ՀԱՅԱՍՏԱՆԻ ԳԻՏՈՒԹՅՈՒՆՆԵՐԻ ԱԶԳԱՅԻՆ ԱԿԱԴԵՄԻԱ
НАЦИОНАЛЬНАЯ АКАДЕМИЯ НАУК АРМЕНИИ
NATIONAL ACADEMY OF SCIENCES OF ARMENIA
ДОКЛАДЫ ԶԵԿՈՒՅՑՆԵՐ REPORTS

Հատոր Том Volume 117 2017 № 1

INFORMATICS

УДК 519

H. A. Sahakyan

Hypergraph Degree Sequence Approximation

(Submitted by corresponding member of NAS RA L.H.Aslanyan 1/XI 2016)

Keywords: *hypergraph degree sequence, (0,1)-matrices,approximation algorithms, greedy algorithm.*

Necessary and sufficient conditions for the existence of a simple hypergraph with given degree sequence is one of the known open problems in the graph theory domain [1-10]. The problem has its interpretation in terms of binary matrices. Existence/construction issues of related matrices with the given parameters/constraints were investigated and an approximation algorithm is constructed [11-12]. In this paper we achieve the performance assessment of that algorithm applying the random set cover technique.

Introduction. A hypergraph H is a pair (V,E), where $V=\{v,\dots,v_n\}$ is the vertex set of H, and E, the set of hyperedges, is a collection of non-empty subsets of V. A hypergraph H is simple if it has no repeated hyper-edges. The degree $d(v)$ of a vertex v of H is the number of hyperedges in H containing v. $d(H)=(d(v_1),\dots,d(v_n))$ is the degree sequence of hypergraph H.

The question of simple necessary and sufficient conditions for the existence of a simple hypergraph with the given degree sequence is a long-standing open problem. The problem has its interpretation in terms of binary matrices. We codethe hyper-edges of H with (0,1) sequences of length n such that j-th component of the sequence equals 1 if and only if j-th vertex of the hypergraph belongs to the given hyper-edge. Hence we get a (0,1) matrix, where the numbers of 1's in rows are cardinalities of hyper-edges; the number of 1s in j-th column is the degree of j-th vertex.Thus, the problem is equivalent to the existence of (0,1)-matrices with distinct rows and with given column sums (number of 1s in the columns). In general, (0,1)-matrices with prescribed row and column sums is a classical object, which appears in many branches of applied mathematics. For example, in Discrete Tomography (0,1) matrices serve for representation of discrete sets [13-15]. The projections of a matrix by the horizontal and vertical directions correspond to the row and column sums of the matrix. There is a known result by Ryser, who obtained a necessary and sufficient condition for a pair of vectors being the row and column sums of a (0,1)-matrix([16]), and a polynomial algorithm that constructs the matrix itself. The requirement of non-repetition of rows makes the problem hard. For such matrices both cases: existence of a (0,1) matrix with a given column sum and with or without row sum, - are algorithmically open problems [17],[18].We consider an optimization version of the problem – to find a (0,1) matrix with a given column sum and with maximal number of pairs of distinct rows, - this leads to a matrix with distinct rows in case when such matrix exists.To find an approximate solution of the problem a greedy algorithm is constructed in [11], which is optimal in local steps. Several properties of the algorithm and experimental results are given in [12]. In this research we estimate the performance of the algorithm using the greedy and randomized set cover technique.

The paper is organized as follows: a brief description of the greedy algorithm is given in Section 2 below. Section 3 is devoted to the evaluation of the algorithm's performance using the greedy set cover technique.

Approximation greedy algorithm for constructing (0,1)-matrices with distinct rows. Consider a (0,1)-matrix of size $m \times n$. Let $S = (s_1, \dots s_n)$ denote the column sum vector of the matrix, where s_j is the number of 1's in j –th column. $U(S)$ denotes the set of all (0,1)-matrices which have m distinct rows and have the column sum vector $S = (s_1, \dots s_n)$. Now we formulate two versions of the problem: existence and optimization.

(P1) Existence of a matrix in $U(S)$. Consider a matrix A of $U(S)$. Clearly any interchange of rows of A keeps the matrix in $U(S)$. Applying certain set of row interchanges we can transform A into another matrix of $U(S)$, in which s_1 ones of the first column are situated in the first $1, \dots, s_1$ positions, and thus form an interval. Then in the same way we can transform the obtained matrix into another one where s_2 ones of the second column compose two intervals (say $s_{2,1}$ and $s_{2,2}$ lengths, where $s_2 = s_{2,1} + s_{2,2}$) situated in the $1, \dots, s_{2,1}$ and $s_1 + 1, \dots s_{2,2}$ positions, respectively. Continuing this process we obtain alternating 1 and 0 intervals (possibly of 0 lengths) in each column. Rows i and j taken from different intervals are distinct, and rows within the same interval coincide with each other.We call this construction matrix of partitioned form. An illustration is in Figure 1 below.

Thus, if $U(S)$ is not empty then it contains at least one matrix of partitioned form. We will search solution of **(P1)** among the matrices of partitioned form, constructing the matrix column-by-column, and providing in each column the given number of ones. If in the last column the matrix has all one length intervals, then all rows are different. Generally, partitioning of intervals can be arbitrary, but it is reasonable to have some objective, for example certain quantitative characteristics leading to the matrix with distinct rows.

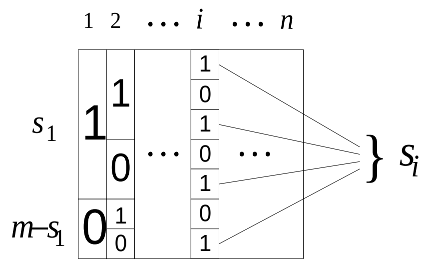

Figure1. The step-by-step partition matrix.

Let $U(S)$ denote the class of (0,1)-matrices of size $m\times n$ which have the column sum vector $S=(s_1,\dots s_n)$. In this way $U(S)\subseteq U(S)$. For a given $A\in U(S)$ let $D(A)$ denote the number of pairs of distinct rows of A.

Consider the following optimization problem:

(P2) **Find** $A_{opt}\in U(S)$ **such that** $D(A_{opt})=\max_{A\in U(S)} D(A)$

Obviously $\binom{m}{2}$ is the lowest upper bound for $D(A)$ and it is achievable for matrices of $U(S)$ only. Therefore if $U(S)$ is not empty, then solutions of *(P2)* are also solution of the existence problem *(P1)*. In this way *(P2)* is not easier than *(P1)*.

Below we give a brief description of the greedy algorithm G introduced in [11] for solving *(P2)***.** G constructs a matrix column-by-column starting from the first column and adding a column in each step.

Algorithm G**.** Without loss of generality we assume that $s_i\ge m-s_i, i=1,\dots,n$.

Step 1. Construction of the first column: we place s_1 ones in the first s_1 positions followed by $m-s_1$ zeros. We get two intervals: s_1-length interval of ones, and $(m-s_1)$-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 numbers contain all zeros. Thus the construction of the first column is in unique way: $\begin{cases} d_{1,1}^G=s_1 \\ d_{1,1}^G+d_{1,2}^G=m \end{cases}$. At this point we get $d_{1,1}^G\cdot d_{1,2}^G=s_1(m-s_1)$ pairs of differing (by the first position) rows.

Let we have constructed the first $k-1$ columns. In general, $(k-1)$-th column consists of 2^{k-1} intervals. Among them 0-length intervals are possible,-

these intervals cannot be used any longer. Assume that $(k-1)$ - th column consists of p non-zero length intervals denoted by $d^G_{k-1,1}, d^G_{k-1,2}, \ldots d^G_{k-1,p}$. 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 constructions can be arbitrary.

Step k.Construction of the k -th column: each $d^G_{k-1,i}$-length interval is partitioned into $d^G_{k-1,i,0}$ and $d^G_{k-1,i,1}$-length intervals filled by zeros and ones respectively such that $\sum_{i=i}^{p} d^G_{k-1,i,0} = m - s_k$ and $\sum_{i=i}^{p} d^G_{k-1,i,1} = s_k$.

The increase of objective function during the k -th step is:

$$\sum_{i=i}^{p} d^G_{k-1,i,1} \cdot d^G_{k-1,i,0}.$$

We will realize partitions having the goal to minimize length differences of intervals. The idea is in the following: if $s_k = m - s_k, k = 1, \ldots, n$, then in each step we would split every interval into 2 equal (± 1) parts and fill by zeros and ones respectively; this would lead to all one length intervals in logarithmic number (minimum possible [19]) steps.Furthermore, among all integer partitions of $d^G_{k-1,i} = d^G_{k-1,i,0} + d^G_{k-1,i,1}$, the largest product $d^G_{k-1,i,0} \cdot d^G_{k-1,i,1}$ is achieved when $d^G_{k-1,i,0} = d^G_{k-1,i,1}$. 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 we 1) distribute the extra ones among intervals keeping a "homogeneous" distribution; and then 2) split each of the remaining intervals into 2 equal parts– putting equal number of zeros and ones.

Theorem 1 [11]

(1) Algorithm G is optimal in local steps: 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.

Performance estimation of the algorithm G. In this section we will use the *greedy set cover technique* to evaluate the performance of G. Consider column-by-column constructions of a (0,1)-matrix of size $m \times n$, with s_i ones in the i -th column. There are $\binom{m}{s_i}$ possible placements of s_i ones in i -th column. Each placement will produce the same $s_i(m - s_i)$ number of pairs of different rows (differing by the i -th column). We enumerate these placements as: $1, 2, \ldots, \binom{m}{s_i}$. By the other hand $\binom{m}{2}$ is the maximum number of distinct pairs of rows. Enumerate these pairs as: $1, 2, \ldots, \binom{m}{2}$.

Now we construct (0,1)-matrix P in the following way: P has $\sum_{i=1}^{n}\binom{m}{s_i}$ rows grouped by $P_1, P_2, \ldots P_n$ blocks/submatrices. Each block P_i consists of $\binom{m}{s_i}$ rows, corresponding to the placements of s_i ones in i-th column. P has $\binom{m}{2}$ columns corresponding to the distinct pairs of rows of M (by the placements). Figure 2 below illustrates the construction.

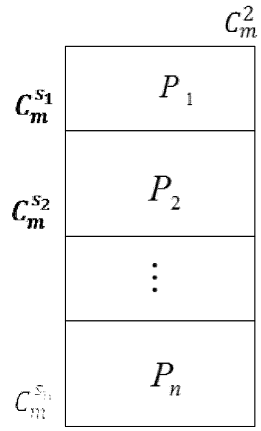

Figure 2. Structure of matrix P.

Let $p_{i,j,k}$ denote the element of (j,k)–th position (j-th row and k-th column) of submatrix P_i:

$$p_{i,j,k} = \begin{cases} 1, \mathit{if\ the\ k-the\ pair\ of\ rows\ of\ M\ is\ differing\ by\ the\ i-th\ column, and\ namely\ by\ the\ j-th\ placement\ of\ s_i\ ones} \\ 0, \mathit{otherwise} \end{cases}$$

In this way j-th row in P_i indicates the pairs which are differing by the j-th placement of s_i ones. Therefore, the *number of ones in each row of* P_i *equals* $s_i(m-s_i)$.

k-th column in P_i indicates those placements of s_i ones, which make distinct the k-th pair of rows of M. Therefore, the number of ones of k-th column is the number of placements of s_i ones making k-th pair of rows distinct by i-th position. Any pair of rows is distinct by i-th position if one the rows has 1, and the other has 0 in this position; or vice versa. In other positions the placement of ones is arbitrary. Thus, the *number of ones in every column of* P_i *equals* $2C_{m-2}^{s_i-1}$.

We get the following relation:

$$s_i(m-s_i)C_m^{s_i} = 2C_{m-2}^{s_i-1}C_m^2 .$$

Now we construct a matrix M, and consider the corresponding construction in P. The construction of each column of M is simply a selection of a row in P_i.

Construction of the first column of M means selection of a certain placement among all $\binom{m}{s_i}$ placements of s_i ones. As a result certain pairs of rows will be differing, and in this way the corresponding row of P_1 will be selected. Ones in this row will indicate $s_1(m-s_i)$ distinct pairs of rows in M. We separate this part in P and exclude it from the further considerations.

Suppose that $i-1$ columns of M are constructed with $s_1 s_2,\ldots s_{i-1}$ ones, and as a result certain pairs of rows are already differing. In Figure 3 these pairs are in the dashed part (without loss of generality, the differing rows are shown in the first part of P). At this moment outside the dashed part there is no 1s. Let t_{i-1} denote the number of row pairs of M which are not yet distinct after the first $i-1$ steps. Now we construct i-th column of M in P, namely, in P_i.

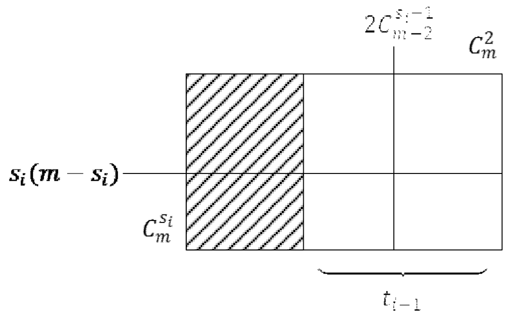

Figure 3. Construction of sub-matrix P_i.

We calculate the number of ones that the not-dashed part should contain, this equals to $t_{t-1}2C_{m-2}^{s_i-1}$. Therefore, $\dfrac{t_{t-1}2\binom{m-2}{s_i-1}}{\binom{m}{s_i}}$ is the average number of ones in rows. It follows that there is a row with at least $\dfrac{t_{t-1}2\binom{m-2}{s_i-1}}{\binom{m}{s_i}}$ ones (or $\left\lceil \dfrac{t_{t-1}2\binom{m-2}{s_i-1}}{\binom{m}{s_i}} \right\rceil$ ones, since we deal with integer numbers). We select this row as a construction of i-th column of M. This means that the construction of i-th column of M will produce at least $\left\lceil \frac{t_{t-1} 2\binom{m-2}{s_i-1}}{\binom{m}{s_i}} \right\rceil$ new pairs of distinct rows.Thus:

$$t_{t-1}-t_i \geq \frac{t_{t-1} 2\binom{m-2}{s_i-1}}{\binom{m}{s_i}} = t_{i-1} \frac{2(m-2)!s_i!(m-s_i)!}{(s_i-1)!(m-s_i-1)!m!} = t_{t-1} \frac{s_i(m-s_i)}{\binom{m}{2}}$$

The achieved important property is stated by the following lemma.

Lemma. Let we construct a matrix M in column-by-column manner, and at i -the step we have:

a) first $i-1$ columns /by some constructions/. Let t_{t-1} denote the number of pairs of rows in M which are not yet differentiated,

b) s_i - the number of ones in i -th column of M, then there is a placement of s_i ones in the i -th column such that new pairs of rows will compose at least $\frac{s_i(m-s_i)}{\binom{m}{2}}$ part of t_{t-1}.

We notice that the result does not depend on the constructions of the first $i-1$ columns, as well as on the order of components of the vector S. Summarizing, we achieve the following estimate of the greedy algorithm:

Theorem 2. For a given vector $S=(s_1, s_2, \ldots s_n)$ let M be a binary matrix of size $m \times n$ with the column sum S, constructed by the greedy algorithm G.

Then the part of not differentiated pairs of rows M is at most $\xi_1 \xi_2 \ldots \xi_n$, where $\xi_i = 1 - \frac{s_i(m-s_i)}{\binom{m}{2}}$.

Further examples are considered in order to understand how tight the estimate is.

Institute for Informatics and Automation Problems of NAS RA
e-mail: hsahakyan@sci.am

H. A. Sahakyan

Hypergraph Degree Sequence Approximation

Necessary and sufficient conditions for the existence of a simple hypergraph with the given degree sequence is one of the known open problems in the graph theory domain. The problem has its interpretation in terms of binary matrices.

Existence/construction issues of related matrices with the given parameters/constraints were investigated and an approximation algorithm is constructed in earlier works. In this paper we achieve the performance assessment of that algorithm applying the random set cover technique.

Հ. Ա. Սահակյան

Հիպերգրաֆի աստիճանային հաջորդականության մոտարկում

Տրված աստիճանային հաջորդականությամբ պարզ հիպերգրաֆի գոյության անհրաժեշտ և բավարար պայմաններ գտնելու խնդիրը գրաֆների տեսության հայտնի բաց խնդիրներից մեկն է: Խնդիրը ունի իր մեկնաբանումը բինար մատրիցների տերմիններով: Նախորդող աշխատանքներում հետազոտվել են տրված սահմանափակումներով մատրիցների գոյության / կառուցման հարցերը և կառուցվել է ապրոքսիմացիոն ալգորիթմ: Ներկա աշխատանքում տրվում է այդ ալգորիթմի աշխատանքի գնահատականը՝ բազմությունների ծածկույթի մեթոդի կիրառմամբ:

А. А. Саакян

Аппроксимация последовательности степеней вершин гиперграфа

Задача нахождения необходимых и достаточных условий существования простого гиперграфа по данной последовательности степеней вершин является известной открытой задачей теории графов. Задача имеет простую интерпретацию в терминах бинарных матриц. В предыдущих работах были исследованы задачи существования и построения бинарных матриц с данными ограничениями и построен аппроксимационный алгоритм. В данной статье приводится оценка работы аппроксимационного алгоритма путем привлечения метода покрытия множеств.

References

1. *Berge C* Hypergraphs: Combinatorics of Finite Sets, North-Holland, 1989.
2. *Colbourn Charles J., Kocay W.L., Stinson D. R.* – Discrete Applied Mathematics. 1986. V. 14. P. 239-254.
3. *Billington D.* – The Journal of Combinatorial Mathematics and Combinatorial Computing. 1988. 3. P. 71-91.
4. *Bhanu Murthy N. L., Murali K. Srinivasan.* –Linear Algebra Appl.2002. V. 350. P. 147–170
5. *Kocay William, Li Pak Ching* – Ars Combin. 2007. V. 82. P. 145–157. MR 2292314 (2007k:05049)
6. *Behrens S, Erbes C., Ferrara M., Hartke S., Reinger B., Spinoza H., Tomilson Ch.* – Electronic Journal of Combinatorics 2013. V. 20. N. 4.
7. *Aslanyan L., Sahakyan H.* – Electronic Notes in Discrete Mathematics. 2006 V. 27. P. 3-4/110.
8. *Sahakyan H.* – Discrete Applied Mathematics, 2009. V. 157. P. 2191-2197.
9. *Sahakyan H.* – Discrete Applied Mathematics. 2014. V. 163, part 2. P. 205-213.

10. *Sahakyan H.* – Applied Mathematical Sciences. 2015. V. 9. N. 5. P. 243-253.
11. *Sahakyan H.* – Information Theories and Applications. 2010-11. V. 17. N 2. P. 124-137.
12. *Sahakyan H., Aslanyan L.* – Information Technologies & Knowledge. 2011. V.5. N 1. P. 55-66.
13. *Gardner R. J., Gritzmann P.*, – Transactions of the American Mathematical Society, 1997. V. 349. N 6. P 2271-2295. S 0002-9947(97)01741-8.
14. *Herman, G.T.,* Kuba, A. (eds.): Advances in Discrete Tomography and its Applications. BirkhËauser. Boston. 2007.
15. *Barcucci, E., Del Lungo, A., Nivat, M., Pinzani,* R. –Theor. Comput. Sci. 1996.V. 155. P. 321–347.
16. *Ryser.* Combinatorial Mathematics. 1966.
17. *Sahakyan H.* – Ninth International Conference on Computer Science and Information Technologies, Revised Selected Papers. IEEE conference proceedings. November. 2013.
18. *Aslanyan L., Gronau H.-D., Sahakyan H., Wagner P.* –CSIT 2015, Revised Selected Papers, IEEE conference proceedings, P. 47-52, **DOI:**10. 1109/ CSITechnol.2015.7358249.
19. *Knuth D.* The Art of Computer Programming. V.3. Sorting and Searching, Addison-Wesley Publishing Company. 1973.