, d$ is not x-critical (3).*

Given a unimodal sequence $a = c \cdot x^{[p]} \cdot d$ that is the form of an element in $\mathrm{SIPM}_k(n)$, the triple $(x, p, w(d))$ is said the *type* of a (by Lemma 10 $p \leq 2k+2$). In other words, a triple of integers (x, y, z) is a type for $\mathrm{SIPM}_k(n)$ if and only if there are a unimodal sequence $a = c \cdot x^{[y]} \cdot d$, with $\bar{c} \in \mathrm{IPM}_k(n - xy - z)$ and $d \in \mathrm{IPM}_k(z)$, and an integer j such that $(a, j) \in \mathrm{SIPM}_k(n)$. Obviously one has $\sqrt{n} - 1 \leq x \leq n$, $1 \leq p \leq \min(\lfloor n/x \rfloor, 2k+2)$ and $0 \leq r \leq kx(x-1)/2 + x - 1$.

4 The algorithm

The idea is that of generating in order (with respect to $<$) all unimodal sequences that are forms of elements in $\mathrm{SIPM}_k(n)$. By Lemma 10, such forms are the product of three sequences, $c \cdot x^{[p]} \cdot d$, which satisfy suitable constraints. Thus, we consider all the triples (x, p, r) corresponding to types, then we generate all the forms associated with each type, and lastly, for each form, we compute all the positions of the symmetric ice piles having that form. So, consider a type (x, p, r) and distinguish three cases depending on p.

When $p \leq 2k$ all the forms can be written as $a = c \cdot x^{[p]} \cdot d$ where $\bar{c}$ and d are arbitrary ice piles in $\mathrm{IPM}_{k,x-1}(n - px - r)$ and $\mathrm{IPM}_{k,x-1}(r)$, respectively. In fact, $x^{[j_1]} \cdot \bar{c}$ and $x^{[j_2]} \cdot d$ are always ice piles for all $j_1, j_2 \leq k$. By choosing two values such that $j_1 + j_2 = p$ we can guarantee the existence of a value i such that $\overline{a_{<i}}$ and $a_{\geq i}$ are ice piles. Then, Lemma 5 states that the unimodal sequence $c \cdot x^{[p]} \cdot d$ is the form of an element in $\mathrm{SIPM}_k(n)$.

Obviously, $c \cdot x^{[p]} d \neq c' \cdot x^{[p]} \cdot d'$ if $h(c), h(d), h(c'), h(d') < x$ and $c' \neq c$ or $d' \neq d$. Thus, we get all the forms of this type, if we generate all the elements of $\mathrm{IPM}_{k,x-1}(r)$ and, for each of these, all the ice piles in $\mathrm{IPM}_{k,x-1}(n - px - r)$.

Consider the case $p = 2k+2$. In order to get all the forms $c \cdot x^{[2k+2]} \cdot d$, by condition (2) in Lemma 10 we need to generate all ice piles $\bar{c}$ and d that are not x-critical. By Corollary 1 these are the ice piles in $G(s)$ and $G(t)$, where s and t are the highest staircases in $\mathrm{IPM}_{k,x-1}(n-px-r)$ and $\mathrm{IPM}_{k,x-1}(r)$, respectively.

Lastly, let $p = 2k+1$. If $r < (k+1)(x-1)$ it is sufficient to generate all d in $\mathrm{IPM}_{k,x-1}(r)$ and (for each of these) all $\bar{c} \in \mathrm{IPM}_{k,x-1}(n-(2k+1)x-r)$. Observe that d is not x-critical and thus condition (3) of Lemma 10 is satisfied. Otherwise ($r \geq (k+1)(x-1)$), let t be the staircase of height $x-1$ in $\mathrm{IPM}_{k,x-1}(r)$ and consider the partition $A \cup B = \mathrm{IPM}_{k,x-1}(r)$ where $A = \{a \in \mathrm{IPM}_{k,x-1}(r) | a <_{\mathrm{nlex}} t\}$ and $B = \{b \in \mathrm{IPM}_{k,x-1}(r) | t \leq_{\mathrm{nlex}} b\}$. By Lemma 2 each element of A is x-critical and then for each $d \in A$ we have to generate all ice piles $\bar{c}$ that are not x-critical (so that $x^{[k+1]} \cdot \bar{c}$ is an ice pile): these are exactly the ice piles $\bar{c} \in G(s)$ where s is the staircase of height $x-1$ in $\mathrm{IPM}_{k,x-1}(n-(2k+1)x-r)$. Then, for each $d \in B$ we generate all $\bar{c} \in \mathrm{IPM}_{k,x-1}(n-(2k+1)x-r)$ ($x^{[k+1]} \cdot d$ is an ice pile). Observe that by Lemma 1 and Corollary 1 one has $A = G(g) \setminus G(t)$, where $g = \min_{<_{\mathrm{nlex}}}(\mathrm{IPM}_{k,x-1}(r))$, and $B = G(t)$.

Once a form $c \cdot x^{[p]} d$ has been generated, all the positions j such that $(c \cdot x^{[p]} \cdot d, j) \in \mathrm{SIPM}_k(n)$ are computed using Corollary 5.

This idea leads immediately to Algorithm 1. We represent ice piles in $\mathrm{IPM}_k(r)$ as structures with six fields: the number of grains r, the current length, the linear partition (an array of integers), the set of moves (an ordered stack of integers), the integer k and a flag which is *on* if and only if the ice pile is x-critical. This last field is included only to simplify the computation of the positions.

The algorithm consists of three nested loops (associated with the three entries of a type (x, p, r)). The repeat-loop is used to set the height x of the symmetric ice pile, starting from the initial value n. The for-loop sets the number p of columns of height x, from the the largest allowed value $\min(\lfloor n/x \rfloor, 2k + 2)$ downto 1. Lastly, the while-loop sets the weight r of the ice pile d in $c \cdot x^{[p]} \cdot d$, from the largest admissible value (the smallest between the number of available grains $n - px$ and either $kx(x-1)/2$ or $kx(x-1)/2 + x - 1$, depending on p) downto the smallest one (i.e. a value r such that (x, p, r) is a type for $\mathrm{SIPM}_k(n)$ while $(x, p, r - 1)$ is not).

Algorithm 1 Exhaustive Generation of $\mathrm{SIPM}_k(n)$.

```
PROCEDURE SIPGENERATION(n,k)
x := n;
repeat
   for p := min(⌊n/x⌋, 2k + 2) downto 1 do
      m := n − p · x;
      if p = 2k + 2 then max := kx(x − 1)/2; else max := kx(x − 1)/2 + x − 1;
      r := min(m, max);
      while ISTYPE(n,k,x,p,r) do
         if p < 2k + 1 then GEN1(m − r,r,x − 1,p,k);
                    else if p = 2k + 2 then GEN2(m − r,r,x − 1,k);
                    else if p = 2k + 1 then GEN3(m − r,r,x − 1,k);
         r:=r − 1;
      end while
   end for
   x:=x − 1;
until TOOLOW(n,k,x);
```

At each iteration, the value p determines which case of Lemma 10 occurs.

When $p < 2k+1$ the call $\textsc{Gen1}(m-r,r,x-1,p,k)$ generates all symmetric ice piles having the form $c \cdot x^{[p]} \cdot d$ with $\bar{c} \in \mathrm{IPM}_{k,x-1}(m-r)$ and $d \in \mathrm{IPM}_{k,x-1}(r)$. Similarly, if $p = 2k + 2$ $\textsc{Gen2}(m - r,r,x - 1,k)$ generates all symmetric ice piles having the form $c \cdot x^{[2k+2]} \cdot d$ with $\bar{c} \in G(s)$, and $d \in G(t)$, where s and t are the highest staircases in $\mathrm{IPM}_{k,x-1}(m - r)$ and $\mathrm{IPM}_{k,x-1}(r)$, respectively. Lastly, when $p = 2k + 1$, the call $\textsc{Gen3}(m-r,r,x-1,k)$ generates all symmetric ice piles having the form $c \cdot x^{[2k+1]} \cdot d$ and such that $\bar{c} \in \mathrm{IPM}_{k,x-1}(m - r)$ or $d \in \mathrm{IPM}_{k,x-1}(r)$ is not x-critical.

The algorithm halts as soon as a value x is reached such that no symmetric ice pile of height x exists (TooLow(n,k,x) at line 24 returns true iff $\mathrm{IPM}_{k,x}(n) = \emptyset$).

Procedures Gen [1-3] are easily developed by means of the following functions and procedures. MinIcePile(x,r,k) costructs the smallest ice pile in $\mathrm{IPM}_{k,x}(r)$ while Staircase(x,r,k) constructs the highest staircase in $\mathrm{IPM}_{k,x}(r)$. Both functions return a reference to an ice pile (the former possibly sets the flag indicating x-criticality). IsStaircase(a) (IsBottom(a)) returns the boolean value *true* iff a is a staircase (fixed point). The generation of the ice piles $\bar{c}$ and d which appear in $c \cdot x^{[p]} \cdot d$ is done by means of Next. This function sets its argument to the next ice pile in the neglex order, and returns a reference to it (see Section 2.1). Once the components c, x, p and d of a form are known, the Procedure Positions computes the range of positions for that form.

Example 3. SIPGeneration$(7, 2)$ produces the following sequence of symmetric ice piles (see Fig. 4): $(7), (6)1, 1(6), (5)2, (5)11, 1(5)1, \ldots, 111(2)11, 11(1)211$.

5 Complexity

With respect to the space complexity, we point out that the algorithm uses $O(\sqrt{kn})$ space, since to represent a form $c \cdot x^{[p]} \cdot d$ we need only two ice piles $\bar{c}$, d (represented by two structures of size $O(\sqrt{kn})$) and two integers x, p. Moreover, recall that for any $a \in \mathrm{IPM}_k(n)$ the generation of $G(a)$ requires $O(|G(a)|)$ time and $O(\sqrt{kn})$ space (see section 2.1).

Procedures MinIcePile and Staircase have a cost which grows as the length $l = O(\sqrt{kn})$ of the returned ice pile, whereas IsBottom and IsStaircase run in time $O(1)$. Indeed, these two functions simply check the stack ST representing the moves of the ice pile ($\mathrm{ST} = \emptyset, \mathrm{ST} \subseteq \{l-2, l-1\}$).
Functions IsType and TooLow run in time $O(1)$ too. In fact, IsType(n, k, x, p, r) when $p \leq 2k$ simply checks whether the two values r and $n - px - r$ are both smaller than $kx(x-1)/2 + x$. Similarly, if $p = 2k + 2$ it verifies that $r, n - (2k+2)x - r \leq kx(x-1)/2$. Lastly, if $p = 2k + 1$ it checks whether $r \leq kx(x-1)/2 + x - 1$ and $n - (2k+1)x - r \leq kx(x-1)/2$ or $r \leq kx(x-1)/2$ and $n - (2k+1)x - r \leq kx(x-1)/2 + x - 1$. TooLow$(n, k, x)$ returns *false* if $2kx(x-1)/2 \leq n - (2k+2)x$, *true* otherwise. At last, Corollary 5 lets us develop a procedure Positions$(\bar{c}, x, p, d)$ which runs in time $O(1)$.

In Procedure SIPGeneration, the repeat-loop iterates $O(n)$ times, the for-loop iterates $O(k)$ times and the while-loop iterates $O(kn^2)$ times. Then the overall cost is $O(k^2n^3)$ plus the cost of $O(k^2n^3)$ calls to Gen[1-3]. Thus, we consider:

Lemma 11. *Let $C(l, r, x, p, k)$ be the number of symmetric ice piles generated by either* Gen1*(l,r,x,p,k) (if $p \leq 2k$) or* Gen3*(l,r,x,k) (if $p = 2k + 1$) or* Gen2*(l,r,x,k) (if $p = 2k + 2$). Then, the running time of* Gen1*(l,r,x,p,k),* Gen2*(l,r,x,k) and* Gen3*(l,r,x,k) is $O(C(l, r, x, p, k)) + O(\sqrt{kn})$.*

We can now prove the main result.

Theorem 1. SIPGENERATION *is a CAT algorithm and uses* $O(\sqrt{kn})$ *space.*

Proof. Note that all the instructions of SIPGENERATION have cost $O(1)$ except the calls to GEN[1-3]. Thus, the cost $T(n)$ of SIPGENERATION(n,k) is $O(k^2n^3)$ (due to the three nested loops) plus the cost of $O(k^2n^3)$ calls to GEN[1-3]. Therefore, by Lemma 11 one has

$$T(n) = O(k^2n^3) + \sum_{x,p,l,r} O(C(l,r,x,p,k)) = O(k^2n^3) + O(|\mathrm{SIPM}_k(n)|).$$

Since $k^2n^3 = O(|\mathrm{SPM}(n)|)$ and $|\mathrm{SPM}(n)| \leq |\mathrm{IPM}_k(n)| \leq |\mathrm{SIPM}_k(n)|$ (see [3] for bounds for $|\mathrm{SPM}(n)|$), SIPGENERATION turns out to be a CAT algorithm. With respect to the space complexity, note that the necessary space for storing two ice piles at a time is $(O(\sqrt{kn}))$. □

As an extension of this work, it is quite natural to ask whether a similar approach can be applied to deal with the exhaustive generating problem for other (similar) discrete models. In particular, the discrete models BSPM and BIPM (Bidimensional Sand and Ice Pile Model) have been introduced in [4] by adding a further dimension to $\mathrm{SPM}(n)$ and $\mathrm{IPM}_k(n)$, respectively. Thus, the elements of BSPM and BIPM are plane partitions, that is, matrices of non-negative integers that are nonincreasing from top to bottom and from left to right. These models are not lattices and admit several fixed points but, unlike $\mathrm{SIPM}_k(n)$, no characterization is known for reachable states and fixed points.

References

1. P. Bak, C. Tang and K. Wiesenfeld, Self-organized criticality, *Phys. Rev. A* 38 (1988), 364–374.
2. T. Brylawski, The lattice of integer partitions, *Discrete Math.* 6 (1973), 201–219.
3. S. Corteel, D. Gouyou-Beauchamps, Enumeration of sand piles, *Discrete Math.* 256 n.3 (2002), 625–643.
4. E. Duchi, R. Mantaci, H. D. Phan and D. Rossin, Bidimensional sand pile and ice pile models, *PU.M.A.* vol. 17 (2007) n.1-2, 71–96.
5. E. Formenti, B. Masson and T. Pisokas, Advances in symmetric sandpiles, *Fundamenta Informaticae* 20 (2006), 1–22.
6. E. Goles and M. A. Kiwi, Games on line graphs and sand piles, *Theoret. Comput. Sci.* 115 (1993), 321–349.
7. E. Goles, M. Morvan and H. D. Phan, Sandpiles and order structure of integer partitions, *Discrete Appl. Math.* 117 (2002), 51–64.
8. M. Latapy, R. Mantaci, M. Morvan and H. D. Phan, Structure of same sand piles model, *Theoret. Comput. Sci.* 262 (2001), 525–556.
9. P. Massazza, A CAT algorithm for sand piles, *PU.M.A.* vol. 19 (2008) n.2-3, 147–158.
10. P. Massazza and R. Radicioni A CAT algorithm for the exhautive generation of ice piles, *RAIRO* Informatique théorique 44 (2010), 525–543.
11. P. Massazza, On the Exhaustive Generation of Symmetric Sand Piles, *Proc. of GASCom 2010*, Montréal, 2-4 September 2010.
12. H. D. Phan, Two sided sand piles model and unimodal sequences, *RAIRO* Informatique théorique 42 (2008), 631–646.