RECONSTRUCTION OF BINARY IMAGES FROM THEIR HORIZONTAL AND DIAGONAL PROJECTIONS[1]

Hasmik Sahakyan, Vladimir Ryazanov, Ani Margaryan

Abstract: *In this paper we consider the problem of reconstruction of binary images from their horizontal and diagonal projections. A large number of publications is devoted to analysis of straight horizontal and/or vertical projections. But reconstruction by the use of incorporated diagonal projections is of a principal difference.*

Keywords: *Discrete tomography, inverse problem, horizontal and diagonal projections.*

ITHEA Keywords: *F.2.2 Nonnumerical Algorithms and Problems: G.2.1 Combinatorics*

Introduction

Discrete Tomography aims at recovering of discrete sets from their projections composed along the given set of directions. *Discrete sets* or *lattice sets* are finite subsets of vertices of the integer lattice Z^d. The *lattice directions* are those, represented by any nonzero vectors of Z^d. A line l in d-dimensional Euclidean space is a *lattice line* if it is parallel to a lattice direction and passes through at least one point in Z^d. A *projection* of a lattice set in a lattice direction u is a function giving the number of its points on each line parallel to the direction u ([HermanKuba, 1999]).

Given a set of lattice directions $\{u_1, u_2, \cdots, u_l\}$ and projections along those directions: $F_1, F_2, \cdots, F_l$, we consider <u>Consistency, uniqueness and reconstruction problems in Discrete Tomography</u>.

Consistency: Does there exist a discrete set $T \in Z^d$ with given projections $F_1, F_2, \cdots, F_l$ in lattice directions $\{u_1, u_2, \cdots, u_l\}$?

Uniqueness: Is a discrete set $T \in Z^d$ uniquely determined by the given projections $F_1, F_2, \cdots, F_l$?

Reconstruction: Construct a discrete set $T \in Z^d$ from its projections $F_1, F_2, \cdots, F_l$.

[1] Partially supported by grants № 18RF-144, and № 18T-1B407 of the Science Committee of the Ministry of Education and Science of Armenia

If we are given dimension count $d \geq 2$, and $l \geq 3$ non-parallel projections in the integer lattice Z^d, then the consistency, reconstruction and uniqueness problems are NP-hard ([GardGrizmPran, 1999]).

Discrete sets of vertices in Z^2 can be considered also as binary images or binary matrices. In the simplest case of horizontal and vertical projections the existence and construction problems of discrete sets by their projections is considered and solved in 1957 in terms of binary matrices ([Ryser 1957], [Gale, 1957]). But in this case, the number of solutions can be exponentially large ([Lungo, 1994).

A commonly used idea to reduce the set of possible solutions is the use of an a priori information/property of the set to be recovered, if such property exists. Two commonly used in this context geometrical properties are *convexity* and c*onnectivity*. The existence problem of a binary matrix is NP-complete for horizontal or vertical convex, as well as for horizontal or vertical convex and connected matrices ([BarcDLungoNivatPinz, 1996]). NP-completeness of the case of 4-connected matrices, as well as of horizontal and vertical convex matrices is proved in [Woeginger, 2001]. The case of horizontal and vertical convex and connected matrices is solvable in polynomial time ([DurrChobrak, 1999]; [Kuba 1999]).

Another idea to reduce the size of set of possible solutions is to take further projections along different lattice directions. Reconstruction problem for the case of horizontal, vertical and diagonal projections is considered and NP-completeness is proved in [GardGrizmPran, 1999]). For some cases (horizontal, vertical, diagonal connected and convex matrices) the problem is solvable in polynomial time ([BarcBrunDeLunNivat, 2001]).

The uniqueness and reconstruction problems for the case of diagonal and anti-diagonal projections are considered in ([SrivansVerma, 2013]).

In this paper, we consider discrete sets in Z^2 and study the reconstruction problem with respect to two directions: one horizontal and one diagonal. In Section 2 we derive necessary conditions for existence of a binary matrix with the given horizontal and diagonal projections. Section 3 introduces a heuristic algorithm of reconstruction of binary matrices from given horizontal and diagonal projections. Experimental results are given in Section 4.

2. Necessary conditions for existence of a binary matrix with given row and diagonal sums

Consider a binary matrix $A = \{a_{i,j}\}$ with n rows and n columns. Let $R = (r_1, \cdots, r_n)$ and $D = (d_1, \cdots, d_{2n-1})$ denote the row sum and the diagonal sum vectors of A respectively, where:

$r_i = \sum_{j=1}^{n} a_{i,j}$, $i = 1, \cdots, n$, and $d_k = \sum_{i+j=k+1} a_{i,j}$, $k = 1, \cdots, 2n-1$. An example is given in Figure 1.

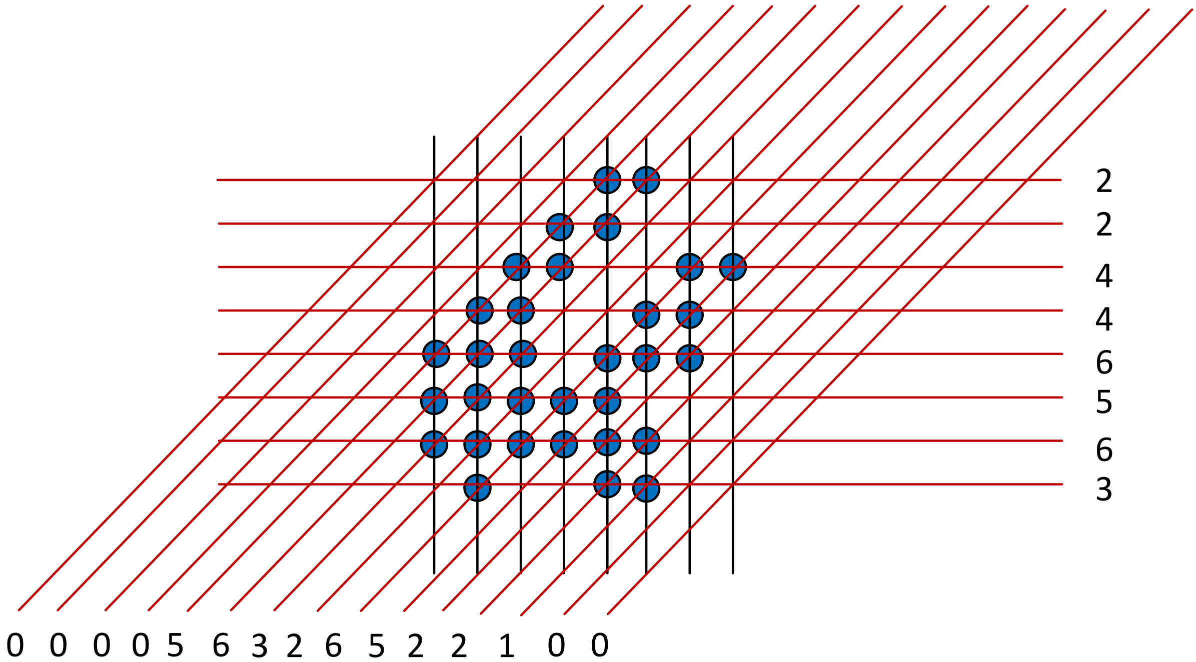


Figure 1. 8x8 size binary image, where $R = (2,2,4,4,6,5,6,3)$ and $D = (0,0,0,0,5,6,3,2,6,5,2,2,1,0,0)$ are horizontal and diagonal projections (row and diagonal sums of the corresponding binary matrix).

Notice that the row and diagonal sum vectors of the matrix must satisfy the following conditions:

$$\sum_{k=1}^{2n-1} d_k = \sum_{i=1}^{n} r_i ,$$

$$0 < r_i \le n, \ \ 1 \le i \le n , \tag{1}$$

$$0 \le d_k \le m_k , \ \ 1 \le k \le 2n-1,$$

where $m_k = \begin{cases} k, & if \ \ 1 \le k \le n \\ n-(k-n), & if \ \ n+1 \le k \le 2n-1 \end{cases}.$

Let $R = (r_1, r_2, \cdots, r_n)$ and $D = (d_1, \cdots, d_{2n-1})$ be non-negative integer vectors. Henceforth we will assume that R and D satisfy the conditions (1) (in this case they are called compatible vectors) and the components of R are arranged in decreasing order: $r_1 \ge r_2 \ge \cdots \ge r_n$. D^{i_j} will denote the following part of the vector $D = (d_1, \cdots, d_{2n-1})$: $D^{i\_j} = (d_i, \cdots, d_j)$, where $1 \le i < j \le 2n-1$.

We define the *maximal matrix* and two its fragments, and introduce so called “majorization” conditions for the fragments and for the whole matrix, which, as one can easily check, are necessary conditions for existing of binary matrix with given horizontal and diagonal projections.

Maximal matrix

Let us compose the binary matrix $\bar{A} = \{\bar{a}_{i,j}\}$ of size $n \times n$ whose rows have the following structure:

$\overbrace{1,1,\cdots,1}^{r_i}\overbrace{0,0,\cdots,0}^{n-r_i}$, for $1 \le i \le n$.

$\bar{A}$ is called *maximal matrix* and is unique for given $R = (r_1, r_2, \cdots, r_n)$ ([Ryser, 1957]).

Let $\bar{D} = \{\bar{d}_1, \cdots, \bar{d}_{2n-1}\}$ denote the diagonal sum vector of $\bar{A}$.

$F1$ - Fragment 1.

For every i, $1 \le i \le n$ let $F1_i$ denote the left part of $\bar{A}$ bounded by the i-th diagonal line as shown in Figure 2. $F1_i$ has i rows and i columns. $S^{F1_i} = (s_1^{F1_i}, s_2^{F1_i}, \cdots, s_i^{F1_i})$ denotes the column sum vector of $F1^i$, where $s_j^{F1_i} = \sum_{k=1}^{i-(j-1)} \bar{a}_{k,j}$, $1 \le j \le i$.

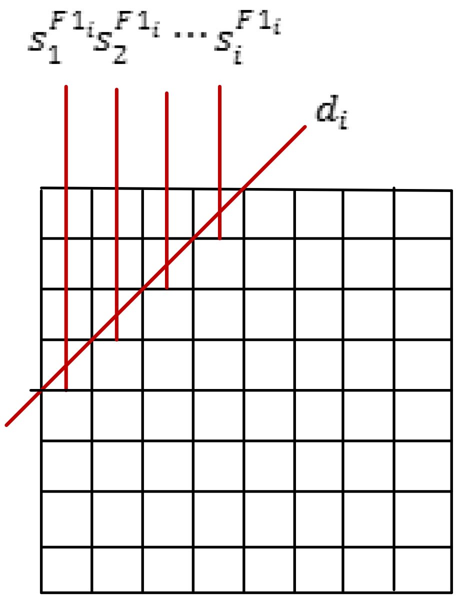


Figure 2. An example of Fragment $F1_i$.

M1 - Majorization condition for the fragment $F1$.

For a given i, $1 \le i \le n$ we say that the column sum S^{F1_i} of the fragment $F1_i$ of the maximal matrix $\bar{A}$ *majorizes* $D^{1\text{-}i}$ (and use the following notation: $S^{F1_i} \succcurlyeq^1 D^{1\text{-}i}$) if for each $1 \le j \le i$ the following conditions hold:

$d_j \le s_1^{F1_i}$,

$d_j + d_{j-1} \le s_1^{F1_i} + s_2^{F1_i}$,

…

$d_j + d_{j-1} + \cdots + d_1 \le s_1^{F1_i} + s_2^{F1_i} + \cdots + s_j^{F1_i}$.

$F2$ - Fragment 2.

For every $i, n \le i \le 2n-1$ let $F2_i$ denote the left part of $\bar{A}$, bounded by the i-th anti-diagonal line as shown in the Figure 3, where components of anti-diagonal sum vector are defined as:

$$d_k^a = \begin{cases} \sum_{i-j=n-k} a_{i,j}\,, & if \;\; 1 \le k \le n \\ \sum_{j-i=k-n} a_{i,j}\,, & if \;\; n+1 \le k \le 2n-1 \end{cases}.$$

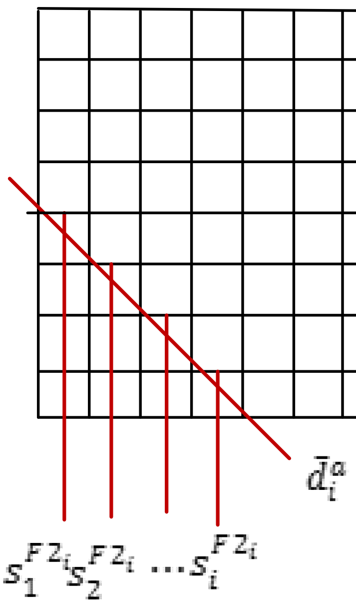


Figure 3. An example of Fragment $F2_i$

$F2_i$ has $(2n-i)$ rows and $(2n-i)$ columns. $S^{F2_i} = (s_1^{F2_i}, s_2^{F2_i}, \cdots, s_{2n-i}^{F2i})$ denotes the column sum vector of $F2_i$ where $s_j^{F2_i} = \sum_{k=i-n+j}^{n} \bar{a}_{k,j}$, $1 \le j \le 2n-i$.

M2 - Majorization condition for the fragment $F2$.

For a given $i, n \leq i \leq 2n-1$ we say that S^{F2_i} *majorizes* D^{i_2n-1} (and use the following notation: $S^{F2_i} \succcurlyeq^2 D^{i\_2n-1}$) if for each $i \leq j \leq 2n-1$ the following conditions hold:

$d_j \leq s_1^{F2_i}$,

$d_j + d_{j+1} \leq s_1^{F2_i} + s_2^{F2_i}$,

...

$d_j + d_{j+1} + \cdots + d_{2n-1} \leq s_1^{F2_i} + s_2^{F2_i} + \cdots + s_{2n-j}^{F2_i}$.

M3 - Majorization condition for the matrix $\bar{A}$

We say that the diagonal sum $\bar{D}$ of the maximal matrix $\bar{A}$ *majorizes* the diagonal sum D (and use the following notation: $\bar{D} \succcurlyeq^3 D$) if the following conditions hold:

$\bar{d}_1 \geq d_1$

$\bar{d}_1 + \bar{d}_2 \geq d_1 + d_2$

...

$\bar{d}_1 + \bar{d}_2 + \cdots + \bar{d}_{2n-2} \geq d_1 + d_2 + \cdots + d_{2n-2}$

$\bar{d}_1 + \bar{d}_2 + \cdots + \bar{d}_{2n-1} = d_1 + d_2 + \cdots + d_{2n-1}$

It is easy to check that if there exists a binary matrix of size $n \times n$ with given row sum vector $R = (r_1, \cdots, r_n)$ and diagonal sum vector $D = (d_1, \cdots, d_{2n-1})$, then the conditions $M1, M2, M3$ hold:

$S^{F1_i} \succcurlyeq^1 D^{1\_i}$ for $i = 1, \cdots, n$,

$S^{F2_i} \succcurlyeq^2 D^{i\_2n-1}$ for $i = n, \cdots, 2n-1$,

$\bar{D} \succcurlyeq^3 D$.

3. Algorithm of reconstructing a binary matrix with given row and diagonal sums

In this section an Algorithm HD is introduced which constructs a binary matrix A with given row sum $R = (r_1, \cdots, r_n)$ and diagonal sum $D = (d_1, \dots, d_{2n-1})$ from the maximal matrix $\bar{A}$.

Construction of A proceeds diagonal by diagonal, starting from the right-bottom corner of the matrix, i.e. from the $(2n-1)$-th diagonal line. In each step k algorithm HD constructs the $(2n-k)$-th diagonal line in A by moving necessary number of 1s from the diagonal lines of the maximal matrix to the $(2n-k)$-th diagonal line.

Let $k-1$ steps be done and $(2n-1)$-th, $(2n-2)$-th, etc. $(2n-k+1)$-th diagonal lines are constructed. $\tilde{A}$ denotes updated after each step maximal matrix. For constructing current $(2n-k)$-th diagonal line, algorithm HD finds the nearest non-zero diagonal line in $\tilde{A}$ (let it be the j-th diagonal with $\tilde{d}_j$ 1s) which intersects by rows with the diagonal line under construction, and moves 1s from one diagonal line to the other (keeping 1s on the same row). The algorithm checks that conditions $M1, M2, M3$ are not violated while moving 1s, otherwise it skips the j-th diagonal line and continue with the next nearest non-zero diagonal $j' < j$, and so on.

Notice that the condition $M3$ provides that there will not be extra 1s on the diagonal line under construction before each step; and $M1$ and $M2$ provide sufficient number of 1s to be moved to the diagonal line under construction.

Algorithm HD:

Input: $R = (r_1, r_2, \cdots, r_n)$ and $D = (d_1, \cdots, d_{2n-1})$ compatible pair of vectors.

1. Construct maximal matrix $\bar{A}$; calculate $\overline{D} = \{\bar{d}_1, \cdots, \bar{d}_{2n-1}\}$; $\tilde{A} := \bar{A}$; $\widetilde{D} := \overline{D}$;
2. if any of conditions $M1, M2, M3$ are violated then return (Algorithm failure);
3. for $(k = 2n-1; k > 0; k--)$

   $\{j := k;$

   while $(\tilde{d}_k \neq d_k)$

   {

   if (<u>Selection of diagonal line</u> j' in $\tilde{A}$ is possible)

   then current_diagonal:=1

   else return (Algorithm failure);

   while (current_diagonal = 1)

     {

     if (<u>Selection of 1 on diagonal line</u> j' is possible)

     then

{

if ($M3$ is violated after replacing selected 1 with 0) return (Algorithm failure);

if ($k > n$)

if ($j' > n$)

if ($M2$ holds after replacing selected 1 with 0)

then { move the selected 1; update $\widetilde{D}$;}

else {current_diagonal:=0; $j := j - 1$;}

else

if ($M2$&$M1$ hold after replacing selected 1 with 0)

then { move the selected 1; update $\widetilde{D}$;}

else {current_diagonal:=0; $j := j - 1$;}

else

if ($M1$ holds after replacing selected 1 with 0)

then { move the selected 1; update $\widetilde{D}$;}

else {current_diagonal:=0; $j := j - 1$;}

}

else {current_diagonal:=0; $j := j - 1$;}

}

}

}

Output: matrix $\tilde{A}$.

<u>Selection of diagonal line j' in $\tilde{A}$</u> is possible: if it can be found diagonal line j', $1 \le j' < j$ in $\tilde{A}$ (nearest possible to the current j-th diagonal line is chosen) which has 1s in those rows intersecting with the k-th diagonal line.

<u>Selection of 1 on diagonal line j'</u> is possible: if it can be found 1 on the diagonal line j' (smallest index of row is chosen), which is possible to move to the k-th diagonal line.

Note. $M1, M2, M3$ conditions have been checking in each step for relevant parts of fragments.

Consider performance of the algorithm on an example: let $n = 6$, $R = (5,5,4,3,2,2)$ and $D = (0,2,2,3,3,5,4,1,0,1,0)$.

First the maximal matrix $\bar{A}$ is constructed:

$$\bar{A} = \begin{matrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{matrix}$$

$\bar{D} = (1,2,3,4,5,5,1,0,0,0,0)$ is the diagonal sum of $\bar{A}$.

$d_{11} = 0$, and hence there is nothing to reconstruct on the 11-th diagonal line. For constructing the next 10-th diagonal with $d_{10} = 1$ the nearest non-zero diagonal line is the 7-th diagonal line with $\tilde{d}_7 = 1$ and the algorithm will move the corresponding 1. Below is matrix after that step.

$$\begin{matrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{matrix}$$

The next non-zero diagonal line to be constructed is the 8-th with $d_8 = 1$; and first non-zero diagonal line from which 1-s will be moved is the 6-th with $\tilde{d}_6 = 5$. Next diagonal line for reconstruction will be the 7-th with $d_7 = 4$, and first non-zero diagonal in $\tilde{A}$ after previous step is $\tilde{d}_6 = 4$. Below are matrices after those steps.

$$\begin{matrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{matrix} \qquad \begin{matrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{matrix}$$

We will skip detailed descriptions of all steps and below is the final reconstructed matrix by Algorithm HD.

$$\tilde{A} = \begin{matrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{matrix}$$

4. Experimental results

In this section experimental results for the provided algorithm are presented.

Software system is created which implements Algorithm HD, and different experiments to check its performance are conducted for the following cases:

1. Input is a pair of random vectors;

In this case random vectors are generated, and then compatibility of the vectors, as well as necessary conditions are checked. For keeping randomness there is an option to insert matrix size and rate of each component of row and diagonal sum vectors comparative to its maximal value.

2. Input is row and diagonal sum vectors of random binary matrices.

For this purpose random matrices are generated and then row and diagonal sums are calculated. To keep randomness in generating process an option is created to insert matrix size and probability of each matrix cell (to be 1).

3. Input is row and diagonal sum vectors inserted manually.

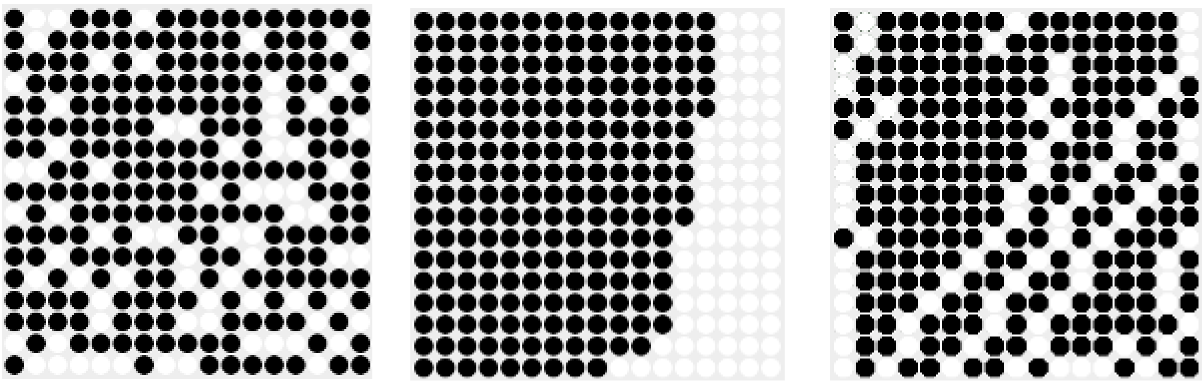
The purpose here is to check the algorithm performance for specially created test cases of row and diagonal sums.

Algorithm performance is checked for up to $50x50$ size matrixes, filled by different probabilities between 0.1 and 0.9.

<u>Experiments for the case 2.</u>

The algorithm is run for more than 1000 samples /random matrices/.

Below in Figure 5 is visualization of an example:



(a) (b) (c)

Figure 5. Stages of algorithm performance
(a) generated random matrix, (b) created maximal matrix, (c) reconstructed matrix

Algorithm has failed only for two samples: it couldn't reconstruct existing matrix from its given projections. Below is an example:

$R = (17,14,14,14,14,13,13,13,13,12,12,11,11,11,11,10,10,10,10,10)$

$D = (1,2,2,3,2,4,4,5,2,7,5,9,9,14,7,15,12,15,11,13,11,9,9,9,8,6,7,6,5,6,7,4,2,3,2,2,2,2,1)$

Experiments for the case 1.

Only 30% of generated vectors passed all conditions ($M1, M2, M3$). For most of them algorithm failed because one of the conditions get violated during some step. For small samples it was possible to check manually and make sure that there is no matrix with given row and diagonal sums.

Conclusion

An algorithm of reconstruction of binary images from their horizontal and diagonal projections is introduced; experimental results are given to measure the algorithm performances.

Bibliography

[HermanKuba, 1999] G.T. Herman and A. Kuba, editors, Discrete Tomography: Foundations, Algorithms and Applications, BirkhÄauser, Boston, 1999.

[GardGrizmPran, 1999] Gardner R. J., Gritzmann P., Prangenberg D., On the computational complexity of reconstructing lattice sets from their X-rays, Discrete Mathematics 202 (1999) 45-71.

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

[Gale, 1957]Gale D., A theorem on flows in networks, Pacific J. Math., 7 (1957), pp. 1073–1082.

[Lungo, 1994] Del Lungo A., Polyominoes defined by two vectors, Theoretical Computer Science, 127, 187-198 (1994).

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

[Woeginger, 2001] Woeginger G.J., The reconstruction of polyominoes from their orthogonal projections, Inform. Process. Lett., 77, pp. 225-229, 2001.

[DurrChobrak, 1999] Chrobak M., Durr C., Reconstructing hv-convex polyominoes from orthogonal projections. Inform, Process. Lett. 69(6), 283–289 (1999); Kuba, A., Reconstruction in different classes of 2D discrete sets, Lecture Notes in Comput., 1999.

[Kuba 1999] Kuba, A., Reconstruction in different classes of 2D discrete sets, Lecture Notes in Comput., 1999.

[BarcBrunDeLunNivat, 2001] Elena Barcucci, Sara Brunetti, Alberto Del Lungo, Maurice Nivat, Reconstruction of lattice sets from their horizontal, vertical and diagonal X-rays, Discrete Mathematics 241 (2001) 65–78.

[SrivansVerma, 2013] T. Srivastana, S.K. Verma, Uniqueness Algorithm with Diagonal and Anti-diagonal Projections, International Journal of Tomography and Simulation 2013.

Authors' Information

Hasmik Sahakyan *– Institute for Informatics and Automation Problems of the National of Science of Armenia; Scientific Secretary. 1 P.Sevak str., Yerevan 0014, Armenia; e-mail: hsahakyan@sci.am*

Major Fields of Scientific Research*: Combinatorics, Discrete tomography, Data Mining*

Vladimir Vasil'evich Ryazanov *- since 1976 has been with the Dorodnitsyn Computing Center, Russian Academy of Sciences. Currently is Head of the Department of Mathematical Problems of Recognition and Methods of Combinatorial Analysis.*
Scientific interests*: recognition theory, cluster analysis, data analysis, optimization of recognition models, and applied systems of analysis and prediction.*

Ani Margaryan *– Institute for Informatics and Automation Problems of the National of Science of Armenia; PhD student. 1 P.Sevak str., Yerevan 0014, Armenia; e-mail: ani.margaryan1991@gmail.com*

Major Fields of Scientific Research*: Discrete tomography algorithms*