An Efficient Algorithm for Generating Symmetric Ice Piles

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

[1]LIAFA, Université Paris Diderot

[2]Dipartimento di Scienze Teoriche e Applicate
Università degli Studi dell'Insubria-Varese

ICTCS 2014
Perugia, September 17-19th 2014

Abstract

Our Interest

The properties of $\text{SIPM}_k(n)$, a new granular dynamical system representing *symmetric ice piles*

Goal

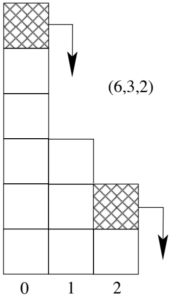
The design of an efficient (CAT) algorithm which generates $\text{SIPM}_k(n)$

The Sand Pile Game

Sand Piles are integer sequences which describe the states of the Sand Pile Game

Initial state: (n), n sand grains in column 0,

RFall Rule: in $(s_0, \ldots, s_l)$ a grain can fall from column i downto $i+1$ iff the height difference is at least 2, $s_i - s_{i+1} \geq 2$



Sand Pile Model

Definition (SPM(n))

SPM(n) is the set of linear partitions of n obtained by closing $\{(n)\}$ w.r.t. RFall.

$$\text{RFall}(s,i) = \begin{cases} (s_0, \ldots, s_{i-1}, s_i - 1, s_{i+1} + 1, \ldots, s_l) & \text{if } 0 \leq i \leq l, \\ & s_i - s_{i+1} \geq 2 \\ \bot & \text{otherwise} \end{cases}$$

- introduced by Back, Tang, Wiesenfeld ['88]
- deeply investigated by Goles, Kiwi ['92]
- used to simulate physical phenomena (e.g. avalanches)
- particular case of the *chip firing game*

Sand Pile Model

Definition (SPM(n))

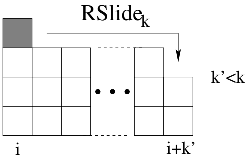
SPM(n) is the set of linear partitions of n obtained by closing $\{(n)\}$ w.r.t. RFall.

$$\text{RFall}(s,i) = \begin{cases} (s_0,\ldots,s_{i-1},s_i-1,s_{i+1}+1,\ldots,s_l) & \text{if } 0 \le i \le l, \\ & s_i - s_{i+1} \ge 2 \\ \perp & \text{otherwise} \end{cases}$$

- introduced by Back, Tang, Wiesenfeld ['88]
- deeply investigated by Goles, Kiwi ['92]
- used to simulate physical phenomena (e.g. avalanches)
- particular case of the *chip firing game*

Ice Pile Model

Ice grains can slide...

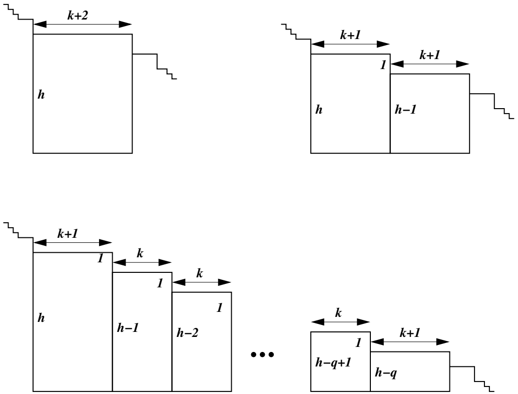


Definition ($IPM_k(n)$)

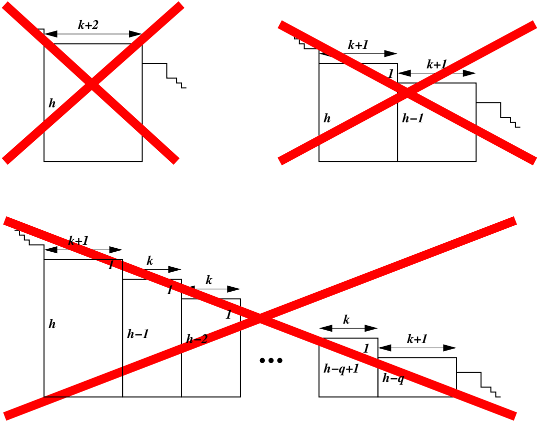
For any $k > 0$, $IPM_k(n)$ is the set of linear partitions of n obtained by closing $\{(n)\}$ w.r.t. RFall and $RSlide_k$.

- introduced by Goles, Morvan, Phan ['98]
- CAT generated by Massazza, Radicioni [2010]

Accessibility in $\text{IPM}_k(n)$



Accessibility in $\mathrm{IPM}_k(n)$



SSPM(n)

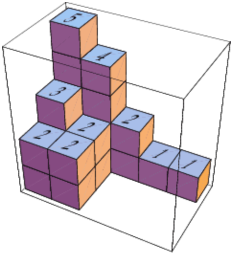
The Symmetric Sand Pile Model SSPM(n)

- introduced by Formenti, Masson, Pisokas ['06]
- studied by Phan ['08]
- symmetric version of SPM(n) (admits also left moves)

Definition (SSPM(n))

SSPM(n) is the set of integer sequences obtained by closing $\{(n)\}$ w.r.t. RFall, LFall (indices of columns can be negative).

BSPM(n)



The Bidimensional Sand Pile Model BSPM(n) introduced by Duchi, Mantaci, Phan, Rossin ['06] as a generalization of SPM(n) to 2D.

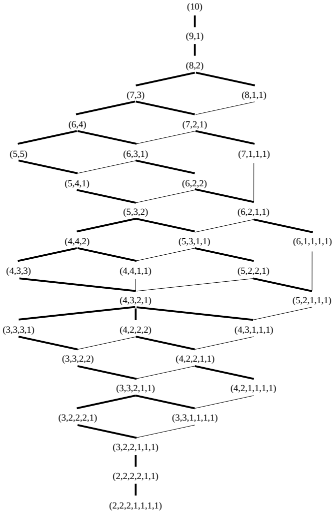
SPM,IPM$_k$, SSPM and BSPM: known results

	SPM(n)	IPM$_k$(n)	SSPM(n)	BSPM
lattice	yes	yes	no	no
characterization	elements fixed point	elements fixed point	elements fixed points	? ?
CAT generation	yes	yes	yes	?

CAT Algorithm for $\mathrm{IPM}_k(n)$ - Spanning Tree

CAT Algorithm (Massazza-Radicioni)

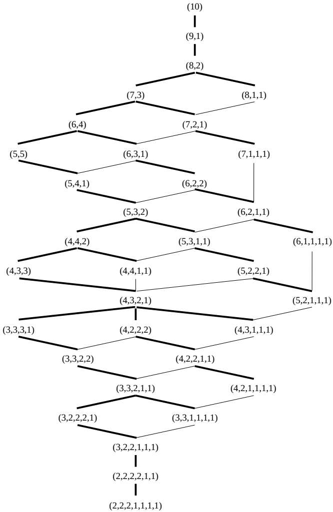
- Spans the poset using a tree;
- Each element is generated applying (the rightmost) $\mathrm{IPM}_k(n)$ move to the grand ancestor of the current partition;
- Partitions are generated in increasing neglex order.



CAT Algorithm for $\mathrm{IPM}_k(n)$ - Spanning Tree

CAT Algorithm (Massazza-Radicioni)

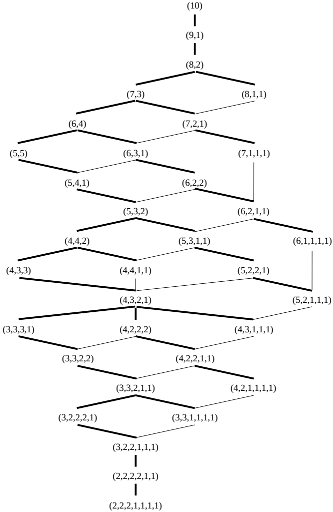
- Spans the poset using a tree;
- Each element is generated applying (the rightmost) $\mathrm{IPM}_k(n)$ move to the grand ancestor of the current partition;
- Partitions are generated in increasing neglex order.



CAT Algorithm for $\mathrm{IPM}_k(n)$ - Spanning Tree

CAT Algorithm (Massazza-Radicioni)

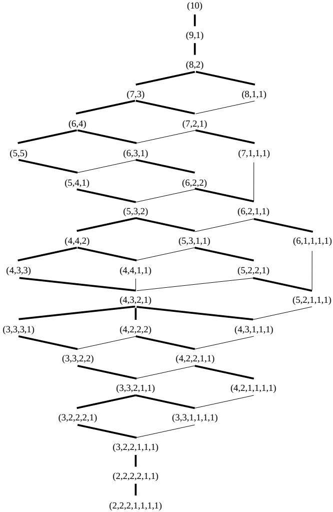
- Spans the poset using a tree;
- Each element is generated applying (the rightmost) $\mathrm{IPM}_k(n)$ move to the grand ancestor of the current partition;
- Partitions are generated in increasing neglex order.



CAT Algorithm for $\mathrm{IPM}_k(n)$ - Spanning Tree

CAT Algorithm (Massazza-Radicioni)

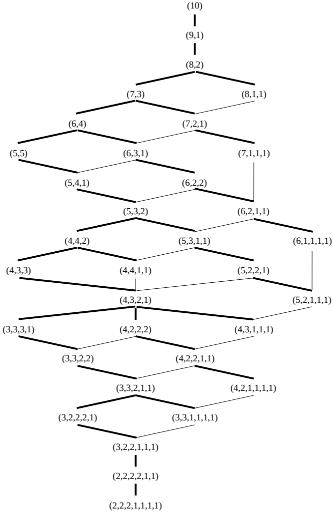
- Spans the poset using a tree;
- Each element is generated applying (the rightmost) $\mathrm{IPM}_k(n)$ move to the grand ancestor of the current partition;
- Partitions are generated in increasing neglex order.



CAT Algorithm for $\mathrm{IPM}_k(n)$ - Spanning Tree

CAT Algorithm (Massazza-Radicioni)

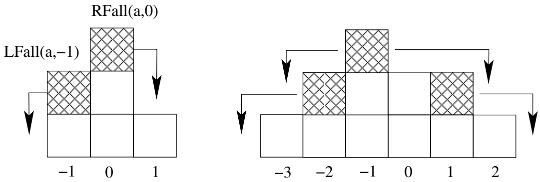
- Spans the poset using a tree;
- Each element is generated applying (the rightmost) $\mathrm{IPM}_k(n)$ move to the grand ancestor of the current partition;
- Partitions are generated in increasing neglex order.



Symmetric Ice piles

Definition ($\mathrm{SIPM}_k(n)$)

$\mathrm{SIPM}_k(n)$ is the set of integer sequences obtained by closing $\{(n)\}$ w.r.t. RFall, LFall, RSlide_k, LSlide_k



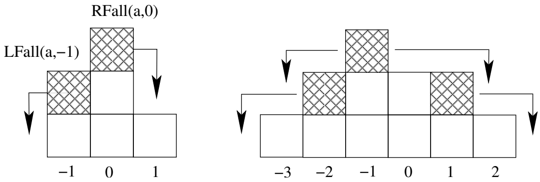
Fall and Slide moves ($k = 3$)

- *unimodal sequence* of n: $a = (a_0, \ldots, a_l)$ such that $\sum_{i=0}^{l} a_i = n$ and $a_0 \leq a_1 \leq \ldots \leq a_j \geq a_{j+i} \geq \ldots \geq a_l$ for some j.
- A generalized unimodal sequence is a pair (a, j) where a is a unimodal sequence (the form) and j an integer (the position).

Symmetric Ice piles

Definition ($\mathrm{SIPM}_k(n)$)

$\mathrm{SIPM}_k(n)$ is the set of integer sequences obtained by closing $\{(n)\}$ w.r.t. $\mathrm{RFall}, \mathrm{LFall}, \mathrm{RSlide}_k, \mathrm{LSlide}_k$



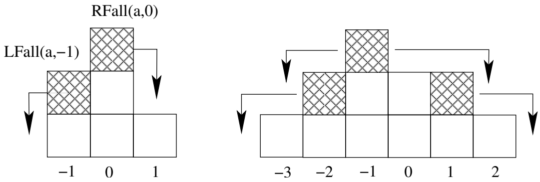
Fall and Slide moves ($k = 3$)

- *unimodal sequence* of n: $a = (a_0, \ldots, a_l)$ such that $\sum_{i=0}^{l} a_i = n$ and $a_0 \leq a_1 \leq \ldots \leq a_j \geq a_{j+i} \geq \ldots \geq a_l$ for some j.
- A generalized unimodal sequence is a pair (a, j) where a is a unimodal sequence (the form) and j an integer (the position).

Symmetric Ice piles

Definition ($\mathrm{SIPM}_k(n)$)

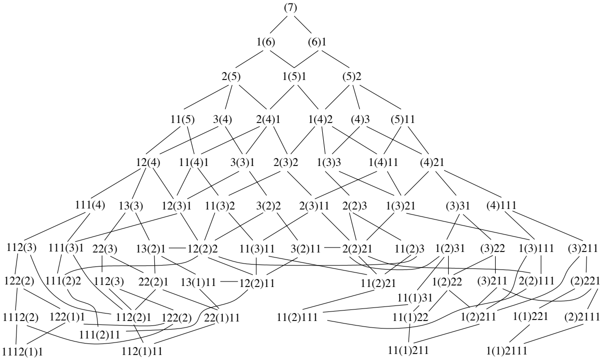
$\mathrm{SIPM}_k(n)$ is the set of integer sequences obtained by closing $\{(n)\}$ w.r.t. $\mathrm{RFall}, \mathrm{LFall}, \mathrm{RSlide}_k, \mathrm{LSlide}_k$



Fall and Slide moves ($k = 3$)

- *unimodal sequence* of n: $a = (a_0, \ldots, a_l)$ such that $\sum_{i=0}^{l} a_i = n$ and $a_0 \leq a_1 \leq \ldots \leq a_j \geq a_{j+i} \geq \ldots \geq a_l$ for some j.
- A generalized unimodal sequence is a pair (a, j) where a is a unimodal sequence (the form) and j an integer (the position).

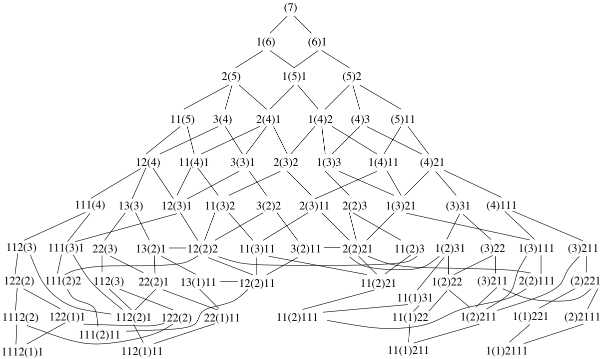
The poset $SIPM_2(7)$



And the order we generate it in:

(7),(6)1,1(6),(5)2,(5)11,1(5)1,2(5),11(5),(4)3,(4)21,(4)111,1(4)2,1(4)11, 2(4)1, 11(4)1, 3(4), 12(4),111(4),3(3)1,(3)31,13(3),1(3)3,3(2)2,(3)22,3(2)11, (3)211,13(2)1, 1(3)21,13(1)11, 1(3)111, 2(3)2,11(3)2, 2(3)11,11(3)11,12(3)1,1(2)31,111(3)1,11(1)31, 22(3),2(2)3,112(3), 11(2)3,22(2)1,2(2)21,(2)221,122(2),12(2)2,1(2)22, 22(1)11, 2(2)111,(2)2111,122(1)1,12(2)11, 1(2)211,112(2)1,11(2)21,1(1)221,1112(2),111(2)2, 11(1)22,112(1)11,11(2)111,1(1)2111, 1112(1)1,111(2)11,11(1)211.

The poset $SIPM_2(7)$



And the order we generate it in:

(7),(6)1,1(6),(5)2,(5)11,1(5)1,2(5),11(5),(4)3,(4)21,(4)111,1(4)2,1(4)11, 2(4)1, 11(4)1, 3(4), 12(4),111(4),3(3)1,(3)31,13(3),1(3)3,3(2)2,(3)22,3(2)11, (3)211,13(2)1, 1(3)21,13(1)11, 1(3)111, 2(3)2,11(3)2, 2(3)11,11(3)11,12(3)1,1(2)31,111(3)1,11(1)31, 22(3),2(2)3,112(3), 11(2)3,22(2)1,2(2)21,(2)221,122(2),12(2)2,1(2)22, 22(1)11, 2(2)111,(2)2111,122(1)1,12(2)11, 1(2)211,112(2)1,11(2)21,1(1)221,1112(2),111(2)2, 11(1)22,112(1)11,11(2)111,1(1)2111, 1112(1)1,111(2)11,11(1)211.

Position

Combinatorial results allow to prove that, for a given form, the possible positions form an integer interval, whose extrema can be computed in $O(1)$.

The formulae for computing these extrema depend on the weight and length of the ice pile, as well as on the width of the plateau and on the result of an operation on ice piles we called completion.

The generating algorithm only has to generate the correct forms, and then, for each of them, compute the possible values for the position.

Position

Combinatorial results allow to prove that, for a given form, the possible positions form an integer interval, whose extrema can be computed in $O(1)$.

The formulae for computing these extrema depend on the weight and length of the ice pile, as well as on the width of the plateau and on the result of an operation on ice piles we called completion.

The generating algorithm only has to generate the correct forms, and then, for each of them, compute the possible values for the position.

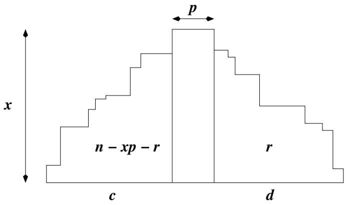
Position

Combinatorial results allow to prove that, for a given form, the possible positions form an integer interval, whose extrema can be computed in $O(1)$.

The formulae for computing these extrema depend on the weight and length of the ice pile, as well as on the width of the plateau and on the result of an operation on ice piles we called completion.

The generating algorithm only has to generate the correct forms, and then, for each of them, compute the possible values for the position.

Type of a form



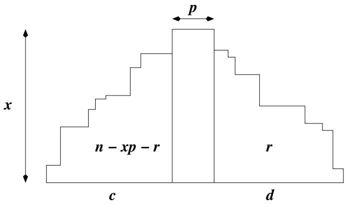
Definition (Type)

The **type** of $a \in US(n)$, $a = c \cdot x^{[p]} \cdot d$, is the triple (x, p, r) with $r = size(d)$ and $height(c) < x$, $height(d) < x$

Some combinatorics allows to characterize types of forms of elements of $\mathrm{SIPM}_k(n)$, in particular :

- bounds for the values of x;
- bounds for the value of r depending on x, p.

Type of a form



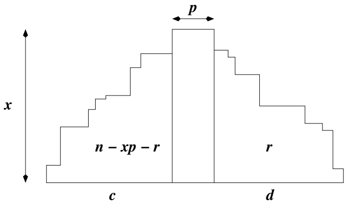
Definition (Type)

The **type** of $a \in US(n)$, $a = c \cdot x^{[p]} \cdot d$, is the triple (x, p, r) with $r = size(d)$ and $height(c) < x$, $height(d) < x$

Some combinatorics allows to characterize types of forms of elements of $\mathrm{SIPM}_k(n)$, in particular :

- bounds for the values of x;
- bounds for the value of r depending on x, p.

Type of a form



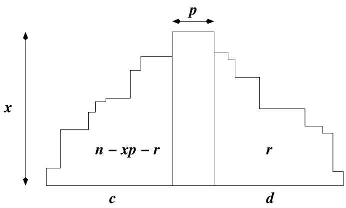
Definition (Type)

The **type** of $a \in US(n)$, $a = c \cdot x^{[p]} \cdot d$, is the triple (x, p, r) with $r = size(d)$ and $height(c) < x$, $height(d) < x$

Some combinatorics allows to characterize types of forms of elements of $\mathrm{SIPM}_k(n)$, in particular :

- bounds for the values of x;
- bounds for the value of r depending on x, p.

Type of a form



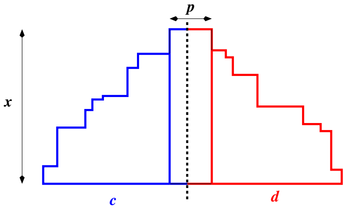
Definition (Type)

The **type** of $a \in US(n)$, $a = c \cdot x^{[p]} \cdot d$, is the triple (x, p, r) with $r = size(d)$ and $height(c) < x$, $height(d) < x$

Some combinatorics allows to characterize types of forms of elements of $\mathrm{SIPM}_k(n)$, in particular :

- bounds for the values of x;
- bounds for the value of r depending on x, p.

Forms

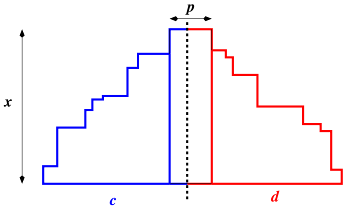


Theorem

$a \in US(n)$ is the form of an element in $SIPM_k(n)$ iff it can be decomposed in to a reverse ice pile and an ice pile (both in IPM_k).

- In particular, $p < 2k + 3$.

Forms

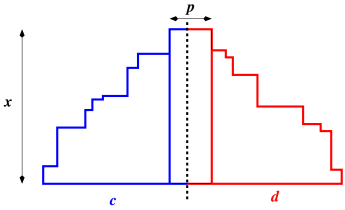


Theorem

$a \in US(n)$ is the form of an element in $SIPM_k(n)$ iff it can be decomposed in to a reverse ice pile and an ice pile (both in IPM_k).

- In particular, $p < 2k + 3$.

Forms

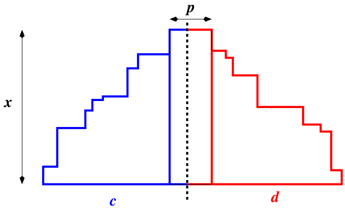


Theorem

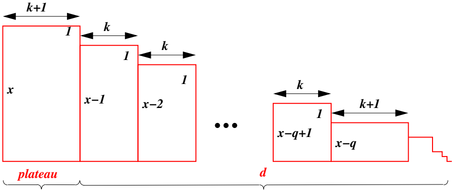
$a \in US(n)$ is the form of an element in $SIPM_k(n)$ iff it can be decomposed in to a reverse ice pile and an ice pile (both in IPM_k).

- In particular, $p < 2k + 3$.

Forms

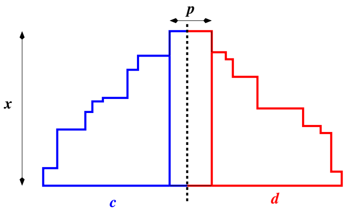


The last theorem also puts constrains on c and d, as both $x^{[b]} \cdot d$ and the reversal of $x^{[a]} \cdot c$ $(a + b = p)$ can not have a prefix like:



(notion of *x*-critical partitions).

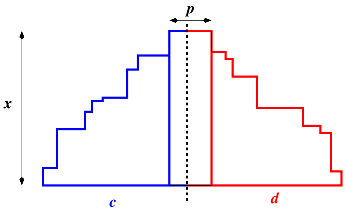
Principle of Algorithm for forms



Our algorithm:

- Loops on x, p, r within the established bounds;
- for each value r, all compatible c's and d's are generated with two nested calls to Massazza-Radicioni algorithm;
- depending on the value of p (three cases $p \leq 2k$, $p = 2k + 1$, $p = 2k + 2$), we know a combinatorial characterization of the minimal ice pile d_0 (resp. c_0), from which the Massazza-Radicioni algorithm has to start the generation (the sought ice piles are precisely those generated by it);

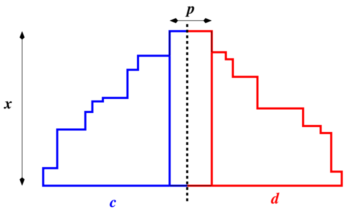
Principle of Algorithm for forms



Our algorithm:

- Loops on x, p, r within the established bounds;
- for each value r, all compatible c's and d's are generated with two nested calls to Massazza-Radicioni algorithm;
- depending on the value of p (three cases $p \leq 2k$, $p = 2k + 1$, $p = 2k + 2$), we know a combinatorial characterization of the minimal ice pile d_0 (resp. c_0), from which the Massazza-Radicioni algorithm has to start the generation (the sought ice piles are precisely those generated by it);

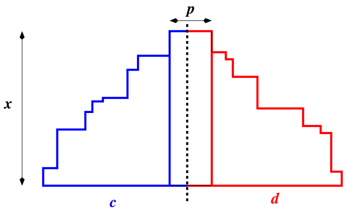
Principle of Algorithm for forms



Our algorithm:

- Loops on x, p, r within the established bounds;
- for each value r, all compatible c's and d's are generated with two nested calls to Massazza-Radicioni algorithm;
- depending on the value of p (three cases $p \leq 2k$, $p = 2k + 1$, $p = 2k + 2$), we know a combinatorial characterization of the minimal ice pile d_0 (resp. c_0), from which the Massazza-Radicioni algorithm has to start the generation (the sought ice piles are precisely those generated by it);

Principle of Algorithm for forms



Our algorithm:

- Loops on x, p, r within the established bounds;
- for each value r, all compatible c's and d's are generated with two nested calls to Massazza-Radicioni algorithm;
- depending on the value of p (three cases $p \leq 2k$, $p = 2k + 1$, $p = 2k + 2$), we know a combinatorial characterization of the minimal ice pile d_0 (resp. c_0), from which the Massazza-Radicioni algorithm has to start the generation (the sought ice piles are precisely those generated by it);

Conclusions

Main results

- a combinatorial characterization of forms and positions of elements of $\mathrm{SIPM}_k(n)$
- a CAT algorithm which generates $\mathrm{SIPM}_k(n)$

Further works

extend the results to $\mathrm{BSPM}(n)$ or to other 2D models inspired to $\mathrm{SPM}(n)$