Available online at www.sciencedirect.com
ScienceDirect
Theoretical Computer Science 370 (2007) 201–217
Theoretical Computer Science
www.elsevier.com/locate/tcs

Binary matrices under the microscope: A tomographical problem

Andrea Frosini[a,*], Maurice Nivat[b]

[a] *Dipartimento di Scienze Matematiche ed Informatiche "Roberto Magari", Università degli Studi di Siena, Pian dei Mantellini 44, 53100 Siena, Italy*

[b] *Laboratoire d'Informatique, Algorithmique, Fondements et Applications (LIAFA), Université Denis Diderot 2, place Jussieu, 75251 Paris 5 Cedex 05, France*

Received 31 August 2005; received in revised form 28 October 2006; accepted 31 October 2006

Communicated by Giorgio Ausiello

Abstract

A binary matrix can be scanned by moving a fixed rectangular window (sub-matrix) across it, rather like examining it closely under a microscope. With each viewing, a convenient measurement is the number of 1s visible in the window, which might be thought of as the *luminosity* of the window. The *rectangular scan* of the binary matrix is then the collection of these luminosities presented in matrix form. We show that, at least in the technical case of a *smooth* $m \times n$ binary matrix, it can be reconstructed from its rectangular scan in polynomial time in the parameters m and n, where the degree of the polynomial depends on the size of the window of inspection. For an arbitrary binary matrix, we then extend this result by determining the entries in its rectangular scan that preclude the smoothness of the matrix.

Keywords: Discrete tomography; Reconstruction algorithm; Computational complexity; Projection; Rectangular scan

1. Introduction and definitions

The aim of *discrete tomography* is the retrieval of geometrical information about a physical structure, regarded as a finite set of points in the integer square lattice $\mathbb{Z} \times \mathbb{Z}$, from measurements, generically known as *projections*, of the number of atoms in the structure that lie on lines with fixed scopes (see [4] for a survey). A common simplification is to represent a finite physical structure as a binary matrix, where an entry is 1 or 0 according to whether an atom is present or absent in the structure at the corresponding point of the lattice. The challenge is then to reconstruct key features of the structure from a small number of scans of projections [6], eventually using some *a priori* information such as convexity [1,2], and periodicity [3].

Our interest here, following [5], is to probe the structure, not with lines of fixed scope, but with their natural two dimensional analogue, rectangles of fixed scope, much as we might examine a specimen under a microscope or magnifying glass. For each position of our rectangular probe, we count the number of visible atoms or, in the simplified binary matrix version of the problem, the number of 1s in the prescribed rectangular window, which we term

* Corresponding author. Tel.: +39 0577233705; fax: +39 0577233730.
E-mail addresses: frosini@unisi.it (A. Frosini), Maurice.Nivat@liafa.jussieu.fr (M. Nivat).

0304-3975/$ - see front matter
doi:10.1016/j.tcs.2006.10.030

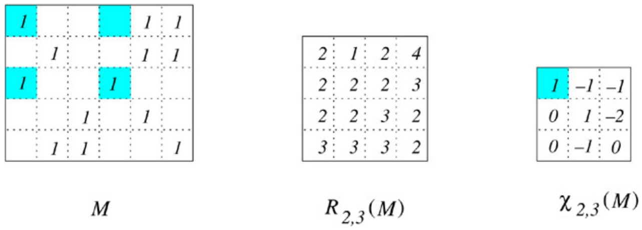


Fig. 1. The matrix M and its corresponding matrices $R_{2,3}(M)$ and $\chi_{2,3}(M)$. The highlighted elements of M are used to compute the highlighted one in $\chi_{2,3}(M)$.

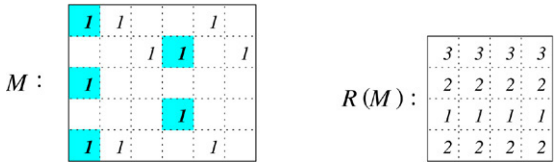


Fig. 2. A non-invariant matrix M whose (2, 3)-rectangular scan has constant rows.

its *luminosity*. In the matrix version of the problem, these measurements can themselves be organized in matrix form, called the *rectangular scan* of the original matrix. Our first objective is then to furnish a strategy to reconstruct the original matrix from its rectangular scan. In the following, we will address this problem as *Reconstruction*(A, p, q), where A is the rectangular scan, and p and q are the dimensions of the rectangular windows. As we also note, our investigation is closely related to results on tiling by translation in the integer square lattice discussed in [5].

To be more precise, let M be a $m \times n$ integer matrix, and, for fixed p and q, with $1 \leq p \leq m$, $1 \leq q \leq n$, consider a $p \times q$ window $R_{p,q}$ allowing us to view the intersection of any p consecutive rows and q consecutive columns of M. Then, the number $R_{p,q}(M)[i, j]$ on view when the top left hand corner of $R_{p,q}$ is positioned over the (i, j)-entry, $M[i, j]$, of M, is given by summing all the entries on view:

$$R_{p,q}(M)[i, j] = \sum_{r=0}^{p-1}\sum_{c=0}^{q-1} M[i + r, j + c], \quad 1 \leq i \leq m - p + 1,\ 1 \leq j \leq n - q + 1.$$

Thus, we obtain an $(m - p + 1) \times (n - q + 1)$ matrix $R_{p,q}(M)$ called the (p, q)-*rectangular scan* of M; when p and q are understood, we write $R(M) = R_{p,q}(M)$, and speak more simply of the *rectangular scan*. (This terminology is a slight departure from that found in [5].) In the special case when $R(M)$ has all entries equal, say k, we say that the matrix M is *homogeneous* of *degree* k, or simply k-homogeneous.

Furthermore, we define an $(m - p) \times (n - q)$ matrix $\chi_{p,q}(M)$ by setting, for $1 \leq i \leq m - p,\ 1 \leq j \leq n - q$:

$$\chi_{p,q}(M)[i, j] = M[i, j] + M[i + p, j + q] - M[i + p, j] - M[i, j + q].$$

As usual, when p and q can be understood without ambiguity, we suppress them as subscripts. In the event that the matrix $\chi(M)$ is a zero matrix, the matrix M is said to be *smooth*. Notice that the homogeneous matrices are *properly* included in the smooth matrices, as shown by the matrix M of Fig. 2, which is smooth, and non-homogeneous.

Simplifying the rules of the game, throughout the paper we will consider the matrix M (representing a physical structure) as a *binary one*; under this assumption, the rectangular scan $R(M)$ turns out to be a positive matrix whose values are in the set $\{0, \ldots, p \cdot q\}$, and the matrix $\chi(M)$ turns out to have values in the integer interval $\{-2, \ldots, 2\}$ (see Fig. 1).

We conclude this introductory section with three observations which are direct consequences of the given definitions and to which we shall have frequent recourse in what follows. Since their proofs are a matter of simple computations, they are omitted.

Lemma 1. *If M_1 and M_2 are two $m \times n$ binary matrices, then*

$$R(M_1 + M_2) = R(M_1) + R(M_2) \quad \textit{and} \quad \chi(M_1 + M_2) = \chi(M_1) + \chi(M_2).$$

Lemma 2. *If M is a binary matrix, then*

$$\chi_{1,1}(R_{p,q}(M)) = \chi_{p,q}(M).$$

Thus the rectangular scan $R(M)$ of a binary matrix M already contains sufficient information to compute $\chi(M)$ and so to decide whether M is smooth. Notice that, with a certain terminological inexactitude, we can also say, in the case where M is smooth, that $R(M)$ is smooth (more precisely, $R(M)$ is (1, 1)-smooth, while M itself is (p, q)-smooth, as our more careful statement of Lemma 2 makes clear).

An appeal to symmetry and induction yields the following generalization of [5, Lemma 2.2].

Lemma 3. *If M is a smooth matrix then, for any integers α and β such that $1 \le i + \alpha p \le m$ and $1 \le j + \beta q \le n$,*

$$M[i, j] + M[i + \alpha p, j + \beta q] = M[i + \alpha p, j] + M[i, j + \beta q].$$

Finally, we say that an entry $M[i, j]$ of the matrix M is (p, q)-*invariant* if, for any integer α such that $1 \le i + \alpha p \le m$ and $1 \le j + \alpha q \le n$,

$$M[i + \alpha p, j + \alpha q] = M[i, j].$$

If all the entries of M are (p, q)-invariant, then M is said to be (p, q)-invariant.

2. A decomposition theorem for binary smooth matrices

In this section we extend the studies about homogeneous matrices started in [5] to the class of smooth matrices: first we furnish a series of simple results which link smoothness and invariance, then we proceed to their reconstruction along a path leading through a decomposition theorem for smooth matrices.

Lemma 4. *If M is a smooth matrix, then each of its elements is $(p, 0)$-invariant or $(0, q)$-invariant.*

Proof. Since M is smooth, for each $1 \le i \le m - p$ and $1 \le j \le n - q$, it holds

$$M[i, j] + M[i + p, j + q] = M[i + p, j] + M[i, j + q].$$

Let us consider the following three possibilities for the element $M[i, j]$:

(i) $M[i, j] \ne M[i + p, j]$: by Lemma 3, for $\alpha = 1$ and for all $\beta \in \mathbb{Z}$ such that $1 \le j + \beta q \le n$, it holds $M[i, j + \beta q] = M[i, j]$ and $M[i + p, j + \beta q] = M[i + p, j]$, so $M[i, j]$ is $(0, q)$-invariant.

(ii) $M[i, j] \ne M[i, j + q]$: by reasoning similarly to (i), we obtain that $M[i, j]$ is $(p, 0)$-invariant.

(iii) $M[i, j] = M[i, j + q] = M[i + p, j]$: if there exists $\alpha_0 \in \mathbb{Z}$ such that $M[i + \alpha_0 p, j] \ne M[i, j]$, again reasoning as in (i), we obtain that $M[i, j]$ is $(0, q)$-invariant.

On the other hand, if for all $1 \le i + \alpha p \le m$ it holds that $M[i + \alpha p, j] = M[i, j]$, then $M[i, j]$ is $(p, 0)$-invariant.

Finally, if $m - p + 1 \le i \le m$ and $n - q + 1 \le j \le n$, a similar reasoning leads again to the thesis. □

The reader can check that each entry of the matrix M in Fig. 2 which is smooth with respect to a 2×3 window, is (2, 0)-invariant (the highlighted ones) or (0, 3)-invariant. A first decomposition result follows:

Theorem 5. *A matrix M is smooth if and only if it can be obtained as the sum of a $(p, 0)$-invariant matrix M_1 and a $(0, q)$-invariant matrix M_2 such that they do not have two entries 1 in the same position.*

Proof. ($\Rightarrow$) Let M_1 and M_2 contain the $(p, 0)$-invariant and the $(0, q)$-invariant elements of M, respectively. By Lemma 4, the thesis is achieved.

($\Leftarrow$) Since M_1 is $(p, 0)$-invariant, then for each $1 \le i \le m - p$, $1 \le j \le n - q$ it holds

$$\begin{aligned}\chi(M_1)[i, j] &= M_1[i, j] + M_1[i + p, j + q] - M_1[i + p, j] - M_1[i, j + q]\\ &= M_1[i, j] + M_1[i, j + q] - M_1[i, j] - M_1[i, j + q] = 0\end{aligned}$$

So, by definition, M_1 is smooth. The same result holds for M_2 and, by Lemma 1, for $M = M_1 + M_2$. □

We can go further on by reformulating this last theorem in terms of the rectangular scans of the matrices M_1 and M_2:

Lemma 6. *The following statements hold:*

(i) *if M is $(0, q)$-invariant, then $R(M)$ has constant rows;*
(ii) *if M is $(p, 0)$-invariant, then $R(M)$ has constant columns.*

Proof. (i) For each $1 \leq i \leq m - p + 1$ and $1 \leq j \leq n - q$, we prove that $R(M)[i, j] = R(M)[i, j + 1]$:

$$R(M)[i, j+1] = \sum_{r=0}^{p-1}\sum_{c=0}^{q-1} M[i+r, j+1+c]$$
$$= \sum_{r=0}^{p-1}\sum_{c=1}^{q-1} M[i+r, j+c] + \sum_{r=0}^{p-1} M[i+r, j+q]$$

since M is $(0, q)$-invariant

$$= \sum_{r=0}^{p-1}\sum_{c=1}^{q-1} M[i+r, j+c] + \sum_{r=0}^{p-1} M[i+r, j] = R(M)[i, j].$$

(ii) The proof is similar to (i). □

After observing that each matrix having constant rows or columns is smooth, a direct consequence of Theorem 5 and Lemma 6 is the following:

Theorem 7. *A binary matrix M is smooth if and only if $R(M)$ can be decomposed into two matrices R_r and R_c having constant rows and columns, respectively.*

Fig. 2 shows that the converse of the two statements of Lemma 6 does not hold in general. However, we can prove the following weaker version:

Proposition 8. *Let M be a binary matrix. The following statements hold:*

(i) *if $R(M)$ has constant columns, then there exists a $(p, 0)$-invariant matrix M_1 such that $R(M) = R(M_1)$;*
(ii) *if $R(M)$ has constant rows, then there exists a $(0, q)$-invariant matrix M_2 such that $R(M) = R(M_2)$.*

Proof. (i) We define the matrix M_1 as follows: the first p rows of M_1 are equal to those of M, and the other entries of M_1 are set according to the desired $(p, 0)$-invariance. It is easy to verify that $R(M_1) = R(M)$.

(ii) A definition of M_2 similar to that in (i) for M_1 can be easily given. □

2.1. Solving Reconstruction(A, p, q) for smooth matrices

A first approach to the general reconstruction problem consists in the definition of the following algorithm which suits only for a binary smooth matrix whose (p, q)-rectangular scan has constant rows:

RECCONSTROWS(A, p, q)

Input: an integer matrix A of dimension $m' \times n'$, having constant rows, and two integers p and q.
Output: a $(0, q)$-invariant matrix M, of dimension $m \times n$, where $m = m' + p - 1$ and $n = n' + q - 1$, having A as (p, q)-rectangular scan, if it exists, else return FAILURE.
Procedure:
Step 1: create the $m \times n$ matrix M and the vector $PEnt$ (storing the Partial number of Entries 1 in each row of M) of dimension m, to support the computation. Initialize the entries both of M and $PEnt$ to 0.
For each row $1 \leq i \leq m' - 1$,
Step 1.1: **if** $A[i, 1] \leq A[i + 1, 1]$ **then**

- $M[i + p, 1] = \cdots = M[i + p, A[i + 1, 1] - A[i, 1] + PEnt[i]] = 1$;
- $PEnt[i + p] = A[i + 1, 1] - A[i, 1] + PEnt[i]$.

If $PEnt[i+p] > q$ **then** FAILURE.

Step 1.2: **if** $A[i,1] > A[i+1,1]$ and $PEnt[i] \geq A[i,1] - A[i+1,1]$ **then**

- $M[i+p,1] = \cdots = M[i+p, A[i+1,1] - A[i,1] + PEnt[i]] = 1$;
- $PEnt[i+p] = A[i+1,1] - A[i,1] + PEnt[i]$.

Step 1.3: **if** $A[i,1] > A[i+1,1]$ and $PEnt[i] < A[i,1] - A[i+1,1]$ **then**

- $k = A[i,1] - A[i+1,1] - PEnt[i]$;
- for each $i' \leq i$, $i' = (i) mod_p$
  - $M[i', PEnt[i']+1] = \cdots = M[i', PEnt[i']+k] = 1$;
  - $PEnt[i'] = PEnt[i'] + k$;
  - **if** $PEnt[i'] > q$ **then** FAILURE.

Step 2: let $k = A[1,1] - PEnt[1] - \cdots - PEnt[p]$.

For each $1 \leq k' \leq k$, search one of the upper leftmost $p \times q$ positions of M, say (i, j), such that, for each $i' = (i) mod_p$, $1 \leq i' \leq m$, it holds $M[i', j] = 0$.

If such a position does not exist **then** FAILURE,

else set all the entries $M[i', j]$ to the value 1, and increase k' by one.

Step 3: complete the entries of M according to the $(0, q)$-invariance constraint, and return M as OUTPUT.

As regard the correctness of this reconstruction algorithm, it relies on the analysis of what is stored in M after Step 1: at that stage, in fact, the entries in the first column of its rectangular scan $R(M)$ differ from those of A by the same constant value, without overcoming.

The formal counterpart of what sketched above is in the following lemmas:

Lemma 9. *After performing Step* 1 *of* RecConstRows(A, p, q), *for each* $1 \leq i < m'$, *it holds*

$$R(M)[i,1] - R(M)[i+1,1] = A[i,1] - A[i+1,1].$$

Proof. Let us first inspect the entries placed in the rows i and $i+p$ of M during Step 1 of RecConstRows(A, p, q), for a generic index $1 \leq i < m'$:

if $A[i,1] \leq A[i+1,1]$, then in row $i+p$ of M are added $A[i+1,1] - A[i,1] + PEnt[i]$ entries 1. Since row i of M contains $PEnt[i]$ entries 1, then, at that step, it holds

$$R(M)[i,1] - R(M)[i+1] = A[i,1] - A[i+1,i]; \tag{1}$$

if $A[i,1] > A[i+1,1]$, and $PEnt[i] \geq A[i,1] - A[i+1,1]$, then in row $i+p$ of M are added $PEnt[i] - A[i,1] + A[i+1,1]$ entries 1, so Eq. (1) still holds;

if $A[i,1] > A[i+1,1]$, and $PEnt[i] < A[i,1] - A[i+1,1]$, then in row i of M are added $A[i,1] - A[i+1,1] - PEnt[i]$ entries 1 in addition to the $PEnt[i]$ ones already present, so Eq. (1) is again satisfied.

For each $i < i' < i+p$, Step 1 eventually changes some entries from row $i+1$ to row $i+p-1$ of M. So, both $R(M)[i,1]$ and $R(M)[i+1,1]$ increase their value by the same amount, without compromising the validity of Eq. (1).

Finally, if $i' \geq i+p$, then Step 1.3 may modify the values of $R(M)[i,1]$ and $R(M)[i+1,1]$, but again by the same amount, since the (eventually) added entries 1 respect the $(p, 0)$-invariance in the rows of index less than i', so Eq. (1) definitively holds, and we obtain the thesis. □

Lemma 10. *Let us consider the vector* $PEnt$ *as updated at the end of Step* 1 *of* RecConstRows(A, p, q). *For each matrix* M *such that* $R(M) = A$, *and for each* $0 \leq i \leq m$, *it holds*

$$M[i,1] + \cdots + M[i,p] \geq PEnt[i].$$

Proof. By contradiction, we assume that there exists an index $1 \leq i \leq m$ and a matrix M' such that

$$M[i,1] + \cdots + M[i,p] + k = PEnt[i],$$

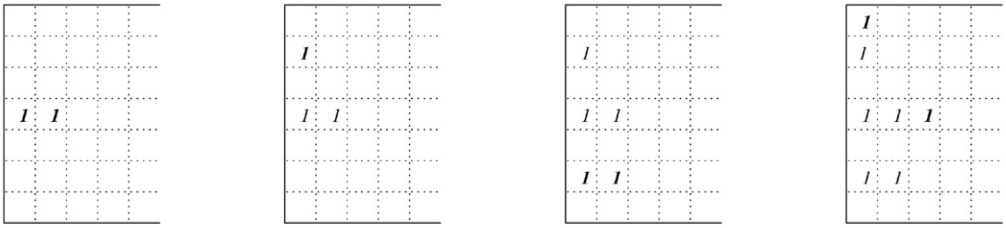
with $R(M) = A$ and $k > 0$. By Lemma 9, the same equation holds for each row $i' = (i) mod_p$.

Let i_0 be the first index such that

A :

5	5	5	5	5	5
7	7	7	7	7	7
6	6	6	6	6	6
8	8	8	8	8	8
5	5	5	5	5	5

Fig. 3. The matrix A of Example 13.



PEnt: (0,0,0,2,0,0,0) (a) **PEnt: (0,1,0,2,0,0,0)** (b) **PEnt: (0,1,0,2,0,2,0)** (c) **PEnt: (1,1,0,3,0,2,0)** (d)

Fig. 4. Step 1 of RecConstRows$(A, 3, 4)$.

- $i_0 = (i) mod_p$;
- for each $i' = (i) mod_p$, it holds $PEnt[i_0] \leq PEnt[i']$.

If $i_0 \leq m - p$, then the minimality of the value of $PEnt[i_0]$ assures that $PEnt[i_0] = 0$ before Step 1 reached row i_0, and, consequently, that $A[i_0, 1] \leq A[i_0 + 1, 1]$.

As soon as Step 1.1 reaches the row index i_0, it eventually increases the value $PEnt[i_0 + p]$, leaving unchanged that of $PEnt[i_0]$. Again the minimality of $PEnt[i_0]$ assures that no changes will be performed to the value of $PEnt[i_0]$ till the end of Step 1.

Hence, the assumption $M[i_0, 1] + \cdots + M[i_0, p] + k = PEnt[i_0]$, with $k > 0$, generates a contradiction.

If $i_0 > m - p$, then a similar argument holds, and so we get the thesis. □

Now, also Step 2 of RecConstRows (A, p, q) can be better understood: the k elements 0 which change their value to 1 and which are added in the upper leftmost $p \times q$ positions of M, fill the gap among $R(M)[i, 1]$ and $A[i, 1]$, so that the output matrix M has the desired property $R(M) = A$.

Corollary 11. *Each row i of a matrix M having A as rectangular scan contains at least $PEnt[i]$ elements which are $(p, 0)$-invariant, and not $(0, q)$-invariant.*

A procedure which reconstructs a smooth matrix whose (p, q)-rectangular scan A has constant columns, say RecConstCols(A, p, q), can be easily inferred from RecConstRows, so, in the following, we will consider it as already defined.

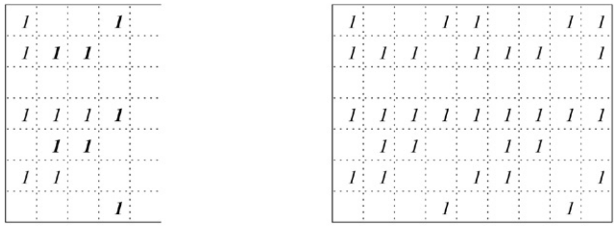
From Lemmas 9 and 10, it is straightforward that

Theorem 12. *The problem Reconstruction(A, p, q) can be solved in $O(m\,n)$, when A has constant rows or constant columns.*

Example 13. Let us follow the computation RecConstRows$(A, 3, 4)$, with A depicted in Fig. 3.

Step 1: the matrix M is created, and its first four columns, together with the vector $PEnt$, are modified as shown in Fig. 4. More precisely,

$A[1, 1] + 2 = A[1, 2]$ requires Step 1.1 to place two entries 1 in (the leftmost positions of) row 4 of Fig. 4(a);
$A[1, 2] = A[1, 3] + 1$ requires Step 1.3 to place one entry 1 in row 2 of Fig. 4(b);
$A[1, 3] + 2 = A[1, 4]$ requires Step 1.1 to place two entries 1 in row 6 of Fig. 4(c);
$A[1, 1] = A[1, 2] + 3$ requires Step 1.3 to add one entry 1 both in row 1 and in row 4 of Fig. 4(d).



(a) (b)

Fig. 5. Steps 2 and 3 of RecConstRows(A, 3, 4).

Step 2 places the remaining $k = A[1, 1] - PEnt[1] - PEnt[2] - PEnt[3] = 3$ entries 1 in the upper leftmost $p \times q$ submatrix of M, and propagates them according to the (3, 0)-invariance, paying attention that no collisions occur (see Fig. 5(a)).

Step 3 completes M according to the (0, 4)-invariance, giving the final solution depicted in Fig. 5(b).

The general reconstruction algorithm

Theorem 7 allows one to foresee the use of the procedures RecConstRows and RecConstCols to solve *Reconstruction*(A, p, q), when A is (1, 1)-smooth: the algorithm at first will split the matrix A into two parts having constant rows and columns, respectively, then it will apply to each of them the appropriate reconstruction procedure, and finally it will merge the two outputs. Performing the merging stage a conflict occurs when the same position in the two output matrices has value 1. To prevent it, small refinements to the outputs of RecConstRows and RecConstCols will be required.

So, let us start by showing in the next lemma a quick way of finding all the possible decompositions of a (1, 1)-smooth matrix into two parts having constant rows and columns, respectively.

Lemma 14. *Let A be a $m \times n$ integer matrix. If A is* (1, 1)*-smooth, then it admits $k + 1$ different decompositions into two matrices having constant rows and columns, with k being the minimum among all the elements of A.*

Proof. The thesis is achieved by defining a procedure which gives as output a complete list of couples of matrices (A_r^t, A_c^t), with $0 \le t \le k$, each of them representing a decomposition of A into two parts having constant rows and columns, and successively, by proving its correctness:

Decompose (A)

Input: an integer $m \times n$ matrix A.

Output: a sequence of different couples of matrices $(A_r^0, A_c^0), \ldots, (A_r^k, A_c^k)$, with k being the minimum element of A, such that, for each $0 \le t \le k$, A_r^t has constant rows, A_c^t has constant columns, and $A_r^t + A_c^t = A$. If such a sequence does not exist, then return FAILURE.

Procedure:

Step 1: initialize all the elements of two $m \times n$ matrices A_c and A_r to the value 0. Let k be the minimum among the entries of A. From each element of A, subtract the value k and store the result in A_c;

Step 2: for each $1 \le i \le m$

Step 2.1: compute
$$k_i = \min_j \{A_c[i, j] \ : \ 1 \le j \le n\};$$

Step 2.2: subtract the value k_i from each element of A_c;

Step 2.3: set all the elements of row i of A_r to the value k_i;

Step 3: **if** the matrix A_c has not constant columns **then** FAILURE

else for each $0 \le t \le k$, create the matrices A_r^t and A_c^t such that
$$A_r^t[i, j] = A_r[i, j] + t \quad \text{and} \quad A_c^t[i, j] = A_c[i, j] + k - t,$$
with $1 \le i \le m$ and $1 \le j \le n$.

Give the sequence $(A_r^0, A_c^0), \ldots, (A_r^k, A_c^k)$ as OUTPUT.

$$A: \quad \begin{array}{|c|c|c|c|c|}\hline 6 & 5 & 5 & 4 & 5\\ \hline 5 & 4 & 4 & 3 & 4\\ \hline 4 & 3 & 3 & 2 & 3\\ \hline 5 & 4 & 4 & 3 & 4\\ \hline\end{array}$$

Fig. 6. The (1, 1)-smooth matrix A of Example 15.

$$A_r^0: \begin{array}{|c|c|c|c|c|}\hline 2 & 2 & 2 & 2 & 2\\ \hline 1 & 1 & 1 & 1 & 1\\ \hline 0 & 0 & 0 & 0 & 0\\ \hline 1 & 1 & 1 & 1 & 1\\ \hline\end{array} \qquad A_r^1: \begin{array}{|c|c|c|c|c|}\hline 3 & 3 & 3 & 3 & 3\\ \hline 2 & 2 & 2 & 2 & 2\\ \hline 1 & 1 & 1 & 1 & 1\\ \hline 2 & 2 & 2 & 2 & 2\\ \hline\end{array} \qquad A_r^2: \begin{array}{|c|c|c|c|c|}\hline 4 & 4 & 4 & 4 & 4\\ \hline 3 & 3 & 3 & 3 & 3\\ \hline 2 & 2 & 2 & 2 & 2\\ \hline 3 & 3 & 3 & 3 & 3\\ \hline\end{array}$$

$$A_c^0: \begin{array}{|c|c|c|c|c|}\hline 4 & 3 & 3 & 2 & 3\\ \hline 4 & 3 & 3 & 2 & 3\\ \hline 4 & 3 & 3 & 2 & 3\\ \hline 4 & 3 & 3 & 2 & 3\\ \hline\end{array} \qquad A_c^1: \begin{array}{|c|c|c|c|c|}\hline 3 & 2 & 2 & 1 & 2\\ \hline 3 & 2 & 2 & 1 & 2\\ \hline 3 & 2 & 2 & 1 & 2\\ \hline 3 & 2 & 2 & 1 & 2\\ \hline\end{array} \qquad A_c^2: \begin{array}{|c|c|c|c|c|}\hline 2 & 1 & 1 & 0 & 1\\ \hline 2 & 1 & 1 & 0 & 1\\ \hline 2 & 1 & 1 & 0 & 1\\ \hline 2 & 1 & 1 & 0 & 1\\ \hline\end{array}$$

Fig. 7. The three decompositions of the matrix A.

Example 15 shows a run of the algorithm. By construction, each couple (A_r^t, A_c^t) is a decomposition of A, and furthermore, A_r^t has constant rows.

What remains to prove is that the matrix A_c updated at the end of Step 2 has constant columns (and, consequently, the same holds for all the matrices A_c^t). Let us denote by r_i the common value of the elements of the i-th row of A_r, and let us proceed by contradiction, assuming that A_c has not constant columns. Since A is the sum of a column constant and a row constant matrix, and for all $1 \leq i \leq m - p + 1$ and $1 \leq j \leq n - q + 1$, it holds $A[i, j] = r_i + A_c[i, j] + k$, then A_c is also the sum of a column constant matrix, and a row constant matrix, this last having at least one row, say i_0, whose elements have value $k'_{i_0} \neq 0$.

This situation generates an absurdity, since k_{i_0} computed in Step 2.1 turns out no longer to be the minimum of row i_0 in A_c, updated to that step.

Since a matrix having constant rows (resp. columns) cannot be obtained as the sum of a matrix having constant rows and a matrix having constant columns unless the latter is a constant matrix, then the $k + 1$ decompositions listed by the algorithm are all the possible ones. □

Example 15. Let us follow the steps of the procedure DECOMPOSE(A), with the matrix A depicted in Fig. 6.

Step 1: we subtract from all the elements of A, the value $k = 2$, i.e. its minimum element, and we store the obtained result in the matrix A_c.

Step 2: for each $1 \leq i \leq m - p + 1$, we find the minimum value k_i among the elements of row i of A_c (Step 2.1), we subtract it from all these elements (Step 2.2), and finally, we set the elements in row i of A_r to the value k_i (Step 2.3). In our case, the minima are $k_1 = 2$, $k_2 = 1$, $k_3 = 0$ and $k_4 = 1$.

Step 3: the matrix A_c updated at the end of Step 2 has constant columns, so the three different decompositions of A can be computed and listed.
The output is depicted in Fig. 7.

Now we are finally able to define a general procedure which solves the problem *Reconstruction*(A, p, q), when A is (1, 1)-smooth:

RECSMOOTH(A, p, q)

Input: an integer (1, 1)-smooth matrix A of dimension $m' \times n'$ and two integers p and q.

Output: a binary matrix M of dimension $m \times n$, with $m = m' + p - 1$ and $n = n' + q - 1$, having A as (p, q)-rectangular scan, if it exists, else return FAILURE.

Procedure:

Step 1: run DECOMPOSE(A), and let $(A_r^0, A_c^0), \ldots, (A_r^k, A_c^k)$ be its output (remind that k is the minimum element of A).

Set $t = 0$;

Step 1.1: run Step 1 of RECCONSTROWS (A_r^t, p, q), and let $PEnt_{\text{row}} = PEnt$, with $PEnt$ updated at the end of the step.

Define $k_{\text{row}} = A_r^t[1, 1] - PEnt[1] - \cdots - PEnt[p]$, and $PEnt_{\equiv p}$ to be the vector having p elements, and such that:

$$PEnt_{\equiv p}[i] = \max\{PEnt_{\text{row}}[i'] : i' = (i)mod_p\},$$

with $1 \le i \le p$, and $1 \le i' \le m$.

Step 1.2: run Step 1 of RECCONSTCOLS(A_c^t, p, q), and let $PEnt_{\text{col}} = PEnt$, with $PEnt$ updated at the end of the step.

Define $k_{\text{col}} = A_c^t[1, 1] - PEnt[1] - \cdots - PEnt[q]$, and let $PEnt_{\equiv q}$ be the vector having q entries, and such that:

$$PEnt_{\equiv q}[j] = \max\{PEnt_{\text{col}}[j'] : j' = (j)mod_q\},$$

with $1 \le j \le q$, and $1 \le j' \le n$;

Step 1.3: among all the possible $p \times q$ matrices whose entries are in $\{P, Q, 1, 0\}$, choose one, say W, such that:

(i) the number of entries Q in its i-th row is $PEnt_{\equiv p}[i]$.
(ii) the number of entries P in its j-th column is $PEnt_{\equiv q}[j]$;
(iii) the number of entries 1 is $k_{\text{row}} + k_{\text{col}}$.

If W does not exist and $t \ne k$ **then** set $t = t + 1$, and return to Step 1.1.
If W does not exist and $t = k$ **then** FAILURE;

Step 2: create the $m \times n$ matrix M, and initialize its entries as follows:

Step 2.1: for each $0 \le i \le m$ and for each $0 \le j \le q$,
if $PEnt_{\text{row}}[i] \ne 0$ and $W[i', j] = Q$, with $i' = (i)mod_p$ **then** set both $M[i, j] = Q$ and $PEnt_{\text{row}}[i] = PEnt_{\text{row}}[i] - 1$;

Step 2.2: for each $0 \le j \le n$ and for each $0 \le i \le p$,
if $PEnt_{\text{col}}[j] \ne 0$ and $W[i, j'] = P$, with $j' = (j)mod_q$ **then** set both $M[i, j] = P$ and $PEnt_{\text{col}}[j] = PEnt_{\text{col}}[j] - 1$;

Step 2.3: for each $0 \le i \le p$ and for each $0 \le j \le q$,
if $W[i, j] = 1$, **then** set $M[i, j] = 1$;

Step 2.4: fill the matrix M imposing the $(0, q)$-invariance of its entries Q and 1, and the $(p, 0)$-invariance of its entries P and 1;

Step 3: change the values P and Q to 1, and set the remaining entries of M with the value 0; finally, give M as output.

Theorem 16. *The problem Reconstruction(A, p, q), with A being $(1, 1)$-smooth, admits a solution if and only if* RECSMOOTH(A, p, q) *does not return FAILURE.*

Proof. $(\Rightarrow)$ Let M be a solution of *Reconstruction*(A, p, q), and let us assume that $M = M_1 + M_2$, with M_1 and M_2 being $(0, q)$-invariant and $(p, 0)$-invariant, respectively.

Let (A_r^t, A_c^t) be one of the decompositions of A such that $R(M_1) = A_r^t$ and $R(M_2) = A_c^t$.

Lemma 10 implies that, for each $1 \le i \le m$, the value $PEnt_{\text{row}}[i]$ indicates the minimum number of elements of M which are $(0, q)$-invariant and not $(p, 0)$-invariant, and which lie in the first q columns of the solution; a symmetrical property holds for $PEnt_{\text{col}}$.

Let us construct a $p \times q$ matrix W' as follows:

– for each $1 \le i \le m$ and $1 \le j \le q$, if $M[i, j] = 1$ is $(0, q)$-invariant and not $(p, 0)$-invariant, then set $W'[i', j] = Q$, with $i' = (i)mod_p$;

$$A_r : \begin{array}{|cccccccc|}\hline 3&3&3&3&3&3&3&3\\ 3&3&3&3&3&3&3&3\\ 4&4&4&4&4&4&4&4\\ 2&2&2&2&2&2&2&2\\ 2&2&2&2&2&2&2&2\\ \hline\end{array} \qquad A_c \begin{array}{|cccccccc|}\hline 4&4&3&2&2&2&3&4\\ 4&4&3&2&2&2&3&4\\ 4&4&3&2&2&2&3&4\\ 4&4&3&2&2&2&3&4\\ 4&4&3&2&2&2&3&4\\ \hline\end{array}$$

Fig. 8. The decomposition of the rectangular scan A used in Example 17.

$$W : \begin{array}{|ccc|}\hline P&P&P\\ Q&1&1\\ Q&Q&1\\ \hline\end{array} \qquad M : \begin{array}{|cccccccccc|}\hline &P&P&&&&&&P&P\\ &1&1&&&&&&&\\ Q&Q&1&&&&&&&\\ &&&&&&&&&\\ Q&&&&&&&&&\\ &&&&&&&&&\\ &&&&&&&&&\\ \hline\end{array} \qquad M : \begin{array}{|cccccccccc|}\hline &P&P&&&&&&P&P\\ &1&1&&1&1&&1&1&\\ Q&Q&1&Q&Q&1&Q&Q&1&Q\\ &P&P&&&&&&P&P\\ Q&1&1&Q&1&1&Q&1&1&Q\\ &&1&&&1&&&1&\\ &P&P&&&&&&P&P\\ \hline\end{array}$$

(a) (b) (c)

Fig. 9. The matrices created in successive steps of RECSMOOTH$(A, 3, 3)$.

– for each $1 \leq i \leq p$ and $1 \leq j \leq n$, if $M[i, j] = 1$ is $(p, 0)$-invariant and not $(0, q)$-invariant, then set $W'[i', j] = P$, with $j' = (j)mod_q$;

– for each $1 \leq i \leq p$ and $1 \leq j \leq q$, if $M[i, j] = 1$ is $(p, 0)$-invariant and $(0, q)$-invariant, then set $W'[i, j] = 1$.

Obviously, by definition of invariance, in W' there are no positions which are first set to a value and then modified to another. So, the existence of matrix W' implies that of a matrix W satisfying the constraints imposed in Step 1.3.

($\Leftarrow$) Immediate. □

Example 17. Let us describe a run of RECSMOOTH$(A, 3, 3)$ starting from the decomposition of A into the couple of matrices depicted in Fig. 8.

Step 1.1 produces the vectors

$$PEnt_{\text{row}} = (0, 0, 2, 0, 1, 0, 0) \quad \text{and} \quad PEnt_{\equiv p} = (0, 1, 2),$$

while Step 1.2 produces the vectors

$$PEnt_{\text{col}} = (0, 1, 1, 0, 0, 0, 0, 0, 1, 1) \quad \text{and} \quad PEnt_{\equiv q} = (1, 1, 1).$$

Among the matrices which are compatible with the requirements of Step 1.3 we choose that depicted in Fig. 9(a).

Steps 2.1, 2.2 and 2.3 produce the matrix M in Fig. 9(b) (notice that the entries P are $(p, 0)$-invariant, while the entries Q are $(0, q)$-invariant as desired). Finally, Step 2.4 produces the matrix in Fig. 9(c), and, consequently, Step 3 the output.

The following theorem holds:

Theorem 18. *The computational complexity of* RECSMOOTH(A, p, q) *is polynomial in m and n.*

Proof. We obtain the thesis by analyzing the complexity of each step of RECSMOOTH:

Step 1: the procedure DECOMPOSE clearly acts in $O(m\, n)$ time (remind that $k \leq p \cdot q$ is the minimum among the elements of A);

Step 1.1 and Step 1.2 are repeated at most k times, and, each time, they ask for a run of RECCONSTROWS and of RECCONSTCOLS which are both performed in $O(m\, n)$. The computation of $PEnt_{\equiv p}$ and $PEnt_{\equiv q}$ does not increase the complexity of these two steps.

Step 1.3 is carried on in constant time with respect to m and n.

Steps 2 and 3 require $O(m\, n)$ to create matrix M.

Hence, the total amount of time is $O(m\, n)$. □

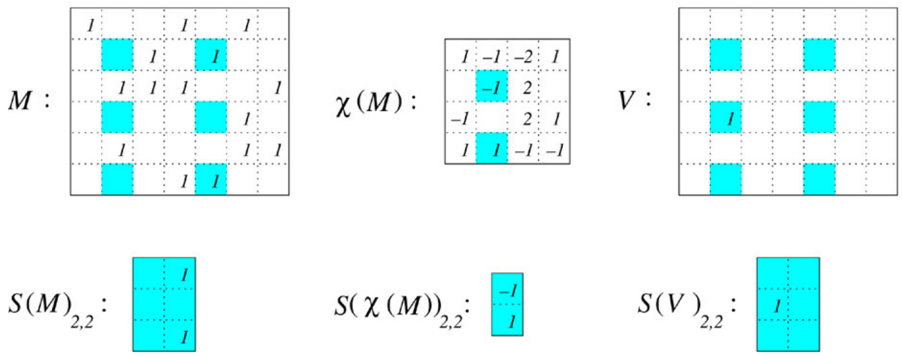


Fig. 10. The subgrids of the matrices M and $\chi(M)$ with respect to the position (2, 2). Matrix V is one of the possible valuations of $S(\chi(M))_{2,2}$.

Remark 19. We are aware that Step 1.3 of RECSMOOTH(A, p, q), i.e. the search of the matrix W, can be carried on in a smarter way, but this will bring no effective contribution to the decreasing of the computational complexity of the reconstruction, and, on the other hand, it will add new lemmas and proofs to the current section.

The first part of the paper devoted to the analysis and the reconstruction of smooth matrices is now completed.

3. Solving *Reconstruction*(A, p, q): Final challenge

This last section concerns the matrices which are not smooth: in particular, for each non-smooth matrix M, we consider the matrix $\chi(M)$ and we define a polynomial time algorithm which lists all the matrices consistent with it. Finally we will integrate it with the algorithm for reconstructing a smooth matrix defined in the previous section, and we will achieve the solution of the general reconstruction problem.

Unfortunately, the definitions introduced up to now are not specific enough to describe these further studies, and a final effort is required to the reader: what follows has the appearance of a standalone part inside this section, but the feeling of a final possible usage will acquire consciousness step by step.

Hence, let a and b be two indexes such that $1 \le a \le p$, $1 \le b \le q$, and A be an integer $m \times n$ matrix. We define the (a, b)-*subgrid* of A to be the submatrix

$$S(A)_{a,b}[i, j] = A[a + (i - 1)\, p, b + (j - 1)\, q]$$

with $1 \le a + (i - 1)p \le m$ and $1 \le b + (j - 1)q \le n$ (see Fig. 10).

If we consider again a binary matrix M, by definition it holds that

$$\begin{aligned}\chi(M)[a + (i - 1)\, p, b + (j - 1)\, q] &= S(\chi(M))_{a,b}[i, j] \\ &= S(M)_{a,b}[i, j] + S(M)_{a,b}[i + 1, j + 1] - S(M)_{a,b}[i + 1, j] \\ &\quad - S(M)_{a,b}[i, j + 1].\end{aligned}$$

A binary matrix V of dimension $m \times n$ is said to be a *valuation* of $S(\chi(M))_{a,b}$ if, for each $1 \le i \le m$, $1 \le j \le n$,

– if $i \ne (a)mod_p$ and $j \ne (b)mod_q$ then $V[i, j] = 0$;
– $S(\chi(M))_{a,b} = S(\chi(V))_{a,b}$ (see Fig. 10).

The notion of valuation extends to the whole matrix $\chi(M)$ as the union of the valuations of all its subgrids.

The following three lemmas are direct consequences of the definition of valuation.

Lemma 20. *Let $S(\chi(M))_{a,b}$ and $S(\chi(M))_{a',b'}$ be two subgrids whose valuations are V and V', respectively. If $a \ne a'$ or $b \ne b'$, then for each $1 \le i \le m$ and $1 \le j \le n$, $V[i, j] = 1$ implies $V'[i, j] = 0$.*

Lemma 21. *Let V be a valuation of $S(\chi(M))_{a,b}$, and let i_0 be a row [column] of $S(V)_{a,b}$ having all the elements equal to 1. The matrix V' such that $S(V')_{a,b}$ is equal to $S(V)_{a,b}$ except in the elements of the row [column] i_0 which are all set to 0, is again a valuation of $S(\chi(M))_{a,b}$.*

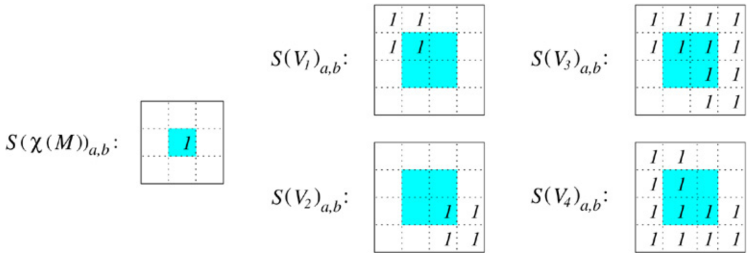


Fig. 11. The four valuations of the point $p_1 = (2, 2)$ of value 1.

If V and V' are two valuations as in Lemma 21, then we say that the valuation V is *greater than* the valuation V'. This relation can be easily extended to a finite partial order on the valuations of the subgrids of $\chi(M)$.

Lemma 22. *Let* $1 \leq i \leq m - p$ *and* $1 \leq j \leq n - q$. *If* $\chi(M)[i, j] = 2$, *then* $M[i, j] = M[i + p, j + q] = 1$, *and* $M[i + p, j] = M[i, j + q] = 0$.

It is straightforward that a symmetric result holds if $\chi(M)[i, j]$ has value -2.

The following lemma turns out to be crucial in this section. A deeper analysis of its implications could furnish material for further studies:

Lemma 23. *Given a binary matrix* M, *for each couple of integers* $1 \leq a \leq p$, $1 \leq b \leq q$, *the number of minimal elements in the partial ordering of the valuations of* $S(\chi(M))_{a,b}$ *is polynomial with respect to the dimensions* m *and* n *of* M. *Furthermore, each minimal element can be reconstructed in polynomial time with respect to* m *and* n.

Proof. Let $S(\chi(M))_{a,b}$ have dimension $m' \times n'$. We order the (positions of the) non-zero elements of $S(\chi(M))_{a,b}$ according to the numbering of its columns (from left to right), and, in the same column, according to the numbering of its rows (from top to bottom), and let $p_1, \ldots, p_t$ be the obtained sequence. We prove the thesis by induction on the number t of elements of the sequence, i.e. we prove that the addition of new non-zero elements in $S(\chi(M))_{a,b}$ do not increase "too much" the number of its possible minimal valuations.

We first observe that, by Lemma 22, the presence of entries 2 or -2 in $S(\chi(M))_{a,b}$ does not increase the number of minimal valuations, so we are allowed to focus our attention exactly on the elements of value 1 or -1. As one can expect, the symmetry of the two cases allows us to show the details of only one of them (in particular when the added element has value 1), letting the reader to infer the other:

Base $t = 1$: if $p_1 = (i, j)$ and $S(\chi(M))_{a,b}[i, j] = 1$, then the four possible valuations of $S(\chi(M))_{a,b}$ can be derived from those depicted in Fig. 11. Two of them are minimal, i.e. $S(V_1)_{a,b}$ and $S(V_2)_{a,b}$, and they can be reached both from $S(V_3)_{a,b}$ and from $S(V_4)_{a,b}$ by deleting the rows or the columns entirely filled with entries 1, as stated in Lemma 21.

Step $t \to t + 1$: let $S(\chi(M))_{a,b}$ have a sequence $p_1, \ldots, p_{t+1}$ of non-zero points, with $p_{t+1} = (i, j)$ of value 1, i.e. $S(\chi(M))_{a,b}[i, j] = 1$. Let V be a valuation of the first $p_1, \ldots p_t$ points in $S(\chi(M))_{a,b}$. It is straightforward that, for all $1 \leq i' \leq m' + 1$ and $j < j' \leq n' + 1$ it holds $S(V)_{a,b}[i', j + 1] = S(V)_{a,b}[i', j']$.

Hereafter, we show *all* the possible ways of extending V to the valuation V' which includes the point p_{t+1}. Some pictures are supplied in order to make the different cases transparent.

Let us call

0-*row*: a row of $S(V)_{a,b}$ whose elements have all value 0;

$*$1-*row*: a row of $S(V)_{a,b}$ whose element in column $j + 1$ has value 1;

$*$0-*row*: a row of $S(V)_{a,b}$ which is neither 0-row nor 1-row.

We examine all the possible configurations of $S(V)_{a,b}$, and for each of them we indicate the desired extension to the *minimal* valuation $S(V')_{a,b}$:

(i) all the rows from 1 to i are 0-rows or $*$1-rows (see Fig. 12(a)).

We define the valuation $S(V')_{a,b}$ as follows: for each $1 \leq i' \leq i$,

if row i' is a 0-row, then change from 0 to 1 the value of each entry of $S(V)_{a,b}$ in position (i', j'), with $1 \leq j' \leq j$, so that it becomes a $*$0-row;

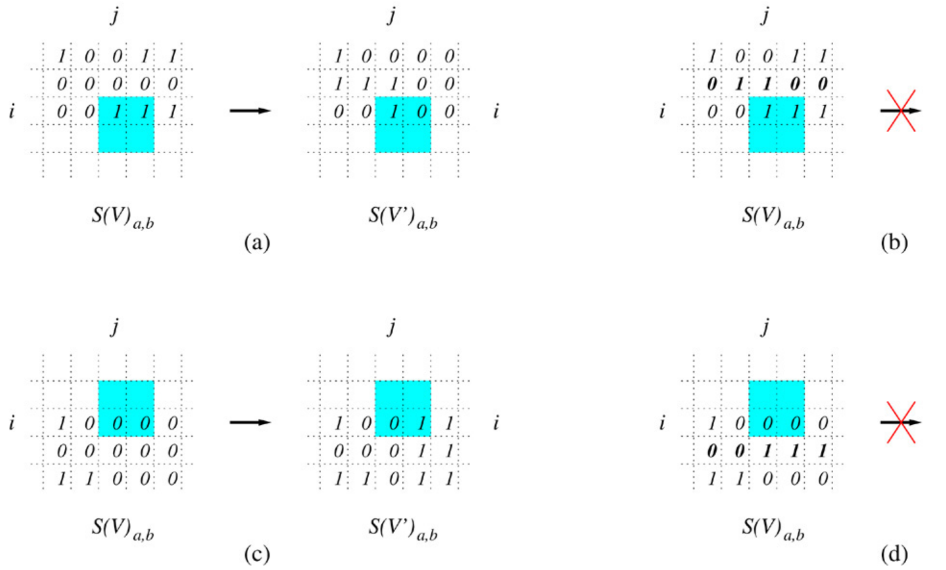


Fig. 12. Examples of the possible ways of extending a valuation V when adding point p_{t+1}. In cases (b) and (d), the boldface rows prevent V from being extended.

if row i' is a $*1$-row, then change from 1 to 0 the value of each entry of $S(V)_{a,b}$ in position (i', j'), with $j+1 \leq j' \leq n'+q-1$, so that it becomes a $*0$-row or a 0-row. If one or more 0-rows are created, then discard this valuation $S(V')_{a,b}$, since it turns out to be redundant with another one which is obtained starting from a different valuation V of $p_1, \ldots, p_t$ (easy check);

(ii) there exists a $*0$-row i', with $1 \leq i' \leq i$ (see Fig. 12(b)).
No changes in the first i rows of $S(V)_{a,b}$ allow the insertion of the new point p_{t+1}, without modifying the elements of the sequence $p_1, \ldots, p_t$;

(iii) all the rows from $i+1$ to $m'+p-1$ are 0-rows or $*0$-rows (see Fig. 12(c)).
We define the valuation $S(V')_{a,b}$ as follows: for each $i+1 \leq i' \leq m'+p-1$, change from 0 to 1 the value of each entry of $S(V)_{a,b}$ in position (i', j'), with $j+1 \leq j' \leq n'+q-1$, so that it becomes a $*1$-row. Discard such a valuation if a row having all the entries equal to 1 has eventually been created, in order to maintain minimality (remind Lemma 21);

(iv) there exists a $*1$-row i', with $i+1 \leq i' \leq n'+q-1$ (see Fig. 12(d)).
No changes in the last $(m'+p-1)-i$ rows of V allow the insertion of the new point p_{t+1}.

Remark. If the point p_{t+1} has value -1, then a further check is needed in the analog of case (i): it may happen that there exists a $*0$-row i', with $1 \leq i' \leq i$, which changes into a $*1$-row after the addition of p_{t+1}. In this case a non-minimal configuration is created.

The four configurations described above are exhaustive with respect to the addition of the single point p_{t+1} of value 1 to the valuation $S(V)_{a,b}$. However, a case has not yet been considered: it appears when two points p_{t+1} and p_{t+2} are added to $S(V)_{a,b}$, under the assumption that they have different values, and they lie in the same column.

Step $t \to t+2$: let us assume that $p_{t+1} = (i, j)$, $p_{t+2} = (i', j)$, $S(\chi(M))_{a,b}[i, j] = 1$, and $S(\chi(M))_{a,b}[i', j] = -1$. If it holds that:

(v) all the rows from i to i' are not $*1$-rows of $S(V)_{a,b}$, and there exists a $*1$-row with index greater than i' (this last condition prevents $S(V)_{a,b}$ from being extended to $S(V')_{a,b}$ by means of (iii)).
We define the valuation V' as follows:
for each $i \leq i_0 \leq i'$, change from 0 to 1 the value of each entry of $S(V)_{a,b}$ in position (i_0, j'), with $j+1 \leq j' \leq n'+q-1$, so that it becomes a $*1$-row.

In the following, when we mention the above described cases (i)–(v), we intent to include also their symmetrical counterparts. It is immediate to check that

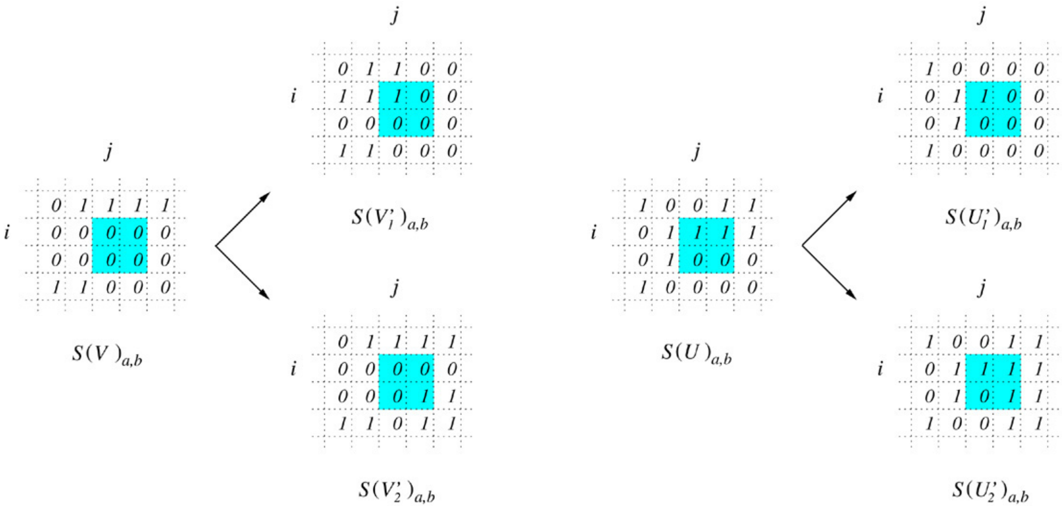


Fig. 13. The valuations $S(V)_{a,b}$ and $S(U)_{a,b}$ extend in two different valuations. Since $S(U)_{a,b}$ does not contain any 0-row, one of its extensions (i.e. $S(U'_2)_{a,b}$) is not minimal.

$$S(\chi(M))_{a,b}: \begin{array}{|cccc|} \hline 1 & & & \\ & -1 & & \\ & & 1 & \\ & & & -1 \\ \hline \end{array}$$

Fig. 14. The matrix $S(\chi(M))_{a,b}$ of Example 28.

– cases (i)–(v) extend $S(V)_{a,b}$ by adding the desired point (or points);
– each extension of $S(V)_{a,b}$ is minimal, since no rows or columns completely filled with entries 1 are added;
– all the minimal valuations for the sequences $p_1, \ldots, p_{t+1}$ or $p_1, \ldots, p_{t+1}, p_{t+2}$ are obtained by means of (i)–(v).

So, what remains to prove is that the number of the different minimal valuations for a given matrix $S(\chi(M))_{a,b}$ is polynomial in its dimensions m' and n' (and consequently in the dimensions m and n of M). We achieve our aim by showing that the number of valuations $S(V)_{a,b}$ which admit more than a single minimal extension is bounded by m'. Some properties are needed:

Proposition 24. *Each valuation $S(V)_{a,b}$ admits at most two different minimal extensions both when adding a single point p_{t+1} (see Fig. 13), and when adding two points p_{t+1} and p_{t+2}, under the assumptions of* (v).

Proposition 25. *If the valuation $S(V)_{a,b}$ does not contain any* 0*-row, then it admits at most one minimal extension via* (i)–(v) *(see Fig. 13, valuation U).*

Proposition 26. *The* 0*-rows of each valuation which extends $S(V)_{a,b}$ are a subset of those of $S(V)_{a,b}$. Furthermore, if $S(V)_{a,b}$ extends in two minimal ways, then the two extensions do not share any* 0*-row.*

Proposition 27. *Two minimal valuations of $S(\chi(M))_{a,b}$ do not share any* 0*-row.*

The proofs of these properties directly follow from the definitions of (i)–(v).

Hence, Proposition 27 assures that each matrix $S(\chi(M))_{a,b}$ has at most m' different minimal valuations containing 0-rows. From Proposition 25, if we add one or two new points to them, then at most m new minimal valuations may arise. As a neat consequence, we obtain that the number of minimal valuations of a given matrix $S(\chi(M))_{a,b}$ is polynomial in m' and n', and so it is the complexity of their reconstruction. □

Example 28. Let us find all the minimal valuations of the matrix $S(\chi(M))_{a,b}$ depicted in Fig. 14.

We proceed from the leftmost entry of $S(\chi(M))_{a,b}$ different from 0, till the rightmost one, and we construct, step by step, all the possible minimal valuations, as described in (i)–(v).

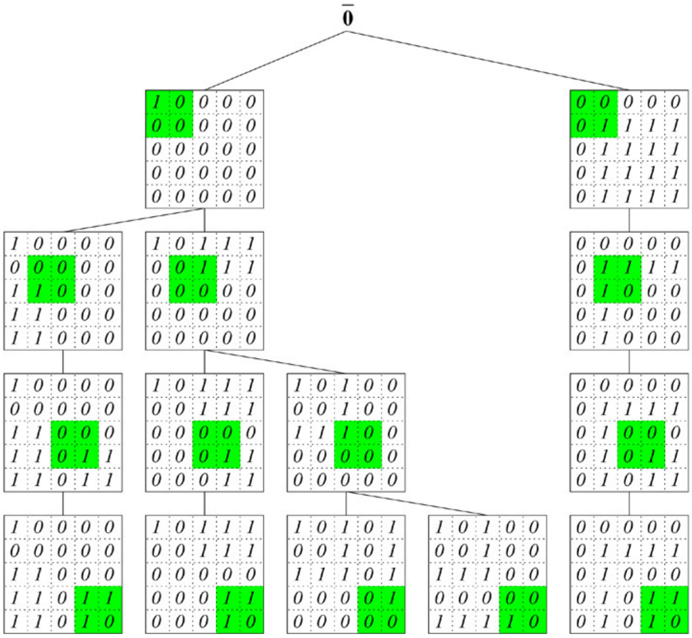


Fig. 15. The computation of the minimal valuations of $S(\chi(M))_{a,b}$.

The computation is represented in Fig. 15, by using a tree whose root is the matrix having all the entries equal to 0, and which represents the (a, b) subgrid of the valuation of a 0-homogeneous matrix. The nodes at level k are all the possible (a, b) subgrids of the valuations of the first k entries different from 0 of $S(\chi(M))_{a,b}$.

On each matrix, the highlighted cells refer to the corresponding entry 1 or -1 of $S(\chi(M))_{a,b}$ which is being considered.

It is easy to check that any further addition of 1 or -1 entries in $S(\chi(M))_{a,b}$ does not increase the number of the minimal valuations.

The following variant of the procedure RECSMOOTH will be used in the final reconstruction algorithm; the details of the procedure which differ from the original ones are given:

RECSMOOTHALL(A, p, q)

Input: an integer matrix A and two integers p and q.

Output: a (eventually void) sequence of $m \times n$ matrices $M_1, \ldots, M_k$ having elements in $\{1_P^{(1)}, \ldots, 1_P^{(k_{\mathrm{col}})}, 1_Q^{(1)}, \ldots, 1_Q^{(k_{\mathrm{row}})}, P, Q, 0\}$, with $1 \leq n \leq p \times q$.

Procedure:

Step 1: ...

Step 1.1: ...

Step 1.2: ...

Step 1.3: list all the possible $p \times q$ matrices such that, for each of them

(i) the number of the entries Q in its i-th row is $PEnt_{\equiv p}[i]$.

(ii) the number of the entries P in its j-th column is $PEnt_{\equiv q}[j]$;

(iii) at least one occurrence of each entry in $1_P^{(1)}, \ldots, 1_P^{(k_{\mathrm{col}})}$ and at least one occurrence of each entry in $1_Q^{(1)}, \ldots, 1_Q^{(k_{\mathrm{row}})}$ is present. Furthermore, all the entries with pedex P [resp. Q], and having the same index, must lie in the same column [resp. row].

If $t \neq k$ **then** set $t = t + 1$, and return to Step 1.1.

Step 2: let $W_1, \ldots, W_K$ be the output list of Step 1. For each $1 \leq i \leq K$, use matrix W_i to create the $m \times n$ matrix M_i whose entries are initialized as follows:
Step 2.1: ...
Step 2.2: ...
Step 2.3: for each $0 \leq i' \leq p$ and $0 \leq j \leq q$,
if $W_i[i', j] \notin \{P, Q\}$, then set $M_i[i', j] = W_i[i', j]]$;
Step 2.4: fill the matrix M imposing the $(0, q)$-invariance of its entries $Q, 1_P^{(1)}, \ldots, 1_P^{(k_{\mathrm{col}})}, 1_Q^{(1)}, \ldots, 1_Q^{(k_{\mathrm{row}})}$, and the $(p, 0)$-invariance of its entries $P, 1_P^{(1)}, \ldots, 1_P^{(k_{\mathrm{col}})}, 1_Q^{(1)}, \ldots, 1_Q^{(k_{\mathrm{row}})}$;

Step 3: return the sequence $M_1, \ldots, M_K$ as output.

This variant of RECSMOOTH inherits its $O(m\, n)$ computational complexity.

As a final observation, one may wonder about the meaning of the indexed entries 1_P and 1_Q inside each matrix: the elements 1_P [resp. 1_Q] having the same index mark the positions where a set of entries whose rectangular scan is 1-homogeneous, can be placed. The choice of all the possible sets of positions for the placement of the $k_{\mathrm{row}} + k_{\mathrm{col}}$-homogeneous part of A constitutes a key point in the definition of the final reconstruction algorithm which follows:

RECONSTRUCTION(A, p, q)

Input: an integer matrix A and two integers p and q.

Output: an $m \times n$ binary matrix M having A as (p, q) rectangular scan, if it exists, else return FAILURE.

Procedure:

Step 1: for each $1 \leq a \leq p$ and $1 \leq b \leq q$, compute the sequence of minimal valuations of $S(\chi_{1,1}(A))_{a,b}$;

Step 2: sum in all possible ways an element from each sequence of valuations computed in Step 1, and let $M_1, \ldots M_v$ be the obtained sequence of binary matrices;

Step 3: for each $1 \leq t \leq v$,
Step 3.1: compute the matrix $A_t = A - R(M_t)$;
Step 3.2: run RECSMOOTHALL(A_t, p, q), and let $M'_1, \ldots, M'_K$ be its output. Set $t' = 1$;
Step 3.3: until $t' \leq K$, compute a matrix M by merging the matrix M_t and the matrix $M'_{t'}$ as follows: initialize $M = M_t$;
for each $1 \leq i \leq m$, $1 \leq j \leq n$
if $M'_{t'}[i, j] \in \{P, Q\}$, **then**
if $M[i, j] = 1$, **then** set $t' = t' + 1$ and return to Step 3.3, **else** $M[i, j] = 1$;
if $M'_{t'}[i, j] = 1_P^{(n_0)}$, with $1 \leq n_0 \leq k_{\mathrm{col}}$, **then**
if $M[i', j] = 0$, for each position (i', j), with $i' = (i) mod_p$, **then** set $M[i', j] = 1$, and change to 0 the remaining entries in column j of $M'_{t'}$ having value $1_P^{(n_0)}$ **else** set $M'_{t'}[i', j] = 0$, and, if no other elements $1_P^{(n_0)}$ are in column j, set $t' = t' + 1$ and return to Step 3.3;
if $M'_{t'}[i, j] = 1_Q^{(m_0)}$, with $1 \leq m_0 \leq k_{\mathrm{row}}$, **then**
if $M[i, j'] = 0$, for each position (i, j'), with $j' = (j) mod_q$, **then** set $M[i, j'] = 1$, and change to 0 the remaining entries in row i of $M'_{t'}$ having value $1_Q^{(m_0)}$ **else** set $M'_{t'}[i, j'] = 0$, and, if no other elements $1_Q^{(m_0)}$ are in row i, set $t' = t' + 1$ and return to Step 3.3;
Return matrix M as output;

Step 4: return FAILURE.

The correctness of the procedure is straightforward, since we create *all* the possible minimal valuations for the entries of A which prevent it from being smooth, and successively, we merge them with *all* the possible solutions for its remaining smooth entries. As a neat consequence we have

Theorem 29. *The problem Reconstruction(A, p, q) admits a solution, if and only if the algorithm* RECONSTRUCTION (A, p, q) *finds it.*

However, one can ask whether such a search always produces an output in an amount of time which is polynomial in the dimensions m and n of the solution. The answer is given in the proof of the following

Theorem 30. *The computational complexity of* RECONSTRUCTION(A, p, q) *is polynomial in the dimension* $m \times n$ *of the solution.*

Proof. The complexity of the algorithm can be computed as the sum of the complexities of its steps, in particular:

Step 1: Lemma 23 assures that the computation of all the valuations of $S(\chi_{1,1}(A))_{a,b}$ can be performed in polynomial time with respect to m and n.

Step 2: the procedure asks for summing in all possible ways an element from each of the $p \cdot q$ sequences of valuations created in Step 1. Since each sum is performed in $O(mn)$, then the total complexity remains polynomially bounded by m and n.

Step 3: the computation of the matrix A_t, the polynomial procedure RECSMOOTHALL and the merging process of M_t with $M'_{t'}$ are performed a polynomial number of times, without increasing the polynomial time complexity of the algorithm. □

References

[1] E. Barcucci, A. Del Lungo, M. Nivat, R. Pinzani, Reconstructing convex polyominoes from horizontal and vertical projections, Theoret. Comput. Sci. 155 (2) (1996) 321–347.

[2] M. Chrobak, C. Dürr, Reconstructing hv-convex polyominoes from orthogonal projections, Inform. Process. Lett. 69 (1999) 283–289.

[3] A. Frosini, M. Nivat, L. Vuillon, An introductive analysis of Periodical Discrete Sets from a tomographical point of view, Theoret. Comput. Sci. 347 (1–2) (2005) 370–392.

[4] G.T. Herman, A. Kuba (Eds.), Discrete Tomography: Foundations Algorithms and Applications, Birkhauser, Boston, MA, 1999.

[5] M. Nivat, Sous-ensembles homogénes de $\mathbb{Z}^2$ et pavages du plan, C. R. Acad. Sci. Paris Ser. I 335 (2002) 83–86.

[6] H. Ryser, Combinatorial properties of matrices of zeros and ones, Canad. J. Math. 9 (1957) 371–377.