APPROXIMATION GREEDY ALGORITHM FOR RECONSTRUCTING OF (0,1)-MATRICES WITH DIFFERENT ROWS

Hasmik Sahakyan

Abstract: *Existence and reconstruction issues are considered for a class of (0,1)-matrices with different rows. Intending to define quantitative characteristics, maximization of which lead to matrices with different rows in case when the later exist, we consider the number of pairs of different rows.A greedy algorithm is introduced for this purpose and then its optimality is proven for local steps.*

Keywords: *(0,1)-matrices, greedy algorithms*

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

Introduction

Matrices with (0,1) elements and prescribed row and column sums is a classical object which appears in many branches of applied mathematics. In combinatorics, such matrices used to encode hypergraphs with prescribed degrees of verticesand related structures, see, for example [LintWilson, 2001]. In statistics, (0,1) matrices with prescribed row and column sums are known as binary contingency tables, see [ChenDiaconisHolmesLiu, 2005]. In Discrete Tomography (0,1) matrices serve for representation of discrete sets. The projections of a matrix in the horizontal and vertical directions correspond to the row and column sums of the matrix. There is a known result of Ryser - a necessary and sufficient condition for a pair of vectors being the row and column sums of a (0,1)-matrix ([Ryser, 1957]). (0,1) matrices with prescribed row and column sums and with special geometrical properties/constraints imposed, are addressed for example in [DurrChrobak, 1999], [BarcucciDel LungoNivatPinzani, 1996], [Woeginger, 2001].

We will consider another additional requirement on (0,1) matrices with given row and column sums - the requirement of non repetition of rows. Such a requirement on rows has its origin in terms of the n dimensional unit cube. Vertices of the cube are presented as n-tuples of 0,1 values, and in this way a vertex subset has been presented as a (0,1)-matrix, where rows correspond to vertices. Row sums indicate the layers of the cube containing the corresponding vertices. Let $R=(r_1,\cdots,r_m)$ and $S=(s_1,\cdots,s_n)$ denote the row and column sum vectors of a $(0,1)$-matrix of size $m\times n$. Then i-th column sum identifies the number of vertices in the vertex subset with 1 value in i-th position. Now existence of a $(0,1)$ matrix is equivalent to the existence of m vertices situated in r_1-th, r_2-th, etc. r_m-th layers such that s_1 vertices/tuplescontain 1 on the first position, s_2 verticescontain 1 on the second position, etc. and s_n contain 1 on the n-th position. In other words s_i and $m-s_i$ are the partition sizesof the vertex subset on i-th direction. In case when the placement of vertices is not

important and we are interested just in existence of a vertex subset with given partition sizes – we search out a subclass of matrices with column sums $S = (s_1, \cdots, s_n)$ and with m rows which are all different.

Both cases (with or without $R = (r_1, \cdots, r_m)$) are known as algorithmically open problems. The current research is focused on the second one and considers issues of *existence* and *construction* of matrices in a constructive way. A greedy algorithm is proposed and then proven its optimality in local steps. Nevertheless there are examples of matrices showing non optimality of the algorithm globally.

$(0,1)$ matrices with 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 sum vectors of the matrix respectively, and let $U(R,S)$ be the class of all $(0,1)$-matrices with row sum R and column sum S.A necessary and sufficient condition for the existence of a $(0,1)$ matrix of the class $U(R,S)$ was found by Ryser. Another result of Ryser is the definition of an interchange operation to be a transformation which replaces the 2x2 submatrix $\begin{bmatrix} 10 \\ 01 \end{bmatrix}$ of a matrix into the 2x2 submatrix $\begin{bmatrix} 01 \\ 10 \end{bmatrix}$ or vice versa. Clearly an interchange (and hence any sequence of interchanges) does notchange the row and column sum vectors of a matrix, and therefore transforms a matrix in $U(R,S)$ into another matrix in $U(R,S)$. Ryser proved that given $A, B \in U(R,S)$ there is a sequence of interchanges which transforms A into B.

We are interested in a subclass of $U(R,S)$ where all the rows of matrices are different and in particular we will consider the class $U(S)$ of all $(0,1)$-matrices with column sum $S = (s_1, \cdots, s_n)$ and with m rows which are all different.

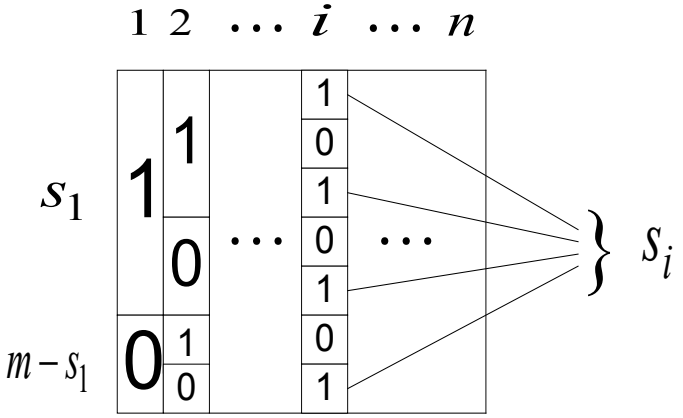
Now we formulate two versions of the problem: decision (P1) and optimization (P2).

(P1) Existence of a $(0,1)$ ***matrix with the given column sum and with different rows***

Remain that no polynomial algorithms are known for solving (P1)and it is known as an open problem. The combinatorial origin is the hypergraph degree sequence problem.

First we define an interchange operation for the class $U(S)$. Let usdefine itin analogous to Ryser's way:the interchange operation replaces the 2x1 submatrix $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ into the 2x1 submatrix $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ or vice versa. Clearly not every interchange operation is admissible: it keeps column sums but it may induce repeating rows and thus may transform a matrix in $U(S)$ into a matrix out of $U(S)$.We call this - simple interchange operation. However performing all interchanges within a pair of rows will keep a matrix within the class. But it is simply the interchange of rows.

Let $S=(s_1,\cdots,s_n)$, and $A\in U(S)$. Applying row interchanges we can transform A into another matrix of $U(S)$ where s_1 ones form an interval in the first column and situated in $1,\cdots,s_1$ rows. Then in the same way we can transform the obtained matrix into another one where s_2 ones compose two intervals in the second column (say $s_{2,1}$ and $s_{2,2}$ lengths, $s_2=s_{2,1}+s_{2,2}$) situated in $1,\cdots,s_{2,1}$ and $s_1+1,\cdots,s_1+s_{2,2}$ rows respectively, and so on. So in each column we will get alternating 1 and 0 intervals (possibly of 0 lengths) as illustrated in the figure below. Rows i and j taken from different intervals are differing; rows within the same interval coincide with each other.



We will call this construction a matrix of *partitioned form*. Starting from some column (at least it is the n-th column) all columns consist of all one length intervals. Depending on partitioning of s_i there can be obtained different matrices of partitioned form in the same class $U(S)$.

Note: two different matrices of partitioned form of the same class $U(S)$ can be transformed into each other applying a sequence of simple interchange operations. Concluding, - given $A,B\in U(S)$ there is a sequence of interchanges which transforms A into B.

So with the help of row interchanges each matrix in $U(S)$ is transformable into a partitioned form. Therefore if the class $U(S)$ is not empty then it contains at leastone matrix of this form. Then it is reasonable to find solution among the matrices of partitioned form. It can be realized constructing a matrix column by column and providing s_i ones in i-th column. If in n-th column we get all one length intervals, then the matrix will not have coinciding rows. During the construction of each column the partitioning of intervals of the previous column can be arbitrary keeping only s_i as the sum of all 1 intervals. But it is reasonable to follow a goal which will lead to the required matrix – with different rows.Let us 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.

We will consider one of such characteristics - the number of pairs of different rows,– it was first considered in [S, 1995].

(P2) $(0,1)$ ***matrices with maximum number of pairs of differing rows***

Let $\Im(S)$ define the class of $(0,1)$-matrices of size $m \times n$, having $S = (s_1, \cdots, s_n)$ as its column sum vector. In this way $U(S) \subseteq \Im(S)$. For a given $A \in \Im(S)$ let $D(A)$ denote the number of pairs of differing rows of A.

Consider the following optimization problem:

P2:Find $A_{opt} \in \Im(S)$ such that $D(A_{opt}) = \max_{A \in \Im(S)} D(A)$.

Obviously C_m^2 is the lowest upper bound for $D(A)$ and it is achievable only for matrices of $U(S)$. Therefore if $U(S)$ is not empty, then a solution of optimization problem (P2)on $\Im(S)$ will serve also as a solution of existence problem(P1) for $U(S)$: $A_{opt} \in U(S)$.

Thus(P2) is not easier than (P1).

We will introduce an approximation algorithm for solving P2. Further we will prove that the algorithm is optimal in local steps.

Greedy approach for solving (P2)

The greedy heuristic is the most used heuristics for optimization problems. The general approach is as follows: repeatedly execute a procedure which minimizes (maximizes) the increase of the objective function. In some cases such a strategy guarantees the optimal solution.

Given $S = (s_1, \cdots, s_n)$. The goal is to construct a matrix $A_{opt} \in \Im(S)$ such that $D(A_{opt}) = \max_{A \in \Im(S)} D(A)$.

Now we describe an algorithm G that constructs a matrix column by column: starting from the first one and adding a column in each step.The objective function is D: $A \in \Im(S) \rightarrow$ number of pairs of differing rows; and the goal is a matrix with the greatest possible value of D.Let A_G denote the constructed by G matrix and let $\Delta D_k(A_G)$ denote the increase of objective function during the k-th stepof G.

Assume without loss of generality that $s_i \geq m - s_i$, $i = 1, \cdots n$.

Algorithm *G*

Step1. Construction of the first column: it consists of s_1 ones placed in the first s_1 rows-positions followed by $m - s_1$ zeros in others. Two intervals is the result: – the s_1-length interval of ones, and the $(m - s_1)$-length interval of zeros. We denote these intervals by $d_{1,1}^G$ and $d_{1,2}^G$. Hereafter the first sub-index will indicate the number of column and the second – the number of interval within the column. Intervals with odd numbers contain all ones, and intervals with even number contain all zeros. So construction of the first column is in unique way:

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

At this point we get $d^G_{1,1} \cdot d^G_{1,2} = s_1 \cdot (m - s_1)$ pairs of differing (by the first position) rows andthus: $\Delta D_1(A_G) = d^G_{1,1} \cdot d^G_{1,2}$.

Let we have constructed the first $k-1$ columns. In general, $(k-1)$-th column consists of 2^{k-1} intervals filled by ones and zeros accordingly. Since among thema presence of 0-length intervalsis possible and they can not be used anymore, let assume that ($k-1$)-th column consists of p non-zero length intervals denoted by $d^G_{k-1,1}, d^G_{k-1,2}, \cdots, d^G_{k-1,p}$. Recall that the rows coincide within the intervals and differ otherwise. If in some column j we get all one length intervals, then at this moment non repetition of all rows, and hence the maximum number of pairs of different rows is already provided. Further constructionsare arbitrary.

Step k. During this step each $d^G_{k-1,i}$ length interval will be partitioned into $d^G_{k-1,i,0}$ and $d^G_{k-1,i,1}$ length intervals filled by zeros and ones respectively: $d^G_{k-1,i,} = d^G_{k-1,i,0} + d^G_{k-1,i,1}$ such that

$\sum_{i=1}^{p} d^G_{k-1,i,0} = m - s_k$ and $\sum_{i=1}^{p} d^G_{k-1,i,1} = s_k$. The increase of objective function during the k-th step is:

$$\Delta D_k(A_G) = \sum_{i=1}^{p} d^G_{k-1,i,1} \cdot d^G_{k-1,i,0}.$$

We will realize partitions having a goal to minimize length differences of intervals.

The idea is in following: if $s_k = m - s_k, k = 1, \cdots n$, then in each step we would split every interval into 2 equal (± 1) parts and fill by zeros and ones respectively which will lead to all one length intervalsin logarithmic number (minimum possible [Knuth,]) steps.Furthermore,among all integer partitions of $d^G_{k-1,i}: d^G_{k-1,i} = d^G_{k-1,i,1} + d^G_{k-1,i,0}$,the largest product $d^G_{k-1,i,1} \cdot d^G_{k-1,i,0}$ is achieved when $d^G_{k-1,i,1} = d^G_{k-1,i,0}$ .Thus following this strategy would bring to the goal, but in general at each step *k*we have $s_k - (m - s_k)$ extra ones. Trying to be closer to equal lengths of intervals we1) distribute the extra $s_k - (m - s_k)$ ones among intervals keeping a "homogeneous" distribution; and then 2) split the remaining intervals into 2 equal parts– puttingequal number of zeros and ones.

Further we will show that this will satisfy the optimization criterion, - the maximum number of new (i, j) pairs of different rows in each step.

Now describe the process in detail.

Let $r_k = s_k - (m - s_k)$ and assume that there are l odd length intervals among the intervals of $(k-1)$-th column. It is easy to check that r_k and l have the same parity and hence $r_k - l$ is even number.Construction of the k-th column is in 2 phases: distribution of r_k "extra" ones during the first, and distribution of remaining ones during the second phases.

Phase 1.

a) $r_k \le l$

Chose arbitrary r_k intervals among the l odd intervals and put an 1 in each.

b) $r_k > l$

All l odd intervals get an 1. After this we put two by two ones in intervals starting from the intervals of even length then altering from odd to even, and continuing cyclically until all r_k ones have been exhausted. If during the process some short intervals have been filled, they do not participate any longer.It is worth to mention that after putting l ones on odd intervals, we get all even lengths, and $r_k - l$ is even as well, so in this way the process of distribution is correct.After this phase there remain equal numbers of zeros and ones.

Phase 2.

a) $r_k \le l$

Half of the remaining $l - r_k$ odd intervals get one 0, others – one 1, after that all intervals have been split into equal parts and receive equal number of zeros and ones.

b) $r_k > l$

all intervals have been split into equal parts and receive equal number of zeros and ones.

Let c_i denote the difference between the distributedones and zeros on i interval: $c_i = d^{G_1}_{k-1,i,1} - d^{G_1}_{k-1,i,0}$, $i = 1, \cdots, p$. Now we will estimate c_i.

The case of $r_k \le l$ is simple:

$$c_i = \begin{cases} 0, & \text{for all even length intervals} \\ 1, & \text{for } (l + r_k)/2 \text{ odd length intervals} \\ -1, & \text{for } (l - r_k)/2 \text{ odd length intervals} \end{cases}$$

Suppose that in case of $r_k > l$, t complete cycles were performed during the two by two distributions of ones. Let D denote the maximum length of those intervals filled during this process. Thus all even intervals of $(\ge D)$-lengths received at least D extra ones ($(< D)$-lengths are filled). Concerning odd intervals - $(\ge D+1)$-lengths received at least $D+1$ extra ones ($(< D+1)$-lengths are filled). Now let d denote the amount of extra ones (above D) received by each not filled interval. Remain that both D and d are even numbers. Thus after t complete cycles, all even intervals of $(> D)$-lengths receive $D + d$ and all odd intervals of $(> D)$-lengthsreceive $D + d + 1$ extra ones. Remaining extra ones (denote this amount by r') is not enough for a next complete cycle. Continue distribution starting from even intervals. Suppose their number is p_1.

Consider cases.

(1) $0 < r' < 2p_1$

Choose $r'/2$ intervals among p_1 and distribute 2 ones on each.

(2) $r' = 2p_1$

All p_1 intervals get 2 ones.

(3) $r' > 2p_1$

All p_1 intervals get 2 ones. Among odd intervals choose $(r' - 2p_1)/2$ and distribute remaining $r' - 2p_1$ ones by two.

Coming back to estimation of c_i, the picture is as follows:

1. $0 < r' < 2p_1$

$$c_i = \begin{cases} D + d + 1, & \text{for odd length intervals} \\ D + d \text{ or } D + d + 2 & \text{for even length interval} \end{cases}$$

2. $r' = 2p_1$

$$c_i = \begin{cases} D + d + 1, & \text{for odd length intervals} \\ D + d + 2 & \text{for even length interval} \end{cases}$$

3. $r' > 2p_1$

$$c_i = \begin{cases} D + d + 1 \text{ or } D + d + 3 & \text{for odd length intervals} \\ D + d + 2 & \text{for even length interval} \end{cases}$$

Resuming: - $\max_{i,j} | c_i - c_j | \leq 2$ for all i, j pairs of even $(\geq D)$-length and odd $(\geq D+1)$-length intervals.

Now the k-th step is completely described.

Thus on k-th column the lengths are the following:

$d^{G_1}_{k-1,i,1} = \frac{d^{G_1}_{k-1,i} + c_i}{2}$, $d^{G_1}_{k-1,i,0} = \frac{d^{G_1}_{k-1,i} - c_i}{2}$, $i = 1, \cdots, p$ filled by 1 and 0 respectively. Each of the intervals may be of 0-length.

Note. (1) and (3) cases may lead to not uniqueness of constructions. Choice of $r'/2$ even intervals in (1) and $(r' - 2p_1)/2$ odd intervals in (3) will cause branching during the first phase.

The goal is to prove that all branches maximize the increase of objective function – pairs of differing rows – in each local step.

Local Optimality

Theorem

(1) Each step of thealgorithm G is optimal: it provides the maximum increase of the objective function – pairs of differing rows;

(2) All optimal constructions of each column are those according to G.

Proof.

(1) Let we have p non zero intervals in $(k-1)$-th column denoted by: $d_{k-1,1}, d_{k-1,2}, \cdots, d_{k-1,p}$, $k = 2, \cdots, n$. Assume thatduring k-th step of G the the i-th interval of $d_{k-1,i}$ length is partitioned into the $d^G_{k-1,i,0}$, $d^G_{k-1,i,1}$ length intervals filled by zeros and ones respectively; and let $d_{k-1,i,1}$, $d_{k-1,i,0}$ bethe corresponding lengths of intervals obtainedas a result of the optimal partition provided by some algorithm during its k-th step, where $\sum_{i=1}^{p} d_{k-1,i,0} = m - s_k$ and $\sum_{i=1}^{p} d_{k-1,i,} = s_k$.

Thus $\Delta D_k(A_G) = \sum_{i=1}^{p} d^G_{k-1,i,1} \cdot d^G_{k-1,i,0}$ and $\Delta D_k(A) = \sum_{i=1}^{p} d_{k-1,i,1} \cdot d_{k-1,i,0}$ are the corresponding increases of the objective function. We intend to prove that $\Delta D_k(A_G) \geq \Delta D_k(A)$.

$$\Delta D_k(A_G) - \Delta D_k(A) = \sum_{i=1}^{p} \left(d^G_{k-1,i,1} \cdot d^G_{k-1,i,0}\right) - \sum_{i=1}^{p} \left(d_{k-1,i,1} \cdot d_{k-1,i,0}\right) =$$

$$\sum_{i=1}^{p} \left(\frac{(d^G_{k-1,i} + c_i)}{2} \cdot \frac{(d^G_{k-1,i} - c_i)}{2} \right) - \sum_{i=1}^{p} \left(d_{k-1,i,1} \cdot d_{k-1,i,0}\right) =$$

$$\frac{1}{4} \sum_{i=1}^{p} \left((d^G_{k-1,i})^2 - (c_i)^2 - 4 \cdot d_{k-1,i,1} \cdot d_{k-1,i,0} \right) =$$

$$\frac{1}{4} \sum_{i=1}^{p} \left(\left(d_{k-1,i,1} + d_{k-1,i,0}\right)^2 - (c_i)^2 - 4 \cdot d_{k-1,i,1} \cdot d_{k-1,i,0} \right) = \frac{1}{4} \sum_{i=1}^{p} \left(\left(d_{k-1,i,1} - d_{k-1,i,0}\right)^2 - (c_i)^2 \right)$$

We denote by α_i the length differences for i-th interval provided by algorithms G and A: $\alpha_i = d_{k-1,i,1} - d^G_{k-1,i,1}$, $i = 1, \cdots, p$. Obviously $d_{k-1,i,0} - d^G_{k-1,i,0} = -\alpha_i$. Hence $d_{k-1,i,1} = \alpha_i + d^G_{k-1,i,1}$ and $d_{k-1,i,0} = -\alpha_i + d^G_{k-1,i,0}$, which implies:

$d_{k-1,i,1} - d_{k-1,i,0} = 2\alpha_i + (d^G_{k-1,i,1} - d^G_{k-1,i,0}) = 2\alpha + c_i$.

Notice that $\sum_{i=1}^{p}\alpha_i = 0$ as $\sum_{i=1}^{p} d_{k-1,i,1} = \sum_{i=1}^{p} d^G_{k-1,i,1} = s_k$.

Thus

$$\Delta D_k(A_G) - \Delta D_k(A) = \frac{1}{4}\sum_{i=1}^{p}\left((c_i + 2\alpha_i)^2 - (c_i)^2\right) = \frac{1}{4}\sum_{i=1}^{p}\left(4\alpha_i c_i + 4\cdot(\alpha_i)^2\right) = \sum_{i=1}^{p}\left(\alpha_i c_i + (\alpha_i)^2\right)$$

and so

$$\Delta D_k(A_G) - \Delta D_k(A) = \sum_{i=1}^{p}\left(\alpha_i c_i + (\alpha_i)^2\right) \quad (1)$$

Consider cases.

1. $\max_{1\le i,j\le p} |c_i - c_j| \le 2$.

Let $\alpha_{j_1} \ge 0,\cdots,\alpha_{j_t} \ge 0,\quad \alpha_{j_{t+1}} < 0,\cdots,\alpha_{j_p} < 0$. For simplification of notations assume that the first are non negative:

$\alpha_1 \ge 0,\cdots,\alpha_t \ge 0,\quad \alpha_{t+1} < 0,\cdots,\alpha_p < 0$. $\sum_{i=1}^{p}\alpha_i = 0$ implies $\sum_{i=1}^{t}\alpha_i = -\sum_{i=t+1}^{p}\alpha_i$.

Thus

$\Delta D_k(A_G) - \Delta D_k(A) = c_{i_0}\cdot\sum_{i=1}^{t}\alpha_i + c_{i_0}\cdot\sum_{i=t+1}^{p}\alpha_i + \sum_{i=1}^{p}(\alpha_i)^2$, where $c_{i_0} = \min_{1\le i\le t} c_i$ and $c_{j_0} = \max_{t+1\le i\le p} c_i$.

We get $c_{i_0}\cdot\sum_{i=1}^{t}\alpha_i - c_{j_0}\cdot\sum_{i=1}^{t}\alpha_i + \sum_{i=1}^{p}(\alpha_i)^2 = (c_{i_0} - c_{j_0})\cdot\sum_{i=1}^{t}\alpha_i + \sum_{i=1}^{p}(\alpha_i)^2 \ge -2\cdot\sum_{i=1}^{t}\alpha_i + \sum_{i=1}^{p}(\alpha_i)^2 =$

$-\sum_{i=1}^{t}\alpha_i - \sum_{i=1}^{t}\alpha_i + \sum_{i=1}^{p}(\alpha_i)^2 = -\sum_{i=1}^{t}\alpha_i + \sum_{i=t+1}^{p}\alpha_i + \sum_{i=1}^{p}(\alpha_i)^2$.

This is non negative since $\alpha_1,\cdots,\alpha_p$ are integers.The proof is completed for case 1.

2. $\max_{1\le i,j\le p} |c_i - c_j| \le 2$ condition may be broken when filled intervals appeared during the complete cycles.

Assume that $d^G_{k-1,i_1},\cdots,d^G_{k-1,i_h}$ are lengths of the filled intervals. So $c_{i_1} = d^G_{k-1,i_1},\cdots,c_{i_h} = d^G_{k-1,i_h}$ takeplace.Thus $d^G_{k-1,i_j} \le D$ for even d^G_{k-1,i_j} and $d^G_{k-1,i_j} \le D+1$ for odd d^G_{k-1,i_j} , $j \in \overline{1,h}$. For remaining intervals: $\max_{i,j\ne i_1,\cdots i_h} |c_i - c_j| \le 2$ is true; and $c_i \ge c_{i_j}$, for $i \ne i_j$, $j \in \overline{1,h}$.

Assume that $\alpha_1 \ge 0,\cdots,\alpha_t \ge 0,\quad \alpha_{t+1} < 0,\cdots,\alpha_p < 0$. Notice that $\alpha_{i_1},\cdots,\alpha_{i_h}$ can not be positive numbers ($\alpha_i = d_{k-1,i,1} - d^G_{k-1,i,1},\quad i = 1,\cdots,p$).

It follows from (1) that

$$\Delta D_k(A_G) - \Delta D_k(A) = \min_{1\le i\le t} c_i \cdot \sum_{i=1}^{t}\alpha_i + \sum_{i=t+1}^{p}\alpha_i c_i + \sum_{i=1}^{p}(\alpha_i)^2 \ge$$

$$\min_{1\le i\le t} c_i \cdot \sum_{i=1}^{t}\alpha_i + \sum_{j=1}^{h}\alpha_j c_j + \max_{\substack{t+1\le i\le p\\ i\ne i_1,\cdots,i_h}} c_i \cdot \sum_{\substack{i=t+1\\ i\ne i_1,\cdots,i_h}}^{p}\alpha_i + \sum_{i=1}^{p}(\alpha_i)^2 = \quad (\text{let } c_{i_0} = \min_{1\le i\le t} c_i \text{ and } c_{j_0} = \max_{\substack{t+1\le i\le p\\ i\ne i_1,\cdots,i_h}} c_i)$$

$$\sum_{j=1}^{h}\alpha_j c_j + c_{i_0}\cdot\sum_{i=1}^{t}\alpha_i + c_{j_0}\cdot\sum_{i=t+1}^{p}\alpha_i - c_{j_0}\cdot\sum_{j=1}^{h}\alpha_j + \sum_{i=1}^{p}(\alpha_i)^2 =$$

$$\sum_{j=1}^{h}\left(\alpha_j\cdot(c_j - c_{j_0})\right) + (c_{i_0} - c_{j_0})\cdot\sum_{i=1}^{t}\alpha_i + \sum_{i=1}^{p}(\alpha_i)^2 .$$

Thus

$$\Delta D_k(A_G) - \Delta D_k(A) = \sum_{j=1}^{h}\left(\alpha_{i_j}\cdot(c_{i_j} - c_{j_0})\right) + (c_{i_0} - c_{j_0})\cdot\sum_{i=1}^{t}\alpha_i + \sum_{i=1}^{p}(\alpha_i)^2 \qquad (2)$$

$(c_{i_0} - c_{j_0})\cdot\sum_{i=1}^{t}\alpha_i + \sum_{i=1}^{p}(\alpha_i)^2 \ge 0$ since $|c_{i_0} - c_{j_0}| \le 2$ and α_i are integers.At the same time $\sum_{j=1}^{h}\left(\alpha_{i_j}\cdot(c_{i_j} - c_{j_0})\right) \ge 0$, since $\alpha_{i_1},\cdots,\alpha_{i_h} \le 0$ and $c_{i_1},\cdots,c_{i_h} \le c_{j_0}$.

Therefore $\Delta D_k(A_G) \ge \Delta D_k(A)$.

Thus we have proven that (1) all branches of algorithm G are optimal in local steps: they provide maximum number of differing rows. Now we prove the converse: (2) all optimal partitions of intervals in each local step appear as a result of algorithm G.

(2) Suppose that $d^G_{k-1,1},\cdots,d^G_{k-1,p}$ are lengths of p intervals ofthe $(k-1)$-th column of the matrix. Let assume that $d^G_{k-1,i,1}$, $d^G_{k-1,i,1}$ are lengths of partitions of the i-th interval appearing as a result of algorithm G at the k-th step, and $d_{k-1,i,1}$, $d_{k-1,i,0}$ are lengths of optimal partitions provided by some algorithm at the k-th step.Let $\Delta D_k(A)$ and $\Delta D_k(A_G)$ denote the increase of objective function at the k-th step according to the optimal partition and partition by algorithm G respectively.It follows from (1) $\Delta D_k(A_G) = \Delta D_k(A)$. All we need to prove that the given optimal partition coincides with some brunch of G.

In (1) we have: $\Delta D_k(A_G) - \Delta D_k(A) = \sum_{i=1}^{p}\left(\alpha_i c_i + (\alpha_i)^2\right)$ where $\alpha_i = d_{k-1,i,1} - d^G_{k-1,i,1}$, $i = 1,\cdots,p$. Hence

$$\sum_{i=1}^{p}\left(\alpha_i c_i + (\alpha_i)^2\right) = 0 \qquad (3)$$

Consider possible cases.

1. $\max\limits_{1\le i,j\le p} | c_i - c_j | \le 2$.

Let

$c_{j_1} = \cdots = c_{j_{t_1}} = c$,

$c_{j_{t_1+1}} = \cdots = c_{j_{t_2}} = c+1$,

$c_{j_{t_2+1}} = \cdots = c_{j_p} = c+2$.

Again for simplification of notifications let assume that

$c_1 = \cdots = c_{t_1} = c$,

$c_{t_1+1} = \cdots = c_{t_2} = c+1$,

$c_{t_2+1} = \cdots = c_p = c+2$:

Putting all this into (3) we get

$$c\cdot\sum_{i=1}^{t_1}\alpha_i + (c+1)\cdot\sum_{i=t_1+1}^{t_2}\alpha_i + (c+2)\cdot\sum_{i=t_2+1}^{p}\alpha_i + \sum_{i=1}^{p}(\alpha_i)^2 =$$

$$c\cdot\left(\sum_{i=1}^{t_1}\alpha_i + \sum_{i=t_1+1}^{t_2}\alpha_i + \sum_{i=t_2+1}^{p}\alpha_i\right) + \sum_{i=t_1+1}^{t_2}\alpha_i + 2\cdot\sum_{i=t_2+1}^{p}\alpha_i + \sum_{i=1}^{p}(\alpha_i)^2 = \sum_{i=t_1+1}^{t_2}\alpha_i + 2\cdot\sum_{i=t_2+1}^{p}\alpha_i + \sum_{i=1}^{p}(\alpha_i)^2 = 0$$

$\sum\limits_{i=1}^{p}\alpha_i = 0$ implies $\sum\limits_{i=t_2+1}^{p}\alpha_i = -\left(\sum\limits_{i=1}^{t_1}\alpha_i + \sum\limits_{i=t_1+1}^{t_2}\alpha_i\right)$.

Hence

$$\sum_{i=t_1+1}^{t_2}\alpha_i + \sum_{i=t_2+1}^{p}\alpha_i - \sum_{i=1}^{t_1}\alpha_i - \sum_{i=t_1+1}^{t_2}\alpha_i + \sum_{i=1}^{p}(\alpha_i)^2 = -\sum_{i=1}^{t_1}\alpha_i + \sum_{i=t_2+1}^{p}\alpha_i + \sum_{i=1}^{p}(\alpha_i)^2 = 0$$

α_i are integers and so $|\alpha_i| \le 1$ for $i \in \overline{1,p}$;and $\alpha_i = 0$ for $i \in \overline{t_1+1, t_2}$ .Thus $\alpha_i = \begin{cases} 1 \\ 0 \end{cases}$ for $i \in \overline{1,t_1}$; $\alpha_i = \begin{cases} -1 \\ 0 \end{cases}$ for $i \in \overline{t_2+1, p}$ where equal numbers of positive and negativeones present. Without loss of generality we assume that no 0 α_i are there. Thus $d_{k-1,i,1} - d^G_{k-1,i,1} = 1$ and hence $d_{k-1,i,0} - d^G_{k-1,i,0} = -1$ for $i \in \overline{1,t_1}$; and

$d_{k-1,i,1} - d^G_{k-1,i,1} = -1$ and hence $d_{k-1,i,0} - d^G_{k-1,i,0} = 1$ for $i \in \overline{1, t_2 + 1, p}$. It follows that the $d^G_{k-1,i,1}$-length interval can not be filled by ones when $i \in \overline{1, t_1}$. Notice also that the lengths of the remaining $i \in \overline{t_1 + 1, t_2}$ intervals are the same in both partitions.

At the same time

$d^G_{k-1,i,1} - d^G_{k-1,i,0} = c$ for $i \in \overline{1, t_1}$ and

$d^G_{k-1,i,1} - d^G_{k-1,i,0} = c + 2$ for $i \in \overline{t_2 + 1, p}$.

Since c and $c + 2$ have the same parity it follows that all intervals in both groups come from either even or odd parts. Since as mentioned above $d^G_{k-1,i,1}$-length intervals are not filled by ones when $i \in \overline{1, t_1}$, then algorithm G had more than one choices, - so we had either (1)or (3)cases. Intervals that receive $c + 2$ extra ones are those that during the last stage of phase 1, received 2 new ones. These 2 ones could be distributed among any other intervals which in our case received c extra ones.

Now all we have to show that this choice made by G coincides with the optimal. Indeed:

$d_{k-1,i,1} - d_{k-1,i,0} = d^G_{k-1,i,1} + 1 - d^G_{k-1,i,0} + 1 = c + 2$ for $i \in \overline{1, t_1}$; and

$d_{k-1,i,1} - d_{k-1,i,0} = d^G_{k-1,i,1} - 1 - d^G_{k-1,i,0} - 1 = c$ for $i \in \overline{1, t_2 + 1, p}$.

So we get intervals where differences between ones are zeros are the same, thus have the same distribution.

2. $\max\limits_{1 \le i, j \le p} | c_i - c_j | \le 2$ condition may be broken if there exist intervals filled during the complete cycles. Let $d^G_{k-1,i_1}, \cdots, d^G_{k-1,i_h}$ are lengths of these intervals. For them $c_{i_1} = d^G_{k-1,i_1}, \cdots, c_{i_h} = d^G_{k-1,i_h}$ is true. Thus $d^G_{k-1,i_j} \le D$ for even d^G_{k-1,i_j}; and $d^G_{k-1,i_j} \le D + 1$ for odd d^G_{k-1,i_j}, where $j \in \overline{1, h}$. For remaining intervals $\max\limits_{i, j \ne i_1, \cdots i_h} | c_i - c_j | \le 2$ take place and $c_i \ge c_{i_j}$, for $i \ne i_j$ and $j \in \overline{1, h}$.

First we show that for the mentioned group of intervals $\alpha_{i_j} = 0$, $j \in \overline{1, h}$.

Recall (2)

$$\Delta D_k(A_G) - \Delta D_k(A) \ge \sum_{j=1}^{h} \left(\alpha_{i_j} \cdot (c_{i_j} - c_{j_0}) \right) + (c_{i_0} - c_{j_0}) \cdot \sum_{i=1}^{t} \alpha_i + \sum_{i=1}^{p} (\alpha_i)^2$$

If c_{j_0} appears among $c_{i_1}, \cdots, c_{i_h}$ then it will imply $c_{i_0} \ge c_{j_0}$ as $c_i \ge c_{i_j}$, for $i \ne i_j$ and $j \in \overline{1, h}$ and hence in (2) we will geta positive summand:

$(c_{i_0} - c_{j_0}) \cdot \sum_{i=1}^{t} \alpha_i + \sum_{i=1}^{p} (\alpha_i)^2 \geq \sum_{i=1}^{p} (\alpha_i)^2 > 0$ in case if a non zero α_i presents. Therefore $\Delta D_k(A_G) > \Delta D_k(A)$ due to $\sum_{j=1}^{h} \left(\alpha_{i_j} \cdot (c_{i_j} - c_{j_0})\right) \geq 0$. But this is acontradiction.

Thus $c_{i_j} - c_{j_0} < 0$ for $i \neq i_j$ and $j \in \overline{1,h}$.

If for some j, $\alpha_{i_j} < 0$ (it can not be positive) then in (2) we geta positive summand $\sum_{j=1}^{h} \left(\alpha_{i_j} \cdot (c_{i_j} - c_{j_0})\right) > 0$.

Therefore $\Delta D_k(A_G) - \Delta D_k(A) > 0$ as $c_{i_j} - c_{j_0} < 0$ and $(c_{i_0} - c_{j_0}) \cdot \sum_{i=1}^{t} \alpha_i + \sum_{i=1}^{p} (\alpha_i)^2 \geq 0$. This again leads to contradiction since $\Delta D_k(A)$ assumed optimal. Therefore $\alpha_{i_j} = 0$, $j \in \overline{1,h}$.

For simplification let $\alpha_j = 0$, $j \in \overline{1,h}$.

Suppose that

$$c_{h+1} = \cdots = c_{t_1} = c,$$

$$c_{t_1+1} = \cdots = c_{t_2} = c + 1,$$

$$c_{t_2+1} = \cdots = c_p = c + 2.$$

Putting into (3) we get:

$$c \cdot \sum_{i=h+1}^{t_1} \alpha_i + (c+1) \cdot \sum_{i=t_1+1}^{t_2} \alpha_i + (c+2) \cdot \sum_{i=t_2+1}^{p} \alpha_i + \sum_{i=1}^{p} (\alpha_i)^2 = 0$$

$$c \cdot \left(\sum_{i=h+1}^{t_1} \alpha_i + \sum_{i=t_1+1}^{t_2} \alpha_i + \sum_{i=t_2+1}^{p} \alpha_i \right) + \sum_{i=t_1+1}^{t_2} \alpha_i + 2 \cdot \sum_{i=t_2+1}^{p} \alpha_i + \sum_{i=1}^{p} (\alpha_i)^2 = 0$$

We have that $\sum_{i=h+1}^{p} \alpha_i = 0$ and $\sum_{i=t_2+1}^{p} \alpha_i = -\left(\sum_{i=h+1}^{t_1} \alpha_i + \sum_{i=t_1+1}^{t_2} \alpha_i \right)$.

Thus

$$\sum_{i=t_1+1}^{t_2} \alpha_i + \sum_{i=t_2+1}^{p} \alpha_i - \left(\sum_{i=h+1}^{t_1} \alpha_i + \sum_{i=t_1+1}^{t_2} \alpha_i \right) + \sum_{i=1}^{p} (\alpha_i)^2 = 0$$

$$\sum_{i=t_2+1}^{p} \alpha_i - \sum_{i=h+1}^{t_1} \alpha_i + \sum_{i=1}^{p} (\alpha_i)^2 = 0$$

α_i are integers which imply that $|\alpha_i| \le 1$ for $i \in \overline{h+1, p}$; and $\alpha_i = 0$ for $i \in \overline{t_1+1, t_2}$. Hence $\alpha_i = \begin{cases} 1 \\ 0 \end{cases}$ for $i \in \overline{h+1, t_1}$, $\alpha_i = \begin{cases} -1 \\ 0 \end{cases}$ for $i \in \overline{t_2+1, p}$, and equal numbers of positive and negative ones present.

The same reasoning as in the previous case will bring that the intervals are the result of a branch of G.

The theorem is proved.

However algorithm G does not provide the global optimum.

Consider the following example: $m = 13, n = 8, S = (11, \cdots, 11)$. All possible realizations of G leave two coinciding rows (see the matrix in left side in figure 1 below). However there exist a matrix in $U(S)$. In the matrix in side in figure 1, the fifth column does not correspond to any realization of G.

$A =$

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	0	1	1
1	1	1	1	0	1	1	1
1	1	1	1	0	1	1	0
1	1	1	0	1	1	1	1
1	1	1	0	1	1	1	0
1	1	0	1	1	1	1	1
1	1	0	1	1	1	0	1
1	0	1	1	1	1	1	1
1	0	1	1	1	1	0	1
0	1	1	1	1	1	1	1
0	1	1	1	1	0	1	1

$A^G =$

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	0
1	1	1	1	1	1	0	1
1	1	1	1	1	0	1	1
1	1	1	1	0	1	1	1
1	1	1	0	1	1	1	1
1	1	1	0	1	1	1	0
1	1	0	1	1	1	1	1
1	1	0	1	1	1	0	1
1	0	1	1	1	1	1	1
1	0	1	1	1	0	1	1
0	1	1	1	1	1	1	1
0	1	1	1	0	1	1	1

Figure 1

Further research concern the study of global properties of the given algorithm and evaluation of results, which is out of scope of the current paper.

Bibliography

[LintWilson, 2001] J.H. van Lint and R.M. Wilson, A Course in Combinatorics. Second edition, Cambridge University Press, Cambridge, 2001

[ChenDiaconisHolmesLiu, 2005], Y. Chen, P. Diaconis, S.P. Holmes, and J.S. Liu, Sequential Monte Carlo methods for statistical analysis of tables, Journal of the American Statistical Association 100 (2005), 109–120,

[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.

[Knuth, 1973] D. Knuth, TheArt of Computer Programming, vol.3. Sorting and Searching, Addison-Wesley Publishing Company, 1973.

[Sahakyan, 1995] H. Sahakyan, Hierarchical Procedures with the Additional Constraints, II Russia, with participation of NIS Conference, Pattern Recognition and Image Analysis: New Information Technologies, Ulianovsk, 1995, pp.76-78.

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*