COMPUTATIONAL ALGORITHMS OF OPTIMIZATION OF CLOSED QUEUING NETWORKS

Z. L. Kruglyi and M. S. Murshtein UDC 62-505:65.012.122

An iteration method is proposed for optimizing closed queuing networks (QUN) that have the property of local balance. By studying the average delay time and the cost of a network as functions of the service rate, it is proved that the functional to be optimized is strongly convex. Iteration algorithms of optimization of closed QUN are studied, as well as the selection of initial points, and how to ensure the highest possible convergence rate.

1. INTRODUCTION

Optimization methods based on models of closed queuing networks (QUN) constitute an efficient tool for solving diverse problems of designing computational systems and networks [1-4]. In developing these methods, and especially in elaborating the corresponding facilities of automatic designing, an important task is to improve the computational algorithms of optimization and to estimate their convergence. This is due to the existing disproportionation between the computational means required for analysis and for design, respectively, as a result of which certain algorithms of optimization of closed QUN cannot be realized in practice.

In [1-4], the designing of closed QUN is based on the use of the method of undetermined Lagrange multipliers. Thus, for optimizing the productivity of a closed QUN, a nonlinear system of partial differential equations has been obtained in [2] that depends on the normalizing constant of the closed network, whereas in [1, 3, 4] the results of a study of the productivity, load, and average queue length as a function of the service rates are used for reducing the problem of optimization of a closed QUN to a system of nonlinear equations that do not contain partial derivatives. The system of nonlinear equations is solved with the aid of an algorithm of multidimensional minimization of the generalized Lagrange function. However, the insufficient convergence of such a universal algorithm constructed without taking into account the specific character of the functional to be optimized, and the large number of iterations in a search for a solution, make it inadequate for designing a QUN of large dimension and also in certain particular cases of optimization of closed networks.

By transforming the original problem of optimization of a closed QUN with constraints in the form of inequalities to a problem of search for an unconstrained extremum, we prove in this paper the strong convexity of the functional to be optimized, so that it is possible to use numerical optimization methods [5]. We propose an iteration method of optimization of closed QUN that makes it possible to increase the convergence rate as compared to conventional methods [1-4]. We also prove the convergence of our algorithm.

2. FORMULATION OF OPTIMIZATION PROBLEM

We shall study methods of solution of optimization problems for separable or locally balanced networks (LBN) [6]. By solving the optimization problem, we select an optimal network configuration that can be uniquely determined by the vector $x = (x_0, \ldots, x_M)$ of

Minsk Scientific-Industrial Society "Tsentrsistem." Translated from Avtomatika i Telemekhanika, No. 7, pp. 86-97, July, 1990. Original article submitted April 25, 1989.

 0005-1179/90/5107-0926$12.50

relative utilization factors or by the vector $\mu = (\mu_0, \ldots, \mu_M)$ of rates of service at the nodes of the network.

Suppose that a closed LBN contains M + 1 nodes and N calls. The service rate at the i-th node in the presence of n calls is equal to

$$\mu_i(n) = \begin{cases} \mu_i n & \text{for} \quad n \leqslant r_i, \\ \mu_i r_i & \text{for} \quad n \geqslant r_i, \quad i=\overline{0,M}, \quad n=\overline{0,N}, \end{cases}$$

where r_i is the number of servers at the i-th node. The circulation of calls in a network can be characterized by the transition probability matrix $P = \|P_{ik}\|$, $i,k=\overline{0,M}\left(\forall i: \sum_{k=0}^{M} P_{ik} = 1,\ 0 \leq P_{ik} \leq 1\right)$. The vector $\mathbf{e} = (e_0, \ldots, e_M)$ of relative rates of the flow of calls can be defined to within a multiplicative constant as a solution of the system

$$\mathbf{e}=\mathbf{e}P. \tag{1}$$

The productivity at the i-th node of a network is defined by the formula

$$\lambda_i(\mathbf{x}) = e_i G_{N-1}(\mathbf{x})/G_N(\mathbf{x}), \quad x_i = e_i/\mu_i, \quad i=\overline{0,M}, \tag{2}$$

where $G_N(\mathbf{x})$ is the normalizing constant of a closed network that contains N calls.

The average delay time of calls in a network can be obtained by the formula

$$t(\mathbf{x}) = \sum_{i=0}^{M}\left[\lambda_i(\mathbf{x}) \Big/ \sum_{i=0}^{M} \lambda_i(\mathbf{x})\right] t_i(\mathbf{x}), \tag{3}$$

where $t_i(\mathbf{x}) = L_i(N)/\lambda_i(\mathbf{x})$ is the average time of sojourn of calls at the i-th node of a network, and $L_i(N)$ the average queue length (taking into account the calls which are being served) at the i-th node.

By virtue of the relation $\lambda_i(\mathbf{x})/e_i = \text{const}$ $(i = \overline{0, M})$, formula (3) is transformed into

$$t(\mathbf{x}) = \sum_{i=0}^{M}\left[e_i \Big/ \sum_{i=0}^{M} e_i\right] \frac{L_i(N)}{\lambda_i(N)} = \frac{N}{\sum_{i=0}^{M} e_i} \frac{G_N(\mathbf{x})}{G_{N-1}(\mathbf{x})}. \tag{4}$$

The cost function of a network is defined as follows:

$$F(\mathbf{x}) = \sum_{i=0}^{M} c_i \mu_i^{a_i} = \sum_{i=0}^{M} c_i' x_i^{-a_i}, \tag{5}$$

where $c_i' = c_i e_i^{a_i}$, $i = \overline{0, M}$; c_i and c_i' are cost coefficients; the a_i's are coefficients of nonlinearity of the network nodes.

The problem of optimization of a closed QUN can be expressed in one of the following formulations.

Formulation 1. Minimize the average delay time with a constraint on the network cost, i.e.,

$$\min t(\mathbf{x}) = \left[N \Big/ \sum_{i=0}^{M} e_i\right] G_N(\mathbf{x})/G_{N-1}(\mathbf{x}) \tag{6}$$

under the constraints

$$F(\mathbf{x})=\sum_{i=0}^{M}c_i'x_i^{-a_i}\leqslant s,\quad 0<\chi<x_i<X<\infty,\quad i=\overline{0,M}. \tag{7}$$

Formulation 2. Minimize the network cost with a constraint on the average delay time, i.e.,

$$\min F(\mathbf{x})=\sum_{i=0}^{M}c_ix_i^{-a_i} \tag{8}$$

under the constraints

$$t(\mathbf{x})=\left[N\Big/\sum_{i=0}^{M}e_i\right]G_N(\mathbf{x})/G_{N-1}(\mathbf{x})\leqslant T,\ \ 0<\chi<x_i<X<\infty,\ \ i=\overline{0,M}. \tag{9}$$

The solution is sought on a set of values of the relative utilization factors $\mathbf{x}$ of the network nodes that are related to the service rates μ by formula (2).

3. ITERATION ALGORITHMS OF OPTIMIZATION OF CLOSED QUN

In this section we substantiate an iteration (gradient) method of solution of the problem of optimization of a closed QUN of guaranteed convergence. Compared to the algorithm based on the use of the method of undetermined Lagrange multipliers realized in [3, 4], the convergence rate of our iteration algorithm is higher in individual cases by more than one order of magnitude.

Let us note that in solving network optimization problems in the formulations 1 and 2, a global minimum is reached for $F(\mathbf{x}) = s$ and $t(\mathbf{x}) = T$, respectively. Indeed, as follows from [3], the functional of problem 1 which has to be minimized is a monotonically increasing function of the argument $\mathbf{x}$. It follows from formula (A.2) of the Appendix that the network cost function $F(\mathbf{x})$ is monotonically decreasing. According to [2], these properties of monotonicity of $t(\mathbf{x})$ and $F(\mathbf{x})$ make it possible to restrict the region of search for a global extremum of problem 1 or 2 to the surfaces $F(\mathbf{x}) = s$ or $t(\mathbf{x}) = T$ instead of the half-spaces defined by the corresponding inequality constraints.

Let us fix a so-called dependent j-th node of the network. The problem of optimization of a closed network in formulation 1 is equivalent to the unconstrained minimization problem

$$z(\mathbf{x}^j)\to\min, \tag{10}$$

where we denoted by $z(\mathbf{x}^j)$ the composite function (6) of M variables

$$\mathbf{x}^j=\{x_0,\dots,x_{j-1},x_{j+1},\dots,x_M\}.$$

The relative utilization factor x_j of a dependent node can be expressed in terms of the relative utilization factors $\mathbf{x}^j$ of the other nodes on the basis of the network cost constraint (7):

$$x_j=\varphi(\mathbf{x}^j)=\psi[s_j(\mathbf{x}^j)]=[s_j(\mathbf{x}^j)/c_j']^{-1/a_j}, \tag{11}$$

where $s_j(\mathbf{x}^j)$ is the portion of the total cost of the network that belongs to the j-th node:

$$s_j(\mathbf{x}^j)=s-\sum_{\substack{i=0\\ i\neq j}}^{M}c_i'x_i^{-a_i}>0,\quad j=\overline{1,M}. \tag{12}$$

Similarly, the problem of optimization in formulation 2 is equivalent to the problem of unconstrained minimization of the composite function

$$u(\mathbf{x}^j)\to\min. \qquad (13)$$

For a dependent node the quantity $x_j = g(\mathbf{x}^j)$ is an implicit function of the arguments $\mathbf{x}^j$ expressed by the relation

$$w(\mathbf{x})=t(\mathbf{x})-T=0. \qquad (14)$$

Let us show that x_j is uniquely determined by (14). By substituting (4) into (14), we obtain

$$\sum_{n=0}^{N} G_n^j(\mathbf{x}^j)x_j^{N-n} \Big/ \sum_{n=1}^{N} G_{n-1}^j(\mathbf{x}^j)x_j^{N-n} = T\sum_{i=0}^{M} e_i/N,$$

where $G_n^j(\mathbf{x}^j)$ is the normalizing constant of a closed network that contains n calls and which differs from the original network by the exclusion of the j-th node. Hence, x_j can be obtained as a solution of the exponential equation

$$x_j^N+\sum_{n=1}^{N}\left[G_n^j(\mathbf{x}^j)-G_{n-1}(\mathbf{x}^j)\frac{T\sum_{i=0}^{M}e_i}{N}\right]x_j^{N-n}=0. \qquad (15)$$

By virtue of Descartes' theorem (on the number of positive roots of an exponential equation), the uniqueness of a solution of Eq. (15) follows from the relation

$$G_n^j(\mathbf{x}^j)/G_{n-1}^j(\mathbf{x}^j)\leqslant T\sum_{i=0}^{M}e_i/N,\ n=\overline{1,N}.$$

Let us prove that as a result of transforming the original LBN optimization problem in the formulation 1 or 2 to a problem of search for an unconstrained extremum, the functional to be minimized in problem (10) or (13) is a strongly convex function of the arguments $\mathbf{x}^j$.

<u>THEOREM.</u> If the service rate at each node of a closed LBN is a nondecreasing function of the number of calls in the queue ($\mu_i(n+1)\geq\mu_i(n)$, $i=\overline{0,M}$, $n=\overline{0,N}$), then the objective functions $z(\mathbf{x}^j)$ and $u(\mathbf{x}^j)$ of the unconstrained minimization problems (10) and (13) will be strongly convex functions of the relative utilization factors $\mathbf{x}^j$ of the network nodes on a set of real positive numbers $0<\chi<x_i<\overline{\chi}<\infty$, $i=\overline{0,M}$.

This theorem is proved in the Appendix.

Taking into account the Appendix, it follows from the theorem that the iteration sequence $\{\mathbf{x}_m^j\}$ defined by the recursion relation

$$\mathbf{x}_{m+1}^j=\mathbf{x}_m-\alpha\,\mathrm{grad}\,z(\mathbf{x}_m^j),\qquad m=1,2,\ldots, \qquad (16)$$

converges, and it yields a solution of the optimization problem in the formulation 1, i.e., $\mathbf{x}=x_j\cup\mathbf{x}^j$.

The initial approximation $\mathbf{x}_1^j=\{x_{i1},\ i=\overline{0,M},\ i\neq j\}$ is selected from the cost constraint (7). At the step m = 1, the scaling during the computer realization of the algorithm was carried out by selecting a multiplicative constant in (1) that satisfies the condition

$$D_j^\varepsilon\leqslant x_{i1}\leqslant 10^w,\ x_{i1}\geqslant 1;\ 10^{-w}\leqslant x_{i1}\leqslant D_j^\zeta,\ x_{i1}<1,$$

where

$$D_j=\sum_{\substack{i=0\\ i\neq j}}^{M}c_i'/s;\ \varepsilon=1/\max_{\substack{i=\overline{0,M}\\ i\neq j}}\{a_i\},\ \zeta=1/\min_{\substack{i=\overline{0,M}\\ i\neq j}}\{a_i\};$$

$[10^{-W}, 10^{W}]$ being the machine order.

At the subsequent steps the scaling was realized by a conventional method [6].

Below we describe the m-th step of the algorithm.

The gradient of the function $z(x_m^j)$ and the coefficient α of the antigradient can be obtained by the formulas

$$\operatorname{grad} z(\mathbf{x}_m^j)=t_i^{(1)}(\mathbf{x}_m)+t_j^{(1)}(\mathbf{x}_m)x_i^{(1)}(\mathbf{x}_m^j),\ i=\overline{0,M},\ i\neq j;\ \alpha=2/K_2,$$

where K_2 is determined from (A.4).

The second differential of the function $z(x^j)$ can be calculated by the formula

$$d^2z(\mathbf{x}_m^j)=[t_{ik}^{2}(\mathbf{x}_m)+t_j^{(1)}(\mathbf{x}_m)x_{ik}^{(2)}(\mathbf{x}_m^j)+t_{ij}^{(2)}(\mathbf{x}_m)x_k^{(1)}(\mathbf{x}_m^j)+$$
$$+t_{jk}^{(2)}(\mathbf{x}_m)x_i^{(1)}(\mathbf{x}_m^j)+t_{jj}^{(2)}(\mathbf{x}_m)x_i^{(1)}(\mathbf{x}_m^j)x_k^{(1)}(\mathbf{x}_m^j)]dx_i\,dx_k,$$
$$i,k=\overline{0,M};\quad i,k\neq j.$$

Here $x_i^{(1)}(x_m^j)$ and $x_{ik}^{(2)}(x_m^j)$ denote the gradient and the Hessian of the function $x_j=\varphi(\mathbf{x}^j)$:

$$x_i^{(1)}=A_j s_j^{-(1/a_j+1)}B_i,\quad i=\overline{0,M},\quad i\neq j;$$

$$x_{ik}^{(2)}=\begin{cases}A_j[(1/a_j+1)s_j^{-(1/a_j+2)}B_i^{(2)}+s_j^{-(1/a_j+1)}(a_i+1)B_i/x_i],\\ i=\overline{0,M},\quad i\neq j,\\ A_j(1/a_j+1)s_j^{-(1/a_j+2)}B_iB_k,\quad i,k=\overline{0,M},\quad i,k\neq j,\quad i\neq k;\end{cases}$$

$$A_j=\frac{1}{a_j}c_j'^{(1/a_j)},\qquad B_i=a_ic_i'x_i^{-(a_i+1)}.$$

The gradient $t_i^{(1)}(x_m)$ and the Hessian $t_{ik}^{(2)}(x_m)$ of the function $t(x)$ are defined by the formulas

$$t_i^{(1)}=\frac{e_i}{\sum_{i=0}^{M}e_i}\frac{N}{\lambda_i(x_m)}\frac{Y_i}{x_i},\quad i=\overline{0,M}, \tag{17}$$

$$t_{ik}^{(2)}=\begin{cases}\dfrac{e_i}{\sum_{i=0}^{M}e_i}\dfrac{N}{\lambda_i(x_m)}[Y_i(Y_i-1)+D_i(N)-D_i(N-1)]/x_i^2,\quad i=\overline{0,M},\\[2ex] \dfrac{e_i}{\sum_{i=0}^{M}e_i}\dfrac{N}{\lambda_i(x_m)}\left[Y_iY_k/(x_ix_k)+\left(\dfrac{\partial L_i(N)}{\partial x_k}-\right.\right.\\[2ex] \left.\left.-\dfrac{\partial L_i(N-1)}{\partial x_k}\right)\Big/x_i\right],\quad i,k=\overline{0,M},\quad i\neq k,\end{cases} \tag{18}$$

where $Y_i = L_i(N) - L_i(N-1)$, $i = \overline{0, M}$; $L_i(n)$ and $D_i(n)$ denote the average queue length and the variance of the number of calls at the i-th node of a network that contains n calls.

According to [3], the partial derivatives $\partial L_i(N)/\partial x_k$ can be calculated by the formula

$$\frac{\partial L_i(N)}{\partial x_k} = -\frac{1}{x_k}\Big[L_i(N)L_k(N) - \sum_{n=1}^{N} L_k^{\,i}(N-n)nP_i(n)\Big], \tag{19}$$

$$i, k=\overline{0, M}, \quad i\neq k,$$

where $L_k^{\,i}(n)$ is the average queue length at the k-th node of a network with n calls that differs from the original network by the exclusion of the i-th node.

For determining the constant K_2, we reduce the inequality (A.4) to the form $Q(x^j) \leq 0$, and we obtain a solution by reducing the quadratic form $Q(dx^j)$ to canonical form.

Another method of determination of the constant K_2 is based on using the following estimates for the partial derivatives [3]:

$$-\frac{D_k(N)}{x_k} \leqslant \frac{\partial L_i(N)}{\partial x_k} \leqslant 0, \quad i, k=\overline{0, M}, \quad i\neq k. \tag{20}$$

For the elements $t_{ik}^{(2)}$ of the Hessian of the function $t(x)$ we obtain from (18) the formula

$$\max\Big\{0, \frac{e_i}{\sum_{i=0}^{M} e_i}\frac{N}{\lambda_i(x_m)}[Y_iY_k - D_k(N)]/(x_ix_k)\Big\} \leqslant \tag{21}$$

$$\leqslant t_{ik}^{(2)} \leqslant \frac{e_i}{\sum_{i=0}^{M} e_i}\frac{N}{\lambda_i(x_m)}[Y_iY_k + D_k(N-1)]/(x_ix_k), \quad i, k=\overline{0, M}, \quad i\neq k.$$

By using estimates (20) and (21) instead of the exact values of the partial derivatives (18) and (19) in calculating $d^2z(x^j)$, it is possible to simplify the obtaining of the constant K_2, since it is not necessary to calculate the normalizing constant for a network in which the node i is absent ($i = \overline{0, M}$). However, this causes a drop in the convergence rate.

Yet another method of determination of the constant K_2 is based on the use of the inequality $(a_1+a_2+\ldots a_m)^2 \leqslant (M+1)(a_1^2+a_2^2+\ldots+a_M^2)$. By setting $a_i = dx_i$, $i = \overline{0, M}$, $i \neq j$, we obtain from (A.4) the formula

$$K_2 = (M+1)\max\{z_{ik}^{(2)}(x_m^j),\ i, k=\overline{0, M},\ i, k\neq j\}, \tag{22}$$

where $z_{ik}^{(2)}(x_m^j)$, $i, k = \overline{0, M}$, $i, k \neq j$, is an element of the Hessian of the function $z(x^j)$.

At the expense of a certain decrease in the convergence rate, it is possible to determine by this method the constant K_2 in explicit form without resorting to numerical algorithms of canonization of matrices and quadratic forms.

A condition of completion of the iteration procedure is the vanishing of $\operatorname{grad} z(x_m^j)$ or the obtaining of the desired value of the error with respect to the functional.

Similarly, the iteration procedure

$$x_{m+1} = x_m^j - \beta \operatorname{grad} u(x_m^j), \quad m=1, 2, \ldots \tag{23}$$

determines the solution of the optimization problem in the formulation 2.

The initial approximation is selected on the basis of the average delay time constraint (9).

Let us consider the m-th step of the algorithm.

The gradient of the functions $u(x_m^j)$ and the coefficient β of the antigradient can be obtained by the formulas

$$\operatorname{grad} u(x_m^j) = F_i^{(1)}(x_m) + F_j^{(1)}(x_m)x_i^{(1)}(x_m^j), \quad i=\overline{0, M},$$

$$i\neq j; \quad \beta = 2/K_2,$$

where K_2 can be determined from (A.4).

The second differential of the function $u(x_m^j)$ is calculated by the formula

$$\partial^2 u(\mathbf{x}_m^j)=[F_{ik}^{(2)}(\mathbf{x}_m)+F_j^{(1)}x_{ik}^{(2)}(\mathbf{x}_m^j)+F_{ij}^{(2)}(\mathbf{x}_m)x_k^{(1)}(\mathbf{x}_m^j)+ +F_{jk}^{(2)}(\mathbf{x}_m)x_i^{(1)}(\mathbf{x}_m^j)+F_{jj}^{(2)}(\mathbf{x}_m)x_i^{(1)}(\mathbf{x}_m^j)x_k^{(1)}(\mathbf{x}_m^j)]dx_i\,dx_k, \quad i,k=\overline{0,M},\ i,k\neq j. \tag{24}$$

Here $x_i^{(1)}(x_m^j)$ and $x_{ik}^{(2)}(x_m^j)$ denote the gradient and the Hessian of the implicit function $x_j = g(\mathbf{x}^j)$ specified by (14):

$$x_i^{(1)}=-t_i^{(1)}(\mathbf{x}_m)/t_j^{(1)}(\mathbf{x}_m), \quad i=\overline{0,M},\ i\neq j,$$

$$x_{ik}^{(2)}=-t_{ik}^{(2)}(\mathbf{x}_m)/t_j^{(1)}(\mathbf{x}_m), \quad i,k=\overline{0,M},\ i,k\neq j,$$

where $t_i^{(1)}$, $i = \overline{0, M}$, and $t_{ik}^{(2)}$, $i, k = \overline{0, M}$, are specified by (17) and (18); $F_i^{(1)}(x_m)$, $i = \overline{0, M}$ and $F_{ik}^{(2)}(x_m)$, $i, k = \overline{0, M}$, denote the gradient and the Hessian of the function $F(\mathbf{x})$ specified by (A.1) and (A.2), respectively.

In solving the inequality (A.4), it is possible to simplify the algorithm by using, instead of the second differential $d^2u(\mathbf{x}^j)$, its upper bound obtained from (24), taking into account (21).

The constant K_2 is expressed by the formula

$$K_2=(M+1)\max\{u_{ik}^{(2)}(\mathbf{x}_m^j),\ i,k=\overline{0,M},\ i,k\neq j\},$$

where $u_{ik}^{(2)}(x^j)$, $i, k = \overline{0, M}$, $i, k \neq j$ is the Hessian of the function $u(x^j)$.

A condition of completion of the procedure (23) is the relation $\mathrm{grad}\,u(\mathbf{x}_m^j) = 0$, or the obtaining of the desired value of the error with respect to the functional.

For estimating the convergence rate of the iteration procedure (16) or (23), it is possible to use Goldstein's theorem for a quasi-Newtonian algorithm [5]. The convergence of the sequence $\{x_m^j\}$ to the optimum value $\mathbf{x}$ is superlinear, with $\|x_m^j - \mathbf{x}\| \to 0$ for $m \to \infty$ faster than any geometric progression $E\theta^m$, $m = 1, 2, \ldots$, $\|\cdot\|$ is the norm in R^M (R^M denotes an M-dimensional Euclidean space), $\theta \in (0, 1]$, and E is a unit matrix.

The iteration procedures (16) and (23) are suitable for solving problems of maximization of the productivity with a constraint on the cost of a network (formulation 3), or minimization of the network cost with a productivity not below the assigned value (formulation 4). The solutions are the same for the formulations 1 and 3, whereas for the formulations 2 and 4 they differ only in the method of determination of the relative utilization factor of a dependent network node which can be obtained from the constraint on the average delay time, or the constraint on the network productivity [4].

4. EXAMPLE OF MINIMIZATION OF AVERAGE DELAY TIME WITH A CONSTRAINT ON THE COST OF A COMPUTATIONAL NETWORK

The original data are: The number of computational network channels $M = 7$; the LBN contains $M + 1$ nodes (the zero node is a "source"); the average length of a packet v_i = 512 bit; the cost coefficients k_i = 11.3 (bit/sec)/rubles; the coefficient of nonlinearity at the network nodes $a_i = 0.67$, $i = \overline{1, M}$; the transition probability matrix for a network with nodes $i = \overline{0, M}$:

$$P=\begin{vmatrix} 0 & 0.217 & 0.1305 & 0.1305 & 0.087 & 0.1305 & 1.174 & 0.1305 \\ 0.9 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.666 & 0 & 0 & 0 & 0 & 0.167 & 0 & 0.167 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.833 & 0 & 0 & 0.167 & 0 & 0 & 0 & 0 \\ 0.875 & 0.125 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.833 & 0 & 0 & 0.167 & 0 & 0 & 0 & 0 \end{vmatrix}.$$

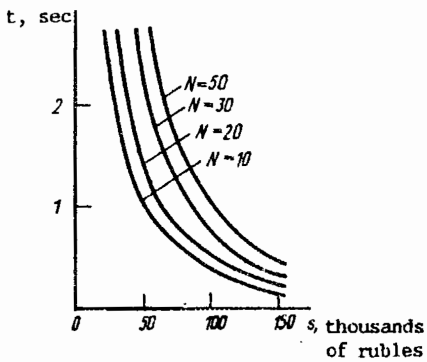

Fig. 1

TABLE 1

Number of calls in network, N	Optimum average delay time, t, sec	Optimum channel capacities, bit/sec						
		w_1	w_2	w_3	w_4	w_5	w_6	w_7
10	0,29	10 933	6433	8638	4553	7712	9220	7715
20	0,47	11 212	6383	8746	4413	7738	9344	7725
30	0,64	11 312	6339	8767	4361	7729	9382	7731
50	1,00	11 364	6286	8756	4258	7699	9380	7692

The solution of Eq. (1) of balance of flows is: $e_1 = 0.2418$, $e_2 = 0.1305$, $e_3 = 0.1844$, $e_4 = 0.087$, $e_5 = 0.1613$, $e_6 = 0.1982$, $e_7 = 0.1613$. Let us take the dependent network node as $j = 0$. As the initial approximation we take $x_i = 1$, $i = \overline{1, M}$. The solution will be obtained by the iteration procedure (16). The relative utilization factor of the node $j = 0$ is calculated by formula (11). The network channel (node) capacities are expressed by the formula $w_i = \mu_i \nu_i$, $i = \overline{1, M}$.

In Table 1 we listed the results of solving the problem of minimization of the average delay time with a constraint on the cost of a computational (or communications) network $s = 100{,}000$ rubles. For $N = 30$ a solution was obtained by the gradient method after 10 iterations with an error for the functional equal to $\Delta = \|x_{m+1} - x_m\| = 10^{-3}$. The method of undetermined Lagrange multipliers [3] yielded an optimum with the same accuracy after 200 iterations. For $N = 50$ a solution with this accuracy was obtained for the first and second methods after 25 and 600 iterations, respectively. In Fig. 1 we plotted the average delay time versus the cost for various numbers of calls in the network.

The above iteration method of optimization of closed QUN has been programmed in the PL/1 language and incorporated in the professional technical guide manual published by Minpribor (Moscow Institute of Instruments) [8]. There we can also find examples of solution of various practical problems of optimization of computational systems and networks.

APPENDIX

Proof of Theorem. The proof of the theorem is preceded by the following four lemmas.

LEMMA 1. The second differential of the function $t(x)$ is nonnegative.

Proof. It is proved in [2] that the ratio $G_N(x)/G_{N-1}(x)$ is a convex function of $\mathbf{x}$. The coefficient $N\Big/\sum_{i=0}^{M} e_i$ of this ratio in (4) is a quantity that does not depend on x, since in solving an optimization problem the number of calls and the network topology are fixed. For a convex twice-continuously differentiable function $t(x)$ we hence obtain $d^2 t(x) \geq 0$.

LEMMA 2. $F(x)$ is a strongly convex function of the relative utilization factors x of the network nodes.

Proof. By virtue of the expressions for the partial derivatives

$$\partial F(\mathbf{x})/\partial x_i = -c_i x_i^{-(a_i+1)} \quad a_i<0, \quad i=\overline{0, M}; \tag{A.1}$$

$$\frac{\partial^2 F(\mathbf{x})}{\partial x_i \partial x_k} = \begin{cases} c_i' x_i - (a_i+2)a_i(a_i+1), & i=\overline{0, M}, \quad i=k; \\ 0, \quad i, k=\overline{0, M}, & i \neq k, \end{cases} \tag{A.2}$$

we obtain for the second differential of the network cost function (5):

$$\partial^2 F(\mathbf{x}) = \sum_{i=0}^{M} c_i' x_i^{-(a_i+2)} a_i(a_i+1)\, dx_i^2.$$

Hence $d^2F(\mathbf{x}) > 0$ and there exists a constant K_1 ($K_1 > 0$) such that

$$K_1(d\mathbf{x})^2 \leq d^2 f(\mathbf{x}), \tag{A.3}$$

where dx denotes the vector $(dx_0, \ldots, dx_M)$.

Since the domain of variation of x is bounded, there exists a boundary K_2 ($K_2 \geq K_1 > 0$):

$$d^2 f(\mathbf{x}) \leq K_2(d\mathbf{x})^2. \tag{A.4}$$

It then follows from the definition that $F(x)$ is a strongly convex function of the argument .

LEMMA 3. $x_j = \varphi(x^j)$ is a strongly convex function of the relative utilization factors x^j of the network nodes.

Proof. By virtue of (11) and (12) we obtain for the second differential of a composite function the expression

$$d^2 x_j = d^2\psi[s_j(\mathbf{x}^j)] + d\psi[s_j(\mathbf{x}^j)]/ds_j(\mathbf{x}^j)\, d^2 s_j(\mathbf{x}^j) > 0. \tag{A.5}$$

Indeed, $d^2 s_j(x^j) < 0$ by being the second differential of a strongly concave function defined by (12);

$$d\psi[s_j(\mathbf{x}^j)]/ds_j(\mathbf{x}^j) = c_j'^{(1/a_j)} s_j(\mathbf{x}^j)^{-(1/a_j+1)}\left(-\frac{1}{a_j}\right) < 0,$$

$$d^2\psi[s_j(\mathbf{x}^j)]/ds_j^2(\mathbf{x}^j) = c_j'^{(1/a_j)} s_j(\mathbf{x}^j)^{-(1/a_j+2)} \frac{1}{a_j}\left(\frac{1}{a_j}+1\right) > 0,$$

hence $d^2\psi[s_j(x^j)] > 0$.

It follows from (A.5) that there exists a constant K_1 such that (A.3) is satisfied. From the boundedness of the domain of variation of x there follows the existence of a boundary K_2. It then follows from the definition that $x_j = \varphi(x^j)$ is a strongly convex function of x^j.

LEMMA 4. $x_j = g(x^j)$ is a concave function of the relative utilization factors x^j of the network nodes.

Proof. By equating to zero the second differential of the left-hand side of (14) which specifies the implicit function $x_j = g(x^j)$, we obtain

$$d^2 w(\mathbf{x}) = d^2 t(\mathbf{x}) + \partial t(\mathbf{x})/\partial g(\mathbf{x}^j)\, d^2 g(\mathbf{x}^j) = 0.$$

From (4), by using (2) and (A.6) [4]:

$$\partial G(N)/\partial x_i = \frac{1}{x_i} G(N) L_i(N), \tag{A.6}$$

we obtain the following formula:

$$\partial t(\mathbf{x})/\partial x_i = \left[e_i \Big/ \sum_{i=0}^{M} e_i\right][N/\lambda_i(\mathbf{x})][L_i(N) - L_i(N-1)]/x_i, \quad i=\overline{0, M}. \tag{A.7}$$

By virtue of Lemma 1 and of (A.7) we obtain

$$d^2g(\mathbf{x}^j)=-d^2t(\mathbf{x})/(\partial t(\mathbf{x})/\partial g(\mathbf{x}^j))\leq 0, \tag{A.8}$$

$$\partial t(\mathbf{x})/\partial g(\mathbf{x}^j)\neq 0. \tag{A.9}$$

The inequality (A.8) proves the result of Lemma 4.

Proof of Theorem. The second differential of the function $z(x^j)$ as the differential of a composite function (j being a dependent node of the network) has the form

$$d^2z(\mathbf{x}^j)=d^2t(\mathbf{x})+\partial t(\mathbf{x})/\partial x_j(\mathbf{x}^j)\,d^2x_j(\mathbf{x}^j)>0. \tag{A.10}$$

Indeed, $d^2t(x) \geq 0$ by virtue of Lemma 1, and $d^2x_j(x^j) > 0$ by virtue of Lemma 3.

By virtue of a lemma from [2] we have

$$L_i(N)-L_i(N-1)\geqslant 0,\quad x_i\geqslant 0,\quad i=\overline{0,M}.$$

It then follows that $\partial t(x)/\partial x_j(x^j) > 0$ under the condition

$$L_j(N)-L_j(N-1)\neq 0. \tag{A.11}$$

It follows from (A.9) that there exists a constant K_1 such that (A.3) is satisfied. By virtue of the boundedness of the domain of variation of x^j there exists a boundary K_2. It then follows from the definition that $z(x^j)$ is a strongly convex function of x^j.

The second differential of the function $u(x^j)$ as the differential of a composite function (j being a dependent node of the network) has the form

$$d^2u(\mathbf{x}^j)=d^2F(\mathbf{x})+\partial F(\mathbf{x})/\partial x_j(\mathbf{x}^j)\,d^2x_j(\mathbf{x}^j)>0. \tag{A.12}$$

Indeed, $d^2F(x) > 0$ (by virtue of Lemma 2); $\partial F(x)/\partial x_j(x^j) < 0$ [by virtue of (A.1)]; $d^2x_j(x^j) \leq 0$ (by virtue of Lemma 4).

It follows from (A.12) that there exists a constant K_1 such that (A.3) is satisfied. It hence follows from the boundedness of the domain of variation x^j that there exists a boundary K_2. By virtue of the definition it then follows that $u(x^j)$ is a strongly convex function of x^j. This completes the proof of the theorem.

It follows from a theorem formulated in [7] that if the function $f(x)$ is strongly convex on a convex set R, then the iteration sequence $\mathbf{x}_{M+1}=\mathbf{x}_m-\alpha\,\mathrm{grad}\,f(\mathbf{x}_m)$, $m = 1, 2, \ldots$ [$0 < \alpha < 2/K_2$, K_2 being a constant specified by (A.4), and $\mathrm{grad}\,f(x_m)$ the gradient of the function $f(x)$ at the point x_m] will converge, beginning with a point x_1 of the set R, to a point of local minimum of the function $f(x)$.

Let us note that the conditions (A.9) and (A.11) are satisfied in closed LBNs for $x_i \neq 0$, $i = \overline{0, M}$, and they do not require special verification. However, in the case of computer realization of optimization algorithms, the difference $L_i(N) - L_i(N - 1)$ may vanish for a dependent node ($i = j$) of the network. Therefore, the node with smallest load should not be taken as a dependent node of the network.

The requirement of boundedness of the domain of variation of x introduced in the formulations 1 and 2 of optimization problems is a consequence of the physical meaning. The value $x_i = 0$ corresponds to closing (short-circuiting), and $x_i = \infty$ to opening the i-th node, $i = \overline{0, M}$, in a closed QUN that contains M + 1 nodes.

LITERATURE CITED

1. V. A. Zhozhikashvili and V. M. Vishnevskii, Queuing Networks. Theory and Applications to Computer Networks [in Russian], Radio i Svyaz', Moscow (1988).
2. K. S. Trivedi and R. A. Wagner, "A decision model for closed queuing networks," IEEE Trans. Soft. Eng., SE-5, No. 4, 328-332, July (1979).
3. V. M. Vishnevskii and Z. L. Kruglyi, "Optimization of closed stochastic networks," Avtomat. Telemekh., No. 2, 32-39 (1987).
4. V. M. Vishnevskii, Z. L. Kruglyi, and M. S. Murshtein, "Designing closed queuing networks according to criteria of response, productivity, and cost," in: Abstracts of Reports of the 2nd All-Union Conference on Distributed Automatic Queuing Systems [in Russian], Institute of Control Problems, Moscow (1986), pp. 75-81.

5. E. Polak, Computational Methods in Optimization, Academic Press, New York (1971).
6. S. C. Bruell and G. Balbo, Computational Algorithm for Closed Queuing Network, North-Holland, New York (1980).
7. V. A. Il'in and É. G. Poznyak, Foundations of Mathematical Analysis [in Russian], Part 1, Nauka, Moscow (1982).
8. Technical Guide Book TGB (RTM) 25 212-3-86, "Institute of Constrol Systems (IASU). Systematic instructions for compiling software. Technique of selection of the configuration, and calculation of productivity and completion of distributed computational systems and networks" [in Russian], Minpribor (Moscow Institute of Instruments), Moscow (1987).