INFORMATICA, 2001, Vol. 12, No. 2, 199–220

The Method for Solving a Piecewise-Linear Multicommodity Flow Problem

Gediminas DAVULIS
Lithuanian University of Law
Ateities 20, 2057 Vilnius, Lithuania
e-mail: fmtk@ltu.lt

Received: December 2000

Abstract. The paper deals with a flow distribution problem with a piecewise-linear cost function. The problem is formulated as a piecewise-linear programming problem which is not separable with respect to separate variable group. The method for solving this problem is based on the extension of the idea of the simplex method to the class of non-separable piecewise-linear problems. It secures finding of a local solution to the problem after a finite number of iterations. The method uses the peculiarities of the problem constraints that make it possible to decompose the matrix of constraints to smaller ones and thus to diminish the volume of calculations.
Key words: piecewise-linear function, critical line, vertex, transport flow, graph, incidence matrix.

1. Introduction

The multicommodity flow problem with a nonseparable piecewise – linear arc cost function ir considered. The problem can be reckoned as a piecewise-linear programming problem. Their property that, analogously to a linear programming problem, the solution of these problems belongs to a finite set of points, is very important. Geometrically these points can be expressed as vertices of a polyhedron that are defined in the same way as in linear programming. However, in contrast to linear programming problems, these vertices may also be inner points of the objective function definition domain. That is why it is natural to strive for the methods that solve the problem considered to be finite, i.e., to ensure finding of the solution (in the convex case) or the local solution (in the general case) affer a finite number of iterations.

Individual cases of the problem considered have been well explored and finite algorithms are known for solving them. If the piecewise-linear cost function in convex and separable, than the problem can be reduced to a linear problem with additional constraints (Assad, 1978; Kennington and Helgasson, 1980; Bertsekas and Tseng, 1988) or to a problem with diserete variables (Kennington and Helgasson, 1980; Bertsekas and Tseng, 1988). To solve the problem with network arc capacity constraints, i.e., the classical multicommodity flow problem, decomposition methods are used (Bertsekas and Tseng, 1988; Golshtein and Sokolov, 1995; 1997).

Generalizations of decomposition and potential methods to a nonlinear case as well as linear approximations and other famous methods, applied in solving nonlinear transport problems (Magnanti and Wong, 1984; Minoux 1988, Melamed 1991), in the considered problem ensure, at best, only the convergence to the local solution and are not finite. Only the method branch-and-bound (Gendron and Crainic, 1994; Zaslavskij and Lebedev, 1995) may serve as an exception. However, this method in usually used to solve problems with complex objective and constraint functions.

In this paper, a finite method for solving the piecewise-linear multicommodity flow problem is presented. It guarantees finding of the local solution affer a finite number of iferations. The method disregards the arc cost function convexity and continuity, taking into account the specific features of constraints of the problem.

2. Statement of the Problem

A transport network presented in the form of a finite connected oriented graph which consists of T nodes and L arcs is given. The flow of $\bar{y}_i^j$ different kinds of products transported in a transport network has to be distributed so that the summary expenditure on transportation be least. An x_i^j-dimensional load vector $a_j = (a_j^1, \ldots, a_j^n)$ to each network node is assigned. Some volume a_j^k of the kth product has to be transported from the j-th node if $a_j^k > 0$, and to it if $a_j^k < 0$. The load of a network arc is expressed by an n-dimensional vector $x_l = (x_l^1, \ldots, x_l^n)$. The arc load by some product is expressed by a positive number, if the product is transported in the direction of arc orientation, or by a negative number, if it is transported in the opposite direction. The transportation expenditure via an arc depends on the summary quantity of the arc load and this dependence is expressed by a non-decreasing piecewise-linear function of one-variable, i.e., $f_i(x_i) = \bar{f}_i(w_i)$, where $w_i = \sum\limits_j |x_i^j|$. The considered problem can be described as a piecewise-linear optimization problem:

$$\min_{x \in \boldsymbol{X}} F(x) = \sum_{l}^{L} f_l(x_l), \tag{1}$$

$$Sx = A, \tag{2}$$

where $x = (x_1, \ldots, x_L)$ is an M-dimensional $(M = n \times L)$ vector of network load; $\boldsymbol{X}$ is a definition of the piecewise-linear cost function $F(x)$, which is found by defining the maximal carrying capacity for each arc; S is a quasidiagonal matrix of size $n \times n$ in the main diagonal of which there are node-arc $(0, 1, -1)$ – incidence matrices of size $T \times L$ for each product and zeros everywhere else; A is an $n \times T$-dimensional load vector of network nodes. We evaluate the arc capacity condition by forming an arc cost function so that its values are increasing sufficiently fast, if the arc load size exceeds the capacity limit and, thus, the summary cost function definition domain is extended onto the whole M-dimensional Euclidean space.

3. The Main Notions

We call as piecewise-linear such a function the definition domain of which can be decomposed into convex polyhedrons, in which it is described by different linear functions in the general case. The smoothness of a piecewise-linear function is violated only at intersection points of polyhedrons. In the case of discontinuity of the function, its value at the discontinuity point is equated to the lowest value of linear functions, corresponding to the polyhedrons, this point belongs to. We call a point critical if linearity of the cost function is violated at it. The sets of critical points are described by linear equations which we call critical equations. We call a critical equation that contains only one variable the main one and this variable – the main variable. We call a point feasible if it satisfies problem condition (2). We call a feasible point a vertex if it is a critical point and satisfies no less than $N = M - R = (L - T + 1) \times n$ linearly independent critical equations (R is a rank of system equations (2)). If the vertex satisfies more than N critical equations, then we call it degenerate. A set of points, satisfying $N - 1$ linearly independent critical equations of those that are satisfied by the vertex x, is called a critical line connected with the vertex. An arc load vector which satisfies at least one critical equation of the arc cost function is called a critical one.

4. Optimality Condition and a Method for Solving the Problem

The vertex x is the local solution to the problem, if the value of the minimized function in it is no higher than its values in the closest vertices which are in critical lines connected with x. The necessity of this condition is obvious, while its sufficiency follows directly from the known theorems of linear programming. For unimodal functions this condition also stands for the global minimum, i.e., the vertex satisfying this condition is the solution to the problem.

The method of problem solving to find a local solution is based on the generation of such a sequence of vertices in which any two adjacent vertices belong to the same critical line, and the values of the objective function are diminishing as long as the vertex satisfying the optimality condition is reached. For this, vertices in critical lines, connected with the next in turn vertex of the sequence, or their segments, are consecutively revised until in some of these lines a new vertex is found with a lower value of the objective function. This procedure is repeated for the new vertex and is terminated only when in no critical line or its segment, there is any other vertex with a lower value of the function.

This idea is realized by performing such stages in the solution of the problem: 1) formation of critical equations of the piecewise-linear function, if they are not directly expressed; 2) determination of the number of critical lines connected with any vertex; 3) finding of the initial vertice; 4) realization of the transition procedure from one vertex to another. A concrete expression of critical equations naturally depends on a concrete piecewise-linear function and it can differ for different functions. The same applies to the number of critical lines, connected with separate vertices. Meanwhile the procedures of stages 3 and 4 are more standardized in the sense that they are less dependent on concrete properties of piecewise-linear functions.

5. Formation of Critical Equations

The linearity of the cost function of any arc and thereby of the whole network in the considered case is violated in the sets of points defined by such conditions:

$$x_i^j = 0, \quad j = 1, \ldots, n, \quad i = 1, \ldots, L, \tag{3}$$

$$\sum_j |x_i^j| - c_i^k = 0, \quad k = 1, \ldots, K_i, \quad i = 1, \ldots, L, \tag{4}$$

where c_i^k are values of the variable w_i that are break (or discontinuity) points of the arc cost function $\bar{f}_i(w_i)$. Critical equations in this case are of the shape:

$$y_{ip} = x_i^j = 0, \quad p = j = 1, \ldots, n, \quad i = 1, ..., L, \tag{5}$$

$$y_{ip} = \sum_j b_{ip}^j x_i^j - c_{ip} = 0, \quad p = M + 1, \ldots, 2^n K_i, \quad i = 1, \ldots, L, \tag{6}$$

where $b_{ip}^j = +1; -1$, and the number of all possible critical equations for each concrete $c_{ip} = c_i^k$ value in the system of equations (6) is equal to 2^n .

The load vector of any arc, whose at least one coordinate is equal to zero, is a critical point. The arc load vector, satisfying non-main equation (6), will be a critical point only in case it also satisfies one of conditions (4) with a respective free term $c_i^k = c_{ip}$, i.e., if its coordinates and that of coefficients of this non-main critical equation will be related by the relations:

$$b_{ip}^j = \mathrm{sign}\,(x_i^j), \quad x_i^j \neq 0. \tag{7}$$

The coefficients of the equation for each zero coordinate can be equal to any of the two numbers: +1 or –1. Therefore the load vector, satisfying condition (4), all the coordinates of which are non-zero, satisfies only one non-main equation with the same free term as in condition (4), because the coefficients of this equation are determined single-valued from condition (7). If this vector has $s < n$ zero coordinates, then it also satisfies 2^s non-main critical equations with a respective free term, since the number of all possible choice variants of coefficients for zero vector coordinates is equal to 2^s. And vice versa, the arc load vector, satisfying more than one critical equation from system (6) with one and the same free term, also satisfies condition (4) with the same free term only if all the vector coordinates, for which the coefficients of these equations are different, are equal to zero.

Thus, the arc load vector, all coordinates of which are equal to zero or it satisfies some condition (4) and $n - 1$ its coordinates are equal to zero, will correspond to the vertex of a piecewise-linear arc cost function. In the latter case, it is a degenerate vertex with which $2(n - 1) + 1 = 2n - 1$ critical lines are connected. Really, any subsystem of critical equations which includes $n - 2$ main critical equations determines two critical lines, because the coefficients for zero coordinates in non-main equations can be equal

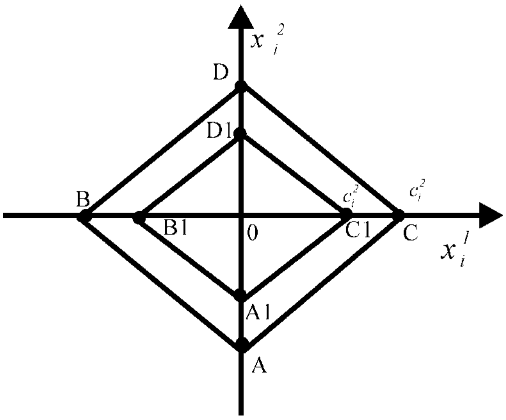

Fig. 1. Sets of critical points and vertices ($O, A, B, C, D, A1, B1, C1, D1$) of the two-variable arc cost function with two break points.

to $+1$ or -1. This degeneracy is unremovable because it is associated with the nature of arc cost functions and cannnot be avoided by changing conditions of the problem. We call an arc a degenerate one if its load vector satisfies some condition (4) and at least one coordinate of the load vector is equal to zero.

Fig. 1 illustrates the sets of arc cost function critical points when two products are transported via the arc and the cost function has two breaks. In the flow distribution problem, critical equations of several arcs also define vertices. Let us have an elementary network of three vertices and three arcs connecting them, in which two kinds of products are transported (Fig. 2). Cost functions of all the arcs have a break each. The sets of vertices and critical points are illustrated for this case in Fig. 3.

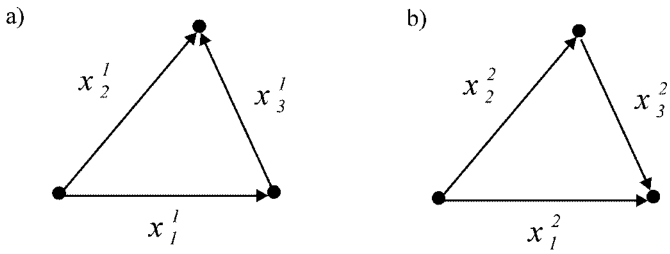

Fig. 2. The network for the first product (a) and for the second product (b).

6. Determination of the Number of Critical Lines Connected with a Vertex

According to the definition of a vertex introduced in Section 3, we regard a flow, satisfying problem conditions (2) and no less than $N = (L - T + 1)n$ critical equations of arc

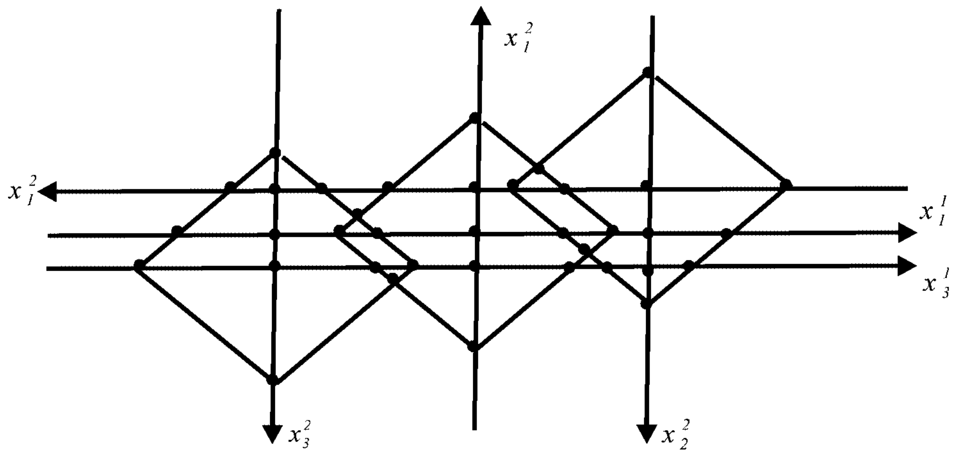

Fig. 3. The set of critical points and vertices of the elementary network described in Fig. 2.

cost functions, as a vertex. We consider such a flow distribution problem, each feasible flow of which fulfils the condition:

$$N_p + N_s \leqslant N, \tag{8}$$

where N_p is the number of main critical equations, and N_s is the number of conditions (4), which are satisfied by the given flow x.

For each vertex inequality (8) turns into an equality. This condition implies that there is no removable degeneracy when coincidence of vertices of the cost functions of several network arcs is possible. Clearly, it does not remove degeneracy related with the specific arc cost function properties described above. If there is no degenerate arc, then only one non-main equation will uniquely express each condition (4). In this case, each vertex will satisfy exactly N critical equations, and the number of critical lines connected with it is also equal to N. As shown in the previous section, each condition (4) for degenerate arcs will be expressed by more than one non-main critical equation that will be satisfied by the given flow, i.e., the given vertex will be degenerate.

Let us vertex x be degenerated and number of critical equations satisfied by it is $\boldsymbol{P}(x)$. We write this critical equations according to increasing of arc numbers. Then the matrix of coefficients of the system of critical equations

$$y_p(x) = 0, \quad \text{for all} \quad p \in \boldsymbol{P}(x), \tag{9}$$

which are satisfied by the degenerate vertex x, is of block-diagonal structure. The coefficients of critical equations of each degenerate arc in it compose a separate diagonal block. Therefore the rank of this system of equations is expressed as follows:

$$\sum_{i \in \boldsymbol{I}_d} r_i + q = N, \tag{10}$$

where $\boldsymbol{I}_d$ is the set of degenerate arc numbers, r_i is the rank of a system of critical equations of the i-th degenerate arc, q is the number of critical equations of nondegenerate arcs.

Any subsystem of the system of equations (9) of rank $N-1$ will define a critical line. One can see from (10) that such a subsystem does not include only one critical equation of a non-degenerate arc or there is a subsystem of the system of critical equations of the rank lower by a unit, i.e., $r_i - 1 = s_i$, of only one degenerate arc. In the first case, a critical line is established uniquely. The second case is more complicated. Subsystems of rank s_i of a degenerate arc consist of the subsystems which include only the main critical equations of this arc or they do not contain one main critical equation. In the latter case, by substituting zero values of main variables into non-main critical equations we get:

$$y_{p(i)} = \sum_{j \notin \boldsymbol{J}_i} b^j_{p(i)} x^j_i + b^{j_0}_{p(i)} x^{j_0}_i - c_{p(i)} = 0, \quad \text{for all} \quad j_0 \in \boldsymbol{J}_i, \tag{11}$$

where $\boldsymbol{J}_i$ is a set of zero coordinate numbers of the i-th arc load vector; the coefficients $b^j_{p(i)}$ and non-zero values of the arc load vector coordinates x^j_i here are related by equations (7).

Since the coefficients $b^{j_0}_{p(i)}$ can assume two different values each $+1$ or -1 for each $j_0 \in \boldsymbol{J}_i$, the number of different equations (11) will be equal to $2s_i$ .We can find all these equations from two systems of equations, each of which includes all main critical equations of the subsystem and one non-main critical equation in each, whose coefficients are obtained as follows:

$$b^j_{p(i)} = \begin{cases} \operatorname{sign}(x^j_i), & \text{if} \quad x^j_i \neq 0, \\ +1 & \text{if} \quad x^j_i = 0, \end{cases} \qquad j = 1, \ldots, n, \tag{12}$$

for one equation, and

$$b^j_{p(i)} = \begin{cases} \operatorname{sign}(x^j_i), & \text{if} \quad x^j_i \neq 0, \\ -1 & \text{if} \quad x^j_i = 0, \end{cases} \qquad j = 1, \ldots, n, \tag{13}$$

for another equation. These systems of equations with the rest equations of system (9) after excluding the block of equations of this degenerate arc will define $2s_i + 1$ diferent critical lines connected with the given vertex x. The total number E of critical lines connected with the vertex x is expressed as follows: $E = 2 \sum_{i \in \boldsymbol{I}_d} s_i + m + q$, where m is the number of degenerate arcs.

From (10) we obtain $\sum_{i \in \boldsymbol{I}_d} s_i + m + q = N$. Hence $\sum_{i \in \boldsymbol{I}_d} s_i = N - m - q$. Then $E = 2(N - m - q) + m + q = 2N - m - q$.

The minimal number of critical lines connected with the given vertex x is obtained when there are no degenerate arcs, i.e., when $m = 0$ and $q = q_{\max} = N$. Then $E_{\min} = 2N - 0 - N = N$. The maximal number of critical lines is got, when there are no non-degenerate arcs, i.e., when $q = 0$ and $m = m_{\min}$. However, the minimal number of degenerate arcs will be, when the degree of degeneracy defined by the number of arc load vector zero coordinates of each of them will be maximal, i.e., equal to $n-1$. It follows from condition (8) for the vertex that $m(n-1)+m=N$. Hence $m_{\min}=N/n$. Consequently,

$$E_{\max}=2N-N/n=2(L-T+1)n-(L-T+1)=(L-T+1)(2n-1).$$

7. Passage from One Vertex to Another

Any vertex may be described by a system of equation:

$$x=\widehat{H}z+\widehat{D}, \tag{14}$$

where the coordinates of the N-dimensional vector of independent variables z are variables y_p described by critical equations (9), i.e., $y_p=y_p(x)$. The system of equations (14) includes condition (2) of the problem. Since the independent variables in the considered case may be expressed by main critical equations (5) that correspond to the basic solution of the system of equations (2), the initial vertex is found by solving the system of equations (2) and equating the value of independent variables to zero.

One can go over from the current vertex $x=\widehat{D}$ to another performing two procedures: uninvariate optimization by varying the value of any independent variable z_k the other values being unchanged, and replacement of variable z_k by a new variable in the system of equations (14) if the optimal value of independent variable z_k defines a new vertex with a lower value of the cost function. The minimized cost function is the sum of a univariate piecewise-linear cost functions of separate arcs, whose loads are changing with a change of independent variable z_k. This function is minimizing in a set of its critical, i.e., break or discontinuity, points since the minimum of the piecewise-linear function will be at the point of its break (or discontinuity). Set of critical points is restricted in the interval in which the coefficients of all critical equations, satisfied by current vertex, satisfies conditions (7). The modified golden section search method (Dievulis, 1999) can be used to find the minimum of the cost function. If the set of critical points is not too large, then the minimum point can be found by revising all the critical points. The non-zero optimal value of independent variable z_k implies that the new vertex satisfies a new critical equation $y_s=0$ instead of that which expresses the variable z_k, i.e., $y_p=0$. Therefore, the independent variable $z_k=y_p$ in the system of equations (14) has to be replaced by another variable y_s expressed by a new critical equation. A sequence of vertices is generated by cyclically repeating that procedures for each coordinate of the independent variable vector z until optimality condition will be satisfied.

If this procedure generate a sequence of vertices defined by main critical equations, then the matrix $\widehat{H}$ of system of equations (14) remains quasidiagonal and replacing any main variable by another basic variable x_i^j one ought to recalculate only one diagonal block of matrix $\widehat{H}$. However, if all or a part of independent variables y_p are expressed by non-main critical equations, then the quasidiagonal structure of matrix $\widehat{H}$ disintegrates.

That is why it would be very irrational to directly apply the procedure of replacement of variables for system of equations (14), because one ought to recalculate not so small sparsely filled up matrix in each iteration of the algorithm.

The described procedure of passage to another vertex is modified so that the vector of independent variables of system (14) is formed only of the variables x_i^j contained by the main or non-main critical equations. Since the rank of node-arc incidence matrix is $T-1$, the number of main and non-main independent variables for each product has to satisfy the conditions:

$$n_1^j + n_2^j = L - T + 1, \quad j = 1, ..., n, \tag{15}$$

where n_1^j, n_2^j are the numbers of main independent variables and independent variables contained by the non-main critical equations, respectively, for the jth flow ingredient.

Since non-main critical equations that express independent variables y_p are linearly independent, we can relate each independent variable y_p in system (14) by a different variable x_i^j contained by this non-main critical equation $y_p = 0$. We fix this relation denoting the variable y_p by the same indices as the non-main variable x_i^j, related with it , i.e., y_i^j. We replace independent variables y_p in the system of equations (14) by non-main variables x_i^jrelated with them, recalculating the elements of the augmented matrix of this system of equations and removing from it the rows with the variables y_p. Thus, the system of equations (14) became quasidiagonal one. Replacement of any independent variable in the system of equations (14) by a basic variable x_i^jwill not destroy a quasi-diagonal structure of matrix $\widehat{H}$ of this system. Therefore one can decompose this system of equations into a system of n independent equations that describe the distribution of separate products in a network.

Thus the system of equations (14) can be replaced by such systems of equations:

$$\bar{x}^j = H^j \widetilde{x}^j + d^j, \quad j = 1, \ldots, n \tag{16}$$

$$\widetilde{\widetilde{x}} = \bar{H} z + \bar{D}, \tag{17}$$

where $\bar{x}^j$ is a $T-1$-dimensional vector of basic variables of the system of equations that describe the j-th product distribution in a network; $\widetilde{x}^j$ is an $L-T+1$-dimensional vector of independent variables of this system of equations; H^j is the incidence matrix of arcs for the j-th product, the elements of which are $-1; 0; 1$; $\widetilde{\widetilde{x}}$ is the vector of non-main variables, consisting of the coordinates of vectors $\widetilde{x}^j$ which are not the main ones; z is the vector of independent variables whose coordinates are main variables x_i^j or variables y_p. The system of equations (17) is obtained from the system of critical equations (9) which satisfy the current vertex x by solving it with respect to non-main variables related with y_p. It is convenient to write the coefficients of these systems of equations into tables. We call the tables of systems (16) as tables of individual products, and that of (17) a table of non-main variables. In the general case, with a change in z_k, non-main and thereby basic variables of systems of equations (16) change, too, if respective coefficients of matrices $\bar{H}$ and H^j are not equal to zero. Values of non-main variables change so that non-main critical equations, contained these variables, be satisfied, i.e., $y_i^j(z) = 0$. Therefore the values of variable z_k will change in the critical line connected with the given vertex.

Thus, quite a large system (14) is replaced by smaller systems of equations (17) for each product and the system of equations (16) of non-main variables relating them. To recalculate matrices of these systems when replacing independent variable we need less calculation amount and computer memory. However in this case the replacement of independent variable z_k became more complicated. In addition, several non-main variables can be replaced in the table of non-main variables and interchanged with basic variables in the tables of individual products so that the number of main n_1^j and non-main n_2^j variables for each product has to satisfy the conditions (15).

We denote the set of indices of non-main variables values of which change with a change in z_k by $\boldsymbol{Z}(k)$. The indices of main variable are included into this set if it is independent variable z_k. Let us denote the indices of main variable or non-main one related to independent variable y_p by (i_0, j_0). We form the sequence of the indices taken from the set $\boldsymbol{Z}(k)$

$$(i_0, j_0), (i_1, j_1), (i_2, j_2), \dots, (i_m, j_m). \tag{18}$$

and second sequence obtained from sequence (18) by shifting indices denoting arc numbers in terms of sequence (18) by one position to the left and writing new index i_{m+1} into the last term of sequence

$$(i_1, j_0), (i_2, j_1), (i_3, j_2), \dots, (i_{m+1}, j_m). \tag{19}$$

so that the interchange of pairs of variables $x_{i_0}^{j_0}$ and $x_{i_1}^{j_0}$ etc. would be possible in respective tables of products $j_0, j_1, \dots, j_m$. It implies that coefficient of table, being in the row of basic variable denoted by indices of the sequence (19) and in the column of independent non-main variable denoted by respective indices of the sequence (18), is not equal to zero and both variables satisfy the same critical equation. The index i_{m+1} denotes the number of arc which load satisfies a new critical equation $y_s = 0$ after performing the optimization procedure. After interchange of the variables in the tables of product the non-main variables denoted by indices (18) also change in table of non-main variables since independent variables y_p which were related by it are now related to other non-main variables denoted by respective indices (19). In order to determine which variables have to be interchanged in the tables of product we form a special graph (Fig. 4) in which the indices of non-main variables, included into the set **Z**(k), are distributed at different levels in this graph as described in below. It is not necessary to form this graph if the numbers of terms in the sequence (18) is not too large. In separate case when sequence (18) consists of only one term, then direct interchange of independent non-main and basic variables is possible in the respect table of product.

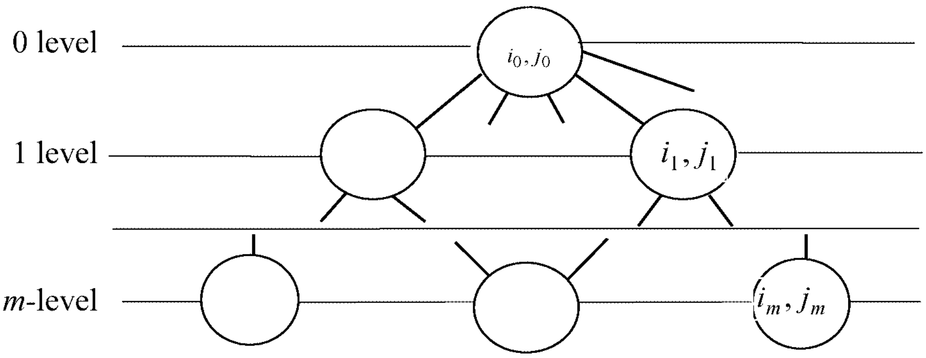

Fig. 4. The graph of contour levels with a selected chain of contours.

8. Usage of Graph Properties

The usage of the properties of graphs which relate the basic variables of the system of equations describing the distribution of a separate product in a network and tree arcs of this network as well as the independent variables and arcs not belonging to the tree, called free arcs, (Christofides, 1976; Dievulis, 1999) enables us to to replace labour-consuming algebraic operations recalculating tables of individual product by easily realizable operations that alter the graph structure and spare computer memory, too. A different free arc corresponds to each column of incidence matrix of the system of equations (16) considered, and non-zero elements of this column to the arcs of the tree which are included into the contour, formed by joining this free arc to the tree. The signs of these elements show the directions of the tree arc in the contour in respect of the direction of the free arc. Making use of this dependence, we can replace systems of equations (16) by lists of network arcs in which the tree arcs of this network and free arcs are separated for each product as well as by the arc load array consisting of arc load vectors. Each arc in the list is represented by its code consisting of three numbers (u, v, l), where u, v are the number of initial and final nodes connected by this arc, and l is the arc number which relates the arc and its load in the array. Each list contains all the information on the network written in the incidence matrices.

It is the most compact way of writing systems of equations (16). However, writing systems of equations in this way requires a special procedure that would be able to distinguish arc codes of any contour from the list of network arcs. Such a procedure can be easily realized if the tree arcs are oriented from the initial node of the tree, called the root of the tree, towards the pendent nodes, i.e., end nodes of the tree that are not initial ones even for a single tree arc (Dievulis, 1999). The arc list of a contour formed by a free arc joined to the tree can be made as follows. We find two paths from both tree nodes, connected by the free arc, to the root following the opposite direction to arc orientation by comparing the number of the initial node of each arc (starting from the last one) with the numbers of final nodes of arcs in the list of arc codes of the tree. There is a single path from each node. The arcs that remain excluding those iterating in both paths make up contour arcs.

In the process of solution of the considered problem any list of network arcs may be altered including into the list of tree arcs the code of free arcs instead of any code of tree arcs that belong to the contour formed by the free arc, and vice versa. Such an alteration is equivalent to the interchanging of the basic and the independent variable in the respective system of equations (16). After this operation the orientation of separate arcs of contour not may correspond to the necessary one. Therefore, the orientation of these arcs is changed so that it be from the root to pendent nodes by interchanging the numbers of the initial and final nodes in the codes of these arcs. The signs of coordinates of these arc load vectors arc changed to the opposite ones in the array, too (Dievulis, 1999).

Since the main or non-main variable x_i^j expresses the load of a free arc, a pair of indices (i, j) of this variable uniquely defines a contour, i.e., the contour which is formed by the i-th free arc in the network of the j-th product. We call two contours connected if they belong to different networks of products and a free arc of one contour, whose load vector satisfies condition (4), belongs to the other contour. A group of contours is called connected, if any contour in it is connected with no less than one contour of this group. A chain of connected contours is called a sequence of contours, in which any contour beginning with the second one is connected with the preceding contour of the sequence. Thus, the non-zero elements of the k-th column in the table of non-main variables and set $\boldsymbol{Z}(k)$, too, define a group of connected contours. According to definition presented above a sequence of indices (18) define a chain of connected contours. We call the contour defined by indices (i_0, j_0) the initial one in the group of connected contour $\boldsymbol{Z}(k)$.

We call the arc i a critical one in the network of the jth product, if its load by the j-th product is described by a main or a non-main variable x_i^j that is related with the variable y_i^j. In the latter case, the summary load of this arc satisfies a non-main critical equation. Since independent variables of each system of equations (8) are main or non-main variables, each free arc is a critical one. So, if condition (8) is fulfilled, then the sets of critical and of free arcs in each vertex have to be identical.

9. Determination of the Initial Vertex Using the Means of Graphs

Determination of the initial vertex means a transformation of the given loads of network nodes into loads of arcs of the chosen tree (one and the same for all the products) so that constraints of problem (2) be satisfied. To make a list of arcs of the initial tree, we can use a formal procedure described in (Dievulis, 1999). In a given network in which a tree has already been isolated an additional (fictitious) node is introduced which is connected by fictitious arcs with the root of the tree and all the network nodes with non-zero loads. Fictitious arcs are oriented from the fictitious node to the network nodes. Their loads are equated to the loads of network nodes with the opposite sign, and that of all the other arcs to zero. In each contour of the extendend network formed by some fictitious arc along with tree arcs and a fictitious arc that connects the fictitious node with the root we choose a flow variation that turns the load of the fictitious arc forming this contour into zero. It means that all the arc loads in this contour are increased by the value of the load of fictitious arc with the opposite sign. After performing this procedure for each contour formed by the fictitious arc, we obtain the loads of tree arcs that satisfy constraints (2) of the problem while the loads of all free arcs are equal to zero, i.e., this flow corresponds to the basic solution of the system of equations (2) which is a vertex.

10. Improvement of Product Distribution in the Network

Improvement of the flow distribution means the passage from one vertex to another with a lower value of the cost function in each iteration of the algorithm. In each its iteration, the product distribution is optimized in one or several contours and, following the fixed rules, the lists of network arcs are altered for one or several products. After determination of the initial vertex we have the lists of network arcs for each product and the initial array of arc loads in which the loads of free arcs are zero. Since for the initial vertex the coordinates of independent vector z are main variables x_i^j, the table of non-main variables is empty. It means that in the first iteration of algorithm we minimized the summary cost function of arcs belonging only to one contour formed by the free arc i in the network of the j-th product. The minimization interval is the minimal change interval of variable x_i^j in which a zero load is obtained for any arc of the contour with the j-th product. However, after one or several iterations of algorithm a table of non-main variables can be formed. If sequence of indices (18) consists of only one term, then we have a case of non-connected contours. In this case the flow distribution of one product is optimizing in the one contour.

The iteration of algorithm for an independent variable z_k is performed in the general case, i.e., in the case of connected contours, as follows:

1) we make up lists of all arcs belonging to the group of connected contours defined by the set $\boldsymbol{Z}(k)$ by using the contour arc isolating procedure;

2) the arc loads of these connected contours are expressed by the formula:

$$x_l^j(\Delta z_k) = \widehat{x}_l^j + \sum_{s \in \boldsymbol{S}(k,j)} e_{ls}^j \bar{h}_{sk}^j \Delta\, z_k, \quad j \in \boldsymbol{J}(k,l), \quad l \in \boldsymbol{L}(k), \tag{20}$$

where $\widehat{x}_l^j$ is the initial load of lth arc by the jth product which is found from the arc load; $\Delta z_k = z_k$; $\boldsymbol{S}(k,j)$ is the set of numbers of free arcs that form the group of connected contours in the network of the jth product; e_{ls}^j is the coefficient the value of which is equal to 0, if the lth arc does not belong to the contour formed by the sth free arc in the network of the jth product, or to $+1$; -1, if the directions of the lth and free arcs are the same in the contour or opposite, respectively; $\bar{h}_{sk}^j$ is the non-zero coefficient in the table of non-main variables present in the kth column and in the row of non-main variable x_s^j; $\boldsymbol{J}(k,l)$ is the set of numbers of products whose flows in the lth arc are changed by the variable z_k; $\boldsymbol{L}(k)$ is the set of numbers of arcs of all connected contours. In the case of non-connected contours sets $\boldsymbol{S}(k,j)$ and $\boldsymbol{J}(k,l)$ consist of one element;

3) by using the univariate minimization procedure the summary cost function of all the arcs of connected contours is minimized in the maximal interval of variation in the

values of z_k in which the non-zero loads of the free arcs, forming connected contours do not change their signs with a change of variable z_k. If the independent variable z_k is a main variable x_i^j and a free arc corresponding to it is degenerate, then only those values of x_i^j are admissible whose signs satisfy the condition (7). In this case, the loads of free arcs forming connected contours will satisfy conditions (4). If the optimal increment Δz_k of variable z_k is not equal to zero, then arc loads of connected contours are recalculated by formula (20) and the iteration is continued by performing the procedure of replacing variables. In the opposite case, the iteration of the algorithm is terminated. In the case of connected contours, the optimization procedure is written as follows:

$$\min_{z_k \in \boldsymbol{\Delta}_{i_0}^{j_0}} F_{i_0}^{j_0}(z_k) = \sum_{l \in \boldsymbol{L}(k)} f_l\big(x_l(z_k)\big), \tag{21}$$

where $\Delta_{i_0}^{j_0}$ is the set of critical values of independent variable z_k; $\boldsymbol{L}(k)$ is the set of numers of arcs of all connected contours or of the initial contour in the case of non-connected contours;

4) to perform the procedure of replacing variables, we determine the chain of connected contours by using special graph (Fig. 4). The connected contours are distributed on different levels in this graph. The zero level contour is the initial contour in the group of connected contours. The first level contours are defined as follows. We find the numbers of all the arcs whose loads satisfy condition (4) in the zero level contour (except the number of arc forming the contour of zero level). Then, the numbers of products are found where these arcs are in the lists of free arcs. The contours formed by these arcs in the networks of respective products are of the first level and graph nodes corresponding to them are connected by edge with the zero level node. This procedure is repeated for each first level contour including the contours not belonging to a lower level to the second level etc. Formation of the graph is completed when we find the contour (i_m, j_m) that contains the tree arc i_{m+1} which becomes a critical one after performing optimization procedure. Fig. 4 shows an example of such a graph.

Afterwards we form a chain of contours $(i_0, j_0), (i_1 j_1), \ldots, (i_m, j_m)$ following from a higher to a lower level node connected with it (if there are several connections, then we choose any of them), starting from the node (i_m, j_m) and finishing with the node (i_0, j_0). A chain of contours being found, the procedure of variable replacement is performed like this:

a) the loads of arcs $i_1, i_2, \ldots, i_{m+1}$ in the network of products $j_0, j_1, \ldots, j_m$, respectively, are expressed by independent variables $z_1, \ldots, z_N$ using the formula:

$$x_i^j(z) = \widehat{x}_i^j + \sum_{s \in \bar{\boldsymbol{S}}(j)} e_{is}^j x_s^j + \sum_{s \in \boldsymbol{S}(j)} e_{is}^j \bigg(\sum_t^N \bar{h}_{st}^j z_t \bigg), \tag{22}$$

where $\bar{\boldsymbol{S}}(j)$ and $\boldsymbol{S}(j)$ are sets of the numbers of free arcs forming contours in the network of the jth product whose load is zero or is expressed by values of non-main variables, respectively. The meaning of the coefficients e_{is}^j and $\bar{h}_{st}^j$ is the same as in formula (20) taking into account the definitions of sets $\bar{\boldsymbol{S}}(j)$ and $\boldsymbol{S}(j)$. The coefficients of these equations are included into the table of non-main variables and the rows of non-main variables denoted by indices of sequence (18) are eliminated from the table;

b) the indices of independent variables $y_{i_1}^{j_1}, \ldots, y_{i_m}^{j_m}$ in the table of non-main variables are respectively replaced by indices of sequence (19) starting from first term;

c) for arc i_{m+1}, main or non-main critical equation $y_s = 0$ is formed the coefficients of which are calculated by formula (12). Variables in this equation are expressed by independent variables z using formula (22). The coefficients of this equation are inserted into the table of non-main variables. The independent variable z_k in the table is replaced by variable y_s denoted by indices of last term in sequence (19), recalculating the elements of the table, respectively. The row of table with variable z_k is removed from it;

d) free arcs are changed in all the contours, belonging to the separated chain in such a way: free arcs $i_0, i_1, \ldots, i_m$ and tree arcs $i_1, i_2, \ldots, i_{m+1}$ are interchanged in the arcs lists of products $j_0, j_1, \ldots, j_m$. Due to this procedure the right structure of the network is preserved for each product, i.e., the number of free arcs in the network of each product is equal to $L - T + 1$ what corresponds to conditions (15).

This procedure is cyclically repeated foe each independent variable $z_1, \ldots, z_N$ in the table of non-main variables as long as it does not change the flow distribution after performing it $N - 1$ times in turn. If there are no degenerate arcs, then calculations are discontinued. In the opposite case, we need an additional procedure. Let arc i be degenerate. Then a new non-main critical equation $\bar{y}_i^j = \bar{y}_i^j(x_i)$ is formed in expression of the former equation $y_i^j = y_i^j(x_i)$ by changing the signs of the coefficients of main variables. This equation is expressed by independent variables $z_1, \ldots, z_N$, using formula (19) and its coefficients are written into the table of non-main variables. Variables y_i^j and $\bar{y}_i^j$ are interchanged and the elements in the table are recalculated. The row with the variable y_i^j is eliminated from the table.

Next we perform the above described procedure for each main variable of the ith arc until we reach a new vertex and the algorithm is repeated from the first or the flow remains unchanged after all the iterations of algorithm for these variables. In the later case, the procedure described is repeated for another degenerate arc. The calculations are terminated, if the flow does not change after performing this additional procedure for each degenerate arc.

Example. Let us illustrate this method by a concrete example. The lists of the tree and free arcs in the network for each product in the example are replaced by the graphs of networks of these products. The replacement is aimed at visual illustration and reduction of calculations. We have to distribute the flow of two products in the network. Fig. 5 depicts the initial structure of the network. It is the same for both products, i.e., arcs 1, 2 and 3 are the tree ones, while 4 and 5 are free arcs. The initial load of arcs is presented in Table 1. It was obtained by equating arc loads of the tree to the loads of their final nodes taken with the opposite sign. The loads of free arcs are zeros, i.e., the initial flow distribution is a vertex. The table of non-main variables (Table 2) is empty, and the independent variables x_4^1, x_5^1 and x_4^2, x_5^2 of the problem are main variables. The cost functions of arcs 1, 3 and 5

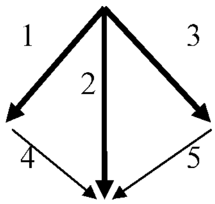

Fig. 5. Initial structure of the graph (a tree is marked out by a bold typed line).

Table 1

The initial array of arc loads

No of arcs	Arc loads	
	by the 1-st product	by the 2-nd product
1	–1	1
2	7	1
3	3	–5
4	0	0
5	0	0

Table 2

The initial table of non-main variables

x_4^1	x_5^1	x_4^2	x_5^2	1
0	0	0	0	0

are identical and expressed as follows:

$$f_i(w_i) = w_i = \sum_j^2 |x_i^j|, \quad i = 1, 3, 5.$$

The cost functions of arcs 2 and 4 are also identical, i.e.,

$$f_i(w_i) = \begin{cases} w_i, & w_i \leqslant 3, \\ 5(w_i - 3) + 3, & w_i > 3, \end{cases} \quad i = 2, 4.$$

The initial value of the cost function is

$$\begin{aligned} F &= \sum_i^5 f_i(w_i) = f_1(2) + f_2(8) + f_3(8) + f_4(0) + f_5(0) \\ &= 2 + 28 + 8 + 0 + 0 = 38. \end{aligned}$$

Iteration 1. The iteration is performed for independent variable $z_1 = x_4^1$.
From the Fig. 5 and Table 1 we define:

$$x_1^1 = -1 + x_4^1, \quad x_2^1 = 7 - x_4^1, \quad \boldsymbol{\Delta}_4^1 = \{-3; 0; 1; 3; 5; 7; 9\}.$$
$$\min_{x_4^1 \in \boldsymbol{\Delta}_4^1} F_4^1(x_4^1) = F_4^1(3) = f_4(3) + f_2(5) + f_1(3) = 3 + 13 + 3 = 19.$$

Under the optimal flow $x_4^1 = 3$ of the first product in the contour which is formed by arc 4, in the network of the first product the load of this arc satisfies a non-main critical equation $x_4^1 + x_4^2 = 3$. In order to fulfil conditions (15) we relate this critical equation with variable x_4^1.

Having written the coefficients of the equation $y_4^1 = x_4^1 + x_4^2 - 3 = 0$ into the table of non-main variables and having replaced the variable x_4^1 by the variable y_4^1 in it we obtain the Table 3.

Table 3

The table of non-main variables after the 1-st iteration of the algorithm

	$\geqslant 0$							$\geqslant 0$				
	x_4^1	x_5^1	x_4^2	x_5^2	1			y_4^1	x_5^1	x_4^2	x_5^2	1
y_4^1	1	0	1	0	–3	$\rightarrow$	x_4^1	1	0	–1	0	3

The loads of arcs 1 and 2 by the first product are recalculated using the given above formulas. The structures of both product networks do not change after the first iteration of the algorithm. The value of the cost function decreases and becomes equal to

$$F = f_1(3) + f_2(5) + f_3(8) + f_4(3) + f_5(0) = 3 + 13 + 8 + 3 + 0 = 27.$$

Arc loads after the 1-st iteration are given in Table 4.

Table 4

Arc loads of the network after the 1-st iteration

No of arcs	Loads of arcs by the 1-st product	Loads of arcs by the 2-nd product
1	2	1
2	4	1
3	3	–5
4	3	0
5	0	0

Iteration 2. The iteration is performed for independent variable $z_2 = x_5^1$.
From the Fig. 5 and Table 4 we define:

$$x_2^1 = 4 - x_5^1, \quad x_3^1 = 3 + x_5^1, \quad \mathbf{\Delta}_4^1 = \{-3; 0; 2; 4; 6\}$$
$$\min_{x_5^1 \in \mathbf{\Delta}_5^1} F_5^1(x_5^1) = F_5^1(2) = f_5(2) + f_2(3) + f_3(10) = 2 + 3 + 10 = 15.$$

When the flow of the first product in the contour formed by arcs 5 in the network of this product $x_5^1 = 2$ is optimal, the load of the arc 2 satisfies the non-main critical equation $x_2^1 + x_2^2 = 3$, which we relate with variable x_2^1. Making use of Fig. 5, Tables 3 and 4, we express the variables x_2^1 and y_2^1 by independent variables:

$$x_2^1 = 4 - x_5^1 - x_4^1 = 4 - x_5^1 - (y_4^1 - x_4^2 + 3 = -x_5^1 - y_4^1 + x_4^2 + 1.$$

Table 5

The table of non-main variables after iteration 2

	≥ 0				
	y_4^1	y_2^1	x_4^2	x_5^2	1
x_4^1	1	0	–1	0	3
x_2^1	0	1	1	1	2

$$\begin{aligned} y_2^1 &= x_2^1 + x_2^2 - 3 = -x_5^1 - y_4^1 + x_4^2 + 1 + (1 - x_4^2 - x_5^2) - 3 \\ &= -y_4^1 - x_5^1 - x_5^2 - 1 = 0. \end{aligned}$$

Having included the coefficients of this equations into the table of non-main variables and having replaced the non-main variable x_5^1 by the variable y_2^1, we get the Table 5. In the network of the first product free arc 5 is replaced by arc 2 respectively.

$$F = f_1(3) + f_2(3) + f_3(10) + f_4(3) + f_5(2) = 3 + 3 + 10 + 3 + 2 = 21.$$

In the Fig. 6 the arcs whose loads satisfy non-main critical equations, are denoted by points in addition.

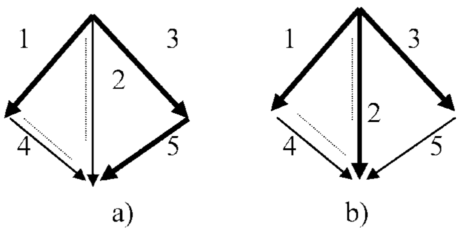

Fig. 6. Structure of the graph (a) – for the 1-st product, (b) - for the 2-nd product after iteration 2.

Table 6

Loads of network arcs after iteration 2

No of arcs	Loads of arcs	
	by the 1-st product	by the 2-nd product
1	2	1
2	2	1
3	5	–5
4	3	0
5	2	0

Iteration 3. The iteration is performed for independent variable $z_3 = x_4^2$.

Table 5 shows that with a change in the values of main independent variable x_4^2 the loads of arcs, belonging to the contours formed by arc 2 in the network of the 2-nd product as well as arcs 2 and 4 in the network of the 1-st product, will also change, i.e., we have the case of connected contours. Using Fig. 6 and Tables 5 and 6 we find:

$$x_4^2 \geqslant 0;\ x_2^2 = 1 - x_4^2; \geqslant 0\ x_1^2 = 1 + x_4^2; x_4^1 = 3 - x_4^2\ 0; x_2^1 = 2 + x_4^2 \geqslant 0;$$
$$x_1^1 = 2 + x_4^1 = 2 - x_4^2; \quad x_5^1 = 2 - x_4^1 - x_2^1 = 2 - (-x_4^2) - x_4^2 = 2;$$
$$x_3^1 = 5 - x_4^1 - x_2^1 = 5 - (-x_4^2) - x_4^2 = 5,$$

i.e., the loads of arcs 3 and 5 do not change when the variable x_4^2 changes.

In view of the conditions $x_2^1 \geqslant 0, x_2^2 \geqslant 0, x_4^1 \geqslant 0, x_4^2 \geqslant 0$ we obtain $\mathbf{\Delta}_4^2 = \{0; 1; 2; 3\}$

$$\min_{x_4^2 \in\ \mathbf{\Delta}_4^2} F_4^2(x_4^2) = F_4^2(0) = f_1(3) + f_2(3) + f_4(3) = 3 + 3 + 3 = 9.$$

Since optimal value of main independent variablex_4^2 equal to 0 the third iteration does not change the flow distribution.

Iteration 4. The iteration is performed for independent variable $z_4 = x_5^2$.

Varying the value of variable x_5^2 in this iteration, the loads of arcs, belonging to the contours formed by arc 5 in the network of the 2-nd product and arc 2 in the network of the 1-st product (Table 5), will also change. Thus:

$$x_5^2 = x_5^2; \qquad x_2^2 = 1 - x_5^2 (\geqslant 0); \quad x_3^2 = -5 + x_5^2;$$
$$x_5^1 = 2 - x_2^1 = 2 - x_5^2; \quad x_2^1 = 2 + x_5^2 (\geqslant 0); \quad x_3^1 = 5 - x_2^1 = 5 - x_5^2.$$

By virtue of the conditions $x_2^1 \geq 0, x_2^2 \geq 0$, we obtain $\mathbf{\Delta}_5^2 = \{-2; 0; 1\}$.

$$\min_{x_5^2 \in \mathbf{\Delta}_5^2} F_5^2(x_5^2) = F_5^2(1) = f_5(2) + f_2(3) + f_3(8) = 2 + 3 + 8 = 13.$$

The zero load of arc 2 by the second product corresponds to the optimal value of independent variable $x_5^2 = 1$. By expressing x_2^2, through independent variables $x_2^2 = 1 - x_4^2 - x_5^2$, inserting the coefficients of the obtained equation into Table 5 of non-main variables, and replacing the independent variable x_5^2 by the variable x_2^2 in it, we obtain a new table of non-main variables (Table 7):

Table 7

The table of non-main variables after replacing the main variable x_5^2 by the main variable x_2^2 in the iteration 4

			≥ 0							≥ 0	≥ 0	
	y_4^1	y_2^1	x_4^2	x_5^2	1			y_4^1	y_2^1	x_4^2	x_2^2	1
x_4^1	1	0	–1	0	3	$\rightarrow$	x_4^1	1	0	–1	0	3
x_2^1	0	1	1	1	2		x_2^1	0	1	0	–1	3
x_2^2	0	0	–1	–1	1							

The structure of the second product network also changes: free arc 5 is replaced by arc 2 (Fig. 7). Arcs load after this iteration are given in Table 8.

$$F = f_1(3) + f_2(3) + f_3(8) + f_4(3) + f_5(2) = 3 + 3 + 8 + 3 + 2 = 19.$$

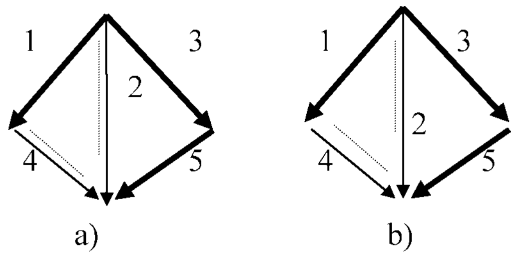

Fig. 7. Structure of the graph after iteration 4: (a) – for the 1-st product (b) – for the 2-nd product.

Table 8

Loads of network arcs after iteration 4

No of arcs	Loads of arcs	
	by the 1-st product	by the 2-nd product
1	2	1
2	3	0
3	4	–4
4	3	0
5	1	1

After performing the iterations 5, 6 and 7 for independent variables y_4^1, y_2^1, x_4^2 the flow distribution in the network does not change.

Since the iterations of the algorithm, performed $N-1=3$ in turn, did not change the flow distribution, in the non-degenerate case the optimality condition would be achieved. In this case, there are two degenerate arcs -2 and 4, and the coefficients of main variables x_2^2 and x_4^2 in non-main equations of these arcs are +1. The optimal flow distribution will be when in the expressions of variables y_2^1 and y_4^1 the signs of coefficients of the variables x_2^2 and x_4^2 are changed by the opposite ones and additional iterations of the main algorithm for the independent variables x_2^2 and x_4^2 do not change the flow distribution.

1-st additional iteration. The iteration is performed for independent variable $z_3 = x_4^2$.

In the table of non-main variables (Table 7), after writing the coefficients of the equation $\bar{y}_4^1 = x_4^1 - x_4^2 - 3 = 0$ into the table and replacing the variable y_4^1 by the variable $\bar{y}_4^1$, we obtain the Table 9.

With a change of variable x_4^2, the loads by the first and the second product of arcs in

Table 9

The table of non-main variables in the first additional iteration

			$0 \leq$	≥ 0	
	$\bar{y}_4^1$	y_2^1	x_4^2	x_2^2	1
x_2^1	0	1	0	–1	3
x_4^1	1	0	1	0	3

the contours, formed by arc 4 in the networks of both products, will change, i.e.,

$$\begin{array}{llll} x_4^2 \leqslant 0; & x_5^2 = 1 - x_4^2; & x_3^2 = -4 - x_4^2; & x_1^2 = 1 + x_4^2; \\ x_4^1 = 3 + x_4^2 \geqslant; & 0x_5^1 = 1 - x_4^2; & x_3^1 = 4 - x_4^2; & x_1^1 = 2 + x_4^2. \end{array}$$

Considering the conditions $x_4^2 \leqslant 0$ and $x_4^1 \geqslant 0$, we get $\mathbf{\Delta}_4^2 = \{-4; -3; -2; -1; 0\}$

$$\min_{x_4^2 \in \mathbf{\Delta}_4^2} F_4^2(x_4^2) = F_4^2(0) = f_4(3) + f_5(2) + f_3(8) + f_1(3) = 3 + 2 + 8 + 3 = 16$$

This additional iteration does not change the flow distribution.

2-nd additional iteration. The iteration is performed for independent variable $z_4 = x_2^2$.

We form an equation $y_2^1 = x_2^1 - x_2^2 - 3 = 0$.

Having performed analogous actions as in the 1-st additional iteration we obtain the Table 10.

Table 10

The table of non-main variables in the second additional iteration

			$0\leqslant$	$0\leqslant$	
	$\bar{y}_4^1$	$\bar{y}_2^1$	x_4^2	x_2^2	1
x_4^1	1	0	1	0	3
x_2^1	0	1	0	1	3

Just like in the first additional iteration we have connected contours which are formed by the second arc in the networks of both products, and the arc loads of these contours change as follows:

$$\begin{array}{lll} x_2^2 \leqslant 0; & x_5^2 = 1 - x_2^2; & x_3^2 = -4 - x_2^2; \\ x_2^1 = 3 + x_2^2 \geqslant 0; & x_5^1 = 1 - x_2^2; & x_3^1 = 4 - x_2^2; \end{array}$$

and $\mathbf{\Delta}_2^2 = \{-4; -3; 0\}$.

$$\min_{x_2^2 \in \mathbf{\Delta}_2^2} F_2^2(x_2^2) = F_2^2(0) = f_2(3) + f_5(2) + f_3(8) = 3 + 2 + 8 = 13.$$

This additional iteration does not change the flow distribution either.

Thus, the flow distribution given in Table 8 satisfies optimality conditions formulated in Section 4 and is the solution to the problem, because the cost functions of all arcs are convex. The lowest possible value of cost function is 19.

The algorithm for solving the problem generalizes the classic multicommodity flow problem is presented in the paper. Described algorithm is suitable for solving various problems of flow distribution that are formulated as piecewise $-$ linear programming problems. The algorithm for solving such problems could be adjusted dependent on the nature of concrete functions by respective changing formulas of coefficients of critical equations (7), (12) and (13).

Acknowledgement. The author expresses his appreciation and respect to Prof. Jonas Mockus whose ideas laid the basis for this article as well as his gratitude to the reviewer whose remarks helped to improve the paper.

References

Assad A. (1978). A multicommodity network flows. *Networks*, **8**(2), 37–91.

Bertsekas D.R., P. Tseng (1988). The relax codes for linear minimum cost network flows problems. *Ann. Operations Research*, **13**, 125–190.

Christofides N. (1976). *Graph Theory. An Algorithmic Approoch*. Academic Press. New York, London, San Francisco. 432 p.p.

Dievulis G. (1999). A method of contour optimization for transport flow distribution. *Statistics in Transition*, **4**(2), 201–212.

Gendron B., T.G. Crainic (1994). Parallel branch-and-bound algoritms: survey and synthesis. *Operations Research*, **42**(6), 1042–1066.

Golshtein E.G., N.A. Sokolov (1995). The new method for solving a multicommodity transport problem. *Ekonomika i Mat. Metody*, **31**(2), 128–136 (in Russian).

Golshtein E.G., N.A. Sokolov (1997). Decomposition method for solving production-transport problems. *Ekonomika ir Mat. Metody*, **33**(1), 112–128 (in Russian).

Kennington J.A., A.V. Helgasson (1980). *Algorithms for Network Programming*. John Wiley and Sons. New York.

Magnanti T.L., R.T. Wong (1984). Network design and transportation planning models and algorithms. *Transportation Science*, **18**(1), 3–55.

Melamed I.I. (1991). Optimization methods for transport. In *Organize of Management for Transport*, Vol 10, Transport, Moscow. pp. 1–164.

Minoux M. (1989). Network synthesis and optimum network design problems: models, solution methods and applications. *Networks*, **19**(2), 313–360.

Zaslavskij A.A., S.S. Lebedev (1995). The use special branch-and-bround method for solving generalized integer transport problem. *Ekonomika i Mat. Metody*, **31**(2), 137–141 (in Russian).

G. Davulis graduated from the Kaunas University of Technology, Lithuania, in 1963. He received doctors degree from the Latvian state university in 1969. He is an associate professor at the Lithuania University of Law. His areas of research interests include macroeconomic modelling, modelling and optimizing economic processes in a transport.

Dalimis tiesinio daugelio produktų transporto uždavinio sprendimo metodas

Gediminas DAVULIS

Nagrinėjamas daugelio produktų transporto uždavinys su dalimis tiesine kainos funkcija. Šis už davinys yra formuluojams kaip neseparabilus atskirų kintamųjų grupių atžvilgiu: dalimis tiesinio programavimo uždavinys. Įprasti metodai, naudojami netiesiniams transporto uždaviniams spręsti, leidžia rasti uždavinio lokalinį sprendinį geriausiu atveju tik duotu tikslumu. Siūlomas metodas, pagrįstas simplekso metodo idėjos išplėtimu neseparabilių dalimis tiesinio programavimo uždavinių klasei, leidžia rasti lokalinį sprendinį po baigtinio iteracijų skaičiaus. Panaudojamos transportinio tipo apribojimų ypatybės, leidžiančios išskaidyti uždavinio apribojimų matricą į eilę mažesnių, tuo sumažinant skaičiavimų apimtis.