

Solving Some Instances of the 2-Color Problem

S. Brocchi¹, A. Frosini¹, and S. Rinaldi²

¹ Dipartimento di Sistemi e Informatica
Università di Firenze, Firenze, Italy

brocchi@dsi.unifi.it, andrea.frosini@unifi.it

² Dipartimento di Scienze Matematiche ed Informatiche
Università di Siena, Siena, Italy
rinaldi@unisi.it

Abstract. In the field of Discrete Tomography, the *2-color problem* consists in determining a matrix whose elements are of two different types, starting from its horizontal and vertical projections. It is known that the one color problem has a polynomial time reconstruction algorithm, while, with $k \geq 2$, the k -color problem is NP-complete. Thus, the 2-color problem constitutes an interesting example of a problem just in the frontier between hard and easy problems.

In this paper we define a linear time algorithm to solve a set of its instances, where some values of the horizontal and vertical projections are constant, while the others are upper bounded by a positive number proportional to the dimension of the problem. Our algorithm relies on classical studies for the solution of the one color problem.

Keywords: Discrete tomography, polynomial time algorithm, k-color problem.

1 Introduction

Recently, a series of valuable improvements in transmission electron microscopy have finally allowed the resolution of an inspected object till reaching the atomic scale (HRTEM). More precisely, since these last years it is possible to collect a huge quantity of data about the density of a material by focusing beams of electrons across it, and then measuring the decreasing of their energies: denser areas of the material will absorb more energy from the beams. The HRTEM has been mainly applied to biological specimens' studies to find structural defects in materials, but it has also furnished the bases for a new technique, called QUANTITEM ([9], [11]), that performs a *quantitative* analysis of highly structured materials like crystals. In particular QUANTITEM allows to measure the exact number of atoms (or other primary constituents) of an object along a set of lines parallel to given directions. Such a measurements, called *projections*, are usually arranged as vectors or matrices having integer entries, and they are used to recover some geometrical properties of the considered object. Usually, the final goal is its faithful reconstruction. This problem is a typical example of *inverse problem*, and fits in the area of computer science called Discrete Tomography,

a branch of the more general Computerized Tomography where only discrete structures are considered. Other classical problems of interest for Discrete Tomography are to determine when one or more projections are consistent with at least one discrete object belonging to a given class, say *consistency problem*, and to search for different objects which share the projections, say *uniqueness problem*.

The choice of representing each physical planar object by means of a discrete set of points in the lattice $\mathbb{Z} \times \mathbb{Z}$ is commonly accepted, and so is also the highly simplified but relevant model, which uses matrices of elements on a finite set $\Sigma = \{x_1, \dots, x_c\} \cup \{0\}$, where each entry $x_i \in \Sigma$ or 0 stands for the presence of an atom of the material x_i or the absence of atoms in the correspondent point of $\mathbb{Z} \times \mathbb{Z}$.

So, a fundamental parameter is the number c of different atoms which compose the object, and that may be detected by the crossing beams' decreasing energies. Actually, the QUANTITEM allows the resolution of polyatomic structures, collecting the obtained measurements in multidimensional vectors.

The simplest case is in presence of an homogeneous material, i.e. $c = 1$: this scenery has been deeply studied in the past years, starting from the classical result by Ryser who showed, in [10], how to reconstruct homogeneous planar sets of points from two projections in polynomial time. Successively, in [6] the authors extended Ryser's result to the lattice \mathbb{Z}^d , with $d \geq 2$, and proved the NP-hardness of the same reconstruction problem in presence of three or more projections.

Obviously, polyatomic discrete sets inherit this last result, and the successive researches focused only on the case of two projections, so it is commonly indicated as *c-color problem* the problem of reconstructing a planar object having c different types of atoms from two projections, w.l.g. the horizontal and the vertical ones.

In [7], the authors furnished a proof of the NP-hardness of the *c-color problem*, with $c \geq 6$. Later and with different techniques, in [2], this result was improved, by showing that the presence of three different types of atoms is sufficient to maintain the *c-color problem* NP-hard. Recently, this result has been definitely extended to $c = 2$ in [5].

The present paper fits into this research line: we consider a significative class of instances the 2-color problem where some constraints on the elements of the two vectors are imposed, and we provide an algorithm that finds a solution to each element in polynomial time. In a few words, the algorithm uses the imposed constraints to reduce a given instance I of 2-color into four solvable 1-color ones, and then acts on their solutions by adding few further elements in order to reconstruct one of the solutions of I .

The paper is organized as follows: in section 2 we recall some basic notions of discrete tomography, and we define the main problem. Then we give some results about the 1-color problem that will be necessary to prove the correctness of the solving algorithm. In section 3 we describe two procedures, one regarding the insertion of elements of a given color in a matrix with some particular

restrictions, and the other concerning the splitting of a multiset of integers into two parts, satisfying a certain balance property. Finally, in section 4, we define a special set of instances of the 2-color problem, and then describe the algorithm which solves such instances. Conclusions are in section 5.

2 Notations and Preliminary Results

Following the standard model, we represent a planar discrete object S made up with c different types of atoms using a $m \times n$ matrix $A = (a_{i,j})$, whose elements belong to the set $\Sigma \cup \{0\}$, with $\Sigma = \{x_1, \dots, x_c\}$. We say that A is a c -colored matrix, and its dimensions are given by the dimensions of the minimal bounding rectangle of S . The elements of A having value 0 can be considered as the background of the object.

For each $1 \leq i \leq m$, let

$$h_i^{x_k} = |\{a_{i,j} : a_{i,j} = x_k\}| \text{ with } 1 \leq j \leq n, \text{ and } x_k \in \Sigma,$$

and analogously, for each $1 \leq j \leq n$, let

$$v_j^{x_k} = |\{a_{i,j} : a_{i,j} = x_k\}| \text{ with } 1 \leq i \leq m, \text{ and } x_k \in \Sigma.$$

We define

$$H = ((h_1^{x_1}, \dots, h_1^{x_c}), \dots, (h_m^{x_1}, \dots, h_m^{x_c})) \text{ and} \\ V = ((v_1^{x_1}, \dots, v_1^{x_c}), \dots, (v_n^{x_1}, \dots, v_n^{x_c}))$$

as the multi-vectors of *horizontal* and *vertical projections* of A , respectively. The matrix A is said to be *consistent* with H and V , and by extension, H is said to be consistent with V . Obviously, when we deal with homogeneous objects, we have $|\Sigma| = 1$, and H and V are integer vectors.

These simple few notions allow us to define the general reconstruction problem:

c-color

Instance: two multi-vectors

$$H = ((h_1^{x_1}, \dots, h_1^{x_c}), \dots, (h_m^{x_1}, \dots, h_m^{x_c})) \text{ and} \\ V = ((v_1^{x_1}, \dots, v_1^{x_c}), \dots, (v_n^{x_1}, \dots, v_n^{x_c})).$$

Task: reconstruct a $m \times n$ c -colored matrix A having H and V as horizontal and vertical projections, if it exists, otherwise give FAILURE.

As a matter of fact, we observe that, by definition, every consistent instance $I = (H, V)$ of c -color, must satisfy the following simple properties: for each $x \in \Sigma$, $1 \leq i \leq m$ and $1 \leq j \leq n$

- (i) $0 \leq h_i^x \leq n$;
- (ii) $0 \leq v_j^x \leq m$;
- (iii) $\sum_{i=1}^m h_i^x = \sum_{j=1}^n v_j^x$.

From now on we consider only instances of c -color that respect these three conditions, and such that the matrices obtained as solutions, if any, have dimension $m \times n$.

On the 1-Color Problem

Now we will start by considering the case of the reconstruction of a single homogeneous object, hence we will deal with matrices having two different values, which means that $\Sigma \cup \{0\}$, with $\Sigma = \{a\}$. This problem has been already extensively studied by Ryser in [10]. In the following we point out some properties common to all couples (H, V) of consistent instances, which will be useful in the rest of the paper. In such a context the superscript a of the elements of the projections H and V of a 1-colored matrix will be omitted, since no ambiguities can occur.

Theorem 1. *Let H be an integer vector. H is consistent with all the integer vectors V which satisfy conditions (ii), with $m = |H|$, and (iii), if and only if all the elements of H belong to a set $\{k, k - 1\}$, with $k > 0$.*

Proof. (\Leftarrow) we will prove that, in this case, the application of the greedy algorithm defined by Ryser always provides a valid solution. We proceed by induction on the number of columns of the matrix A :

Base. If the matrix is formed by a single column and the projections respect our assumptions, the result trivially holds.

Inductive step. If the matrix A has more than one column, then set v_1 elements in the first column in priority in the lines where $h_i = k$. This operation is possible as for (iii) it stands that $|h_i : h_i > 0| \geq v_1$.

At this point the matrix formed by the columns $2, \dots, m$ of A and with projections H' and v_2, \dots, v_m , where H' contains the projections of H minus the projections of column 1, can be correctly filled by inductive hypothesis, since its values belong to the set $\{k, k - 1\}$ or $\{k - 1, k - 2\}$.

(\Rightarrow) let us proceed by contradiction assuming that there exists a vector H , which is compatible with each vector V satisfying (ii) and (iii), and having two elements that differ at least by two. Let us consider the array $\vec{V} = \{n, \dots, n, j, 0, \dots, 0\}$, that satisfies (iii). Moreover it is easy to verify that any matrix having \vec{V} as vector of vertical projections must have any couple of horizontal projections differing of at most by 1, leading to a contradiction. \square

3 Two Useful Procedures

In this paragraph we show two procedures that will be used in the main algorithm for solving particular instances of the 2-color problem. The first one, called *Insertion*, acts on a 2-colored matrix, and changes the positions of some elements inside it. The second, called *Balance* splits an integer vector into two parts, maintaining balanced the sums of their elements.

3.1 Procedure 1: *Insertion*

Let A be a 2-colored matrix of dimension $m \times n$ on the alphabet $\Sigma = \{a, b\} \cup \{0\}$. The matrix A is said to be *c-sparse* if each of its rows contains only elements 0

except for at most c positions, which are filled with elements a , with $0 < c \leq n$. The procedure *Insertion* we are going to define performs the following:

Task. *given a set R of at most $\min\{m, n - c\}$ rows of a c -sparse matrix A , replace, inside each row of R , an element 0 with an element b so that there exists at most one element b for each column of A .*

Procedure. *Insertion*

```

for each  $i \in R$  do
  set  $ok = false$ 
  for each  $j$  such as  $1 \leq j \leq n$  do
    if  $a_{i,j} = 0$  and no element  $b$  is present in column  $j$  then
      set  $a_{i,j} = b$ 
      set  $ok = true$ 
      exit  $j$  loop
    end if
  next  $j$ 
  if  $ok = false$  then give FAILURE;
next  $i$ 
give matrix  $A$  as OUTPUT.

```

Theorem 2. *The procedure Insertion having as input a set R of rows of a c -sparse 2-colored matrix M never gives FAILURE.*

The proof directly follows from the observation that, that if a row i contains no elements 0 to be changed into a b , then at least $n - c$ elements b have already been placed in the rows of R , and this is a contradiction.

Lemma 1. *The procedure Insertion acts in $O(mn)$ computational time.*

The proof is immediate, since the elements of R are $O(m)$, and for each of them, at most $O(n)$ elements of A are scanned.

3.2 Procedure 2: Balance

Now, let us consider an integer vector $S = (s_1, \dots, s_d)$, and two integers min and Max , such that $0 < min \leq s_k \leq Max$, with $k \in 1, \dots, d$.

We define the procedure *Balance* which performs the following:

Task. *determine a rearrangement \tilde{S} of the elements of S such that*

$$0 \leq \sum_{k=1}^{d'} \tilde{s}_k - \sum_{k=d'+1}^d \tilde{s}_k \leq Max, \quad \text{with } d' = \lceil \frac{d}{2} \rceil. \tag{1}$$

Procedure. *Balance*

Step 1. Let \hat{S} be the vector S increasingly ordered, and set $d' = \lceil \frac{d}{2} \rceil$;

Step 2. for each $1 \leq k \leq d'$, set $\tilde{s}_k = \hat{s}_{2k-1}$;

Step 3. for each $d' + 1 \leq k \leq d$, set $\tilde{s}_k = \hat{s}_{2(k-d')}$;

Step 4. give vector \tilde{S} as OUTPUT.

It is easy to see that the vector \tilde{S} satisfies the required task, in fact, if d is odd it holds

$$\begin{aligned} 0 \leq \sum_{k=1}^{d'} \tilde{s}_k - \sum_{k=d'+1}^d \tilde{s}_k &= \sum_{k=1}^{(d-1)/2} (\hat{s}_{2k-1} - \hat{s}_{2k}) + \hat{s}_d \leq \sum_{k=1}^{(d-1)/2} (\hat{s}_{2k-1} - \hat{s}_{2k+1}) + \hat{s}_d = \\ &= \hat{s}_1 - \hat{s}_d + \hat{s}_d = \hat{s}_1 \leq Max, \end{aligned}$$

otherwise, if d is even we can similarly check that

$$0 \leq \sum_{k=1}^{d'} \tilde{s}_k - \sum_{k=d'+1}^d \tilde{s}_k \leq Max - min.$$

Lemma 2. *The procedure Balance acts in $O(d + Max)$ computational time.*

The result is immediate, since $O(d + Max)$ is required to order the vector S using counting sort, and $O(d)$ to scan S and create \tilde{S} .

Finally, we define $\sigma_{Balance}$ the permutation of the elements of S performed by *Balance* to obtain \tilde{S} .

4 Solving Some Instances of the 2-Color Problem

In this paragraph we define a subset of instances of 2-color, and then we present a polynomial algorithm to solve the problem. These instances are defined by some constraints on their values: in a word, some elements of H and V are constant, while the remaining elements are smaller than a given number which is somehow proportional to the minimum between m and n , the dimensions of the reconstructed matrix. Our approach to the problem can be naturally generalized to the whole set of instances of 2-color, unfortunately without being able to polynomially bound the computational complexity. So, let us define the following problem:

2c-restricted

Instance. two consistent vectors of projections

$$H = ((h_1^a, h_1^b), \dots, (h_m^a, h_m^b)) \text{ and } V = ((v_1^a, v_1^b), \dots, (v_n^a, v_n^b))$$

such that

- (1) all the elements of H and V are bounded by an integer M such that
 - $M = \lfloor \min\{m, n\}/3 \rfloor$ if m and n are both even,
 - $M = \lfloor \min\{m, n\}/4 \rfloor$ otherwise;
- (2) there are two positive integers c_1 and c_2 such that for all $1 \leq i \leq m$, $1 \leq j \leq n$ we have $h_i^a = c_1$ and $v_j^b = c_2$.

Task. reconstruct a $m \times n$ 2-colored matrix A having H and V as horizontal and vertical projections, if it exists, otherwise give FAILURE.

The imposed constraints on the projections of $2c - restricted$ allow a fast reconstruction algorithm. The algorithm is mainly based on the fact that, for each of its instances, there exists a solution which can be divided into four disjoint parts, say NW , NE , SW , and SE , each of them containing only one color, say the dominant one, except for a fixed number of elements.

More precisely, given an instance I we split the non constant part H^b of H [resp. V^a of V], into two parts which differ at most by M , using the procedure *Balance*. Successively we arrange these parts into four instances of 1-color which are used to set the dominant color in the four parts NW , NE , SW , and SE . At this step it is not guaranteed that all these instances satisfy (iii), so we have to slightly modify them by setting aside some elements, at most M for each one, to make them consistent and, furthermore, solvable by Lemma 1.

Once we have reconstructed the four submatrices NW , NE , SW , and SE , we obtain a partial solution which turns into the final one, which we call A , after placing the previously removed elements by procedure *Insertion*. Theorem 2 assures that this process successfully ends.

Now we define the four submatrices NW , NE , SW , and SE of a $m \times n$ matrix A

- NW is the submatrix formed by the intersection of the first $\lfloor m/2 \rfloor$ rows with the first $\lceil n/2 \rceil$ columns;
- NE is the submatrix formed by the intersection of the first $\lfloor m/2 \rfloor$ rows with the last $\lfloor n/2 \rfloor$ columns;
- SW is the submatrix formed by the intersection of the last $\lceil m/2 \rceil$ rows with the first $\lceil n/2 \rceil$ columns;
- SE is the submatrix formed by the intersection of the last $\lceil m/2 \rceil$ rows with the last $\lfloor n/2 \rfloor$ columns.

So, let us define the algorithm.

Algorithm. *Reconstruct*

Step 1. let $H^b = (h_1^b, \dots, h_m^b)$ [resp. $V^a = (v_1^a, \dots, v_n^a)$].

Apply procedure *Balance* to H^b [resp. V^a], and let \tilde{H}^b [resp. \tilde{V}^a] be its output. In this way the sums of the elements in the first and second half of \tilde{H}^b [resp. \tilde{V}^a] have a bounded difference.

Let $\sigma_{Balance}^b$ [resp. $\sigma_{Balance}^a$] be one of the permutations of the elements of H^b [resp. V^a] that leads to \tilde{H}^b [resp. \tilde{V}^a].

Compute the vector $\tilde{H} = ((c_1, \tilde{h}_1^b), \dots, (c_1, \tilde{h}_m^b))$ [resp. $\tilde{V} = ((\tilde{v}_1^a, c_2), \dots, (\tilde{v}_n^a, c_2))$]. Let $m' = \lceil m/2 \rceil$ and $n' = \lceil n/2 \rceil$.

Step 2.

now we compute and store in D_{SW}^a or D_{NE}^a the (number of) elements a that lie in the submatrices SW or NE of the final solution, where the dominant color is b .

The symmetrical computation of D_{SE}^b or D_{NW}^b is carried with respect to the color b and the submatrices SE and NW .

These numbers are computed in such a manner to allow the instances of the 1-color problems, which partially reconstruct the four submatrices of the final solution, to satisfy condition (iii).

if $\sum_{k=1}^{n'} \tilde{v}_k^a > m' c_1$

then set $D_{SW}^a = \sum_{k=1}^{n'} \tilde{v}_k^a - m' c_1$, and $D_{NE}^a = 0$

else set $D_{NE}^a = m' c_1 - \sum_{k=1}^{n'} \tilde{v}_k^a$, and $D_{SW}^a = 0$.

If $\sum_{k=1}^{m'} \tilde{h}_k^b > (n-n') c_2$ then set $D_{NW}^b = \sum_{k=1}^{m'} \tilde{h}_k^b - (n-n') c_2$, and $D_{SE}^b = 0$

else set $D_{SE}^b = (n-n') c_2 - \sum_{k=1}^{m'} \tilde{h}_k^b$, and $D_{NW}^b = 0$.

W.l.g. assume that $D_{SW}^a = D_{SE}^b = 0$, and, as a consequence, $D_{NW}^b \geq 0$ and $D_{NE}^a \geq 0$.

Step 3. (where the submatrix NW is reconstructed)

Step 3.1: create the instance $I_{NW}^a = (H^{NW}, V^{NW})$ of 1-color such that

- $h_i^{NW} = c_1 - 1$, with $1 \leq i \leq D_{NE}^a$;
- $h_i^{NW} = c_1$, with $D_{NE}^a < i \leq m'$;
- $v_j^{NW} = \tilde{v}_j^a$ with $1 \leq j \leq n'$.

Step 3.2: solve the instance I_{NW}^a of 1-color (by means of Ryser's algorithm, as described in [10]) and insert the values a in the correspondent submatrix NW of A .

Step 3.3: run the procedure *Insertion* with $R = \{1, \dots, D_{NW}^b\}$ (notice that the submatrix NW is M -sparse). Store the indexes of the columns where the D_{NW}^b elements b are placed in R_{NW} .

Step 4. (where the submatrix NE is reconstructed);

Step 4.1: create the instance $I_{NE}^b = (H^{NE}, V^{NE})$ of 1-color such that

- $h_i^{NE} = \tilde{h}_i^b - 1$, with $1 \leq i \leq D_{NW}^b$;
- $h_i^{NE} = \tilde{h}_i^b$ with $D_{NW}^b < j \leq m'$;
- $v_j^{NE} = c_2$ with $1 \leq j \leq n - n'$.

Step 4.2: solve the instance I_{NE}^b of 1-color and insert the values b in the correspondent submatrix NE of A .

Step 4.3: run the procedure *Insertion* with $R = \{1, \dots, D_{NE}^a\}$. Store the indexes of the columns where the D_{NE}^a elements a are placed in R_{NE} .

Step 5,6: (where the submatrix SW [resp. SE] is reconstructed) now act similarly as in Step 3.2 [resp. Step 4.2] to reconstruct the two remaining submatrices, with the following remark: in Step 5.1 [resp. Step 6.1], the two projections of the instance $I_{SW} = (H_{SW}, V_{SW})$ [resp. $I_{SE} = (H_{SE}, V_{SE})$], can be computed by simply subtracting from \tilde{H} and \tilde{V} the elements a [resp. b] stored in R_{NW} [resp. R_{NE}], i.e. those already placed in NW [resp. NE].

Step 7: permute the rows of A according to $(\sigma_{Balance}^b)^{-1}$, then permute the columns of A according to $(\sigma_{Balance}^a)^{-1}$, and finally return the updated matrix A as OUTPUT.

4.1 An Example of the Execution of the Algorithm

To clarify the steps of the reconstruction procedure we furnish the following example:

Let us represent a $m \times n$, 2-colored matrix by means of a $m \times n$ set of cells on a squared surface having two different colors, say red and blue; red, blue and void cells correspond to the elements a , b , and 0 of the matrix, respectively. Let us consider the instance $I = (H, V)$ such that

$$\begin{aligned}
 H &= ((2, 3), (2, 3), (2, 4), (2, 3), (2, 1), (2, 2), (2, 4), (2, 4), (2, 1), (2, 4), (2, 4), (2, 3)) \\
 V &= ((2, 3), (3, 3), (1, 3), (3, 3), (3, 3), (1, 3), (2, 3), (2, 3), (3, 3), (2, 3), (1, 3), (1, 3))
 \end{aligned}$$

It is easy to check that H and V respect conditions (i), (ii) and (iii) for each color, and that I is an instance of $2c - restricted$. Since H^{blue} and V^{red} are already balanced, i.e. they satisfy equation (1), there is no need to perform procedure *Balance*, and so for sake of simplicity we skip Step 1 and consider $H = \tilde{H}$, and $V = \tilde{V}$. The steps of the procedure *Reconstruction* are sketched in Fig. 1, starting from a), where the instance I is depicted; furthermore, we have $m' = n' = 6$. In Step 2 we compute $D_{SW}^{red} = 1$, and $D_{NE}^{red} = 0$, then we compute $D_{NW}^{blue} = 0$ $D_{SE}^{blue} = 2$. These numbers represent the red and blue elements that have to be added to each of the four submatrices.

Since $D_{SW}^{red} \neq 0$ and $D_{SE}^{blue} \neq 0$, we start the reconstruction process from one of these two submatrices, say the SW submatrix, then we proceed with the other. So, two blue cells are moved from the submatrix SW to SE , and we create the instance of 1-color $I_{SW}^{blue} = (H^{SW}, V^{SW})$, with $H^{SW} = (3, 3, 1, 4, 4, 3)$, and $V^{SW} = (3, 3, 3, 3, 3, 3)$. Finally we solve I_{SW}^{blue} , as in Fig. 1, b), and we add, by means of the procedure *Insertion*, one red cell in the first free position, as required by D_{SW}^{red} ; this concludes Step 3.

Similarly to Step 3, we perform Step 4, where we reconstruct the submatrix SE , and, successively we add two blue cells as required by D_{SE}^{blue} , as in Fig. 1, d).

Now, we perform Steps 5 and 6 after creating the instances $I^{NW} = (H^{NW}, V^{NW})$ and $I^{NE} = (H^{NE}, V^{NE})$, respectively, where $H^{NW} = (2, 2, 2, 2, 2, 2)$, $V^{NW} = (1, 3, 1, 3, 3, 1)$, $H^{NE} = (3, 3, 4, 3, 1, 2)$, and $V^{NE} = (2, 2, 3, 3, 3, 3)$.

The reconstruction of the two submatrices NW and NE , together with the whole solution of I , are depicted in Fig. 1, e).

4.2 Correctness

Now we prove that the algorithm *Reconstruct* having as input a generic instance I of $2c - restricted$ always leads to a solution by showing that:

- i) each instance of 1-color related to the four submatrices NW , NE , SW and SE admits a solution;
- ii) each reconstructed submatrix satisfies the conditions required by the procedure *Insertion*.

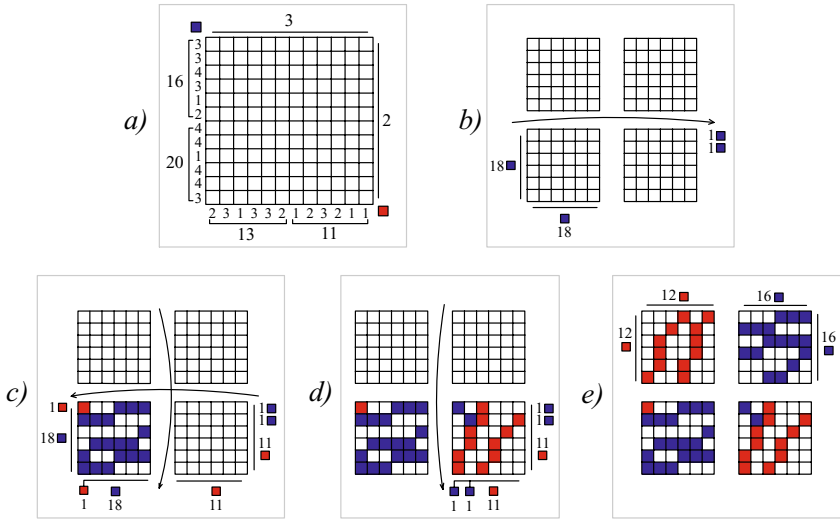


Fig. 1. The steps of *Reconstruction* on the instance I

Assertion i) is straightforward since each instance of 1-color related to the submatrices NW , NE , SW and SE created in Steps 3, 4, 5 and 6 of *Reconstruction* has at least one among the horizontal or the vertical projections whose values are in $\{k - 1, k\}$, where $k = c_1$ or $k = c_2$. Further, the number of cells in the horizontal and vertical projections are consistent thanks to the use of the variables D . So, the hypothesis of Theorem 1 are satisfied (with respect to each instance), and consequently each instance admits a solution.

Assertion ii) follows after observing that

- each reconstructed submatrix is M -sparse;
- if m or n are odd, the values D_{NW}^b , D_{NE}^a , D_{SW}^a , and D_{SE}^b are positive and each of them bounded by $\lfloor n/2 \rfloor - M$. As an example, let us consider D_{NW}^b : its value is either zero or it holds $D_{NW}^b \leq \lceil (M - 1)/2 + c_2/2 \rceil$ (remind that the application of the procedure *Balance* to H^b produces $\sum_{k=1}^{m'} \tilde{h}_k^b - \sum_{k=m'+1}^m \tilde{h}_k^b \leq M - 1$). Let $min = \min\{m, n\}$, and observe that $c_2 \leq min/4$. Consequently

$$D_{NW}^b \leq \lceil (M - 1)/2 + c_2/2 \rceil \leq (min - 4)/8 + min/8 + 1 = min/4 + 1/2$$

and assuming that $min \geq 3$, we finally reach

$$D_{NW}^b \leq min/4 + 1/2 \leq min/2 \leq \lfloor n/2 \rfloor - M.$$

The same argument can be used to prove the analogous result for D_{NE}^a , D_{SW}^a , and D_{SE}^b .

With similar arguments, we can prove that if the dimensions m and n of the solution matrix are both even, the maximum value M of the elements of H^b and V^a becomes $\lfloor \min\{m, n\}/3 \rfloor$ as, in such a setting, the four values D_{NW}^b , D_{NE}^a , D_{SW}^a , and D_{SE}^b are upper bounded by $\lceil (M-1)/2 \rceil$.

- each submatrix has more than D_{NW}^b [resp. D_{NE}^a , D_{SW}^a , and D_{SE}^b] rows.

So, all the hypotheses to apply the procedure *Insertion* to each reconstructed submatrix NW , NE , SW and SE are satisfied, as desired.

4.3 Complexity

Theorem 3. *The algorithm Reconstruct finds a solution to a generic instance I of $2c$ – restricted in $O(mn)$ computational time.*

Proof We will analyze the complexity of each step of *Reconstruct*

Step 1: The procedure *Balance* acts in $O(m+n)$ (see Lemma 2) computational time, and the same for the computation of the vectors \tilde{H} and \tilde{V} .

Step 2: All the computations are clearly performed in $O(n+m)$.

Steps 3 – 6: Each step takes $O(mn)$ computational time, since Ryser’s algorithm for solving the 1-color problem takes $O(mn)$, and the same holds for procedure *Insertion* (see Lemma 1).

Steps 7: The output matrix A is computed in $O(nm)$ computational time.

So, *Reconstruct* takes $O(mn)$ computational time as desired. \square

5 Further Research

In this paper we show a class of instances of the 2-color problem which can be solved in linear (computational) time with respect to the dimensions of the solution matrix. The algorithm we propose is mainly based on the existence of a family of 2-colored matrices which, after an appropriate organization of its rows and columns, can be divided into disjoint submatrices each with only one color, so that we are allowed to use, for each of these cases, the standard Ryser’s reconstruction algorithm for 1-colored matrices.

The proposed result gives birth to a new way of challenging subclasses of instances of the 2 – color problem, searching for all those solutions where the two colors are somehow disjoint in each row and column. As a consequence, the proposed algorithm *Reconstruct* seems to allow a further series of refinements by using techniques that decompose each instance into smaller ones in the *divide et impera* fashion, and which may lead to the definition of the non polynomial set of instances of 2 – color, to the limit; obviously the hardest task will be to keep polynomially bounded the amount of time needed for the whole process.

Our result suggests that the hard part of 2-color has to contain only those instances having non uniform projections for both colors, and it furnishes hints and inspiration for future researches.

References

1. Brocchi, S., Frosini, A., Picouleau, C.: Reconstruction of binary matrices under fixed size neighborhood constraints. *Theoretical Computer Science* 406, 1-2, 43-54 (2008)
2. Chrobak, M., Durr, C.: Reconstructing polyatomic structures from discrete X-rays: NP-completeness proof for three atoms. *Theoretical computer science* 259, 81-98 (2001)
3. Costa, M.C., de Werra, D., Picouleau, C., Schindl, D.: A solvable case of image reconstruction in discrete tomography. *Discrete Applied Mathematics* 148(3), 240-245 (2005)
4. Costa, M.C., de Werra, D., Picouleau, C.: Using graphs for some discrete tomography problems, CEDRIC report (2004), <http://cedric.cnam.fr/>
5. Dürr, C., Guíñez, F., Matamala, M.: Reconstructing 3-colored grids from horizontal and vertical projections is NP-hard. In: Fiat, A., Sanders, P. (eds.) *ESA 2009*. LNCS, vol. 5757, pp. 776-788. Springer, Heidelberg (2009)
6. Gardner, R.J., Gritzmann, P., Pranenberg, D.: On the computational complexity of reconstructing lattice sets from their X-rays. *Discrete Mathematics* 202(1-3), 45-71 (1999)
7. Gardner, R.J., Gritzmann, P., Pranenberg, D.: On the computational complexity of determining polyatomic structures by X-rays. *Theoretical computer science* 233, 91-106 (2000)
8. Herman, G., Kuba, A.: *Discrete Tomography: Foundations, Algorithms and Applications*. Birkhauser, Basel (1999)
9. Kisielowski, C., Schwander, P., Baumann, F.H., Seibt, M., Kim, Y., Ourmazd, A.: An approach to quantitative high-resolution transmission electron microscopy of crystalline materials. *Ultramicroscopy* 58, 131-155 (1995)
10. Ryser, H.J.: Combinatorial properties of matrices of zeros and ones. *Canad. J. Math.* 9, 371-377 (1957)
11. Schwander, P., Kisielowski, C., Seibt, M., Baumann, F.H., Kim, Y., Ourmazd, A.: Mapping projected potential interfacial roughness, and composition in general crystalline solids by quantitative transmission electron microscopy. *Phys. Rev. Lett.* 71, 4150-4153 (1993)