ISSN 1064-2307, Journal of Computer and Systems Sciences International, 2008, Vol. 47, No. 2, pp. 214–220. © Pleiades Publishing, Ltd., 2008.
Original Russian Text © L.G. Afraimovich, M.Kh. Prilutskii, 2008, published in Izvestiya Akademii Nauk. Teoriya i Sistemy Upravleniya, 2008, No. 2, pp. 57–63.

Multicommodity Flows in Tree-Like Networks

L. G. Afraimovich and M. Kh. Prilutskii

Nizhni Novgorod State University, pr. Gagarina 23, Nizhni Novgorod, 603600 Russia

Received May 14, 2007

Abstract—A problem of finding multicommodity minimum-cost flows in tree-like networks is considered. To solve this problem, an algorithm based on the reduction to a problem of finding a single-commodity flow in a network of arbitrary structure is proposed. For single-commodity flows in tree-like networks, an algorithm using a procedure of border reduction is used.

DOI: 10.1134/S106423070802007X

INTRODUCTION

Search for multicommodity flows in networks is needed to solve a wide class of application problems, including the problems of volumetric calendar scheduling [1], distribution of capacities of data-transmission channels by Internet providers [2], balanced loading of distributed computer networks [3], transportation problem with intermediate points [4], etc.

Examples of classical flow problems are the search for a maximum flow and the search for a minimum-cost flow. The most study has been received by single-commodity problems, which are formalized as linear programming problems with an absolutely unimodular matrix of the system of constraints. These problems are polynomially solvable, including in the integer case. A survey of algorithms for solving the single-commodity maximum-flow problem can be found, for example, in [5]; the algorithms for solving the problem of a minimum-cost single-commodity flow were considered in [6].

The class of problems of finding multicommodity flows has been studied to a lesser extent. The algorithms for multicommodity (nonintegral) flows can be found, for example, in [7, 8]. The problem of finding a maximum integer-valued multicommodity flow has been shown to be NP-hard already in the two-commodity case [9]. The same paper also proposed a polynomial algorithm for the finding a maximum two-commodity flow, constructing an integer-valued flow accurate up to the 1/2 (for example, 7/2, 15/2, 16/2). There are approximate algorithms for solving the problem of a maximum integer-valued multicommodity flow [10, 11]. Some generalizations of the models of multicommodity flows in networks are discussed in [12, 13]. Of special interest is the class of multicommodity flows in tree-like networks. It follows from [14] that, in the general case, the finding an integer-valued multicommodity flow in a tree-like unoriented network is an NP-hard problem.

In this paper, we consider a special subclass of problems of finding a multicommodity flow in oriented networks of a root tree-like structure for the case when the root is the source of different commodities and the leaves of the tree are their sinks. For the network under consideration, we propose an algorithm of finding a minimum-cost multicommodity flow, based on the reduction to the problem of single-commodity flow in a network of arbitrary structure. We prove that this problem is polynomially solvable. For the case of single-commodity flows in tree-like networks, we developed a method using a procedure of border reduction.

1. TREE-LIKE NETWORK STRUCTURES

The problems considered in this paper emerge in the design and management of tree-like structures simulating economic, industrial, and technical systems. The problems include, for example, volumetric multi-commodity planning for plant subdivisions, multicommodity tree-like transportation problem with intermediate points, breaking up nodes in the design of manufacturing of new products for small plants, nomenclature planning for plants with a continuous production cycle, etc.

By their content, the problems of volumetric multicommodity planning for plant subdivisions can be formulated in the following way. A plant consists of subdivisions (workshops), in turn, each of which has workplace sectors. For a given design period, it is required to distribute the existing orders in volumetric terms by workshops and sectors, so that the given indices and total profit of the plant be a maximum.

Let I be the set of plant subdivisions, J_i be the set of sectors of the subdivision i, $i \in I$, and let K be the set of orders to be filled by the plant. Let us consider the following planning indices: A is the workload remained uncompleted for a given period on all orders; B_i and C_i are the minimum-possible and maximum-allowable workloads, respectively, that can be completed in the planning period by the subdivision i on all orders, $i \in I$; D_{ij} and E_{ij} are the minimum-possible and maximum-allowable workloads, respectively, that can be completed in the design period by the sector j of the subdivision i on all orders, $j \in J_i$, $i \in I$; G_{ijk} is the workload remaining uncompleted on order k by the sector j of the subdivision i, $j \in J_i$, $i \in I$, $k \in K$; and v_{ijk} is the profit to be gained by the plant per unit work done on order k by the sector j of the subdivision i, $j \in J_i$, $i \in I$, $k \in K$. Then, the problem of design is formalized as the following linear programming problem:

$$\begin{gathered}
\sum_{i \in I} \sum_{j \in J_i} \sum_{k \in K} x_{ijk} \le A, \\
B_i \le \sum_{k \in K} \sum_{j \in J_i} x_{ijk} \le C_i, \quad i \in I, \\
D_{ij} \le \sum_{k \in K} x_{ijk} \le E_{ij}, \quad j \in J_i,\ i \in I, \\
0 \le x_{ijk} \le G_{ijk}, \quad j \in J_i,\ i \in I, \quad k \in K, \\
\sum_{i \in I} \sum_{j \in J_i} \sum_{k \in K} v_{ijk} x_{ijk} \to \max,
\end{gathered} \tag{1.1}$$

where x_{ijk} is the workload to be done in the design period by the sector j of the subdivision i on order k, $j \in J_i$, $i \in I$, $k \in K$.

If the design indices are A_k (the workload remained uncompleted for the given design period on order k, $k \in K$), B_i and C_i (the minimum-possible and maximum-allowable workloads, respectively, that can be completed in the design period by the subdivision i on all orders, $i \in I$), D_{ijk} (the maximum-possible workload that can be completed in the design period by the sector j of the subdivision i on order k, $j \in J_i$, $i \in I$, $k \in K$), g_k (the profit to be gained by the plant per unit work done on order k, $k \in K$), h_{ij} (the costs covered by plant for conducting a unit work by the sector j of the subdivision i, $j \in J_i$, $i \in I$), then the problem of volumetric design is transformed into the following linear programming problem:

$$\begin{gathered}
\sum_{i \in I} \sum_{j \in J_i} x_{ijk} \le A_k, \quad k \in K, \\
B_i \le \sum_{k \in K} \sum_{j \in J_i} x_{ijk} \le C_i, \quad i \in I, \\
0 \le x_{ijk} \le D_{ijk}, \quad j \in J_i,\ i \in I,\ k \in K, \\
\sum_{k \in K} g_k \cdot \sum_{i \in I} \sum_{j \in J_i} x_{ijk} - \sum_{i \in I} \sum_{j \in J_i} h_{ij} \cdot \sum_{k \in K} x_{ijk} \to \max.
\end{gathered} \tag{1.2}$$

These optimization problems are of multicommodity character, which is explained by the fact that the plant works simultaneously on different orders. The network model of this industrial system associates the whole plant with the root of the oriented graph (the source of heterogeneous products corresponding to different orders), the subdivision sectors with the graph leaves (the sinks of heterogeneous products corresponding to different orders), the plant subdivisions with the remaining nodes of the oriented graph (the elements transmitting heterogeneous products related to different orders).

2. STATEMENT OF THE PROBLEM

Let us consider an oriented q-commodity network of a tree-like structure, with its root being the source of all q commodities and its leaves being the sinks. Let $G = (V, A)$ be the root oriented tree, $A \subseteq V^2$ and $|V| = n$. Let us select a special node s (the tree root), $s \in V$, and a subset T (the set of leaves), $T \subseteq V$, among the set of tree nodes. We denote by $\pi(i)$ the node that is the direct predecessor of the node i, $i \in V \backslash \{s\}$.

Let us introduce the following parameters: u_s^k, and w_s^k are the minimum-possible and maximum-possible total amount of outgoing flows of commodity k from source s, $k = \overline{1, q}$; u_i^k, and w_i^k are the minimum-possible and maximum-possible total amount of the incoming flow of commodity k into node i, $i \in V \backslash \{s\}$, $k = \overline{1, q}$; l_{ij} and c_{ij} are the lower and upper capacities of the arc (i, j), $(i, j) \in A$; and e_{ij}^k is the cost of the unit flow of commodity k passing through the arc (i, j), $(i, j) \in A$, $k = \overline{1, q}$.

Then, the problem of finding a minimum-cost multicommodity flow (hereafter, L_q^{tree} problem) is to determine the values x_{ij}^k of the flow of commodity k passing through the arc (i, j), $(i, j) \in A$, $k = \overline{1, q}$, for which the following constraints are satisfied:

$$\begin{gathered}
u_s^k \le \sum_{j | \pi(j) = s} x_{sj}^k \le w_s^k, \quad k = \overline{1, q}, \\
u_i^k \le x_{\pi(i)i}^k \le w_i^k, \quad k = \overline{1, q},\ i \in V \backslash \{s\}, \\
l_{ij} \le \sum_{k=1}^{q} x_{ij}^k \le c_{ij}, \quad (i, j) \in A, \\
x_{\pi(i)i}^k - \sum_{j | \pi(j) = i} x_{ij}^k = 0, \quad k = \overline{1, q},\ i \in V \backslash (T \cup \{s\}), \\
x_{ij}^k \ge 0, \quad k = \overline{1, q}, \quad (i, j) \in A,
\end{gathered}$$

and the criterion $\sum_{k=1}^{q} \sum_{(i, j) \in A} x_{ij}^k e_{ij}^k$ characterizing the total flow cost takes its minimum value.

The L_q^{tree} problem stated above is a linear programming problem and can be solved by using general methods (for example, by the simplex method). The transportation particularity of the mathematical model (the coefficients of the constraint matrix take values from

0,1,-1) makes it possible to solve this problem by the Agmon–Motzkin modified iteration method of orthogonal projections [15]. In this paper, we propose more efficient solution algorithms based on the reducibility of the original multicommodity ($q \geq 2$) problem of finding a maximum single-commodity flow in the transportation network and, for the single-commodity case ($q = 1$), on the method of border reduction [16]. Here, it is interesting also to consider the problem of finding an integer-valued minimum-cost multicommodity flow (the $L_{q,Z}^{tree}$ problem), which requires an additional assumption that $x_{ij}^k \in Z, (i, j) \in A, k = \overline{1, q}$.

3. THE SOLUTION ALGORITHM FOR $\mathbf{L_1^{tree}}$

For a single-commodity flow, the system of constraints of the problem L_1^{tree} is transformed into the form

$$\begin{gathered} u_s \leq \sum_{j|\pi(j) = s} x_{sj} \leq w_s, \\ u_i \leq x_{\pi(i)i} \leq w_i, \quad i \in V\backslash\{s\}, \\ l_{ij} \leq x_{ij} \leq c_{ij}, \quad (i, j) \in A, \\ x_{\pi(i)i} - \sum_{j|\pi(j) = i} x_{ij} = 0, \quad i \in V\backslash(T \cup \{s\}), \\ x_{ij} \geq 0, \quad (i, j) \in A, \end{gathered} \tag{3.1}$$

and the criterion characterizing the total flow cost can be written as

$$\sum_{(i, j) \in A} x_{ij} e_{ij} \to \min. \tag{3.2}$$

For convenience, the subscript $k = 1$ is omitted here.

Let us transform system (3.1) in the following way. For a node j, we determine the minimum-possible and maximum-possible total amounts of commodity flow as $U_j = \max(u_j, l_{\pi(j)j})$ and $W_j = \min(w_j, c_{\pi(j)j})$, $j \in V\backslash\{s\}$; $U_s = u_s$, $W_s = w_s$. Let

$$y_j = x_{\pi(j)j}, \quad j \in V\backslash\{s\}; \quad y_s = \sum_{i|\pi(i) = s} x_{si}.$$

Then, (3.1) can be transformed into the form

$$\begin{gathered} U_i \leq y_i \leq W_i, \quad i \in V, \\ y_i = \sum_{j|\pi(j) = i} y_j, \quad i \in V\backslash T, \\ y_i \geq 0, \quad i \in V. \end{gathered} \tag{3.3}$$

The optimality criterion is set as

$$\sum_{i \in V\backslash\{s\}} y_i e_i' \to \min, \tag{3.4}$$

where $e_i' = e_{\pi(i)i}$, $i \in V\backslash\{s\}$. It is obvious that problem (3.1), (3.2) is equivalent to the formulation (3.3) and (3.4).

Let us apply the procedure of "border reduction" for the system of constraints (3.3); i.e., based on the following recurrent relations, define the values U_i^p, W_i^p, $i \in V$

$$\begin{aligned} U_i^p &= U_i, \quad i \in T, \\ W_i^p &= W_i, \quad i \in T, \\ U_i^p &= \max\Bigl(U_i, \sum_{j|\pi(j) = i} U_j^p\Bigr), \quad i \in V\backslash T, \\ W_i^p &= \min\Bigl(W_i, \sum_{j|\pi(j) = i} W_j^p\Bigr), \quad i \in V\backslash T. \end{aligned} \tag{3.5}$$

Theorem on border reduction. System (3.3) is consistent if and only if $U_i^p \leq W_i^p$, $i \in V$.

Proof. The necessity of the theorem conditions is obvious. The sufficiency is proved constructively. Let us show that, if the theorem is true, a specially constructed vector $\vec{y}^0 \in R^n$ is a feasible solution of system (3.3). By the theorem conditions, we have $U_s^p \leq W_s^p$; therefore, there is a value such that $y_s^0 \in [U_s^p, W_s^p]$, and, thus (due to the fact that $[U_s^p, W_s^p] \subseteq [U_s, W_s]$) y_s^0 satisfies condition (3.3).

The components y_j^0 of the vector $\vec{y}^0$, for which $\pi(j) = s$, are determined from the following expressions

$$\sum_{j|\pi(j) = s} y_j^0 = y_s^0; \quad U_j^p \leq y_j^0 \leq W_j^p, \quad \pi(j) = s.$$

This system is consistent because $y_s^0 \in [U_s^p, W_s^p]$, and it follows from the recurrent relations that

$$U_s^p \geq \sum_{j|\pi(j) = s} U_j^p, \quad W_s^p \leq \sum_{j|\pi(j) = s} W_j^p.$$

Finding similarly the value of the components of the vector $\vec{y}^0$, we can construct a feasible solution of system (3.3). This completes the proof of the theorem.

It follows from (3.5) that the procedure of border reduction has a computational complexity $O(n)$. Then, the constructive scheme of the proof of the theorem yields a series of corollaries.

Corollary 1. There exists an algorithm of finding a feasible solution of the problem L_1^{tree} with the computational complexity of $O(n)$.

Corollary 2. There exists an algorithm for solving the problem L_1^{tree} with the computational complexity $O(n^2)$.

Proof. The scheme of the proof of the border reduction theorem makes it possible to construct an efficient procedure for solving the problem L_1^{tree}. First, we apply the border reduction to the system of constraints (3.3). If the conditions of the theorem are satisfied, the problem L_1^{tree} has a solution. Let us find all the $|T|$ chains that connect the tree root with leaves. For each chain $(s, j_1, j_2, \ldots, j_t)$, $j_t \in T$, we determine the value $\sum_{i=1}^{t} e'_{j_t}$ of the cost of unit commodity flow along this chain. Starting from the chain for which this cost is a maximum in modulus, we pass a minimum-possible flow along this chain (without breaking the given borders) if the cost of the unit flow is positive, and we pass a maximum-possible flow otherwise. Let us decrease the left and right borders by the amount of flow in all nodes consistent with the chosen chain. We eliminate this chain, and the conditions of the theorem will be satisfied. Now, we choose the next chain with a maximum (in modulus) cost of unit flow. We pass along this chain a minimum-possible (maximum-possible) flow and decrease the left and right borders by the amount of flow of the corresponding nodes. This process is repeated $|T|$ times. The total flow obtained will determine an optimal solution of the problem L_1^{tree}. If the maximum length of the chain connecting the root with leaves (for the tree-like network under consideration) is equal to k, the algorithm examines no more than $(k + 1)|T|$ nodes, which determines an estimate of computational complexity.

Remark 1. The estimate $O(n^2)$ of the solution algorithm of the problem L_1^{tree}, described in the proof of *Corollary 2,* is attainable. Indeed, this estimate holds for a tree consisting of $2n$ nodes and n leaves connected with the root by chains of lengths 1, 2, …, n, respectively. On the other hand, for a wide class of applications for which the tree-like network model includes no chains (between the root and leaves) exceeding in their length some given constant, the computational complexity of the algorithm becomes linear. Among these are some multi-index problems with transportation-type constraints. Indeed, among the above-mentioned problems of multi-commodity planning (1.1) and (1.2), which are of the class L_q^{tree}, formulation (1.1) is reduced to the problem L_1^{tree}, for which the algorithm described in Corollary 2 has a linear computational complexity.

4. A SOLUTION ALGORITHM FOR $L_{\mathbf{q}}^{\mathrm{tree}}$

The algorithm proposed for solving the problem L_q^{tree} of finding a multicommodity minimum-cost flow in a tree-like oriented network is based on its reducibility to the problem of finding a (single-commodity) minimum-cost circulation in a network of arbitrary structure with two-side arc capacities (this problem is denoted by L_1). We use the following conception of reducibility.

Definition 1. A matrix A', $A' = \|a'_{ij}\|_{m' \times n'}$, is reduced to a matrix A'', $A'' = \|a''_{ij}\|_{m'' \times n''}$, if there exists a mapping $\alpha : \{1, 2, \ldots, n'\} \longrightarrow \{1, 2, \ldots, n''\}$ such that, for any linear programming problem $f'(\vec{x}'^0) = \max\{(\vec{x}', \vec{c}') \mid A'\vec{x}' \le \vec{b}', \vec{x}' \ge 0\}$ (denoted as L'), one can construct a linear programming problem $f''(\vec{x}''^0) = \max\{(\vec{x}'', \vec{c}'') \mid A''\vec{x}'' \le \vec{b}'', \vec{x}'' \ge 0\}$ (denoted as L''); in this case,

if $\vec{x}''^0 = (x''_1, x''_2, \ldots, x''_{n''})$ is an optimal (feasible) solution of L'', then $\vec{x}'^0 = (x''_{\alpha(1)}, x''_{\alpha(2)}, \ldots, x''_{\alpha(n')})$ is an optimal (feasible) solution of L';

the problem L' is inconsistent if L'' is inconsistent;

all components of the vector $\vec{b}''$ are integers if all components of $\vec{b}'$ are integers.

Definition 2. The linear programming problem L' is reduced to the linear programming problem L'' if the constraint matrix of L' is reduced to that of L''.

This conception of reducibility of problems of linear programming makes it possible to ensure that, if L' is reduced to L'', then

by solving the problem L'', one can obtain the solution of L' as a subset of components of the solution of L'', which are determined by the mapping α;

the matrix A'' and mapping α remain constant for any values of the vectors $\vec{b}'$ and $\vec{c}'$, and depend only on the form of the matrix A'.

Reducibility theorem. The problem L_q^{tree} is reduced to the problem L_1.

Proof. Let us prove this theorem constructively, based on the construction scheme of the problem L_1 corresponding to the problem L_q^{tree}. Let L_q^{tree} be defined by the graph $G = (V, A)$. We use an auxiliary node v and, for each node i of G, introduce q additional nodes i^k, $i = V$, $k = \overline{1, q}$. Let us construct a set of nodes and a set of arcs specifying the structure of the problem L_1.

The set of nodes is $V \cup \{i^k | i \in V, k = \overline{1, q}\} \cup \{v\}$. The set of arcs is $A \cup A_q \cup A_{T,q} \cup A_{s,q} \cup \{(v, s)\}$, where $A_q = \{(j^k, i^k) | (i, j) \in A, k = \overline{1, q}\}$, $A_{T,q} = \{(i, i^k) | i \in T, k = \overline{1, q}\}$, and $A_{s,q} = \{(s^k, v) | k = \overline{1, q}\}$.

Let us determine the capacities of arcs for the problem L_1: the lower and upper capacities of an arc (i, j), $(i, j) \in A$, are equal to l_{ij} and c_{ij}, respectively; the capacities of an arc (j^k, i^k), $(j^k, i^k) \in A_q$, are equal to u_j^k and w_j^k; the capacities of an arc (s^k, v), $(s^k, v) \in A_{s,q}$, are set to u_s^k, and w_s^k; and the lower and upper capacities of the remaining arcs are zero and infinity, respectively. The cost of unit flow passing through an arc (j^k, i^k), $(j^k, i^k) \in A_q$, is equal to e_{ij}^k and the remaining arcs are set to zero costs.

By introducing the mapping α that transforms the set of variables of the problem L_q^{tree} into that of L_1, we associate each variable x_{ij}^k, $(i, j) \in A$, $k = \overline{1, q}$, with the amount of flow passing through the arc (j^k, i^k), $(j^k, i^k) \in A_q$; thus, the problem L_q^{tree} is reduced to L_1. This completes the proof of the theorem.

*Corollary 3***.** There exists a solution algorithm for the problem L_q^{tree} with the computational complexity $O(n^3q^3\log^2(nq))$.

Proof. By the reducibility theorem, the solution of L_q^{tree} can be determined as a subset of components of solutions of L_1. It follows from the constructive scheme of the proof that the network for the problem L_1 contains $O(nq)$ nodes and $O(nq)$ arcs.

To solve the problem L_1, one can use the well-known algorithms (for example, the strongly polynomial algorithm of Galil and Tardos [17]). Then, the computational complexity of the solution of L_1 and, consequently, of the original problem L_q^{tree}, will be $O((nq)^2\log(nq)(nq + nq\log(nq)))$, which yields the needed estimate for the computational complexity. This completes the proof corollary.

Proposition. If a linear programming problem is reduced to the problem L_1, the constraint matrix of the original problem is absolutely unimodular.

Proof. We prove this proposition by contradiction. Assume that a linear programming problem is reduced to the problem L_1, but the corresponding constraint matrix is not absolutely unimodular. The conditions of absolute unimodularity for the case of integer-valued parameters of the linear programming problem are necessary and sufficient for all nodes of the polygon corresponding to the system of constraints of the linear programming problem to be integer-valued [18]. Then, one can construct such a linear target function that the linear programming problem will have a unique optimal noninteger solution. This contradicts the theorem on integer flow [19] (according to which, if the problem is solvable, there always exists its optimal integer-valued solution). This completes the proof of the proposition.

It immediately follows from the proposition and the reducibility theorem that the proposed solution algorithm for the problem L_q^{tree}, used in proving corollary 3, can be also applied to solving the problem $L_{q,Z}^{tree}$.

Corollary 4. The problem $L_{q,Z}^{tree}$ is polynomially solvable.

Remark 2. The theorem on the border reduction can be used as necessary conditions for the consistency of the problem L_q^{tree}. To do this, it is necessary to convolve the constraints of the multicommodity problem L_q^{tree} by transforming it into the single-commodity problem L_1^{tree}. The procedure of reduction is to pass from the constraints of L_q^{tree} to constraints (3.3) of the problem L_1^{tree}. The initial parameters for these constraints are derived from the relations

$$U_j = \max\left(\sum_{k=1}^{q} u_j^k, l_{\pi(j)j}\right),$$

$$W_j = \min\left(\sum_{k=1}^{q} w_j^k, c_{\pi(j)j}\right), \quad j \in V\backslash\{s\};$$

$$U_s = \sum_{k=1}^{q} u_s^k, \quad W_s = \sum_{k=1}^{q} w_s^k.$$

5. EXAMPLES OF PROBLEM SOLUTION

Let us consider a numerical example of the solution of the problem L_q^{tree} for $q = 2$ and $q = 1$. Let $V = \{s, i, j, l, p\}$, $T = \{j, l, p\}$, and $A = \{(s, i), (s, j), (i, l), (i, p)\}$.

For the case $q = 2$, we have the following values of the initial parameters of the problem L_2^{tree}:

$$u_s^1 = 5, \; w_s^1 = 20, \; u_s^2 = 7, \; w_s^2 = 18;$$

$$u_i^1 = 3, \; w_i^1 = 7, \; u_i^2 = 6, \; w_i^2 = 15;$$

$$u_j^1 = 2, \; w_j^1 = 4, \; u_j^2 = 3, \; w_j^2 = 6;$$

$$u_l^1 = 1, \; w_l^1 = 4, \; u_l^2 = 4, \; w_l^2 = 6;$$

$$u_p^1 = 0, \; w_p^1 = 2, \; u_p^2 = 2, \; w_p^2 = 5;$$

$$l_{si} = 8, \; c_{si} = 17, \; e_{si}^1 = -2, \; e_{si}^2 = 1;$$

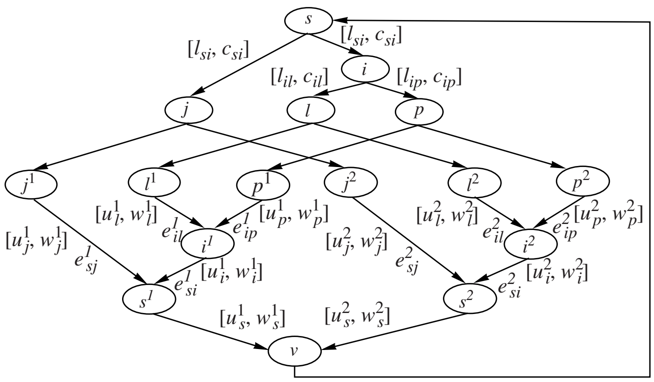

Figure.

$$l_{sj} = 2, \quad c_{sj} = 8, \quad e^1_{sj} = 2, \quad e^2_{sj} = 0;$$

$$l_{il} = 6, \quad c_{il} = 10, \quad e^1_{il} = 3, \quad e^2_{il} = -4;$$

$$l_{ip} = 4, \quad c_{ip} = 9, \quad e^1_{ip} = -3, \quad e^2_{ip} = 1.$$

Let us construct the corresponding problem L_1 described in the proof of the reducibility theorem (see figure). The circulation of the minimum cost in L_1 yields (in line with the mapping α) the following solution if L_2^{tree}:

$$x^1_{si} = 3, \quad x^1_{sj} = 2, \quad x^1_{il} = 1, \quad x^1_{ip} = 2;$$

$$x^2_{si} = 8, \quad x^2_{sj} = 3, \quad x^2_{il} = 6, \quad x^2_{ip} = 2.$$

Here the optimal value of the criterion is equal to –19.

Let $q = 1$. We determine the values parameters of L_1^{tree} by convolving the constraints of the problem L_2^{tree}:

$$U_s = 12, \quad W_s = 38;$$

$$U_i = 9, \quad W_i = 17, \quad e'_i = -1;$$

$$U_j = 5, \quad W_j = 8, \quad e'_j = 2;$$

$$U_l = 6, \quad W_l = 10, \quad e'_l = -1;$$

$$U_p = 4, \quad W_p = 7, \quad e'_p = -2.$$

After the border reduction, the initial parameters of the system will have the form

$$U^p_s = 15, \quad W^p_s = 25; \quad U^p_i = 10, \quad W^p_i = 17;$$

$$U^p_j = 5, \quad W^p_j = 8; \quad U^p_l = 6, \quad W^p_l = 10;$$

$$U^p_p = 4, W^p_p = 7.$$

The system is consistent and it optimal solution is $y_s = 22$, $y_i = 17$, $y_j = 5$, $y_l = 10$, $y_p = 7$. Here, the optimal value of the criterion is equal to –31.

CONCLUSIONS

The algorithms developed by us were used to create the program system "Allocation of limited resources in hierarchic transportation-type systems". The system was tested on a series of plants in Nizhni Novgorod. The computational experiments performed allow us to claim that these algorithms are efficient in solving large-scale application problems. Indeed, the average time of search for a maximum multicommodity flow in a network with 520 nodes and 30 commodities (which corresponds to the solution of a volumetric calendar scheduling for subdivisions of the pilot factory of the Federal State Unitary Enterprise "Pilot Engineering Department of Machine Building after I. I. Afrikantov" in Nizhni Novgorod) was about 6 minutes (the computer characteristics were Pentium IV 1200 MHz processor, 256 Mb DDR RAM, Windows 2000 OS).

Note that the mathematical model of L_{30}^{tree} contained 15570 variables, 6090 two-side constraints on subsums, and 570 balance equations.

REFERENCES

1. M. Kh. Prilutskii, "Multicriteria Multi-Index Resource Scheduling Problems," Izv. Ross. Akad. Nauk, Teor. Sist. Upr., No. 1, 78–82 (2007) [Comp. Syst. Sci. **46** (1), 78–82 (2007)].
2. L. G. Afraimovich and M. Kh. Prilutskii, "Multicriteria Problems of Resource Allocation in Hierarchical Systems," Avtom. Telemekh., No. 6, 194–205 (2006).
3. R. Deikmann, A. Frommer, and B. Monien, "Efficient Schemes for Nearest Neighbor Load Balancing," Parallel Computing **25**, 789–812 (1999).
4. A. Corban, "Transportation Problem with Intermediate Centers," Rev. Roum. Mat. Pures. Appl. **16** (9), 128–134 (1971).
5. A. V. Goldberg and S. Rao, "Beyond the Flow Decomposition Barrier," J. ACM **45** (5), 783–797 (1998).
6. A. V. Goldberg and R. E. Tarjan, "Solving Minimum-Cost Flow Problems by Successive Approximation," Math. Operat. Res. **15** (3), 430–466 (1990).
7. E. G. Gol'shtein and N. A. Sokolov, "A New Method for Solving a Multi-Commodity Transportation Problem," Ekon. Mat. Met. **31** (2), 128–136 (1995).
8. A. Kamath and O. Palmon, "Improved Interior Point Algorithms for Exact and Approximate Solution of Multicommodity Flow Problems," in *Proceedings of 6th Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, USA, 1995*, pp. 502–511.
9. A. Itai, "Two-Commodity Flow," J. ACM **25** (4), 596–611 (1978).
10. O. Kenji, "Approximate Max-Integral-Flow/Min-Multicut Theorems," in *Proceedings of 36th Annual ACM Symposium on Theory of Computing, Chicago, USA, 2004*, pp. 539–545.
11. N. Garg and J. Konemann, "Faster and Simpler Algorithms for Multicommodity Flow and Other Fractional Packing Problems," in *Proceedings of 39th Annual Symposium on Foundations of Computer Science, Washington, USA, 1998*, pp. 300–309.

12. Yu. E. Malashenko and N. M. Novikova, *Uncertainty Models in Many-User Networks* (Editorial URSS, Moscow, 1999) [in Russian].
13. T. C. Hu, *Integer Programming and Network Flows* (Addison Wesley, New York 1969; Mir, Moscow, 1974).
14. N. Garg, V. V. Vazirani, and M. Yannakakis, "Primal-Dual Approximation Algorithms for Integral Flow and Multicut in Trees," Algorithmica **18** (1), 3–20 (1997).
15. M. Kh. Prilutskii, "Multicriteria Allocation in Hierarchical Systems," Avtom. Telemekh., No. 2, 24–29 (1996).
16. M. Kh. Prilutskii, "Allocation of Homogeneous Resource in Hierarchical Systems of Treelike Structure," in *Proceedings of International Conference on Systems Identification and Control Problems SISRRO'2000"*. (IPU im. V.A. Trapeznikova RAN., Moscow, 2000), pp. 2038–2049 [in Russian].
17. Z. Galil and E. Tardos, "A Min-Cost Flow Algorithm," J. ACM *35* (2), 374–386 (1988).
18. A. D. Gofman and D. B. Kraskal, "Integer Boundary Points of Convex Polyhedrons," in *Linear Inequalities and Associated Problems* (Inostrannaya Literatura, Moscow, 1959), pp. 325–347 [in Russian].
19. L. R. Ford and D. R. Fulkerson, *Flows in Networks* (Princeton Univ. Press, Princeton 1962; Mir, Moscow, 1966).