



HAL
open science

Polynomial Time Reconstruction of Regular Convex Lattice Sets from their Horizontal and Vertical X-Rays

Yan Gerard

► **To cite this version:**

Yan Gerard. Polynomial Time Reconstruction of Regular Convex Lattice Sets from their Horizontal and Vertical X-Rays. 2018. hal-01854636

HAL Id: hal-01854636

<https://hal.science/hal-01854636>

Preprint submitted on 6 Aug 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Polynomial Time Reconstruction of Regular Convex Lattice Sets from their Horizontal and Vertical X-Rays

Yan Gérard

LIMOS, University Clermont Auvergne, France
yan.gerard@uca.fr

Abstract. We consider a problem of Discrete Tomography that has been open for 20 years: the reconstruction of convex lattice sets from their horizontal and vertical *X-rays* (X-rays is the mathematical term for the number of points of a set in a sequence of consecutive lines). We prove that it can be solved in polynomial time for the subclass of the *regular* lattice sets. Regularity is a property related to the relative position of the points of the set with extreme abscissa and ordinate. This algorithm that we call **ConvexTomo** follows the classical strategy initiated by E. Barcucci et al. for the reconstruction of horizontally and vertically convex 4-connected lattice sets. The approach introduced for the reconstruction of this class of lattice sets can be adapted until the creation of combinatorial structures called *switching components*. They are used to express horizontal and vertical convexity as a conjunction of 2-clauses. Then polynomial time algorithms solving 2-SAT provide polynomial time algorithms of reconstruction. The difficulty to overcome is that convexity (and no more directional convexities) requires 3-clauses which makes this approach no more polynomial.

In this paper, we present a new approach encoding the research of a convex configuration of the switching components in the research of a path between two sets of vertices in a Directed Acyclic Graph. This reduction passes through the introduction of a new class of problems of computational and discrete geometry that we call *Convex Aggregation*: given a convex lattice set $A \subset \mathbb{Z}^2$ and an ordered finite family of lattice sets $B^i \subset \mathbb{Z}^2$ called *blocks* (blocks are around A), does there exist a non empty subset of blocks such that their union with A remains convex? We reduce the question to the research of a path connecting two sets of vertices in a Directed Acyclic Graph. Then we investigate its variant related to the research of a convex configuration of the switching components. This problem is made of four related problems of Convex Aggregation. We reduce it again in a more complex manner to the research of a path in Discrete Acyclic Graph. It provides the final step of the algorithm **ConvexTomo** with a polynomial time complexity whereas the clauses approaches might be exponential.

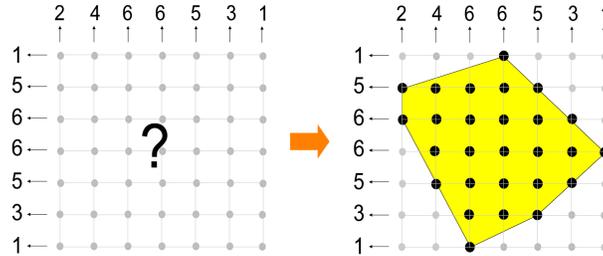


Fig. 1. Considered problem of Discrete Tomography: Find a convex polygon with given numbers of interior lattice points on the horizontal and vertical lines. The solution is a convex lattice set.

1 **Keywords:** Digital Geometry, Discrete Tomography, Convex lat-
 2 **tice sets, Filling operations, Switching Component, 3-SAT, Di-**
 3 **rected Acyclic Graph**

4 **1 Introduction**

5 **1.1 About Discrete Tomography**

6 In the mid 1990s, researchers in Material Science and especially in three
 7 dimensional Electron Microscopy previewed the development of an up-
 8 coming technology able to count the number of atoms of a material
 9 crossed by a beam of straight lines [8]. Under the same principle than
 10 Computerized Tomography, they intended to use this process in order to
 11 reconstruct the 3D structure of different materials (proteins, crystals...)
 12 with a very high level of precision. They started to use the algorithms
 13 of Computerized Tomography well-known in Medical Imaging. They dis-
 14 covered that these algorithms designed for the investigation of materials
 15 at a scale where it can be assumed to be continuous were absolutely not
 16 well-suited at a level where the set of atoms is closer to a discrete set
 17 of points. The discrete nature of the objects to be reconstructed is the
 18 first difficulty which makes CT algorithms ineffective at the atomic scale.
 19 A second difficulty comes from the very low number of X-rays -from 2
 20 to 10- which can be used in Material Science since the X-rays damage
 21 the atomic structure. As comparison, CT-scans provide usually hundreds
 22 of X-rays. The third difference with Computerized Tomography is that
 23 for the reconstruction of the atomic structure of crystals (see [3, 26] for
 24 crystalline structures of nano-particules computed with Discrete Tomog-
 25 raphy in the 2010s years), the atoms are centered on a lattice so that the

26 problem becomes the reconstruction of a lattice set, namely in dimension
 27 2 a binary matrix.

28 The development of the technology for counting the number of atoms
 29 on straight lines took finally more time than expected but the impulse
 30 was given to explore this new range of questions dealing with the re-
 31 construction of discrete sets of points. The sequence of cardinalities of
 32 the intersections of a discrete set with consecutive parallel lines has been
 33 called by keeping the physical term of *X-ray* while the reconstruction of
 34 a discrete set from X-rays took the name of *Discrete Tomography* [15, 20,
 35 21]. Due to the technical principle providing the measurements and the
 36 complexity of the considered problems, a special attention has been given
 37 on the problem in dimension 2.

38 1.2 Problem Statement

39 An *X-ray* is the sequence of the cardinalities of the intersection between
 40 a given lattice set and the consecutive diophantine lines in a chosen direc-
 41 tion. In the two-dimensional case of the vertical and horizontal directions,
 42 it leads to the following definition:

43 **Definition 1.** Given a finite lattice set $S \subset [1..m] \times [1..n]$, its vertical X-
 44 ray $V(S) \subset \mathbb{Z}^m$ is the vector of coordinates $v_i(S) = |\{(x, y) \in S | x = i\}|$
 45 for $1 \leq i \leq m$ and its horizontal X-ray $H(S) \subset \mathbb{Z}^n$ is the vector of
 46 coordinates $h_j(S) = |\{(x, y) \in S | y = j\}|$ for $1 \leq j \leq n$ (Fig.2).

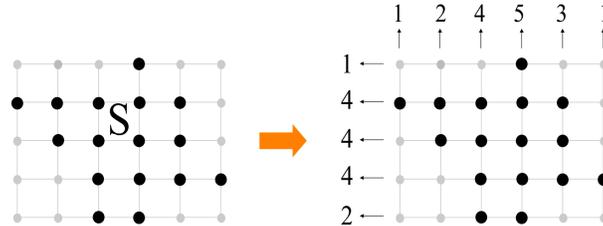


Fig. 2. The horizontal and vertical X-rays of the lattice set S are the vectors $V(S) = (1, 2, 4, 5, 3, 1)$ and $H(S) = (2, 4, 4, 5, 1)$.

47 It leads to introduce a generic problem of Discrete Tomography. The
 48 question is the existence of a lattice set with given X-rays and belonging
 49 to a given class A of lattice sets:

50 *Problem 1* ($DT_{\mathcal{A}}(h, v)$).
 51 Given a class \mathcal{A} of finite lattice sets,
 52 **Input:** two vectors $v \in \mathbb{Z}^m$ and $h \in \mathbb{Z}^n$.
 53 **Output:** does there exist a lattice set $S \in \mathcal{A}$ included in the rectangle
 54 $[1..m] \times [1..n]$ with $V(S) = v$ and $H(S) = h$?

55 The class \mathcal{A} is a parameter of Problem $DT_{\mathcal{A}}(h, v)$. We introduce the
 56 class \mathcal{C} of convex lattice sets.

57 **Definition 2.** A lattice set $S \subset \mathbb{Z}^d$ is convex if it is equal to its intersec-
 58 tion with its real convex hull $S = \text{conv}_{\mathbb{R}^d}(S) \cap \mathbb{Z}^d$ (Fig.3). The class of
 59 the convex lattice sets is denoted \mathcal{C} .

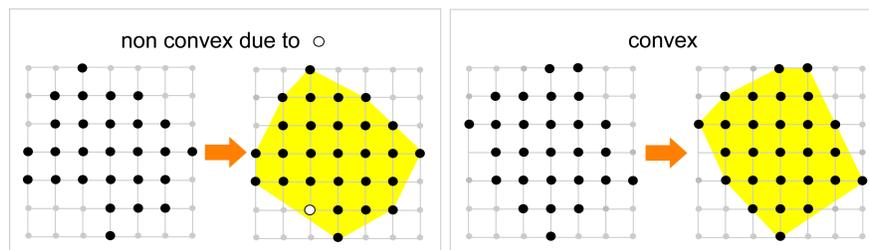


Fig. 3. A convex lattice set is equal to the intersection of its convex hull (in yellow) with the lattice \mathbb{Z}^2 .

60 In other words, the convex lattice sets are the intersections of con-
 61 vex polygons with the lattice \mathbb{Z}^2 . The complexity of their reconstruction
 62 $DT_{\mathcal{C}}(h, v)$ is a twenty years old open question (Fig.1). The purpose of the
 63 paper is to break the status quo and provide a partial answer opening
 64 new perspectives.

65 2 State of the Art

66 While Computerized Tomography has been stated on the prior works of J.
 67 Radon (1917) [24] or Fourier Analysis [4], Discrete Tomography found its
 68 basis in results of D. Gale and H.J. Ryser (1957) [13, 25] or in the more
 69 general theory of flows in networks by L.R. Ford and D.R. Fulkerson
 70 (1956) [12].

71 According to these fundamental results, if we consider the whole class
 72 denote \mathcal{W} of all lattice sets, the problem $DT_{\mathcal{W}}(h, v)$ can be solved in

73 polynomial time [13, 25]. Another way to consider the problem is on a set
 74 of edges of a complete bipartite graph. It can be solved by any max-flow
 75 algorithm (Fig.4).

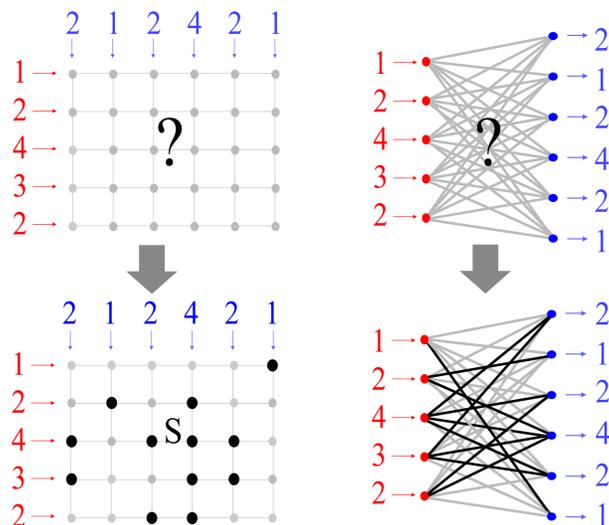


Fig. 4. Classical problem of reconstruction. The problem of reconstruction of a lattice set with prescribed horizontal and vertical X-rays can be reformulated in terms of flows in a bipartite graph. It can also be solved with the polynomial time algorithm of H.J. Ryser [25].

76 Many variants of this problem have been investigated, not only with
 77 horizontal and vertical X-rays but in different dimensions, with different
 78 directions of X-rays and different kinds of atoms. In dimension 3, the
 79 problem is related with timetables or data security. Both variants are
 80 NP-complete [11, 22] and this extension is related with multi-commodity
 81 flow problems [18]. The reconstruction of sets with different kinds of atoms
 82 can also be considered with one X-ray per type of material. The problem
 83 becomes again NP-hard from two different kinds of atoms [17, 10]. Still
 84 in dimension 2, the number of X-rays can be increased with a result of
 85 NP-completeness from three X-rays [16].

86 In the two-dimensional case with horizontal and vertical X-rays, the
 87 complexities of the problem $DT_{\mathcal{A}}(h, v)$ have already been determined
 88 for many classes \mathcal{A} . The problem is NP-complete for the class of the 4-
 89 connected lattice set (4-connected finite subsets of \mathbb{Z}^2 are called *polyomi-*

90 *noes* and their class is denoted \mathcal{P}). We have also results with directional
 91 convexities. By denoting \mathcal{H} the class of horizontally convex (H-convex)
 92 lattice sets i.e with consecutive points in any row (Fig.??), the problem
 93 is NP-complete [2]. We have of course the same result for the class de-
 94 noted \mathcal{V} of the vertically convex (V-convex) lattice sets. If we consider
 95 the class of the horizontally and vertically convex (HV-convex) lattice
 96 sets $\mathcal{H} \cap \mathcal{V}$, the problem remains again NP-complete [27] (Fig.??). No-
 97 tice that the reconstruction of HV-convex lattice sets is a particular case
 98 of puzzle games called nonograms. In summary, while the initial prob-
 99 lem $DT_{\mathcal{W}}(h, v)$ without complementary constraints on the solutions can
 100 be solved in polynomial time, all the variants $DT_{\mathcal{A}}(h, v)$ with \mathcal{P} , \mathcal{H} , \mathcal{V} ,
 101 $\mathcal{H} \cap \mathcal{V}$ as class \mathcal{A} are NP-complete. These complexities in NP are how-
 102 ever counter-balanced by two major results of the field published in two
 103 seminal papers [2, 14].

- 104 – Horizontally and vertically 4-connected subsets of \mathbb{Z}^2 can be recon-
 105 structed in polynomial time: $DT_{\mathcal{A}}(h, v)$ is polynomial for the class
 106 $\mathcal{A} = \mathcal{H} \cap \mathcal{V} \cap \mathcal{P}$ [2].
- 107 – On the other side, results of uniqueness have been obtained for the
 108 class \mathcal{C} of convex lattice sets with different number of directions of
 109 X-rays. R. Gardner and P. Grizmann characterized the sets of d di-
 110 rections for which any convex lattice set is uniquely determined by
 111 its X-rays [14]. For $n = 2$ or $n = 3$ directions, for any directions,
 112 there exist ambiguous pairs or triplet of X-rays. For $n \geq 7$ directions,
 113 all convex lattice sets are uniquely determined by their X-rays. For
 114 $3 < n < 7$, the so-called cross-ratios of the directions provide a charac-
 115 terization of the sets of direction providing uniqueness or ambiguous
 116 X-rays [14]. With the directions of X-rays providing uniqueness, these
 117 results have been completed by a polynomial time algorithm of recon-
 118 struction [6]. This algorithm follows the same principle than the one
 119 used for the reconstruction of HV-convex polyominoes [2].

Class \mathcal{A}	H and V X-rays	4 directions or more
$\mathcal{H} \cap \mathcal{V} \cap \mathcal{P}$ (HV-convex polyominoes)	$DT_{\mathcal{H} \cap \mathcal{V} \cap \mathcal{P}}(h, v)$ polynomial time [1]	another open question
\mathcal{C} (convex lattice sets)	$DT_{\mathcal{C}}(h, v)$ open question	polynomial time (if uniqueness) [14, 6]

Table 1. Milestones results

120 The problem $DT_{\mathcal{C}}(h, v)$ that we consider in the paper is very close
 121 to the two milestones results (Tab.1). It deals with very simple objects,
 122 convex lattice sets, and the most simple directions of X-rays: horizontal
 123 and vertical. After twenty years of silence, the question of its complexity
 124 became recently subject of a new attention [9].

125 The complexity of $DT_{\mathcal{C}}(h, v)$ has not been yet determined because the
 126 principles used for providing the polynomial time algorithms of Tab.1 [2,
 127 6] do not hold. First, we don't have the uniqueness property used for the
 128 polynomial time algorithm from 4 directions of X-rays [6]. There are many
 129 ambiguities expressed by boolean variables. Secondly, the combinatorial
 130 expression of the convexity constraint requires 3-clauses whereas HV-
 131 convexity is expressed by a conjunction of 2-clauses which can be solved
 132 in a polynomial time.

133 2.1 Main Result

134 We prove in the paper that the problem $DT_{\mathcal{C}}(h, v)$ of the reconstruction
 135 of convex lattice sets with given X-rays can be solved in polynomial for
 136 the subclass of the *regular* convex lattice sets (Fig.6). Regularity is re-
 137 lated with the positions of the points of the lattice set with minimal and
 138 maximal abscissa and ordinate. These extreme points are the *feet* of the
 139 lattice set (Fig.5).

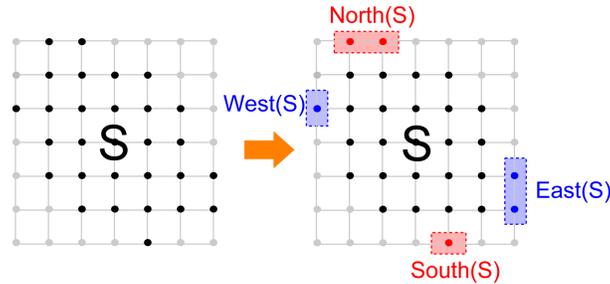


Fig. 5. The four feet of a lattice set S are denoted South, West, North and East.

140 **Definition 3.** Given a lattice set $S \subset \mathbb{Z}^2$, the South, West, North and
 141 East feet are its four subsets

142 $\text{South}(S) = \{(x, y) \in S \mid \forall (x', y') \in S, y' \geq y\},$

143 $\text{West}(S) = \{(x, y) \in S \mid \forall (x', y') \in S, x' \geq x\},$

144 $\text{North}(S) = \{(x, y) \in S \mid \forall (x', y') \in S, y' \leq y\}$,
 145 $\text{East}(S) = \{(x, y) \in S \mid \forall (x', y') \in S, x' \leq x\}$.
 146 The lattice set S is said irregular if there exists $(X, Y) \in \mathbb{Z}^2$ verifying
 147 either $x(\text{South}(S)) < X < x(\text{North}(S))$ and $y(\text{West}(S)) < Y < y(\text{East}(S))$,
 148 or $x(\text{South}(S)) > X > x(\text{North}(S))$ and $y(\text{West}(S)) > Y > y(\text{East}(S))$.
 149 Otherwise, S is said regular (Fig.6).

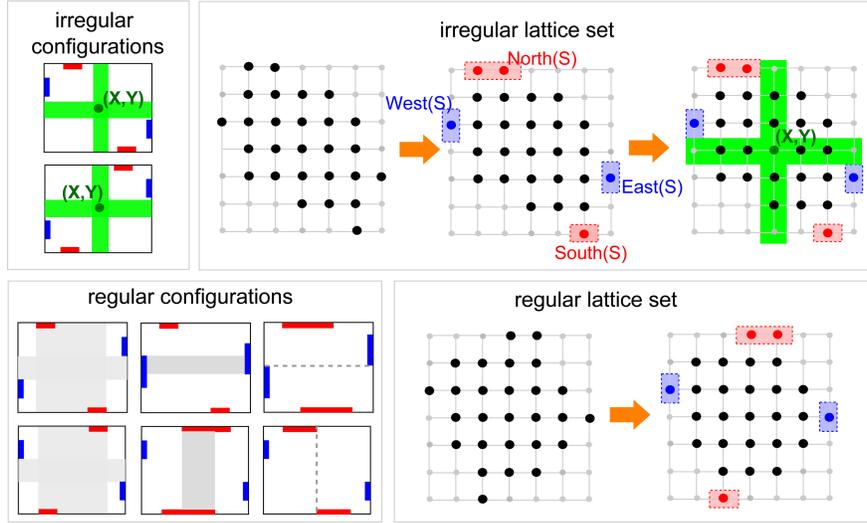


Fig. 6. Regular lattice sets are the lattice sets such that there exists no integer point (X, Y) (represented by the green cross) separating the pairs of feet in opposite corners.

150 The class of the regular lattice sets (Fig.6) is denoted \mathcal{R} (then the
 151 class of the regular convex lattice sets is $\mathcal{C} \cap \mathcal{R}$). The main result of the
 152 paper is the following theorem:

153 **Theorem 1.** *The algorithm `ConvexTomo` solves $DT_{\mathcal{C} \cap \mathcal{R}}(h, v)$ with a worst
 154 case time complexity in $O(m^4 n^4 + m^{11} n^2)$ where we can assume $m \leq n$.*

155 The time complexity of `ConvexTomo` is high but polynomial.

156 2.2 A Strategy Passing through Intermediary Results on a 157 New Convex Aggregation Problem

158 The algorithm `ConvexTomo` follows the guidelines of the three first steps
 159 of the classical polynomial time algorithm designed for the reconstruction

160 of HV-convex polyominoes [2]. The first step fixes the feet in a regular
 161 position (there are a polynomial number of possible positions). Under this
 162 assumption, the second step determines some points belonging to all the
 163 solutions (the set In) and some other points which can be excluded from
 164 all solutions (the set Out). It ends while the so-called *filling operations* do
 165 not allow to determine any new point or if a contradiction follows from a
 166 non empty intersection $\text{In} \cap \text{Out} \neq \emptyset$ (in this case, the considered position
 167 of the feet admits no solution).

	Data: $m, n, h \in \mathbb{Z}^n, v \in \mathbb{Z}^m$	
	Result: Regular convex lattice set $\text{In} \subset [1, m] \times [1, n]$ verifying $H(\text{In}) = h$ and $V(\text{In}) = v$	
1	for <i>regular configuration</i> South, East, North, West do	
2	/* Step 1 - Initialization of the feet	*/
3	Out \leftarrow <i>RectangleBorder</i> \ (South \cup East \cup North \cup West)	
4	In \leftarrow $\text{conv}_{\mathbb{R}^2}(\text{South} \cup \text{East} \cup \text{North} \cup \text{West}) \cap \mathbb{Z}^2$	
5	NW \cup NE \cup SE \cup SW \leftarrow <i>Decomposition</i> _{In} (Shell)	
6	/* Step 2 - Filling operations	*/
7	FillTomo($m, n, h, v, \text{NW}, \text{NE}, \text{SE}, \text{SW}, \text{In}, \text{Out}$)	
8	/* Step 3 - Compute Extended Switching Components	*/
9	Extended Switching Components \leftarrow <i>SwitchingComponents</i> ($m, n, h, v, \text{In}, \text{Out}$)	
10	/* Step 4 - Convex Aggregation	*/
11	In \leftarrow <i>ConvexAggregation</i> ($m, n, v, h, \text{NW}, \text{NE}, \text{SE}, \text{SW}, \text{In}, \text{Extended Switching Components}$)	
12	end	

Algorithm 1: Main algorithm $\text{ConvexTomo}(m, n, h, v)$

168 The set of the remaining undetermined points is a shell between the
 169 points of In and the ones of Out. It is denoted Shell. It is the set of the
 170 points which might belong to some solutions and be outside from oth-
 171 ers. These ambiguities are formalized with a decomposition of the shell
 172 in combinatorial structures called *switching components*. After the fill-
 173 ing operations, the prescribed X-rays are guaranteed but it remains to
 174 determine a configuration of the switching components providing a con-
 175 vex lattice set. Unfortunately, the classical approach encoding convexities
 176 with clauses is unable to provide a polynomial algorithm.

177 This combinatorial difficulty is the main challenge to overcome. Our
 178 main contribution is a new approach which allows to solve it in polyno-
 179 mial time in the case of the regular lattice sets. This result requires first
 180 recent results on regular switching components stated in Property 2 [19].
 181 It requires also a better understanding of the combinatorial problem. It

182 can be seen as a specific problem of *Convex Aggregation* of either the odd
183 parts, or the even parts of the switching components to the set In. This
184 problem being quite hard to handle, we provide first results on a more
185 generic problem of same nature:

186 Given a finite convex lattice set $A \subset \mathbb{Z}^2$ and q finite lattice sets B^i called
187 blocks, is it possible to add blocks to In so that their union remains
188 convex?

189 It is the problem denoted *ConvexAggregation*(A, B^i). It is stated
190 more precisely in Sect.4 in Problem 2. Under the assumption that the
191 blocks B^i can be ordered (their rows and columns are ordered), we pro-
192 vide a polynomial time algorithm for solving it although its expression
193 with boolean variables requires again 3-clauses. The result is stated in
194 Theorem 2. The main idea is that convexity can be controlled through
195 the local property that all the turning angles of the border have the same
196 orientation. The only information that we need to keep in memory for
197 building a convex contour is the last edge. This deep property allows to
198 solve *ConvexAggregation*(A, B^i) in polynomial time by reducing it to
199 the research of a path connecting two subsets of vertices in a Directed
200 Acyclic Graph (DAG) (Property 3).

201 If we come back to the problem of Discrete Tomography $DT_{\mathcal{C} \cap \mathcal{R}}(h, v)$,
202 it can be seen as four related problems of Convex Aggregation. We build
203 one DAG per problem and call them the *slave* DAGs. Then the relations
204 between the four solutions that we search for are controlled by building
205 a fifth DAG that we call *master* DAG. It allows to encode the existence
206 of a solution of $DT_{\mathcal{C} \cap \mathcal{R}}(h, v)$ with the considered feet in the existence of
207 a path joining two regions of the master DAG (Theorem 3). It provides
208 the fourth and final step of the algorithm **ConvexTomo** reconstructing a
209 convex solution of $DT_{\mathcal{C} \cap \mathcal{R}}(h, v)$ in polynomial time if there exists one.

210 The benefit of the paper is to solve this highly non trivial combinato-
211 rial problem $DT_{\mathcal{C} \cap \mathcal{R}}(h, v)$ with a completely new approach. The reduction
212 of the problem of Convex Aggregation to the research of path in a DAG
213 (Property 3) is an intermediary result which has its own interest.

214 **2.3 Plan**

215 The polynomial time algorithm **ConvexTomo** is presented in the following
216 order. We start with the presentation of its three first steps in Sec.3. Then,
217 the problem can be reformulated into a particular question of Convex
218 Aggregation. Before solving it, we need first to investigate a more simple
219 problem of the same kind. Sec.4 is devoted the presentation of generic
220 Convex Aggregation and its reduction to the research of a path in a DAG

221 (Property 3). We adapt the approach for the fourth and last step of the
 222 algorithm `ConvexTomo` with a master DAG and four slave DAGs in Sec.5.
 223 Its final complexity analysis provides a proof of Theorem 1.

224 **3 ConvexTomo - Steps 1/2/3 - From X-rays to Extended** 225 **Switching Components**

226 Following the guidelines of the original algorithm presented in [2] for
 227 $DT_{\mathcal{H} \cap \mathcal{V} \cap \mathcal{P}}(h, v)$, we start by fixing the position of the feet.

228 **3.1 Step 1 - Fixing the Position of the Feet and Initialization**

229 The algorithm `ConvexTomo` investigates all the regular configuration of
 230 the feet. One of them being chosen, we can determine that some points
 231 are necessary in a solution, if it exists one, while others can be excluded
 232 from any solution. This principle is formalized by working with a partition
 233 of $[1..m] \times [1..n]$ in three sets of points:

- 234 – The set `In` contains the points which are known to belong to all solu-
 235 tions.
- 236 – The set `Out` contains the points which are known to be excluded from
 237 all solutions.
- 238 – The set `Shell` is the set of the undetermined points.

239 Once that the feet have been fixed, we add in `Out` the points with $x = 1$
 240 or $x = m$ or $y = 1$ or $y = n$ which are not in the feet since under the
 241 assumption of the considered feet, we are sure that they don't belong to
 242 any solution. As we search for a convex solution, we initialize the set `In`
 243 with the convex hull of the feet (Fig.7).

244 The convex hull of `In` provides a partition of the undetermined points
 245 (the shell) in four subsets: the North West, North East, South East and
 246 South West borders. They are respectively denoted NW, NE, SE and SW
 247 (Fig.7). The assignation of the points of the shell to its four borders NW,
 248 NE, SE and SW is done by the function denoted `DecompositionIn`(Shell)
 249 in Alg.1.

250

251 *Complexity Analysis*

252 There are at most $m - 1$ possible positions for each one of the South
 253 and the North feet and $n - 1$ cases for the West and East feet. It makes
 254 less than $m^2 n^2$ configurations of the feet to explore.

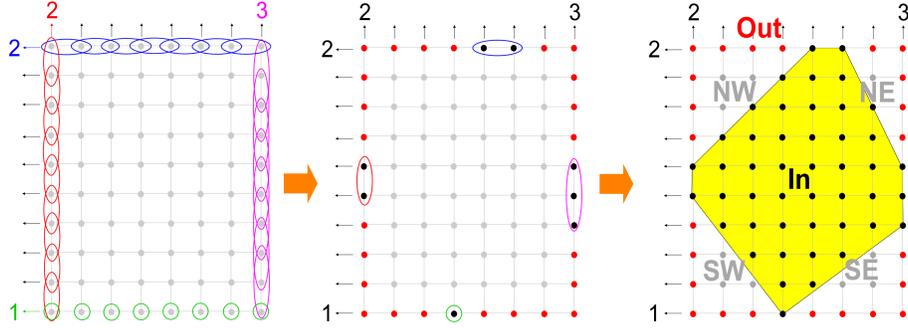


Fig. 7. Fix the feet and Initialize In, Out, NW, NE, SE and SW. On the left, the different possible positions of the feet. We do not consider the irregular configurations. In the middle, a regular position of the feet is chosen. The other points with $x = 1$ or $x = m$ or $y = 1$ or $y = n$ are added in Out (in red). The shell (in grey) is the set of the points which are not yet determined. On the right, due to convexity, we can add to In not only the four feet but directly the lattice points of their convex hull. The shell is thus decomposed in four subsets NW, NE, SE and SW according to their position relatively to the convex hull of In.

255 With the configuration of the feet, the computation of the convex hull
 256 of In takes a constant time. The assignation of the points of the lattice
 257 to In, Out, NW, NE, SE and SW requires less than $O(mn)$ operations.

258 **Proposition 1.** *The initialization takes $O(mn)$ time.*

259 3.2 Step 2 - Filling Operations - FillTomo

260 Filling operations are widely used in Discrete Tomography and we refer
 261 to [7, 6] for a more complete presentation of the operations with suitable
 262 data structures.

263 Starting from the feet, according to the X-rays, the first run of the
 264 filling operations fills directly the rows of the West and East feet and
 265 the columns of the South and North feet so that it does not remain any
 266 undetermined points on these lines. They are all either in In, or in Out.
 267 Notice that just with HV-convexity (we recall that convexity implies HV-
 268 convexity), a whole part of the lattice can be quickly determined (Fig.8).

269 A run of the filling operations is organized as follows.

- 270 1. Fill the rows with the procedure `FillRows(In, Out, h)`. It includes the
- 271 four filling operations illustrated Fig.10.
- 272 2. Fill the columns with the procedure `FillColumns(In, Out, v)` under
- 273 the same principle than for rows.

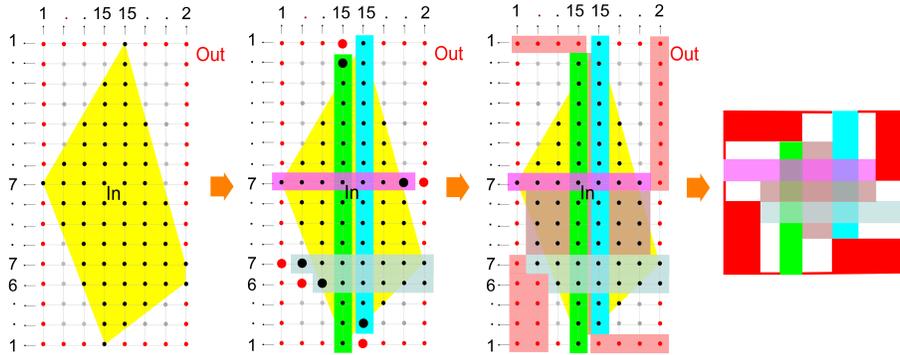


Fig. 8. Filling from the feet. Starting from the feet, the filling operations allow to fill directly an important part of the lattice. In one run, we obtain a figure that can be summarized in the right drawing, where the undetermined points are only in the white zones.

- 274 3. As we search for convex solutions, if In has been updated in the previous steps, we complete the update by replacing it by its discrete convex hull ($\text{In} \leftarrow \text{convv}(\text{In}) \cap \mathbb{Z}^2$).
- 275
- 276
- 277 4. If In or Out have been updated in the three previous steps, we complete Out by all the hidden points (a point x is hidden if $\text{conv}_{\mathbb{R}^2}(\{x\} \cup \text{In}) \cap \text{Out} \neq \emptyset$ - Fig.10).
- 278
- 279
- 280 5. Update the four borders SE, NE, NW and SW of the shell.

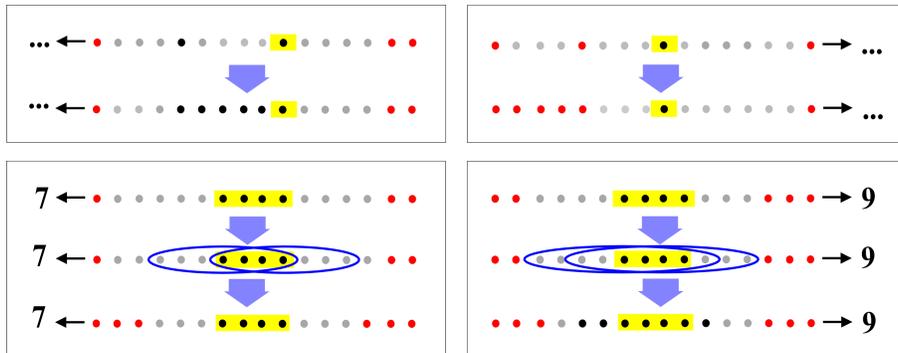


Fig. 9. The filling operations used in $\text{FillRows}(\text{In}, \text{Out}, h)$ and $\text{FillColumns}(\text{In}, \text{Out}, v)$. Notice that even if there no point of In in a row, there is always some points of its convex hull (in yellow) that can be used in a similar way.

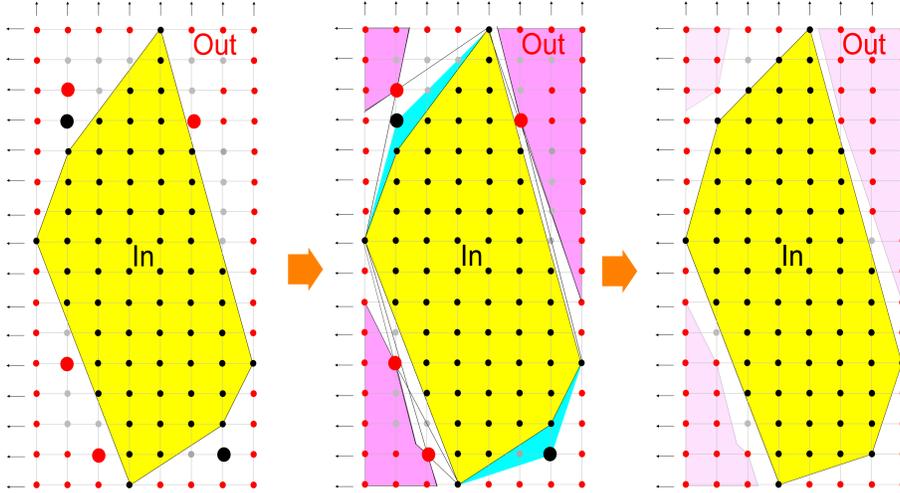


Fig. 10. The filling operations including the points of the convex hull of In and excluding the undetermined points hidden by a point of Out. On the left, we assume that new points of In and Out have been determined (the six large disks). Then we compute the new convex hull with the new points of In and use the tangent lines of the points of Out to compute the hidden zones (in pink).

281 We run the filling operations until falling in one of the two following
 282 cases:

- 283 – We stop if a run does not allow to determine any new point. It is
 284 for instance the case if we have no more undetermined point. In this
 285 case, we just have to check the solution. We might also have a set
 286 of undetermined points expressing the ambiguity of the input. They
 287 express the possibility that different lattice sets can be solutions.
- 288 – We stop if a point of In is added in Out or conversely. This contradic-
 289 tion means that the considered position of the feet does not provide
 290 solutions.

291 *Complexity Analysis*

292 With suitable data structures, each run of the classical filling oper-
 293 ations can be performed in $O(mn)$ time [7, 6]. The complexity do not
 294 differ for Algorithm FillTomo. We refer to [5] for the dynamic update of
 295 the convex hull as well as the computation of the hidden points (through
 296 the tangents). The time of computation of these two parts can be easily
 297 bounded by $O(mn)$ per new determined point.

298 As FillTomo stops if no new point is determined, the number of runs
 299 is at most the number of undetermined points which is bounded by mn .

```

1 FillTomo Data:  $m, n, v \in \mathbb{Z}^m, h \in \mathbb{Z}^n$  and four borders In, Out, NW, NE, SE,
   SW
   Result: Sets In, Out, SE, NE, NW and SW under the assumption of the given
   feet
2 while SE, NE, NW or SW have been decreased do
3   /* Filling operations */
4   FillRows(In, Out, h);
5   FillColumns(In, Out, h);
6   In  $\leftarrow \text{conv}_{\mathbb{R}^2}(\text{In}) \cap \mathbb{Z}^2$ ;
7   Out  $\leftarrow \text{Shadow}_{I_n}(\text{Out}) \cap \mathbb{Z}^2$ ;
8   /* Remove new determined points from SE, NE, NW or SW */
9   Decrease(SE, NE, NW, SW, In, Out);
10  if In  $\cap$  Out  $\neq \emptyset$  then
11    | return("no solution");
12  end
13 end

```

Algorithm 2: FillTomo($m, n, v, h, \text{In}, \text{Out}, \text{NW}, \text{NE}, \text{SE}, \text{SW}$)

300 Therefore, the overall time complexity of the function FillTomo is in
301 $O(m^2n^2)$.

302 **Proposition 2.** Algorithm FillTomo requires $O(m^2n^2)$ operations.

303 3.3 Step 3 - Computing the Switching Components - 304 SwitchingComponents

305 The third step of ConvexTomo occurs only if it remains undetermined
306 points. Their set, the shell has a lot of properties. If an undetermined
307 point $\underline{p} = (i, j)$ is in a South border $\text{SW} \cup \text{SE}$, then the point denoted
308 $\bar{p} = (i, j + v_i)$ is a North undetermined point (otherwise the filling op-
309 erations would have determined (i, j)). We define them as *vertical corre-*
310 *spondents*. In the same way, any West undetermined point $|p = (i, j)$ has
311 an horizontal East correspondent $p| = (i + h_j, j)$ (Fig.11). Horizontal and
312 vertical correspondences are symmetric relations.

313 Their main property is the following:

314 *Property 1.* We consider an instance $DT_C(h, v)$ with a position of the feet
315 leading to undetermined points at the end of the filling operations. For
316 any solution S of $DT_C(h, v)$, an undetermined point p is in S if and only
317 if its correspondents are not in S .

318 *Proof.* Corresponding points cannot be both in S because their distance
319 is h_j (horizontally) or v_i (vertically) would lead to have too many points

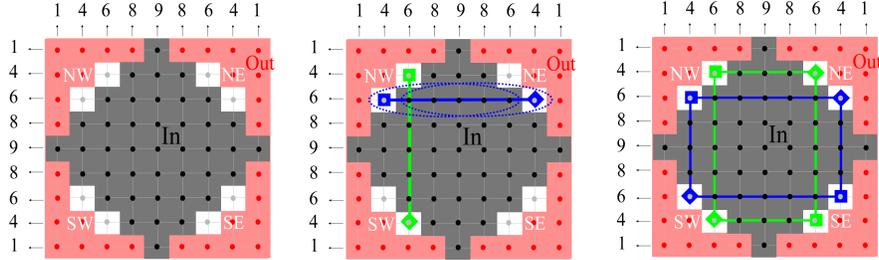


Fig. 11. Corresponding points. On the left, the undetermined points are drawn in grey in white cells. In the middle, a pair of vertical correspondents (green) and a pair of horizontal correspondents (blue). We represent the corresponding points alternatively with squares or diamonds. Notice that the segment represented by the dotted ellipse has only two possible positions. Due to its length, if it contains the square, it does not contain the diamond and conversely. On the right, the correspondences define closed paths called switching components which provide a partition of the undetermined points. For each switching components, either the squares, or the diamonds belong to a solution.

320 in the row or column of p . Conversely, they cannot be both outside from
 321 S because otherwise, it does not remain enough points between them to
 322 have the prescribed number of points on its row or column. \square

323 Starting from an undetermined point p_1 , a sequence of correspondents
 324 can be defined by induction: the point p_{2k} is the horizontal correspondent
 325 of p_{2k-1} while p_{2k+1} is the vertical correspondent of p_{2k} . As the set of the
 326 undetermined points is finite, the sequence is cyclic.

327 **Definition 4.** A switching component P is a closed path of alternatively
 328 horizontal and vertical corresponding undetermined points (Fig.12).

329 The switching components provide a partition of the shell. Due to
 330 Property 1, either the points with even indices, or the points with odd
 331 indices belong to a solution S . This binary state of the switching compo-
 332 nent with regard to a solution S can be encoded by a boolean variable
 333 denoted $P(S)$. We choose $P(S) = 1$ if the points with odd indices are in
 334 S and $P(S) = 0$ otherwise.

335 The classical approach developed in [2] for reconstructing HV-convex
 336 polyominoes is to search for an assignment of the boolean variables leading
 337 to an HV-convex solution S . The approach passes through the encoding
 338 of the HV-convexity constraint in a conjunction of 2-clauses. It leads to
 339 a 2-SAT instance (Fig.13).

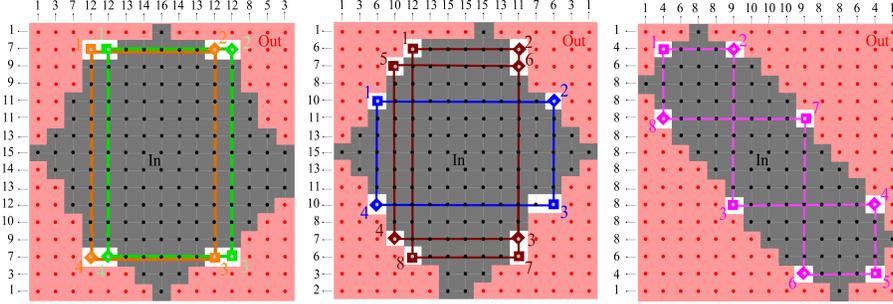


Fig. 12. Switching components. In each switching component, the squares represent the points with odd indices and the diamonds the ones with even indices. Either the squares, or the diamonds of a switching component are in a solution. In the two left cases, the feet are in a regular position, while we have an irregular position of the feet on the right. In this case, the switching components might have a different structure with turning sometimes clockwise and sometimes anticlockwise. Such switching components are said irregular but they don't occur in the regular case that we investigate.

340 Convexity can be also be encoded with clauses (Fig.13) but 3-clauses
 341 might be necessary with the difficulty that 3-SAT is no more polyno-
 342 mial but NP-complete. This obstruction is the main difficulty of the re-
 343 construction of convex lattice sets. The new approach presented in next
 344 Sec.5 requires some properties proved very recently on regular switching
 345 components [19]. We summarize these properties as follows.

346 *Property 2.* (i) With a regular position of the feet, all the switching com-
 347 ponents have a constant turning angle (they are called regular).

348 (ii) By choosing p_1 in NW (it is always possible), for any k , we have
 349 $p_{1+4k} \in \text{NW}$, $p_{2+4k} \in \text{NE}$, $p_{3+4k} \in \text{SE}$, $p_{4k} \in \text{SW}$.

350 (iii) The switching components which have two points at Euclidean
 351 distance 1 are considered as connected. This symmetric relation leads to
 352 define the connected components of switching components that we call
 353 extended switching components. The extended switching component of
 354 a switching component P is denoted \bar{P} . HV-convexity enforces all the
 355 switching component of \bar{P} to be equal to P . Either the points of the NW
 356 and SE borders of \bar{P} (represented by squares in Fig.12) belong to S , or the
 357 points of $\text{NE} \cup \text{SW}$ (represented by diamonds) are in S . For the following,
 358 we assume that the extended switching components are unstructured sets
 359 obtained by union of the points of the connected sequences.

360 (iv) The points of different extended switching components can share
 361 neither a row, nor a column.

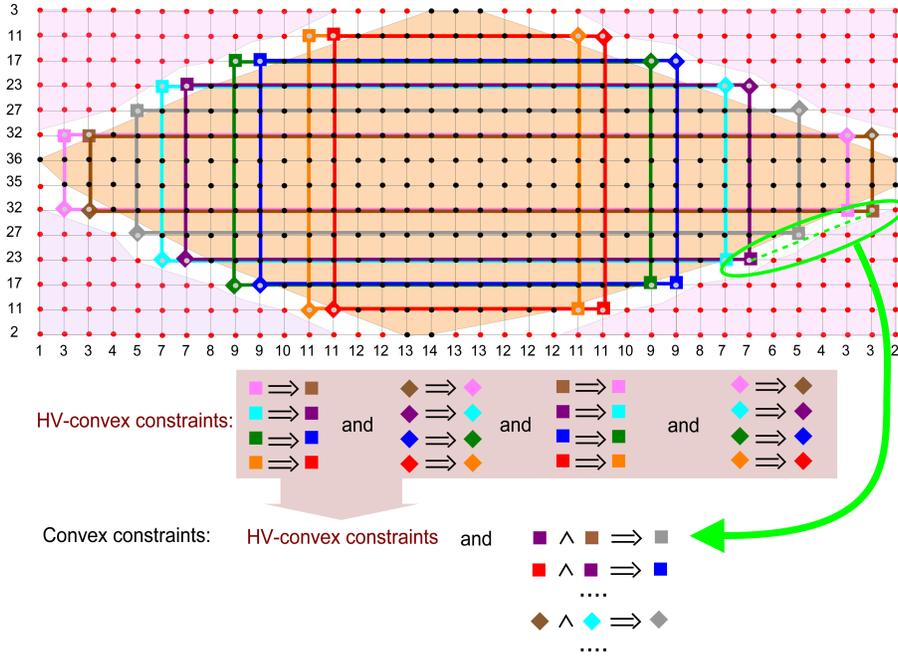


Fig.13. 2 and 3 clauses encoding convexity. HV-convexity is expressed by a conjunction of 2-clauses. Thus, the research of an HV-convex solution is reduced to a 2-SAT instance that can be solved in polynomial time. Expressing convexity in the same manner, might require 3-clauses with the difficulty that 3-SAT is NP-complete.

362 (v) The switching components can be ordered according to their rows
 363 (or equivalently according to their columns). They cannot interlace.

364 The algorithm `SwitchingComponents` decomposes the shell in switch-
 365 ing components and then merges them in extended switching components.

366 *Complexity analysis*

367 The computation is linear in the number of undetermined points. It
 368 requires no more than $O(mn)$ operations.

369 **Proposition 3.** *Algorithm `SwitchingComponents` takes $O(mn)$ time.*

370 4 Problem of Convex Aggregation

371 The three first steps of Algorithm `ConvexTomo` led to a stage where any
 372 assignment of the switching components provides the requested X-rays.

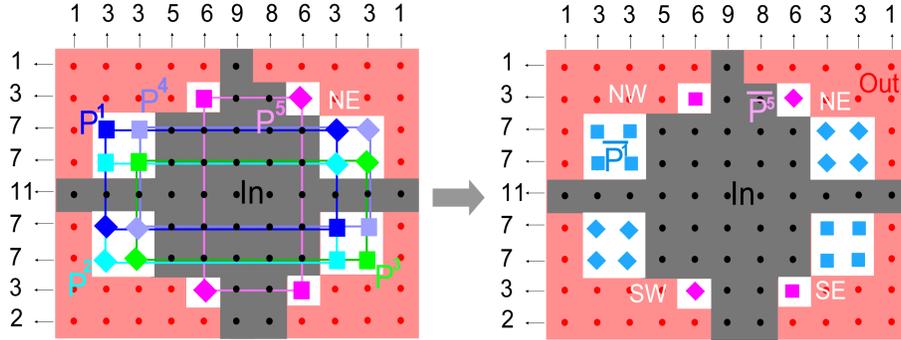


Fig. 14. Extended switching components. On the left, due to convexity, the switching components P^1, P^2, P^3, P^4 are necessarily equal while P^5 is independent. On the right, we merge P^1, P^2, P^3, P^4 in the extended switching component \bar{P}^1 and P^5 becomes \bar{P}^5 . The extended switching components are on different rows and columns. They can also be ordered according to their rows (or columns).

373 The convexity of the solution is however not yet guaranteed. The ap-
 374 proaches of the state of the art do not allow to overcome this difficulty.
 375 Therefore, we explore a new class of problems dealing with Convex Ag-
 376 gregation. We reduce it to a question of reachability in a Directed Acyclic
 377 Graph. In the following section, we use this tool to provide the final step
 378 of Algorithm ConvexTomo.

379 4.1 Problem Statement

380 We consider a new problem of Convex Aggregation of blocks along a
 381 convex set. The problem can be stated as follows:

382 *Problem 2 (ConvexAggregation(A, B^i)).*

383 **Input:** - A convex lattice set $A \subset [0..s] \times [0..t]$ with $\{(0, 0), (s, t)\} \subset A$
 384 - a finite sequence of q lattice sets that we call *blocks* $B^i \subset [0..s] \times [0..t]$
 385 disjoint from A , above the diagonal from $(0, 0)$ to (s, t) , with increasing
 386 abscissa and ordinates so that for any $i < j$, $(x, y) \in B^i, (x', y') \in B^j$, we
 387 have $x < x'$ and $y < y'$.

388 **Output:** Does there exist a convex union $A \cup (\cup_{i \in I} B^i)$ where I is a non
 389 empty subset of the indices from 1 to q (Fig.15)?

390 We could also generalize the problem with a given non convex lattice
 391 set A . In this case, the first step of an approach would be to fill the
 392 non-convex parts of A with blocks and repeat this process until finding

393 a Convex Aggregation. There is no combinatorial difficulty with such an
 394 instance. The problem becomes of interest if we start from a set which is
 395 already convex and search for a non trivial Convex Aggregation of blocks.

396 We can assume without loss of generality that the set A contains the
 397 point $(s, 0)$ and fills the triangle below the diagonal from $(0, 0)$ to (s, t) .
 398 The reason is that the blocks B^i being above the diagonal, the edges of the
 399 convex hull of A below the diagonal remain unchanged. The combinatorial
 400 problem is above.

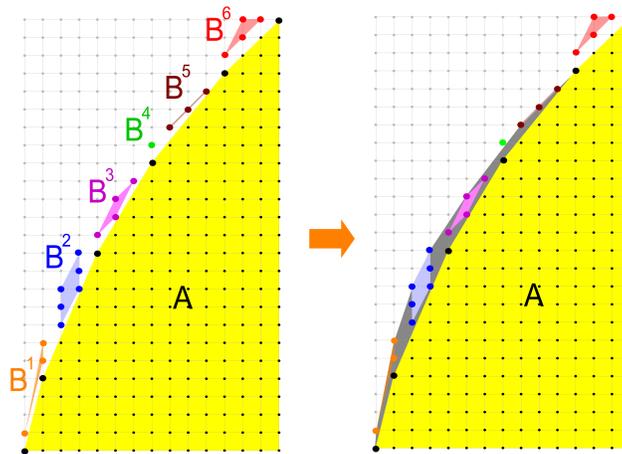


Fig. 15. An instance of $ConvexAggregation(A, B^i)$ and its solution. On the left, the input is a convex lattice set A and a sequence of lattice sets B^i . On the right the output is a non empty union of B^i so that their union with A is still convex. It means that some of the sets B^i have to be aggregated with A and some others discarded (for instance the green and the red sets in the suggested solution).

401 We could encode the choice to add or reject a block B^i by a boolean
 402 variable but as previously (Fig.13), the convexity is expressed by a con-
 403 junction of 3-clauses that 3-SAT algorithms cannot necessarily solve in
 404 polynomial time. It is the same difficulty than the one presented in the
 405 framework of $DT_C(h, v)$.

406 4.2 Rewriting $ConvexAggregation(A, B^i)$ with a Directed 407 Acyclic Graph

408 We provide a new approach by reducing the problem to the research of a
 409 path joining two subsets in a Directed Acyclic Graph. Let us consider a

410 solution $A \cup_{i \in I} B^i$ of the instance $ConvexAggregation(A, B^i)$ as drawn in
 411 Fig.15. The possible vertices of the upper hull of a solution are necessarily
 412 upper vertices of A or upper vertices of a block B^i . The set of all these
 413 upper vertices is denoted U . Then the set of the upper edges of a solution
 414 is included $U \times U$. We orient them and call them *bipoints* in order to avoid
 415 the possible ambiguities with other edges considered in the following.

416 Let us consider a bipoint $\overrightarrow{vv'}$ between two vertices $v =$
 417 (x_0, y_0) and $v' = (x_1, y_1)$ in U with $y_0 \leq y_1$. We define its left $\text{Left}(\overrightarrow{vv'})$
 418 as the set of points $p \in [0..m] \times [y_0..y_1]$, verifying $\det(\overrightarrow{vv'}, \overrightarrow{vp}) > 0$ (strict
 419 inequality) while its right is the set of points $p \in [0..m] \times [y_0..y_1]$ verifying
 420 $\det(\overrightarrow{vv'}, \overrightarrow{vp}) \leq 0$ (large inequality) (Fig.16).

421 To complete the notations, given a block B^i , we denote H^i the minimal
 422 horizontal strip containing all the points of the block ($H^i = \{(x, y) |$
 423 $\exists(x', y') \in B^i, \exists(x'', y'') \in B^i, y' \leq y \leq y''\}$). Notice that due to the
 424 increasing assumption on the blocks in $ConvexAggregation(A, B^i)$, the
 425 horizontal strips H^j are ordered and disjointed.

426 There are bipoints with vertices in U which can clearly not appear
 427 in the contour of a solution of $ConvexAggregation(A, B^i)$. We want to
 428 exclude them. We define the set V as the set of bipoints of $U \times U$ obtained
 429 by removing from $U \times U$ any bipoint $\overrightarrow{pp'} \in V \times V$ (Fig.16)

- 430 1. with $x > x'$ or with $x = x'$ and $y > y'$ (we keep only the bipoints going
 431 to the right or upward),
- 432 2. or having on its right (in $\text{Right}(\overrightarrow{pp'})$) a lattice point which is not in
 433 $A \cup_{1 \leq i \leq n} B^i$ (such point are called *outliers*),
- 434 3. or with a point of A on its left (in $\text{Left}(\overrightarrow{pp'})$),
- 435 4. or with points of the same block B^i on its right ($B^i \cap \text{Right}(\overrightarrow{pp'})$ non
 436 empty) and on its left ($B^i \cap \text{Left}(\overrightarrow{pp'})$ non empty).

437 Let us come back to a solution $A \cup_{i \in I} B^i$ of the instance $Convex$
 438 $Aggregation(A, B^i)$. The upper path connecting the origin to the point
 439 (s, t) is the concatenation of bipoints of V . Convexity is expressed by the
 440 property that two consecutive bipoints $\overrightarrow{pp'}$ and $\overrightarrow{p'p''}$ turn always clockwise.
 441 This condition is the key to build the DAG that we use to solve Problem
 442 $ConvexAggregation(A, B^i)$ (Fig.17).

443 We have however to take care to avoid inconsistent concatenation of
 444 bipoints as for instance drawn in Fig.17. A path which is passing on the
 445 left of some points of a block B^i should not be able to pass further on the
 446 right of B^i . A solution to forbid such inconsistent concatenation is first
 447 to duplicate the bipoints included in the horizontal strip H^i of a block

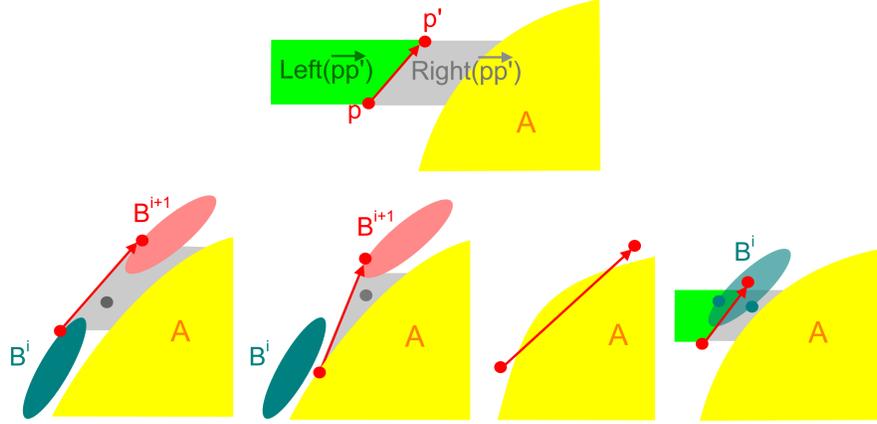


Fig. 16. The excluded bipoints. Let us precise that, although the shapes drawn in this figure appear to be continuous, they represent discrete lattice sets. Above, the sets $\text{Left}(\overrightarrow{pp'})$ and $\text{Right}(\overrightarrow{pp'})$. Below, we exclude the bipoints with an outlier on their right, with a point of A on their left, and with a pair of points of the same block on both sides. They are excluded because they are surely not contour bipoints of a solution of $\text{ConvexAggregation}(A, B^i)$.

448 B^i but also, without any point of B^i neither on its left, nor on its right.
 449 This case is illustrated in the lower part of Fig.17. Such duplication leads
 450 to define a new set of bipoints that we denote V' . We complete now the
 451 definitions:

452 **Definition 5.** Let U be the set of the upper vertices of the convex hulls
 453 of the sets A and of the blocks B^i . Let $p(x, y)$ and $p'(x', y')$ be a pair of
 454 vertices in U .

455 The set V is the set of bipoints $\overrightarrow{pp'} \in V \times V$ satisfying the four condi-
 456 tions:

457 (i) $y' > y$ or if $y' = y$, $x' > x$ where (x, y) and (x', y') are respectively the
 458 coordinates of p and p' ,

459 (ii) there is no outlier on the right of $\overrightarrow{pp'}$ ($([0..s] \times [0..t] \setminus (A \cup_{1 \leq i \leq n} B^i)) \cap$
 460 $\text{Right}(\overrightarrow{pp'}) = \emptyset$),

461 (iii) there is no point of A on the left of $\overrightarrow{pp'}$ ($A \cap \text{Left}(\overrightarrow{pp'}) = \emptyset$),

462 (iv) there are no pair of points of the same block B^i on the right and
 463 on the left of $\overrightarrow{pp'}$ (for any index i , $B^i \cap \text{Left}(\overrightarrow{pp'})$ or $B^i \cap \text{Right}(\overrightarrow{pp'})$ is
 464 empty).

465 The final set V' is obtained from V by duplicating the bipoints $\overrightarrow{pp'}$ in-

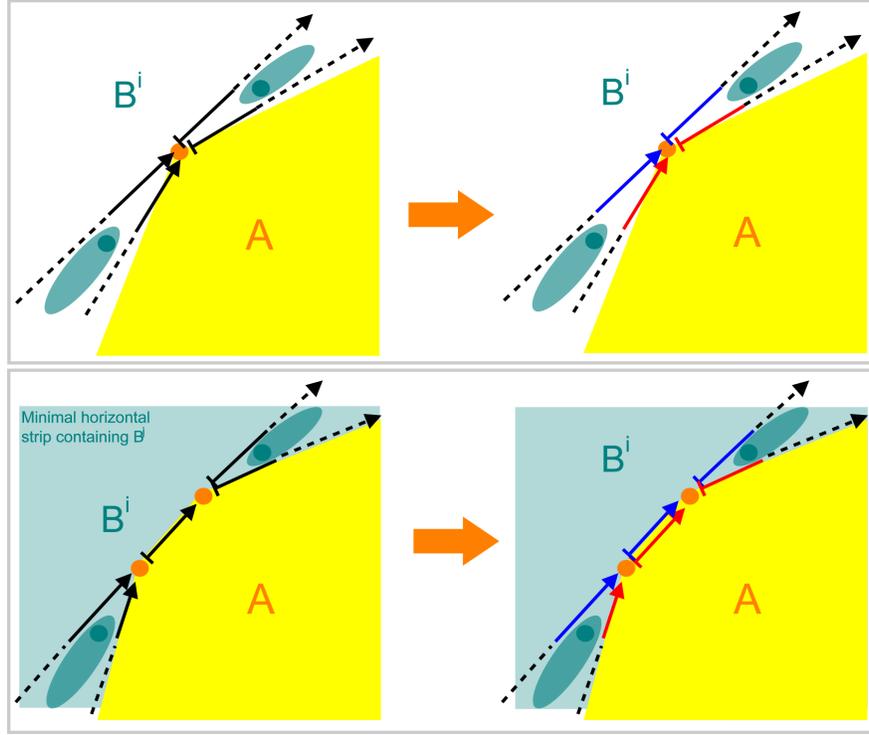


Fig. 17. Critical cases. On the left, we have two configurations which might lead to inconsistent concatenation of bipoints with points of B^i on both sides of the path. We avoid this problem by adding labels (or colors). Labeling is sufficient to solve the case above, but not the one below. The reason is that the intermediary bipoint might be used for a path with the block B^i on its left or on its right (we did no assumption on the connectivity or convexity of the blocks). To avoid inconsistency, we duplicate this bipoints and provide a copy with the two possible labels (or colors).

466 *cluded in the horizontal strip H^j of a block B^i but with no point of B^i in*
 467 *$\text{Left}(\vec{pp}') \cup \text{Right}(\vec{pp}')$. The copy is denoted \vec{pp}^* .*

468 We label now the bipoints of V' in order to avoid inconsistent paths.
 469 The labels are registered in a vector of dimension q where q is the number
 470 of blocks B^i .

471 **Definition 6.** *For any index i from 1 to q , the label of the bipoint $\vec{pp}' \in V$*
 472 *of index i is denoted $\vec{pp}'[i]$ and we have: $\vec{pp}'[i] = 1$ if $B^i \cap \text{Right}(\vec{pp}')$ is*
 473 *not empty,*
 474 *$\vec{pp}'[i] = -1$ if $B^i \cap \text{Left}(\vec{pp}')$ is not empty,*

475 $\overrightarrow{pp'}[i] = 0$ otherwise, except if the bipoint is included in the strip H^i . In
 476 this specific case, $\overrightarrow{pp'}[i] = -1$ for the original bipoint $\overrightarrow{pp'}$ and $\overrightarrow{pp'^*}[i] = 1$
 477 for its copy.

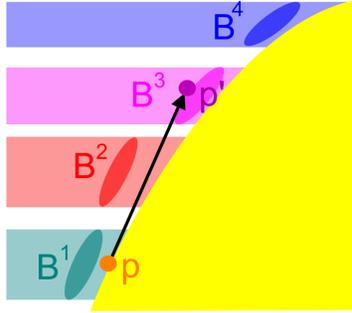


Fig. 18. Labels. The labels of the bipoint $\overrightarrow{pp'}$ are $[-1, -1, 1, 0]$ since it passes to the right of the two previous blocks B^1 and B^2 , to the left of B^3 and does not cross the horizontal strip H^4 of B^4 .

478 The labels allow to avoid the concatenation of inconsistent bipoints.
 479 Concatenation of the bipoint $\overrightarrow{pp'}$ with $\overrightarrow{p'p''}$ is only accepted if they turn
 480 clockwise and if $\overrightarrow{pp'}[i]$ and $\overrightarrow{p'p''}[i]$ are equal in the case where the point p'
 481 in the horizontal strip H^i of block B^i . As example, the concatenation of a
 482 bipoint $\overrightarrow{pp'}[i]$ of labels $[0, 1, -1, 0, 0, 0]$ with $\overrightarrow{p'p''}[i]$ of label $[0, 0, 1, -1, 1, 0]$ is
 483 excluded since the third label is different. The clockwise constraint guar-
 484 antees the convexity of the path while the label consistency guarantees
 485 that the blocks are either on the left, or on the right of a path obtained
 486 by concatenation. Formally, we build the following DAG.

487 **Definition 7.** We consider the DAG $G_{A, B^i} = (V', E)$. Its vertices are
 488 the bipoints of V' . We have an edge from the bipoint $\overrightarrow{pp'} \in V'$ to the
 489 bipoint $\overrightarrow{p'p''} \in V'$ if
 490 - the angle $(\overrightarrow{pp'}, \overrightarrow{p'p''})$ turns clockwise ($\det(\overrightarrow{pp'}, \overrightarrow{p'p''}) < 0$),
 491 - and, if p' is in the horizontal strip H^i of the block B^i , then $\overrightarrow{pp'}[i] =$
 492 $\overrightarrow{p'p''}[i]$.

493 We reduce the problem $ConvexAggregation(A, B^i)$ to the research of
 494 a path from a bipoint starting at the origin to a bipoint ending at point
 495 (s, t) .

496 *Property 3.* The set of bipoins $\overrightarrow{pp'}$ where $p = (0, 0)$ is denoted V^0 while
 497 the set of bipoins $\overrightarrow{pp'}$ where $p' = (m, n)$ is denoted V^1 .

498 The instance $ConvexAggregation(A, B^i)$ admits a solution if and only
 499 if the DAG G_{A, B^i} admits a non trivial path from the set V^0 to the set V^1
 500 (the trivial path is the convex border of A excluding all the blocks B^i .
 501 All labels are -1).

502 *Proof.* First, a solution of $ConvexAggregation(A, B^i)$ has a convex path
 503 from the origin to (s, t) . By taking its consecutive pairs of vertices as
 504 bipoins, we have a sequence of bipoins turning clockwise, starting from
 505 a bipoint with the origin in $(0, 0)$ and going to a bipoint with its end in
 506 (s, t) . If they are all in V^1 and verify the conditions of concatenation, they
 507 provide a path going from V^0 to V^1 in the DAG. Both conditions follow
 508 directly from the definitions.

509 Secondly, we consider a path going from V^0 to V^1 in the DAG G_{A, B^i} .
 510 The concatenation condition on the labels guarantees that the path passes
 511 either on the left, or on the right of any block. The orientation condition
 512 (clockwise angle) guarantees the convexity of the path, and the condition
 513 that each bipoint has no outlier on its right guarantees that the set of the
 514 points on the right of the path is the union of A with the blocks providing
 515 labels equal to $+1$.

516 It remains to notice that the trivial solution of $ConvexAggregation(A,$
 517 $B^i)$ corresponds to the excluded path with all labels equal to -1 . \square

518 According to Property 3, we can solve $ConvexAggregation(A, B^i)$ by
 519 searching for a non trivial path from V^0 to V^1 in the DAG G_{A, B^i} . Starting
 520 from the bipoins of V^0 , we use a depth-first search to determine if V^1
 521 is reached. At each vertex (bipoins) of the graph, we give the priority of
 522 exploration to the following bipoins having a first non null label equal to
 523 $+1$ or before -1 . With this strategy, the trivial path is the last one to be
 524 explored, so that the requested result is obtained before considering it.

525 *Complexity Analysis* The number of upper vertices of the sets $A \subset$
 526 $[0..s] \times [0..t]$ and of the disjointed blocks $B^i \subset [0..s] \times [0..t]$ is at most $s + s$.
 527 Therefore, the number of bipoins in V is in $O(s^2)$. With duplications, it
 528 provides the number of vertices of the DAG in $O(s^2)$. It remains to count
 529 the number of edges. Edges are no more than triangles. Their number is
 530 in $O(s^3)$.

531 About the time to create the DAG G_{A, B^i} , we need first to compute all
 532 the upper vertices of the convex hulls of A and the blocks. We can assume
 533 that the lattice points are ordered so that the total time is bounded by
 534 $O(st)$ (ordered lattice sets provide simple polygons whose convex hulls

535 can be computed in linear time [23]). It provides the set U .
 536 Then we have to check the conditions required for a bipoint of $U \times U$ to
 537 belong to V . The condition (i) is trivial while each one of the conditions
 538 (ii) (iii) and (iv) can be resolved in $O(st)$ with a naive algorithm. The
 539 labels are also computed in $O(st)$. With $O(s^2)$ possible bipoints in V , we
 540 obtain $O(s^3t)$ for the computation of V' .

541 The determination of the valid edges of the DAG (clockwise turning angle
 542 and label's consistency) can be done in constant time for each pair of
 543 bipoints (or triangle) and therefore with a total time in $O(s^3)$.

544 The depth-first search algorithm used to determine whether V^1 can
 545 be reached from V^0 takes a linear time in the number of vertices and
 546 edges i.e $O(s^2 + s^3) = O(s^3)$. By counting the time necessary to build
 547 the graph and to search for a solution, we have (without optimization) a
 548 worst-case time complexity in $O(s^3t)$. It proves the following theorem.

549 **Theorem 2.** *The problem $ConvexAggregation(A, B^i)$ in $[0..s] \times [0..t]$
 550 can be solved in $O(s^3t)$.*

551 5 ConvexTomo - Steps 4 - Convex Aggregation of the 552 Extended Switching Components

553 5.1 Reformulation of the Problem in Terms of Convex 554 Aggregation

555 At the end of the third step of the algorithm `ConvexTomo`, the set `In`
 556 and the extended switching components have been computed. It remains
 557 to find a assignment of the extended switching components providing a
 558 convex solution. According to Property 2, for each extended switching
 559 component, we have the choice between aggregating its North West and
 560 South East parts (represented with squares if the figures) or its South
 561 West and North East parts (the diamonds in the figures) to the set `In`.
 562 Thus, the blocks are the subsets of the extended switching components in
 563 each border NW, NE, SE and SW. The block B^i of the North West border
 564 is for instance $B_{NW}^i = P^i \cap NW$. Property 2 (different extended switching
 565 components have neither rows, nor columns in common) guarantees the
 566 growing property of the blocks assumed in $ConvexAggregation(A, B^i)$.

567 Without the relations induced by the switching components on each
 568 border, we would have four independent problems of Convex Aggregation
 569 $ConvexAggregation(A, B^i)$ as solved in previous section (Fig.19). The
 570 difficulty comes from the fact the four problems are not independent but
 571 deeply related: If the block B_{NW}^i is added in the North West border NW,

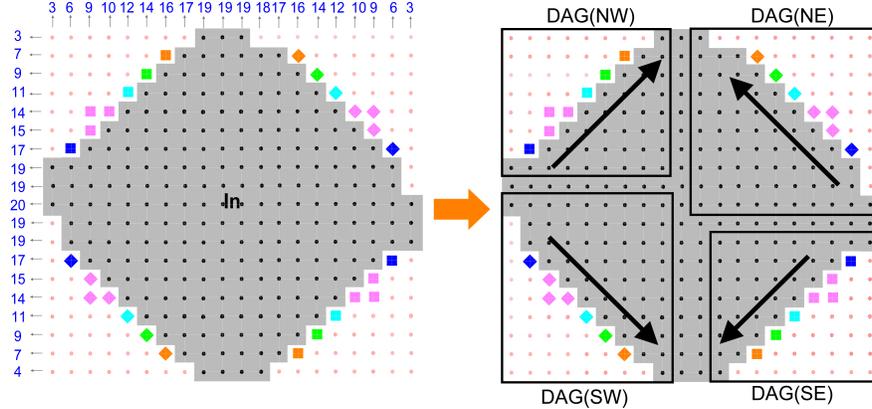


Fig. 19. Convex Aggregation of switching components and corresponding DAGs. After the third step, the algorithm `ConvexTomo` provides a decomposition of the undetermined points in extended switching components (with colored squares and diamonds). For each color, we aggregate either the squares, or the diamonds to the set `In` with the goal to provide a convex lattice set. It's the problem of Convex Aggregation that we address to conclude the paper. We start by building four DAGs G_{NW} , G_{NE} , G_{SE} and G_{SW} corresponding to the four rectangles drawn on the right. The four problems are considered upward for the North borders and downward for the South borders so that the switching components provide blocks in the same order.

572 the block B_{SE}^i of the same extended switching component in SE has also
 573 to be aggregated while the corresponding blocks B_{NE}^i and B_{SW}^i in NE and
 574 SW have to be discarded (Fig.19). We could think about exploring the sets
 575 of solutions on the four border in order determine whether a combination
 576 of solutions might be consistent but the potential exponential number
 577 of solutions makes this approach potentially non polynomial. We have
 578 to solve simultaneously the four related problems of Convex Aggregation.
 579 The principle that is we use is to build a new DAG that we call the *master*
 580 DAG coordinating the aggregation on each border.

581 5.2 Building the Master DAG

582 We place us after the third step of the algorithm `ConvexTomo` for an
 583 instance an instance $DT_{C \cap \mathcal{R}}(h, v)$ with a regular position of the feet. At
 584 this stage, the extended switching components P^i and the convex set `In`
 585 have been computed. We start by defining the four DAGs G_{NW} , G_{NE} ,
 586 G_{SE} and G_{SW} with a subset of `In` as convex set A and with the switching
 587 components $B_{NW}^i = P^i \cap NW$ as block B^i and the same for the three
 588 other borders. According to the ordering of the blocks, the fours DAGs

589 are oriented upward for the North borders and downward for the South
590 borders (Fig.19) so that the blocks B^i of the switching components appear
591 in the same order. The vertices of the DAGs are V'_{NW} , V'_{NE} , V'_{SE} and V'_{SW}
592 and their edges E_{NW} , E_{NE} , E_{SE} and E_{SW} as defined in Definition 7. We
593 call them the *slave* DAGs. Then we define the master DAG as follows:

594 **Definition 8.** *The master DAG $G = (V, E)$ has a set of vertices V con-*
595 *taining the 4-tuples of bipoints $(\overrightarrow{w_{NW}}, \overrightarrow{w_{NE}}, \overrightarrow{w_{SE}}, \overrightarrow{w_{SW}}) \in V'_{NW} \times V'_{NE} \times$*
596 *$V'_{SE} \times V'_{SW}$ verifying two conditions (letter w represents bipoints):*

- 597 – *Label’s consistency-* for any index i from 1 to q , the labels $\overrightarrow{w_{NW}}[i]$,
598 $\overrightarrow{w_{SE}}[i]$ are of same sign (with the convention that 0 is both positive
599 and negative). The labels $\overrightarrow{w_{NE}}[i]$ and $\overrightarrow{w_{SW}}[i]$ are also of same sign and
600 it is the opposite from the previous pair.
- 601 – *Label’s continuity-* If there is null label for the four bipoints ($\overrightarrow{w_{NW}}[i] =$
602 $\overrightarrow{w_{NE}}[i] = \overrightarrow{w_{SE}}[i] = \overrightarrow{w_{SW}}[i] = 0$), either all the labels with indices $i \leq i_0$
603 are null ($\forall i \leq i_0, \overrightarrow{w_{NW}}[i] = \overrightarrow{w_{NE}}[i] = \overrightarrow{w_{SE}}[i] = \overrightarrow{w_{SW}}[i] = 0$), either
604 all labels with indices $i \geq i_0$ are null ($\forall i \geq i_0, \overrightarrow{w_{NW}}[i] = \overrightarrow{w_{NE}}[i] =$
605 $\overrightarrow{w_{SE}}[i] = \overrightarrow{w_{SW}}[i] = 0$).

606 We have an edge from $(\overrightarrow{w_{NW}}, \overrightarrow{w_{NE}}, \overrightarrow{w_{SE}}, \overrightarrow{w_{SW}})$ to $(\overrightarrow{w'_{NW}}, \overrightarrow{w'_{NE}}, \overrightarrow{w'_{SE}}, \overrightarrow{w'_{SW}})$
607 if three of the bipoints are unchanged (for instance $\overrightarrow{w_{NW}} = \overrightarrow{w'_{NW}}$, $\overrightarrow{w_{NE}} =$
608 $\overrightarrow{w'_{NE}}$, $\overrightarrow{w_{SE}} = \overrightarrow{w'_{SE}}$) and the pair of different bipoints is an edge of the
609 corresponding slave DAG: for instance, $(\overrightarrow{w_{SW}}, \overrightarrow{w'_{SW}}) \in E_{SW}$.

610 The vertices of the master DAG are made of four bipoints, one on
611 each border. The edges between these vertices do not allow to progress
612 simultaneously on different border. The contour advances from the East
613 and West feet border per border until reaching or not the South and
614 North feet.

615 The condition of label’s consistency is not sufficient to guarantee the
616 coherence of the contour obtained this way. Without the condition of
617 continuity, we could imagine that the contour could progress first in the
618 North West border until reaching the North foot, before advancing on the
619 other borders. In such a case, there would be no guarantee that the choices
620 (or labels) done in the North West border would be consistent with the
621 ones done afterwards. The condition of continuity avoids such a lost of
622 information. The contour does not leave a switching component before
623 all the four paths arrived to it. It’s a keypoint to provide the following
624 equivalence.

625 **5.3 Equivalence between a Path in the Master DAG and a**
626 **Convex Aggregation of the Related Blocks**

627 We prove the following theorem:

628 **Theorem 3.** *We consider an instance $DT_{\mathcal{C} \cap \mathcal{R}}(h, v)$ with a regular posi-*
629 *tion of the feet providing after the third step of algorithm `ConvexTomo` q*
630 *switching components P^i .*

631 *The slave DAGs G_{NW} , G_{NE} , G_{SE} and G_{SW} and the master DAG $G =$*
632 *(V, E) being defined according to Definition 8, we denote V^0 the 4-tuples*
633 *of vertices $(w_{\text{NW}}, w_{\text{NE}}, w_{\text{SE}}, w_{\text{SW}})$ with all bipoints w starting from the*
634 *West and East feet $V^0 \subset V_{\text{NW}}^0 \times V_{\text{NE}}^0 \times V_{\text{SE}}^0 \times V_{\text{SW}}^0$. In the same way, V^1*
635 *is the set the valid 4-tuples of bipoints ending at the North and South feet*
636 *$V^1 \subset V_{\text{NW}}^1 \times V_{\text{NE}}^1 \times V_{\text{SE}}^1 \times V_{\text{SW}}^1$.*

637 *The instance $DT_{\mathcal{C} \cap \mathcal{R}}(h, v)$ admits a solution if and only if there is path*
638 *in the master DAG $G = (V, E)$ from V^0 to V^1 .*

639 *Proof.* We have to prove that a solution of $DT_{\mathcal{C} \cap \mathcal{R}}(h, v)$ provides a path
640 from V^0 to V^1 in the master DAG G and conversely that a path in the
641 DAG provides a convex lattice set with the prescribed X-rays.

642 For the first assertion, we have a convex solution with a contour be-
643 tween the four feet. The main point is to show that we can advance on the
644 different borders from the East and West feet to the South and North feet
645 by using only valid 4-tuples of bipoints and valid edges (valid according
646 to the definition of the master DAG). The consistency of the 4-tuples of
647 bipoints that can be extracted from the contour is straightforward but
648 not the continuity. There are however strategies which allow to guarantee
649 this property at each step: we move the bipoint with the less advanced
650 starting point with respect to the horizontal strips H^i of the switching
651 components (as illustrated in Fig.20). By induction, the continuity of 4-
652 tuples of bipoints (which is true in V^0) is guaranteed. It provides the first
653 assertion.

654 For the converse, each path in the master DAG from V^0 to V^1 provides
655 a path in the slave DAGs. Theorem 2 insures that each one of these paths
656 defines a convex contour with blocks on the left and blocks on the right.
657 It remains to prove that the contours on each border are consistent.

658 We have a first property: following the path in the DAG until a 4-tuple
659 of bipoints $(\overrightarrow{w_{\text{NW}}}, \overrightarrow{w_{\text{NE}}}, \overrightarrow{w_{\text{SE}}}, \overrightarrow{w_{\text{SW}}})$, if a label of one of the bipoints (for
660 instance $\overrightarrow{w_{\text{NW}}}[i]$) is not null, then all the labels which have been visited
661 and determined by the path after the index i have non null values for at
662 least one of the bipoints of the 4-tuple. This property denoted (i) can be

663 established by induction. It is obvious for the initial 4-tuple of bipoints
664 (due to the consistency condition) and it remains true by induction due
665 to the continuity condition of the 4-tuple of bipoints.
666 Let us show now that the contour gradually built from the path in the
667 DAG is consistent. We prove it again by induction. With the consistency
668 condition, it is true for the initial 4-tuple of bipoints. Then we assume
669 that the contour generated by the path is consistent until the vertex
670 $(\overrightarrow{w_{NW}}, \overrightarrow{w_{NE}}, \overrightarrow{w_{SE}}, \overrightarrow{w_{SW}})$. According to the set of edges E , we can assume
671 w.l.g that the next vertex is $(\overrightarrow{w'_{NW}}, \overrightarrow{w'_{NE}}, \overrightarrow{w'_{SE}}, \overrightarrow{w'_{SW}})$. If the new bipoint
672 $\overrightarrow{w'_{SW}}$ is inconsistent for the switching component of index j , it means that
673 the index already had in the path a non null label. Due to property (i),
674 this label is also non null value in the current 4-tuple. As there was no
675 inconsistency before, it follows that $\overrightarrow{w'_{SW}}$ has an inconsistent label with
676 one of the three bipoints $\overrightarrow{w_{NW}}, \overrightarrow{w_{NE}}, \overrightarrow{w_{SE}}$. It's in contradiction with
677 the consistency of the 4-tuple $(\overrightarrow{w_{NW}}, \overrightarrow{w_{NE}}, \overrightarrow{w_{SE}}, \overrightarrow{w_{SW}})$. It proves that the
678 contour built from the path in the master DAG is a valid solution and
679 therefore provides a convex lattice set with the prescribed X-rays.

680 5.4 Algorithm ConvexAggregation

681 The fourth step of Algorithm ConvexTomo solves the problem of Convex
682 Aggregation issued from the switching components through the research
683 of a path in the master DAG G . The algorithm ConvexAggregation starts
684 by computing the four slave DAGs. Then it considers all the 4-tuples of
685 bipoints in $V_{NW} \times V_{NE} \times V_{SE} \times V_{SW}$ and checks their validity. Afterward,
686 it computes the edges of the master DAG by checking the validity and
687 the turning angle of the pairs of vertices.

688 The final part of the algorithm is the research of a path starting
689 from V^0 and reaching V^1 . It can be done with depth-first or breadth-first
690 search.

691 *Complexity Analysis*

692 We start with the computation of the slave DAGs. According to the
693 previous analysis, by replacing (s, t) by the upper bound (m, n) , it can
694 be done in $O(m^3n)$.

695 For the master DAG, as we have at most m^2 bipoints in each slave DAG,
696 the number of 4-tuples of bipoints in V is bounded by $O(m^8)$. For each
697 one of them, testing their validity requires $O(q)$. The computation of the
698 vertices of the master DAG takes $O(qm^8)$.

699 For going from the vertex $(\overrightarrow{p_{NW}p'_{NW}}, \overrightarrow{p_{NW}p'_{NW}}, \overrightarrow{p_{NW}p'_{NW}}, \overrightarrow{p_{NW}p'_{NW}})$ to one
700 of its successors, we have to introduce a new upper point, for instance

<p>Data: m, n, In and the extended switching components $(P^i)_{1 \leq i \leq q}$</p> <p>Result: An assignment of the extended switching component providing a convex aggregated set</p> <pre> 1 /* Compute the slave DAGs $G_{\text{NW}}, G_{\text{NE}}, G_{\text{SE}}$ and G_{SW} with labels */ 2 $(V_{\text{NW}}, E_{\text{NW}}) \leftarrow \text{DAG}(\text{In}, (P^i)_{1 \leq i \leq q} \cap \text{NW});$ 3 $(V_{\text{NE}}, E_{\text{NE}}) \leftarrow \text{DAG}(\text{In}, (P^i)_{1 \leq i \leq q} \cap \text{NE});$ 4 $(V_{\text{SE}}, E_{\text{SE}}) \leftarrow \text{DAG}(\text{In}, (P^i)_{1 \leq i \leq q} \cap \text{SE});$ 5 $(V_{\text{SW}}, E_{\text{SW}}) \leftarrow \text{DAG}(\text{In}, (P^i)_{1 \leq i \leq q} \cap \text{SW});$ 6 /* Compute the master DAG */ 7 $(V, E) \leftarrow \text{masterDAG}((V_{\text{NW}}, V_{\text{NE}}, V_{\text{SE}}, V_{\text{SW}}, E_{\text{NW}}, E_{\text{NE}}, E_{\text{SE}}, E_{\text{SW}}, \text{labels});$ 8 $V^0 \leftarrow V \cap V_{\text{NW}}^0 \times V_{\text{NE}}^0 \times V_{\text{SE}}^0 \times V_{\text{SW}}^0;$ 9 $V^1 \leftarrow V \cap V_{\text{NW}}^1 \times V_{\text{NE}}^1 \times V_{\text{SE}}^1 \times V_{\text{SW}}^1;$ 10 /* Search for a path from V^0 to V^1 */ 11 $\text{path} \leftarrow \text{searchPath}(V, E, V^0, V^1);$ 12 $\text{readLabels}(\text{path})$ </pre>

Algorithm 3: ConvexAggregation($m, n, \text{In}, (P^i)_{1 \leq i \leq q}$)

701 $p''_{\text{NW}} \in \text{NW}$, and then checking if $\overrightarrow{(p'_{\text{NW}}p''_{\text{NW}}), \overrightarrow{(p_{\text{NW}}p'_{\text{NW}}), \overrightarrow{(p_{\text{NW}}p'_{\text{NW}}), \overrightarrow{(p_{\text{NW}}p'_{\text{NW}})$
702 is a valid vertex. It checks also whether the pair $(\overrightarrow{(p_{\text{NW}}p'_{\text{NW}}), \overrightarrow{(p'_{\text{NW}}p''_{\text{NW}})})$ is
703 in E_{NW} .

704 The number of edges is bounded by the number of vertices $O(m^8)$ times
705 the number of possible new points $2m$. It makes a number of edges in
706 $O(m^9)$. The time necessary to check their validity is constant with a suit-
707 able data structure containing the edges of the slave DAGs.

708 With the number of vertices and edges, the breadth-first search can be
709 done in $O(m^9)$. As q is lower than m , it provides a total time of compu-
710 tation in $O(m^9)$. It proves the following theorem.

711 **Theorem 4.** *Algorithm ConvexAggregation($m, n, \text{In}, (P^i)_{1 \leq i \leq q}$) solves*
712 *the problem of Convex Aggregation of the blocks issued from the switching*
713 *components in $[0..m] \times [0..n]$ with a worst case time complexity in $O(m^9)$.*

714 Notice that if $n < m$, with a rotation of $\frac{\pi}{2}$, it could be as well $O(n^9)$.

715 5.5 Proof of Theorem 1

716 Theorem 4 provides the time complexity in $O(m^9)$ of the algorithm **Convex**
717 **Aggregation**. According to the propositions 1, 2, 3, the three first steps
718 (**Initialization**, **FillTomo**, **SwitchingComponents**) require respectively
719 $O(mn)$, $O(m^2n^2)$ and $O(mn)$ operations. Since there are at most m^2n^2
720 regular positions of the feet to explore, the overall time complexity of
721 Algorithm **FillTomo** is $O(m^4n^4 + m^{11}n^2)$, as stated in Theorem 1.

722 6 Conclusion

723 The problem of the reconstruction of convex lattice sets with prescribed
724 horizontal and vertical X-rays is the most challenging problem of com-
725 plexity in the framework of Discrete Tomography. The question is to
726 determine whether we can handle with ambiguous X-rays and convexity
727 constraints in polynomial time.

728 The paper is a first step in the direction of a better understanding
729 of the relations between ambiguities and convexity. The ambiguities are
730 expressed by the switching components. After the three first steps of
731 the classical algorithm, they lead to reduce the problem to a question
732 of Convex Aggregation. The new idea that we present is that discrete
733 convexity is a local constraint. It can be reduced to the property that
734 consecutive edges have a clockwise turning angle. Such a constraint can
735 be handled with a DAG or more generally with Dynamic Programming.
736 The crucial information is the last reconstructed edge so that we can try
737 to go further. These two elements (switching components and encoding
738 of the last edge) allowed us to develop the polynomial time algorithm
739 `ConvexTomo` in the case of a regular position of the feet. It is a good
740 news but why not a more general result ? What happens in the case of
741 an irregular position of the feet which is different than for the regular
742 positions ? The answers can be reduced to only one word: "structure".

743 The structure of the regular switching components can be simplified
744 by merging them into extended switching components. According to re-
745 cent combinatorial results (Property 2 [19]), extended switching compo-
746 nents can be ordered (their rows and columns are ordered) so that the
747 consecutive blocks to consider in a Convex Aggregation framework are
748 also well ordered. This order (or increasing property) is the only assump-
749 tion that we did on the blocks in problem *ConvexAggregation*(A, B^i).
750 Without it, there is no doubt that Convex Aggregation would become
751 NP-hard. Structural properties are necessary to provide polynomial time
752 algorithms of Convex Aggregation. This remark leads to the following
753 question: with irregular positions of the feet, do switching components
754 (Fig.12) have enough structures to be handled with polynomial time al-
755 gorithms ? It's an open question whose answer could as well lead to a
756 polynomial time algorithm as to a result of NP-completeness.

757 References

- 758 1. E. Barcucci, P. Dulio, A. Frosini, and S. Rinaldi. Ambiguity results in the char-
759 acterization of hv-convex polyominoes from projections. In *Discrete Geometry for*

- 760 *Computer Imagery - 20th IAPR International Conference, DGCI 2017, Vienna,*
761 *Austria, September 19-21, 2017, Proceedings*, pages 147–158, 2017.
- 762 2. E. Barcucci, A. D. Lungo, M. Nivat, and R. Pinzani. Reconstructing convex poly-
763 ominoes from horizontal and vertical projections. *Theor. Comput. Sci.*, 155(2):321–
764 347, 1996.
- 765 3. K. Batenburg, S. Bals, J. Sijbers, C. Kübel, P. Midgley, J. Hernandez, U. Kaiser,
766 E. Encina, E. Coronado, and G. van Tendeloo. 3d imaging of nanomaterials by
767 discrete tomography. *Ultramicroscopy*, 109:730–40, 2009. 43.01.05; LK 01.
- 768 4. R. N. Bracewell. Strip integration in radio astronomy. *Australian Journal of*
769 *Physics*, 9(2):198–217, 1956.
- 770 5. G. S. Brodal and R. Jacob. Dynamic planar convex hull. In *43rd Symposium on*
771 *Foundations of Computer Science (FOCS 2002), 16-19 November 2002, Vancouver,*
772 *BC, Canada, Proceedings*, pages 617–626, 2002.
- 773 6. S. Brunetti and A. Daurat. Reconstruction of convex lattice sets from tomographic
774 projections in quartic time. *Theoretical Computer Science*, 406(1):55 – 62, 2008.
775 Discrete Tomography and Digital Geometry: In memory of Attila Kuba.
- 776 7. S. Brunetti, A. Daurat, and A. Kuba. Fast filling operations used in the recon-
777 struction of convex lattice sets. In A. Kuba, L. G. Nyúl, and K. Palágyi, editors,
778 *Discrete Geometry for Computer Imagery*, pages 98–109, Berlin, Heidelberg, 2006.
779 Springer Berlin Heidelberg.
- 780 8. J. M. Carazo, C. O. Sorzano, E. Rietzel, R. Schröder, and R. Marabini. *Discrete*
781 *Tomography in Electron Microscopy*, pages 405–416. Birkhäuser Boston, Boston,
782 MA, 1999.
- 783 9. P. Dulio, A. Frosini, S. Rinaldi, L. Tarsissi, and L. Vuillon. First steps in the
784 algorithmic reconstruction of digital convex sets. In *Combinatorics on Words -*
785 *11th International Conference, WORDS 2017, Montréal, QC, Canada, September*
786 *11-15, 2017, Proceedings*, pages 164–176, 2017.
- 787 10. C. Dürr, F. Guíñez, and M. Matamala. Reconstructing 3-colored grids from hori-
788 zontal and vertical projections is np-hard. In *Algorithms - ESA 2009, 17th Annual*
789 *European Symposium, Copenhagen, Denmark, September 7-9, 2009. Proceedings*,
790 pages 776–787, 2009.
- 791 11. S. Even, A. Itai, and A. Shamir. On the complexity of time table and multi-
792 commodity flow problems. In *Proceedings of the 16th Annual Symposium on Foun-*
793 *dations of Computer Science, SFCS '75*, pages 184–193, Washington, DC, USA,
794 1975. IEEE Computer Society.
- 795 12. D. R. Ford and D. R. Fulkerson. Maximal flow through a networks. *Canadian*
796 *Journal of Mathematics*, 8:399–404, 1956.
- 797 13. D. Gale. A theorem on flows in networks. *Pacific J. Math.*, 7:1073–1082, 1957.
- 798 14. R. Gardner and P. Gritzmann. Determination of finite sets by x-rays. *Transactions*
799 *of the American Mathematical Society*, 349(6):2271—2295, 1997.
- 800 15. R. J. Gardner. *Geometric Tomography*. Encyclopedia of Mathematics and its
801 Applications. Cambridge University Press, 1995.
- 802 16. R. J. Gardner, P. Gritzmann, and D. Prangenberg. On the computational com-
803 plexity of reconstructing lattice sets from their x-rays. *Discrete Mathematics*,
804 202(1-3):45–71, 1999.
- 805 17. R. J. Gardner, P. Gritzmann, and D. Prangenberg. On the computational complex-
806 ity of determining polyatomic structures by x-rays. *Theor. Comput. Sci.*, 233(1-
807 2):91–106, 2000.
- 808 18. Y. Gérard. About the complexity of timetables and 3-dimensional discrete to-
809 mography: A short proof of np-hardness. In *Combinatorial Image Analysis, 13th*

- 810 *International Workshop, IWCIA 2009, Playa del Carmen, Mexico, November 24-*
811 *27, 2009. Proceedings*, pages 289–301, 2009.
- 812 19. Y. Gérard. Regular switching components. *preprint Hal-01832674*, 2018.
- 813 20. G. T. Herman and A. Kuba. *Discrete Tomography - Foundations, Algorithms and*
814 *Applications*. Birkhauser, 1999.
- 815 21. G. T. Herman and A. Kuba. *Advances in Discrete Tomography and Its Applica-*
816 *tions*. Birkhauser, 2007.
- 817 22. R. W. Irving and M. Jerrum. Three-dimensional statistical data security problems.
818 *SIAM J. Comput.*, 23(1):170–184, 1994.
- 819 23. D. McCallum and D. Avis. A linear algorithm for finding the convex hull of a
820 simple polygon. *Information Processing Letters*, 9(5):201–206, 1979.
- 821 24. J. Radon. Über die bestimmung von funktionen durch ihre integralwerte langs
822 gewisser mannigfaltigkeiten. *Berichte über die Verhandlungen der Königlich-*
823 *Sächsischen Akademie der Wissenschaften zu Leipzig, Mathematisch-Physische*
824 *Klasse*, 69:262–277, 1917.
- 825 25. H. Ryser. Combinatorial properties of matrices of zeros and ones. *Can. J. Math.*,
826 9:371–377, 1957.
- 827 26. S. Van Aert, K. J. Batenburg, M. D. Rossell, R. Erni, and G. Van Tendeloo.
828 Three-dimensional atomic imaging of crystalline nanoparticles. *Nature*, 470:374–
829 377, 2011.
- 830 27. G. J. Woeginger. The reconstruction of polyominoes from their orthogonal projec-
831 tions. *Information Processing Letters*, 77(5):225 – 229, 2001.

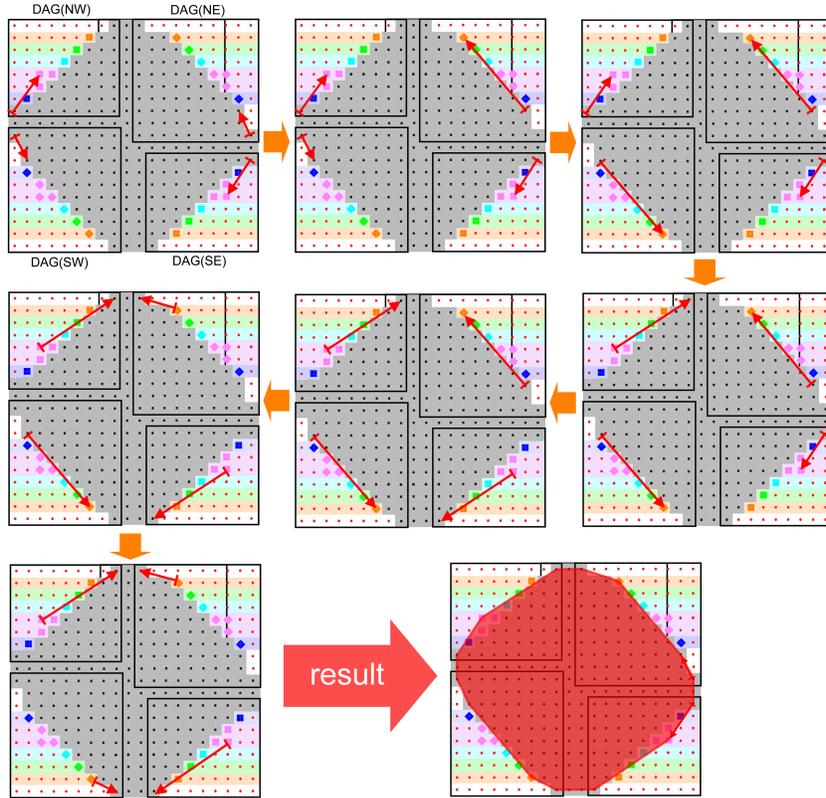


Fig.20. A path in the master DAG and the corresponding solution of $DT_{C \cap \mathcal{R}}(h, v)$. We show why any solution of $DT_{C \cap \mathcal{R}}(h, v)$ can be obtained through a path from V^0 to V^1 in the master DAG. The path is obtained with the strategy to advance the bipoint $\overrightarrow{vv'}$ with the less advanced point v regarding the horizontal strips H^i of the switching components. With this strategy, the continuity of the labels is guaranteed by induction.