LINEAR PROGRAM FORM FOR RAY DIFFERENT DISCRETE TOMOGRAPHY

Hasmik Sahakyan, Levon Aslanyan

Abstract: *A special quality of discrete tomography problem solutions that requires the ray difference is considered. Two classes of reconstruction tasks of* $(0,1)$ *-matrices with different rows are studied: matrices with prescribed column and row sums and matrices with prescribed column sums only. Both cases are known as algorithmically open problems. We reformulate them as integer programming problems. Depending on parameters obtained, the Lagrangean relaxation model and then variable splitting technique, or a greedy heuristics approaches are applied for getting approximate solutions. In later case an optimization version is considered, where the objective is to maximize the number of pair-wise different row rays, which in case of existence of a matrix, is equivalent to the requirement of row differences.*

Keywords: *discrete tomography, (0,1)-matrices, integer programming*

ACM Classification Keywords: *F.2.2 Nonnumerical Algorithms and Problems: Computations on discrete structures*

Introduction

Reconstruction algorithms that solve the so called inverse structural problems, have many applications in image processing, medicine, computer tomograph assisted engineering and design, electron microscopy, etc. There are a number of well known medical problems that require discrete reconstruction. For example, Onnasch and Prause [PrauseOnnasch, 1996] described an application of the discrete tomography technique, to routinely reconstruct the acquired biplane cardiac angiograms. Their model based reconstruction approach aims to recover the three-dimensional shape of the left or right chambers of the heart. In [SlumpGerbrands, 1982] Slump and Gerbrands presented a method based on a network flow approach that reconstructs the left ventricle of the heart from two projections.

Reconstruction of discrete sets (finite subsets of the two-dimensional integer lattice) from given projections, - is one of the main tasks of Discrete Tomography. Discrete sets can be presented as binary images. The line sum of a line through the image is the sum of the values of the points on this line. The projection of the image in a certain direction consists of all the line sums of the lines through the image in this direction. Any binary image with exactly the same projections as the original image is *a reconstruction of the image*.

Opposite to methods of Computerized Tomography which use several hundreds of projections in Discrete Tomography a few projections are available. The main problem arising here is that the reconstruction task is usually extremely underdetermined, i.e. there may be many different binary images with the same projections.

On the other hand for any set of more than two directions, the problem of reconstructing a binary image from its projections in those directions is NP-complete. For exactly two directions, the horizontal and vertical ones, it is possible to reconstruct an image in polynomial time. Already in 1957, Ryser found a necessary and sufficient condition for a pair of vectors being the horizontal and vertical projections of a discrete set ([Ryser, 1957]). In the proof of his theorem, Ryser also described a reconstruction algorithm. Another result of Ryser is the definition of the switching operation by which discrete sets having the same projections can be transformed into each other.

So, the problem of reconstructing a binary image from a small number of projections generally leads to a large number of solutions. To reduce the number of possible solutions, a priori information on the image being reconstructed is used. Two special geometrical properties/constraints are often imposed, - convexity and connectivity:

A matrix is horizontal convex (h-convex) if in every row the 1's form an interval, similarly, vertical convex (v-convex), - when in every column the 1's form an interval; and a matrix is hv-convex if it is h-convex and v-convex. A matrix is connected if the set of 1's is connected with respect to the adjacency relation, where every pixel is adjacent to its two vertical neighbours and to its two horizontal neighbours.

It is proven ([BarcucciDelLungoNivatPinzani,1996], [Woeginger,2001], [DurrChrobak,1999] that the existence problems of h-convex, v-convex, hv-convex matrices and the existence problem for connected matrices (polyominoes) are NP-complete; and the reconstruction problem for horizontal and vertical convex polyominoes can be solved in polynomial time.

In [DahlFlatberg, 2002] G. Dahl and T. Flatberg consider a variant of reconstructing hv-convex (0,1)-matrices, where instead of requiring the ones to occur consecutively in each row and column, they maximize the number of neighboring ones. Then the problem is reformulated as an integer programming problem and a solution method based on variable splitting is proposed.

We will consider another relevant concept in this context, - the requirement of different rows on matrix to be reconstructed. Two cases are studied with the requirement of row differences. First - reconstruction of $(0,1)$-matrices with prescribed row and column sums and different rows, and the second - reconstruction with prescribed column sums and different rows. Both are known as algorithmically open problems. The first problem we reformulate as an integer programming problem and use the Lagrangean relaxation and variable splitting technique for an approximate solution. Solving the second problem, we try to find in a constructive way a matrix with given parameters. Additionally we consider an optimization problem - instead of requiring all different rows, we maximize the number of different pairs of rows, which in case of existence of a matrix, is equivalent to the requirement of different rows. Then an approximation greedy algorithm is applied for constructing matrices with the maximum number of different pairs of rows.

$(0,1)$ matrices with different rows

In this section we consider a special case of discrete tomography, - to reconstruct the $(0,1)$-matrix with prescribed row and/or column sums and with all different rows.

Consider a $(0,1)$-matrix of size $m \times n$. Let $R = (r_1, \cdots, r_m)$ and $S = (s_1, \cdots, s_n)$ denote the row and column sums of the matrix respectively, and let $U(R,S)$ be the class of all $(0,1)$-matrices with row sums R and column sums S. A necessary and sufficient condition for the existence of a $(0,1)$ matrix of the class $U(R,S)$ was found by Ryser. Now we formulate the two basic problems:

P1. Existence of a $(0,1)$ matrix with the given row and column sums and with different rows

Given integer vectors $R = (r_1, \cdots, r_m)$ and $S = (s_1, \cdots, s_n)$. Is there a binary matrix $X = \{x_{i,j}\}$ in the class $U(R,S)$ with different rows?

P2. Existence of a $(0,1)$ matrix with the given column sums and with different rows

Given an integer vector $S = (s_1, \cdots, s_n)$. Is there a binary matrix $X = \{x_{i,j}\}$ with different rows whose column sums are given by $S = (s_1, \cdots, s_n)$?

No polynomial algorithms are known for solving P1 or P2, and they are known as open problems. The combinatorial origin of P1 is the hypergraph degree sequence problem, where the complexity is open even for the case $r_i = 3, i = 1,\cdots,m$. P2 comes from the n -dimensional unit cube subsets partitioning.

Below in this section we reformulate P1 as an integer programming problem and show the way of use of integer programming techniques to find the approximate solutions.

Let X be a $(0,1)$ -matrix of size $m \times n$. Obviously $\sum_{j=1}^{n} x_{i,j} = r_i, i = 1,\cdots,m$ and $\sum_{i=1}^{m} x_{i,j} = s_j, j = 1,\cdots,n$ are the row and column sums of the matrix.

The requirement of row difference in X which initially is a combinatorial property, will be presented algebraically, using the intersections of pairs of binary rows. Consider two rows i' and i'' that have the same row sum r . If these rows are different, then they intersect (by 1s) in less than r places. For determining the intersection size of this pair of rows we introduce additional n binary variables $y_{p(i',i''),j}$, so that these variable satisfy requirements $(y_{p(i',i''),j} = 1) \Leftrightarrow (x_{i',j} = 1) \,\&\, (x_{i'',j} = 1)$, where $p(i',i'')$ indicates the number (enumeration) of pair (i',i'') . Obviously this summarized picture can be provided by the following conditions:

$$\begin{cases} y_{p(i',i''),j} \le x_{i',j} \\ y_{p(i',i''),j} \le x_{i'',j} \\ y_{p(i',i''),j} \ge x_{i',j} + x_{i'',j} - 1 \end{cases}$$

In these terms the row pair intersection size is presented by formula $\sum_{j=1}^{n} y_{p(i',i''),j}$, and $\sum_{j=1}^{n} y_{p(i',i''),j} < r$ is the condition of distinction of these rows.

Indeed such conditions are necessary for only pairs of rows with the same sum. It is satisfactory to separate the row pairs with equal sums imposing these restrictions only for such rows. Below, in fact, we prefer to compose a system of inequalities for all pairs of rows.

Enumerate pairs of rows and let $p(i',i'')$ is the number of the pair (i',i'') , for $1 \le i' < i'' \le m$. Assuming that $r_1 \le \cdots \le r_m$, the requirement of different rows has been easily replaced with the following property: intersection size for each pair (i' , i''), $1 \le i' < i'' \le m$ is less than row sum of i'' -th row.

Now P1 can be formulated as follows: given the following system

$$(P1) \begin{cases} (1) \sum_{i=1}^{m} x_{i,j} = s_j, j = 1,\cdots,n \\ (2) \sum_{j=1}^{n} x_{i,j} = r_i, i = 1,\cdots,m \\ (3) \begin{cases} y_{p(i',i''),j} \le x_{i',j} \\ y_{p(i',i''),j} \le x_{i'',j} \\ y_{p(i',i''),j} \ge x_{i',j} + x_{i''j} - 1 \end{cases} \quad 1 \le i' < i'' \le m,\ j = 1,\cdots,n \\ (4) \sum_{j=1}^{n} y_{p(i',i''),j} < r_{i''} \quad 1 \le i' < i'' \le m \\ (5) x_{i,j} \in \{0,1\}, y_{i,j} \in \{0,1\} \end{cases}$$

where $R=(r_1,\cdots,r_m)$ and $S=(s_1,\cdots,s_n)$ are integer vectors and $r_1\le\cdots\le r_m$. $X=\{x_{i,j}\}_{m\times n}$ and $Y=\{y_{i,j}\}_{C_m^2\times n}$ are unknown variables. Is there a $(0,1)$ solution to this system?

Our immediate goal is to represent (P1) in the canonical form of a liner program instance: $Az\le b$.

First we explain the structure of vector z. z is introduced as concatenation of two parts: x and y:

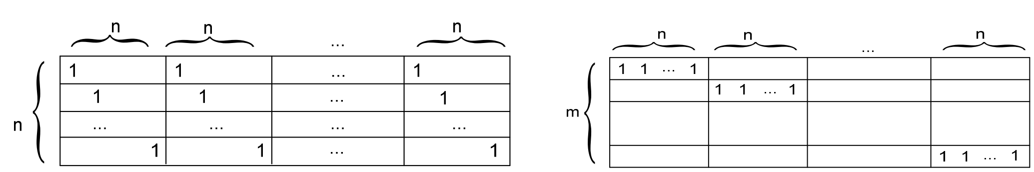
$$x=\begin{pmatrix} x_{1,1}\\ \vdots\\ x_{1,n}\\ \vdots\\ x_{m,1}\\ \vdots\\ x_{m,n}\end{pmatrix} \text{ and } y=\begin{pmatrix} y_{1,1}\\ \vdots\\ y_{1,n}\\ \\ \vdots\\ \\ y_{C_m^2,1}\\ \vdots\\ y_{C_m^2,n}\end{pmatrix},\quad z=\begin{pmatrix} x\\ y\end{pmatrix}, \text{ with } m\cdot n+C_m^2\cdot n \text{ coordinates.}$$

Vector b is a column vector consisting of several fragments: b_1, b_2, b_3, b_4, b_5, where b_1 and b_2 are transposed forms of S and R respectively, b_3 is $2n\cdot C_m^2$-length column vector consisting of all 0s and b_4 is $n\cdot C_m^2$-length column vector consisting of all 1s. C_m^2 components of b_5 are composed by the pairs $p(i',i'')$, as the maximum values of $(r_{i'},r_{i''})$. $n+m+(3n+1)\cdot C_m^2$ is the total length of vector b.

Now compose the matrix A. It will have $n+m+(3n+1)\cdot C_m^2$ rows and $m\cdot n+C_m^2\cdot n$ columns and can be presented in form:

$$\begin{pmatrix} A_1 & B_1\\ A_2 & B_2\\ A_3 & B_3\\ A_4 & B_4\\ A_5 & B_5\end{pmatrix}$$

The structures of $(A_1)_{n\times(m\cdot n)}$ and $(A_2)_{m\times(m\cdot n)}$ is given below in figure 1



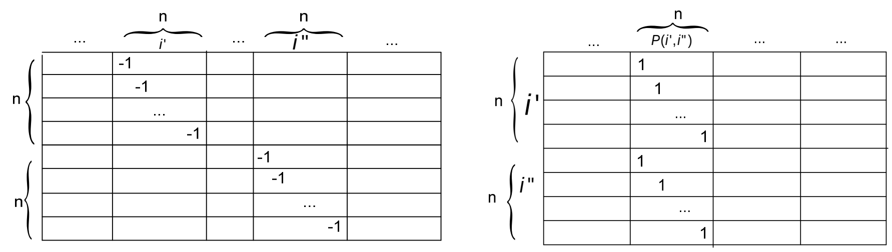
(a) (b)

Fig.1

(a) shows the placement of 1s in $(A_1)_{n\times(m\cdot n)}$ and (b) - in $(A_2)_{m\times(m\cdot n)}$.

$(B_1)_{n\times(C_m^2\cdot n)}$ and $(B_2)_{m\times(C_m^2\cdot n)}$ consist of all 0 components.

$(A_3)_{2n \cdot C_m^2 \times (m \cdot n)}$ and $(B_3)_{2n \cdot C_m^2 \times (C_m^2 \cdot n)}$ consist of C_m^2 submatrices (vertically) of the following structures (one for each pair of rows (i', i'')) given in figure 2.

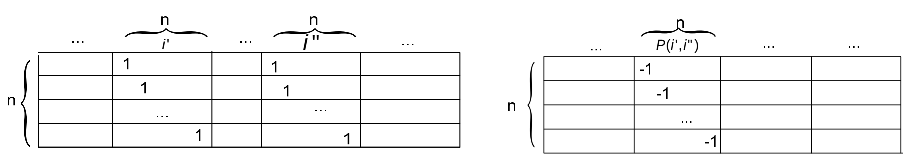


(a) (b)

Fig.2

(a) gives the placement of 1s in $(A_3)_{2n \cdot C_m^2 \times (m \cdot n)}$ and (b) - in $(B_3)_{2n \cdot C_m^2 \times (C_m^2 \cdot n)}$.

Similarly, $(A_4)_{n \cdot C_m^2 \times (m \cdot n)}$ and $(B_4)_{n \cdot C_m^2 \times (C_m^2 \cdot n)}$ are the vertical concatenation of the following structures given below in figure 3 that are constructed for each pair of rows (i', i'').



(a) (b)

Fig.3

(a) corresponds to $(A_4)_{n \cdot C_m^2 \times (m \cdot n)}$ and (b) - to $(B_4)_{n \cdot C_m^2 \times (C_m^2 \cdot n)}$.

$(A_5)_{C_m^2 \times (m \cdot n)}$ consists of all 0 components and $(B_5)_{C_m^2 \times (C_m^2 \cdot n)}$ has the form given in figure 4:

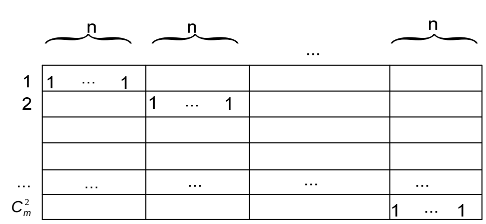


Fig.4

Now A_1 and A_2 (with b_1 and b_2) provide column and row sums respectively, - (1) and (2) conditions in (P1). A_3, B_3 and A_4, B_4 (with b_3, b_4) are composed for presenting intersections of row pairs, - (3). And finally, B_5 with b_5 is to provide the condition (4), - row differences.

Below in figure 5 we present the whole scheme.

This sparse matrix codes the discrete tomography P1 problem in terms of integer linear programming. Several classes of integer programming are known solvable in polynomial time – problems with total unimodular matrices, set network problem, etc. Matrix A is not as simple but the given form helps to see and construct the appropriate relaxations to receive and apply approximate solution algorithms.

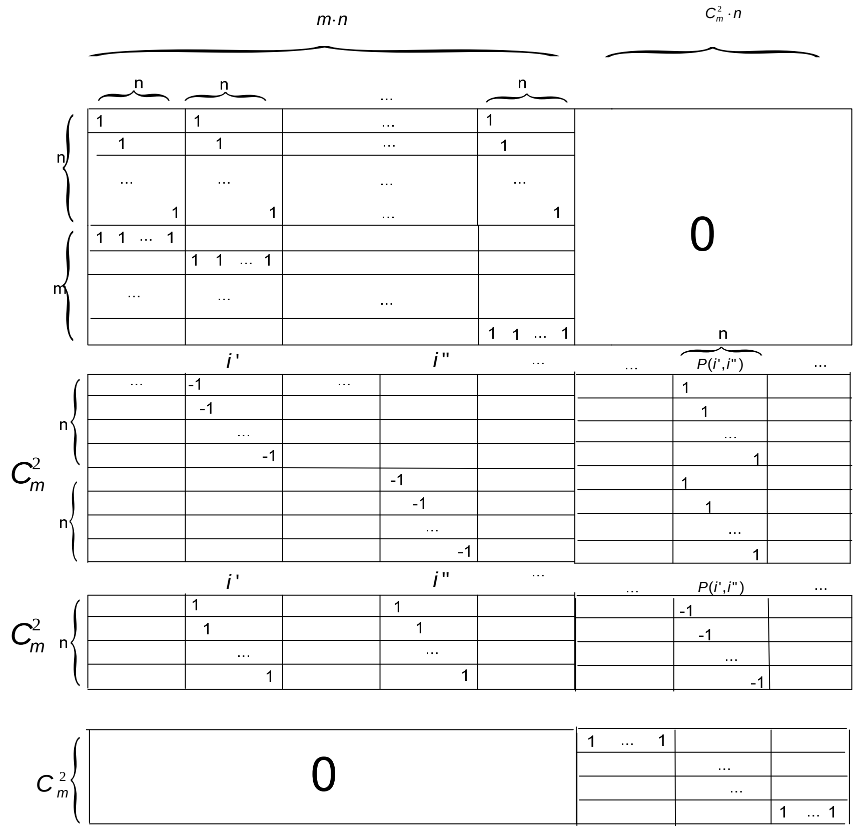


Fig.5

Let us consider now also the problem P2 and the problem of representing it in a linear program system.

Let X be a $(0,1)$-matrix of size $m \times n$ for problem P2. $\sum_{i=1}^{m} x_{i,j} = s_j, j = 1,\cdots,n$ provides the column sums of the matrix. For presenting the condition of different rows of X, we are considering the following representation of rows. i-th row of X, as a binary vector of size n can be represented in form $2^{n-1}x_{i,1} + 2^{n-2}x_{i,2} + \cdots + 2^0 x_{i,n}$. If two rows are different then the numerical values of the corresponding representations are different. We get the following system:

$$(P2)\begin{cases}(1)\sum_{i=1}^{m} x_{i,j} = s_j, j = 1,\cdots,n \\ (2)\sum_{j=1}^{n} 2^{n-j} x_{i',j} \neq \sum_{j=1}^{n} 2^{n-j} x_{i'',j} \quad 1 \le i' < i'' \le m \\ (3) x_{i,j} \in \{0,1\}\end{cases}$$

Enumerate pairs of rows and let $p(i',i'')$ be the number of the pair (i',i''). We introduce integer variables $y_1,\cdots,y_{C_m^2}$, where $y_{p(i',i'')} = \sum_{j=1}^{n} 2^{n-j} x_{i',j} - \sum_{j=1}^{n} 2^{n-j} x_{i'',j}$.

Now P2 can be formulated in the following way: given the system

$$
(P2)\begin{cases}
(1)\sum_{i=1}^{m} x_{i,j} = s_j,\ j = 1,\cdots,n \\
(2) y_{p(i',i'')} = \sum_{j=1}^{n} 2^{n-j}(x_{i',j} - x_{i'',j}) \quad 1 \le i' < i'' \le m \\
(3)\ y_j \ne 0, \quad j = 1,\cdots,C_m^2 \\
(4) x_{i,j} \in \{0,1\} \\
(5)\ y_j \in Z
\end{cases}
$$

where $S = (s_1,\cdots,s_n)$ is an integer vector and $X = \{x_{i,j}\}_{m\times n}$ and $Y = \{y_j\}_{C_m^2}$ are unknown variables. Is there a solution of this system?

In the same way as in the previous case, this system can be presented in form $Az \le b$.

Thus we get two cases of integer programming problems and now techniques of integer programming can be applied for solving them. First the linear programming relaxation is considered and experiments are done in existing linear programming software environment. The results were not adequate. Afterward an experimentation software system is created which provides an environment for treatment of combinatorial problems. In particular P1 and P2 are relaxed in Lagrangean manner and experimented for different classes of (0,1)-matrices. The results are satisfactory for a number of specific cases.

Lagrangean relaxation and variable splitting for P1 and P2

We consider Lagrangean relaxation of P1. There are many ways in which a given problem can be relaxed in a Lagrangean fashion. We will use variable splitting technique - we split our problem into separate vertical and horizontal subproblems, then the horizontal subproblem is further separated into subproblems for each pair of rows.

We duplicate variables $x_{i,j}$, getting 2 independent sets of variables $x_{i,j}^h$ and $x_{i,j}^v$, and then dualize the copy (duplication) constraint using Lagrangean multipliers $\lambda_{i,j}$.

$$
(P1LR)\begin{cases}
\max\{\sum \lambda_{i,j}(x_{i,j}^h - x_{i,j}^v)\} \\
(1)\sum_{i=1}^{m} x_{i,j}^v = s_j,\ j = 1,\cdots,n \\
(2)\sum_{j=1}^{n} x_{i,j}^h = r_i,\ i = 1,\cdots,m \\
(3)\begin{cases}
y_{p(i',i''),j} \le x_{i',j}^h \\
y_{p(i',i''),j} \le x_{i'',j}^h \\
y_{p(i',i''),j} \ge x_{i',j}^h + x_{i'',j}^h - 1
\end{cases} \quad 1 \le i' < i'' \le m,\ j = 1,\cdots,n \\
(4)\sum_{j=1}^{n} y_{p(i',i''),j} < r_{i''} \quad 1 \le i' < i'' \le m \\
(5) x_{i,j}^h, x_{i,j}^v \in \{0,1\},\ y_{i,j} \in \{0,1\}
\end{cases}
$$

Split the problem into sub problems – horizontal and vertical:

$$
\text{(P1LR-h)} \begin{cases} \max\{\sum \alpha_{i,j} x^h_{i,j}\} \\ (1)\sum_{j=1}^{n} x^h_{i,j} = r_i, \quad i = 1,\cdots,m \\ (2)\begin{cases} y_{p(i',i''),j} \le x^h_{i',j} \\ y_{p(i',i''),j} \le x^h_{i'',j} \\ y_{p(i',i''),j} \ge x^h_{i',j} + x^h_{i'',j} - 1 \end{cases} \quad 1 \le i' < i'' \le m,\ j = 1,\cdots,n \\ (3)\sum_{j=1}^{n} y_{p(i',i''),j} < r_{i''} \quad 1 \le i' < i'' \le m \\ (4) x^h_{i,j} \in \{0,1\}, y_{i,j} \in \{0,1\} \end{cases}
\qquad
\text{(P1LR-v)} \begin{cases} \max\{\sum \beta_{i,j} x^v_{i,j}\} \\ (1)\sum_{i=1}^{m} x^v_{i,j} = s_j,\ j = 1,\cdots,n \\ (2) x^v_{i,j} \in \{0,1\} \end{cases}
$$

Using similar reasons P1LR-h is then split into subproblems for each pair of rows.

Further we apply an iterative procedure to find the optimisation coefficients $\lambda_{i,j}$. On each iteration we consider C_m^2 +1 separate subproblems (C_m^2 horizontal and 1 vertical).

The obtained vertical and horizontal subproblems are simple and can be easily solved. Solutions of subproblems of this and similar kind are gathered in special library in the above mentioned experimentation software.

An analog reasoning is true for P2 .

Construction of a $(0,1)$ matrix with the given column sums and with different rows

In this section we consider the problem P2 from another point of view. This time we look for a possible solution - by constructing a special solution/matrix which is in a specific standard format.

Let an integer vector $S = (s_1,\cdots,s_n)$, $0 \le s_i \le m$, $i = 1,\cdots,n$ is given. If there exists an $m \times n$ $(0,1)$ -matrix with all different rows and with s_i column sums, then after a finite number of row transpositions it can be transformed into the matrix with the same column sums, which has the “canonical form” – where each column consists of continuous intervals of 1’s (higher part) and 0’s (lower part) such that they split the intervals of the previous column in 2 parts. Therefore if P2 has a solution, it can be found (constructed) by algorithms which compose matrices in column-by-column fashion. We will refer to the construction version of P2 as P2-C.

The first column is being constructed by allocating s_1 1’s to the first s_1 rows-positions followed by the $m - s_1$ 0’s in others. Two intervals is the result: – the s_1 interval of 1’s, and the $m - s_1$ interval of 0’s. We denote these intervals $d_{1,1}$ and $d_{1,2}$. Hereafter the first index will indicate the number of column and the second – the number of interval within the column. Intervals with odd numbers consist of 1s, and intervals with even number consist of 0s. So for the first column we have the following system:

$$
\begin{cases} d_{1,1} + d_{1,2} = m \\ d_{1,1} = s_1 \end{cases}
$$

We construct the second column putting s_2 ones and $m - s_2$ zeros in $d_{2,1}$, $d_{2,2}$, $d_{2,3}$ and $d_{2,4}$ intervals such that $d_{2,1}$ and $d_{2,3}$ are filled by 1's, and $d_{2,2}$, $d_{2,4}$ - by 0's, and

$$\begin{cases} d_{2,1} + d_{2,2} + d_{2,3} + d_{2,4} = m \\ d_{2,1} + d_{2,2} = s_1 \\ d_{2,1} + d_{2,3} = s_2 \end{cases}$$

In general, k -th column consists of $d_{k,1}, d_{k,2}, \cdots, d_{k,2^k}$ intervals (among them can be 0-length intervals) filled by 0's and 1's accordingly, such that

$$\begin{cases} \sum_{i=1}^{2^k} d_{k,i} = m \\ \sum_{i=0}^{2^{k-(j+1)}} (d_{k,2^{j+1}\cdot i+1} + d_{k,2^{j+1}\cdot i+2} + \cdots + d_{k,2^{j+1}\cdot i+2^j}) = s_{k-j}, \quad j = 0, \cdots, k-1 \end{cases}$$

So we formulate the following problem:

Given a system of $d_{n,1}, d_{n,2}, \cdots, d_{n,2^n}$ variables

$$\text{(P2-C)} \begin{cases} \sum_{i=1}^{2^n} d_{n,i} = m \\ \sum_{i=0}^{2^{n-(j+1)}} (d_{n,2^{j+1}\cdot i+1} + d_{n,2^{j+1}\cdot i+2} + \cdots + d_{n,2^{j+1}\cdot i+2^j}) = s_{n-j}, \quad j = 0, \cdots, n-1 \\ d_{i,j} \in \{0,1\} \end{cases}$$

Is there a solution of the system? As in previous cases we reformulate our problem as an integer programming task. But in this case we get s a system with the exponential number of variables.

Below we will consider an optimization version of P2-C and bring an approximation greedy algorithm for its solution.

During the construction of matrices in column-by-column fashion, partitioning of the intervals in each step can be performed by different ways - following different goals. Let assume that the partitioning of intervals aims to maximize some quantitative characteristics, which leads to the matrices with different rows in case when the later exists. One of such characteristics - the number of pairs of different rows – is considered in [S, 1995]. An approximation greedy algorithm, which constructs the target (0,1)-matrices in the above described column-by-column fashion of partitioning is considered. The algorithm provides the optimal construction of each column – i.e. the construction, which provides the maximal number of new (i, j) pairs of different rows in each step. It is proven that the optimal construction of each column is provided by partitioning, which distributes the difference $s_k - (m - s_k)$ "homogeneously" on all current non atomic intervals.

Bibliography

[PrauseOnnasch, 1996] G. P. M. Prause and D. G. W. Onnasch, Binary reconstruction of the heart chambers from biplane angiographic image sequence, IEEE Transactions Medical Imaging, 15 (1996) 532-559

[SlumpGerbrands, 1982] Slump, C.H., Gerbrands, J.J., A network flow approach to reconstruction of the left ventricle from two projections. Comput. Gr. Im. Proc., 18, 18.36 (1982).

[DurrChrobak, 1999] Durr Ch., Chrobak M., Reconstructing hv-convex polyominoes from orthogonal projections, Information Processing Letters 69 (1999) pp. 283-291.

[BarcucciDel LungoNivatPinzani, 1996] E. Barcucci, A. Del Lungo, M. Nivat, and R. Pinzani, Reconstructing convex polyominoes from horizontal and vertical projections, Theoret. Comput. Sci., 155:321{347, 1996.

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

[Ryser, 1966] H. J. Ryser. Combinatorial Mathematics, 1966.

[DahlFlatberg, 2002] G. Dahl and T. Flatberg, Lagrangian decomposition for reconstructing hv-convex (0,1)-matrices, Report 303, ISBN 82-7368-255-2, ISSN 0806-3036, December 2002

Authors' Information

Hasmik Sahakyan *– Leading Researcher, Institute for Informatics and Automation Problems, NAS RA, P.Sevak St. 1, Yerevan 14, Armenia, e-mail: hasmik@ipia.sci.am*

Levon Aslanyan *– Head of Department, Institute for Informatics and Automation Problems, NAS RA, P.Sevak St. 1, Yerevan 14, Armenia, e-mail: lasl@.sci.am*