

## MATRIX ALGORITHM OF SOLVING GRAPH CUTTING PROBLEM

V.K. Pogrebnoy

TPU Institute «Cybernetic centre»

E-mail: vk@ad.cctpu.edu.ru

*Matrix algorithm of solving graph cutting problem has been suggested. The main algorithm points based on matrix graph presentation were considered. Formalization of the main algorithm procedures – defining estimations for selecting relocatable matrix elements and matrix conversion by reciprocal transfer of columns and lines was given. Algorithm operation was considered by the example of data transmission graph between the stations of local computer system network.*

### Introduction

The problem of graph cutting into minimally connected parts may be referred to a number of classical problems of graph theory widely used in practical applications. Among the earliest and the most developed applications is the problem of cutting computer circuit given as a graph [1] or in the form of more adequate model – hypergraph [2]. For an ordinary graph [3] the problem of cutting is stated in the following way. Graph  $G=(S, V)$  should be cut into parts  $G_f=(S_f, V_f)$ ,  $f=1, 2, \dots, F$ , where  $F$  is the number of parts into which the graph is cut;  $S_f$  is the vertex set belonging to  $f$  part;  $V_f$  is the set of arcs incident to vertices  $S_f$ .

The set of parts  $B(G)$  is called cutting of graph  $G$ , if

$$\begin{aligned} \forall G_f \in B(G) [G_f \neq \emptyset \ \& \ \cup G_f = G]; \\ \forall G_f, G_q \in B(G) [G_f \neq G_q \ \& \ S_f \cap S_q = \\ = \emptyset \ \& \ V_f \cap V_q = V_{f_q}], \ f, q = 1, 2, \dots, F. \end{aligned}$$

Here  $V_{f_q}$  is the set of arcs connecting pairs of vertices one of which belongs to the part  $G_f$ , and another one –  $G_q$ .

Let us denote  $|V_{f_q}|=g_{f_q}$  and call it by a number of arc attachment of parts  $G_f$  and  $G_q$ . Then a number of arc attachment of cutting graph  $G$  is determined by  $g$ :

$$g = \sum_{f=1}^F \sum_{q=1}^F g_{f_q}, \ f \neq q.$$

Traditional criterion of solving the problem of cutting graph  $G$  is minimization of a number of arc attachment  $g$  at restriction per vertex quantity in parts  $G_f$ .

Among the known algorithms of graph cutting there are exact algorithms using techniques of solving the problems of discrete programming and the approximate ones. The approximate algorithms became more widespread. Among them the sequential, iteration and compound algorithms are singled out.

In approximate algorithms of sequential type firstly, the graph vertex is selected by a certain criterion then other vertices are attached to it till the first part obtaining. Then the second part and next parts are formed from the rest graph vertices till complete cutting.

Iteration algorithms take a certain cutting obtained, for example, by one of sequential algorithms as the original one and then vertices are exchanged from one part to another in serially connected pairs of parts so that performance criterion quality is improved.

The analysis of the given algorithm results in the following conclusions:

- presence of a large variety of algorithms which is generated by a tendency to take into account the specific character of the investigated objects and graphs used for their description;
- algorithm alignment on operation directly with a list of vertices, arcs and their attributes;
- performing operations of optimization of arc attachment between two parts of graph is of local character and «does not see» what happens to arc attachment between other parts of cutting.

The first two conclusions reflect both positive and negative properties of algorithms. So, accounting specific character results in algorithm uniqueness but allows increasing their efficiency. Using lists is preferable for graphs with a large number of vertices and small number of arcs. As for the third conclusion it reflects the disadvantage of such algorithm which is difficult to be surmounted. It is when optimization is fulfilled by local enhancements of arc attachments inside an ordinary pair of cutting parts.

The cutting algorithm proposed in the given paper is based on graph representation by vertex connection matrix and more complete analysis of estimating efficiency of vertex assignment into subgraphs of cutting.

### The fundamentals of matrix algorithm

The cutting problem solution is considered for an ordinary weighted graph  $G=(S, V, \mathbf{R})$  represented by the matrix  $\mathbf{R}=\|r_{ij}\|$ , where  $r_{ij}$  is the weight of the arc  $v_{ij} \in V$ ,  $i, j=1, 2, \dots, n$ . Such graph is, for example, a data transfer graph [4] between vertices  $s_i \in S$ , which correspond in this case to the stations of local area network and weights  $r_{ij}$  are the volumes of data transferred between stations  $s_i$  and  $s_j$  in network. Matrix  $\mathbf{R}$  is symmetrical as the weight  $r_{ij}$  includes the volume of data which transferred from station  $s_i$  to  $s_j$  and from  $s_j$  to  $s_i$ .

Constructing local network of computer system on the basis of several backbones the problem of station distribution on backbones so that the volume of data transferred between stations connected to different backbones is minimal occurs. Such problem fits to the problem of graph cutting into minimal subgraphs. Vertices included into one subgraph correspond to the stations connected to on backbone.

The result of solving cutting problem may be presented by the matrix  $\mathbf{R}$  divided into blocks as it is shown in Fig. 1. Each  $f$  block is a connection matrix of subgraph  $G_f$  vertices. Elements of matrix  $\mathbf{R}$  being outside the blocks determine a number of arc attachment of cutting graph  $G$  and form a section area. Matrix shown in Fig. 1 relative to matrix  $\mathbf{R}$  may be considered as a matrix – pattern specifying cutting conditions – graph dimension, block composition indicating the dimension of each of them, block binding to numbers of lines and columns of the pattern. Blocks are considered in this case as potential targets of graph vertices at their inclusion into proper subgraph.

If such pattern is overlapped on matrix  $\mathbf{R}$  of graph  $G$  the original cutting in which graph vertex numbers coincide with numbers of lines and columns of the pattern is obtained. In Fig. 1 numbers of lines and columns of the pattern are shown on the left and on the top respectively and numbers of vertices are shown on the right and below. In this case index sets  $J_f$  and  $J_q$  include numbers of lines and columns of the pattern and  $J_f^*$  and  $J_q^*$  – of matrix  $\mathbf{R}$ . For initial cutting  $J_f = J_f^*$ ,  $J_q = J_q^*$ .

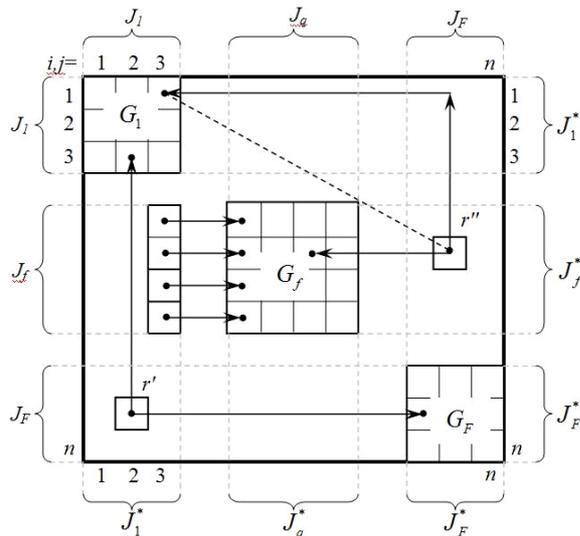


Fig. 1. Matrix-pattern for cutting problem

Quality of cutting for the considered application is determined by a sum of weights of arc attachment of cutting that corresponds to a sum of elements of matrix  $\mathbf{R}$  situated in the range of pattern section. Let us denote the sum of weights by value  $r$  and element set of section region by value  $V_c$ . It follows from this that  $r = \sum_{v_{ij} \in V_c} r_{ij}$ .

It is possible to decrease value  $r$  by equivalent conversion of matrix  $\mathbf{R}$  moving elements from set  $V_c$  with higher weights into the region of pattern blocks. Matrix  $\mathbf{R}$  and matrix  $\mathbf{R}_k$  obtained as a result of  $k$  transformation are equivalent if each of them is the vertex matrix of one and the same graph  $G$ . On the basis of this the cutting problem consists in conversion of matrix  $\mathbf{R}$  to the form in which a sum of weights of section region elements is minimal and a sum of weights of block region elements is maximal, respectively, while moving elements from section region into block region and saving the equivalence of matrix  $\mathbf{R}$ .

To implement this approach it is necessary to develop a policy of selection and moving elements of matrix  $\mathbf{R}$  fulfilling the conditions of equivalence for solving cutting problem. A proper algorithm is called matrix and includes two main procedures:

- determination of estimates for selecting relocatable elements;
- equivalent conversion of matrix  $\mathbf{R}$  implementing the selected relocation.

Implementing this policy two variants of matrix algorithm are singled out. In the first pattern blocks are considered as free places for assignment of selected elements. The second variant of algorithm uses pattern overlap on the matrix  $\mathbf{R}$ . The initial cutting variant obtained in this case is improved by reciprocal element relocation between section and block regions. The second simpler variant of algorithm is suggested further.

#### Matrix element relocation

Relocation is performed over the elements of matrix  $\mathbf{R}$  on which matrix – pattern is overlapped. The necessity of element relocation is stipulated by a criterion of solving cutting problem according to which a minimal sum of element weights should remain in section region. At pattern overlapping on matrix  $\mathbf{R}$  we obtain the initial cutting variant. So, operations of element relocation are performed in conditions when all places in blocks are occupied. Therefore, to perform the relocation, the target for moved elements should be released. If the released elements are not carried out of the matrix for saving and arranged on the places of moved elements then reciprocal element relocation occurs; such rule of relocation fits to the suggested variant of matrix algorithm.

Matrix element may move along a line or a column as it is shown in Fig. 1 for element  $r'$ , situated in the 2<sup>d</sup> column of the matrix. If element moves with the change of line and column as it is shown for the element  $r''$  situated in  $(n-1)$  column then two relocations are fulfilled, for example, firstly, along the column and then on a line or on the contrary.

To save matrix equivalence the proper line and column should be moved together with element relocation. If relocation is performed on a line then a column and appropriate line are moved. So relocation of element  $r'$  into block  $G_f$  results in transfer of 2<sup>d</sup> column into  $(n-2)$  column and 2<sup>d</sup> line into  $(n-2)$  line. Similarly, at relocation of element  $r'$  along the column the  $(n-1)$  line is transferred into 3<sup>d</sup> line and the  $(n-1)$  column is transferred into 3<sup>d</sup> column.

An object of relocation is often not one but several elements. A number of elements are determined as a rule by dimension of the block where they move. Relocation of 4 elements of the 3<sup>d</sup> column into the 1<sup>st</sup> column of the block  $G_f$  with the dimension  $4 \times 4$  is shown in Fig. 1. One of relocated elements falls on block diagonal element in this case it is the 1 element from the top. If this element has the weight  $r_{ij} > 0$  then it is transferred to the place of diagonal element of relocated column. Similarly, if nonzero element falls on diagonal element at

line relocation then it is transferred to diagonal element of the relocated line. Such rule of interaction of non-zero element and diagonal element is also used at transfer of lines and columns with released elements.

### Element selection for relocation

On the basis of tendency to move elements with maximal total weights from section region into block region let us introduce preference judgment of elements for their reciprocal transfer. For this purpose each column of matrix is divided into groups of elements according to their belonging to the lines of one block so that  $f$  group of elements corresponds to lines  $J_f$ , belonging to block  $G_f$ , for example, for the matrix in Fig. 1, 1<sup>st</sup> group of all columns consists of first three elements and the last one – of elements  $\{n-2, n-1, n\}$ . At such partitioning one of the groups of each column belongs to a block and all the rest belong to section region.

Let us introduce the following notations. Numbers of lines and columns of matrix-pattern are denoted by indices  $i$  and  $j$ ;  $j$  is the number of column or line moved into block;  $i$  is the number of column or line excluded of the block. Block numbers of matrix-pattern are denoted by indices  $f$  and  $q$ .  $q$  is the number of block into which column  $j$  is relocated;  $f$  is the number of block from which column  $j$  is excluded;  $J_q$  is the set of line numbers of block  $q$ ;  $J_f$  is the set of line numbers of block  $f$ .

Preference judgments are calculated relative to the groups of elements situated in section region. Sum of weights of elements in group  $J_q$  of column  $j$  which is transferred into block  $q$  is denoted by value  $\tilde{\alpha}_{qj}$ ,

$$\tilde{\alpha}_{qj} = \sum_{i \in J_q} r_{ij}, \quad \alpha_{qji} = \tilde{\alpha}_{qj} - \xi_{qji}. \quad (1)$$

Here  $\xi_{qji}$  is the weight  $r_{ij}$  of element  $(i,j)$  of group  $J_q$  which at transfer of column  $j$  into block  $q$  on the place of column  $i$ , falls on its diagonal element;  $\alpha_{qji}$  is the sum of weights of elements in column  $j$  after its relocation into block  $q$  on the place of column  $i$ .

Similarly for column  $i$  which moves into block  $f$  on the place of column  $j$ , the values  $\tilde{\alpha}_{fi}$  and  $\alpha_{fji}$  are calculated:

$$\tilde{\alpha}_{fi} = \sum_{i \in J_f} r_{ij}, \quad \alpha_{fji} = \tilde{\alpha}_{fi} - \xi_{fji}. \quad (2)$$

Decrease of element weight sums which occurs in blocks  $q$  and  $f$  excluding appropriate columns  $i$  and  $j$  from them is determined by values  $\beta_{qi}$  and  $\beta_{fj}$ :

$$\beta_{qi} = \sum_{i \in J_q} r_{ij}, \quad \beta_{fj} = \sum_{i \in J_f} r_{ij}. \quad (3)$$

The diagram of reciprocal transfer of columns  $j$  and  $i$  and determination of proper estimates in agreed notations is given in Fig. 2. Diagonal elements of columns  $i$  and  $j$  in blocks  $q$  and  $f$  are hatched. Elements of columns  $i$  and  $j$  which fall on diagonal elements in blocks being transferred are marked out as well. Before column transfer the elements marked out in them move in the column to the place of diagonal element that is shown by arrows in Fig. 2. The places of marked out elements are taken as diagonal elements after column transfer.

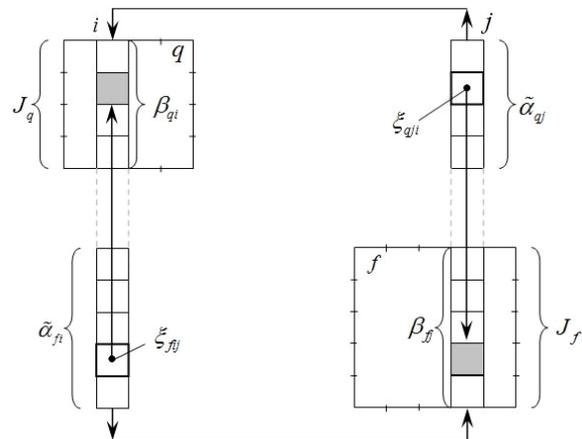


Fig. 2. Diagram of column reciprocal transfer

Estimates introduced in (1)–(3) allow obtaining sum preference judgments  $\mu_{ij}$  for pairs of columns  $i$  and  $j$  relative to their reciprocal transfer,

$$\mu_{ij} = \alpha_{qji} - \beta_{fj} + \alpha_{fji} - \beta_{qi}. \quad (4)$$

Estimates  $\mu_{ij} > 0$  show how much element sum weight increases in blocks after reciprocal transfer of columns  $j$  and  $i$  or, respectively, how much element sum weight decreases in section region.

Estimates  $\mu_{ij}$  are calculated only for elements of pattern section region as element move in blocks does not change the sum of their weights. If matrix  $\mathbf{R}$  is symmetrical in this case then estimates  $\mu_{ij}$  are calculated for one part of section region – top or bottom.

### Matrix algorithm and example of its application

After presentation of graph by adjacency matrix  $\mathbf{R}$  algorithm operation includes fulfillment of the following stages.

1. Pattern formation and its overlap on matrix  $\mathbf{R}$ . Pattern is formed on the basis of information about a number of cutting blocks, dimension of each of them and their place in the pattern. Places of block situation in the pattern are determined in a random sequence regardless of block dimension. As a result of pattern overlapping on the matrix  $\mathbf{R}$  the initial variant of cutting in which  $J_f = J_f^*$ ;  $J_q = J_q^*$ ;  $f, q = 1, 2, \dots, F$  is obtained.
2. Calculation of estimates  $\mu_{ij}$ . Estimates are calculated by the expressions (1)–(4). If matrix  $\mathbf{R}$  is symmetrical then calculation is carried out for one part of section region symmetrical relative to matrix diagonal – top (over diagonal) and bottom (under diagonal). Among estimates  $\mu_{ij} > 0$  a maximal one is selected and appropriate columns  $i$  and  $j$  are preferred for their reciprocal transfer. In this case the conversion to fulfilling stage 3 is carried out. Without estimates  $\mu_{ij} > 0$  algorithm stops operating and fixes the obtained variant of cutting.
3. Reciprocal transfer of columns  $i$  and  $j$ . Columns and proper lines are transferred by the rules given above and the diagram given in Fig. 2. Reciprocal transfer of lines and columns is attended by updating of index sets  $J_f^*$  and  $J_q^*$ . After that the conversion to fulfilling the second stage of algorithm is carried out.

Let us show algorithm operation by the example of weighted ordinary graph containing 10 vertices. The graph should be divided into 3 subgraphs one of which contains 4 vertices and two other ones contain 3 vertices. Matrix  $\mathbf{R}$  of the given graph and matrix – pattern constructed for the given cutting conditions matched with it are given in Fig. 3, a. The initial variant of cutting obtained in this case is characterized by element weight sums in blocks (numerator) and in section region (denominator) equal 15/50.

To improve the ratio 15/50 the estimates  $\mu_{ij}$  are calculated; the estimate with maximal value is selected among them. In this case it is estimates  $\mu_{5,3}=\mu_{9,3}=9$ . Let us select any of them, for example  $\mu_{5,3}$  that assumes relocation of element (2,5) with weight  $r_{2,5}=2$  into block  $J_1$  and element (4,3) with weight  $r_{4,3}=7$  into block  $J_2$ . To relocate these elements the reciprocal transfer of columns  $j=5$  and  $i=3$  and appropriate lines is carried out. The result of transfer is given in Fig. 3, б. The ratio of weight sum is equal to 24/41.

$\mu_{9,1}=14$  is maximal estimate  $\mu_{ij}$  for the given variant of cutting. After reciprocal transfer of column  $j=9$  and  $i=1$  the cutting variant given in Fig. 3, в, with weight sum ratio 38/27 turns out.  $\mu_{8,6}=3$  is maximal estimate  $\mu_{ij}$  in this case. As a result of reciprocal transfer of columns  $j=8$  and  $i=6$  we obtain cutting given in Fig. 3, г, for which there are no estimates  $\mu_{ij}>0$ . Therefore, algorithm completes its operation with variant of cutting  $J_1^*=\{0,9,2,5\}$ ,  $J_2^*=\{4,3,8\}$ ,  $J_3^*=\{7,6,1\}$  and ratio of weight sum 41/24.

If the graph used as an example is interpreted as a graph of data transfer [4] the vertices of which are stations of local network  $s_i, i=0,1,2,\dots,9$ , and arc weights  $r_{ij}$  correspond to volumes of data transferred between stations  $s_i$  and  $s_j$  then as a result of solving cutting problem we obtain three subsets of stations  $\{s_0, s_9, s_2, s_5\}$ ,  $\{s_4, s_3, s_8\}$ ,  $\{s_7, s_6, s_1\}$ , each of that is connected to one of three network backbones. At such variant of network construction the majority of data, in this example it is 41 units, is

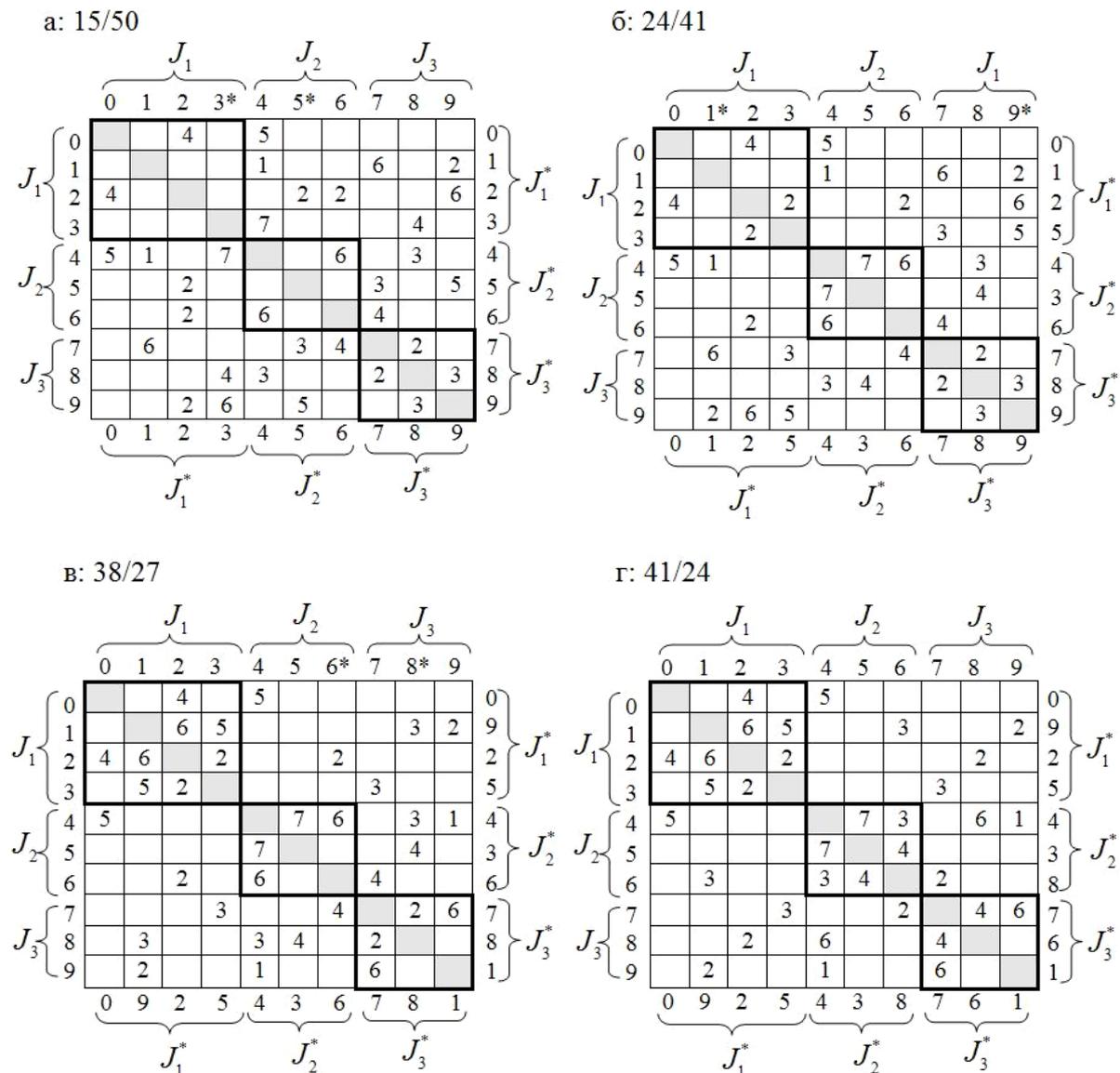


Fig. 3. Example of solving the problem of graph cutting

transferred between stations inside sets loading appropriate backbone. In this case data may be transferred simultaneously (parallel) in all backbones. Other data at the rate of 24 units are transferred between stations connected to different backbones. At such transfers two or three backbones are loaded simultaneously that results in increasing network load and, respectively, time for data transfer.

#### Conclusion

Advantages of suggested algorithm in comparison with the known ones operating with arc list are stipulated by graph presentation in matrix form at which section region is specified in explicit form. It allows easily observing («seeing») the changes occurring in section region at matrix conversion and calculating preference judgments for fulfilling regular transformation on the basis of analysis of current state of the whole section region.

The simplest scheme of element relocation from section region into block region based on reciprocal transfer of columns and appropriate lines is implemented in algorithm. Other more complicated schemes connected, for example, with the formation of reloca-

tion close circuits are the subject of further investigations and they are not considered in the given article.

Matrix algorithm is applicable for graphs with asymmetrical matrix including an unweighted one. It is obvious that use of this algorithm turns out to be more preferable for graphs with high degree of vertices connectivity as the volume of calculations carried out by algorithm does not depend on coefficient of matrix filling with nonzero elements.

Along with achieving the main goal – formalization of matrix algorithm of solving cutting problem the matrix technique has another no less important value. It consists in the fact that matrix form of cutting problem presentation creates possibilities for deeper understanding of its nature, improving analysis techniques of section region at optimization, studying algorithm dependence on various strategies of determining estimates for making decision.

Matrix method may be taken as a basis of algorithm development for various types of graphs and cutting conditions. In particular, bipartite graphs are of interest; their matrix presentations, in comparison with the examined ones, assume more efficient operations of matrix conversion when solving cutting problem.

#### REFERENCES

1. Shtein M.E., Shtein B.E. Methods of computer-aided design of digital equipment. – Moscow: Sovetskoe Radio, 1973. – 296 p.
2. Kornienko A.V., Pogrebnoy V.K. Model and algorithm for cutting digital computing devices into functional blocks // *Upravlyayushie Systemy i Mashiny*. – 1976. – № 5. – P. 94-98.
3. Kristofides N. Graph theory. Algorithm approach. – Moscow: Mir, 1978. – 432 p.
4. Pogrebnoy A.V. Determining data transfer volumes in computer system network for the specified model of program load // *Bulletin of the Tomsk Polytechnic University*. – 2007. – V. 310. – № 3. – P. 103–107.

*Received on 31.10.2007*