Proceedings of the Steklov Institute of Mathematics, Vol. 239, 2002, pp. 306–314.
Translated from Trudy Matematicheskogo Instituta imeni V.A. Steklova, Vol. 239, 2002, pp. 323–331.
Original Russian Text Copyright © 2002 by Shchepin.
English Translation Copyright © 2002 by MAIK "Nauka/Interperiodica" (Russia).

On the Geometry of Multiprocessor Distributions

E. V. Shchepin[1]

Received April 2001

Abstract—An algorithm for solving the linear programming problem known as the multiprocessor distribution (or scheduling) problem is suggested. The problem is to distribute a given set of tasks among given processors so as to minimize the load time of the most loaded processor. Dividing the tasks into parts and distributing the parts among different processors is allowed. The algorithm constructed uses the specifics of the multiprocessor distribution problem and can therefore operate substantially more efficiently than the general linear programming algorithm. The author was unable to answer the question about the polynomiality of the algorithm.

1. INTRODUCTION

The problem of multiprocessor distribution is an interpretation of the following matrix optimization problem: Given a positive $m \times n$ matrix, construct a stochastic $n \times m$ matrix such that the maximum diagonal element of the product of these matrices is minimal. A matrix is stochastic if it is nonnegative (i.e., has nonnegative elements) and the sum of elements in each row equals 1.

If the solution is required to be an integer stochastic matrix (which corresponds to an interruption-free distribution), then the problem is NP-hard [1] even for $m = 2$. In the case of arbitrary stochastic matrices (distribution with interruptions), the problem can be solved in polynomial time (see [2–4]). In this case, it reduces to the following linear programming problem, where $C_{\max}$ is to be minimized:

$$\sum_{j=1}^{n} p_i^j x_i^j \le C_{\max}, \qquad i = 1, 2, \dots, m, \tag{1}$$

$$\sum_{i=1}^{m} x_i^j = 1, \qquad j = 1, 2, \dots, n, \tag{2}$$

$$x_i^j \ge 0. \tag{3}$$

Here, x_i^j and $C_{\max}$ are unknowns, and the coefficients p_i^j are positive.

This paper suggests an algorithm for solving the above problem. The algorithm uses the specifics of the problem and is an alternative to linear programming methods.

2. MULTIPROCESSOR DISTRIBUTIONS

We call a linearly ordered set of processors $\mathcal{P} = \{P_1, \dots, P_m\}$ an *m-multiprocessor*. A linearly ordered set of tasks $\mathcal{J} = \{J^1, J^2, \dots, J^n\}$ is called an *m-multitask*. A *processor form* is a mapping $\mathcal{P} \times \mathcal{J} \to \mathbb{R}^+$ to the set of positive real numbers. The value of the processor form at a pair P, J is denoted by $P(J)$ and interpreted as the time required to completely accomplish the task J on the processor P.

[1] Steklov Institute of Mathematics, Russian Academy of Sciences, ul. Gubkina 8, Moscow, 117966 Russia.
E-mail: scepin@mi.ras.ru; scepin@mail.ru

A *processor–task system* is a triplet consisting of a multiprocessor, a multitask, and their processor form.

A *distribution* of a multitask $\mathcal{J}$ on a multiprocessor $\mathcal{P}$ is a function $\delta\colon \mathcal{P}\times\mathcal{J}\to\mathbb{R}^+$ that assigns a nonnegative number $\delta(P,J)\le 1$ to each pair P,J (processor–task). This number indicates the share of the task J distributed on the processor P in such a way that the sum of shares of every task over all processors equals 1. Thus, we consider distributions with *interruptions*, which admit dividing tasks into parts and distributing these parts among different processors.

For a distribution δ and a processor–task pair P,J, the *processing time* of the task J on the processor P under the distribution δ is the product $\delta(P,J)P(J)$; it is denoted by $P^\delta(J)$.

For a given processor, the sum of all processing times over all tasks under a given distribution is called the *load time*. The maximum load time over all processors is the *distribution load*. The distribution under which all processors are loaded equally is called *uniform*. A distribution with minimum possible load is said to be *optimal*.

3. TRANSMISSIONS

We define the *signature* of a distribution δ to be the function $\operatorname{sgn}\delta$, where $\operatorname{sgn}x=\frac{x}{|x|}$ if $x\ne 0$ and $\operatorname{sgn}0=0$. For given processor P and distribution δ, we write $\delta(P)=\{J\mid\delta(P,J)>0\}$ to denote the set of tasks distributed on P by δ.

For any pair of processors P,Q and any task J, we denote the ratio $\frac{Q(J)}{P(J)}$ by $[P,Q]^J$; the minimum of $[P,Q]^J$ for $J\in\delta(P)$ is called the *coefficient of transmission* from P to Q and denoted by $[P,Q]_\delta$. Note that the transmission coefficient depends only on the signature of δ.

By $J(P,Q)$, we denote the task J with minimum number for which $[P,Q]^J=[P,Q]_\delta$. We omit the subscript δ if there is no risk of confusion.

For a sequence of processors $P_1,P_2,\dots,P_k$, the product $[P_1,P_2]_\delta[P_2,P_3]_\delta\dots[P_{k-1},P_k]_\delta$ is denoted by $[P_1,P_2,\dots,P_k]_\delta$ and called the transmission coefficient of the processor sequence $P_1,P_2,\dots,P_k$. A sequence of processors is called a *chain* if all processors in this sequence, possibly except the first and last, are different. If $P_k=P_1$, then the chain is called a *cycle*. A cycle with transmission coefficient less than one is said to be *reducing*.

Suppose that $P_1,\dots,P_k$ is a processor chain and ε is a positive number. Set $\varepsilon_i=[P_1,P_2,\dots,P_i]\varepsilon$. We say that a distribution δ' is obtained from a distribution δ by a *simple ε-transmission* along the chain $P_1,\dots,P_k$ if

(i) $P_j^{\delta'}(J(P_j,P_{j+1}))=P_j^{\delta}(J(P_j,P_{j+1}))-\varepsilon_j$ for $j<k$;

(ii) $P_j^{\delta'}(J(P_{j-1},P_j))=P_j^{\delta}(J(P_{j-1},P_j))+\varepsilon_j$ for $j>1$;

(iii) $\delta'(P_j,J)=\delta(P_j,J)$ for J different from $J(P_{j-1},P_j)$ and $J(P_j,P_{j+1})$.

Thus, under a simple ε-transmission, the load of the first machine in the chain decreases by ε, the load of the last increases by ε_k, and the loads of all other machines remain unchanged. At the ith processor of the chain, the portion of task $J(P_{j-1},P_j)$ arrives from the preceding processor, and the portion $J(P_j,P_{j+1})$ with the same load time departs (it is transmitted to the next machine).

The distribution δ' determined by the above formulas is well-defined only if $P_j^\delta(J(P_j,P_{j+1}))-\varepsilon_j\ge 0$ for any j. We define the *transmission capacity* of a chain $P_1,\dots,P_k$ as the largest ε for which the simple ε-transmission is well-defined. The transmission capacity of the chain is then determined by the formula

$$\min_j\frac{P_j^\delta(J(P_j,P_{j+1}))}{[P_1,P_2,\dots,P_j]}.$$

A *maximal ε-transmission*, that is, a simple ε-transmission for which ε equals the capacity of the chain, changes the task signature of the distribution and, as a rule, affects the transmission coefficient.

If the chain is a cycle, then the first machine coincides with the last and the load of this machine changes by $\varepsilon_k - \varepsilon$. This value is positive if the transmission coefficient of the chain is greater than 1, and it is negative otherwise.

A cycle $P_1, \dots, P_k$ is called *degenerate* if $J(P_i, P_{i+1})$ is the same for all $i < k$. For a degenerate cycle, the transmission coefficient equals 1. A cycle is called a *signature* cycle if $\delta(P_{i+1}, J(P_i, P_{i+1})) > 0$ for any $i < k$. A nonmaximal simple ε-transmission along a signature cycle does not change the distribution signature because the tasks are transmitted only to machines that have already been assigned them.

A distribution for which all signature cycles are degenerate is said to be *acyclic*.

Lemma 1 [5, Lemma 5.1]**.** *Any distribution can be transformed into an acyclic distribution without increasing the loads of processors.*

Proof. First, we transform the distribution in such a way that any two different machines share no more than one task. If a pair of machines P, Q share a task J, then $[P, Q] \geq \frac{Q(J)}{P(J)}$ and $[Q, P] \geq \frac{P(J)}{Q(J)}$; therefore, $[P, Q, P] \leq 1$. Thus, we can perform a maximal transmission along the cycle P, Q, P. In this case, a certain task moves from one machine to the other. If the machines still share some task after this maximal transmission, then we again perform such a transmission, and so on. Since every maximal transmission changes the distribution signature and no task transfers between the machines more than once, the process terminates after a finite number of steps.

For a distribution where no two machines share tasks, consider the signature graph whose vertices are processors and whose edges correspond to the tasks shared by pairs of processors [3]. If a pair of processors P, Q shares a task I, then $[P, Q] \geq \frac{Q(I)}{P(I)}$ and $[Q, P] \geq \frac{P(I)}{Q(I)}$. Therefore, $[P, Q][Q, P] \leq 1$. For the same reason, for any cycle in the signature graph, the product of the transmission coefficients of the cycle and its inverse (the cycle with opposite orientation) is less than or equal to 1. Hence, we can choose an orientation of an arbitrary cycle in such a way that the cycle with this orientation is nonenhancing. A maximal transmission along such a cycle destroys it. In this way, we can destroy all nondegenerate signature cycles without increasing the loads of the machines. $\square$

4. AN OPTIMALITY CRITERION

A distribution is called *free* if any cycle with unit transmission is degenerate.

Lemma 2. *Any acyclic distribution δ can be made free by an arbitrarily small change of the processor form while preserving its uniformity.*

Proof. Indeed, it suffices to replace the values $P(J)$ for which $\delta(P, J) = 0$ by close algebraically independent numbers. This does not affect the machine loads. For any nondegenerate cycle, every coefficient $[P_i, P_{i+1}]$ for which $\delta(P_{i+1}, J(P_i, P_{i+1})) = 0$ becomes a number algebraically independent of the other transmission coefficients of the cycle; therefore, the product of all these coefficients cannot equal 1. $\square$

Theorem 1 (optimality criterion)**.** *A distribution has the minimum load among all possible distributions for a given multiprocessor distribution problem if and only if it is uniform and has no reducing cycles.*

Proof. If a distribution is not uniform, then there is a machine that is loaded lighter than the machine with maximum load, and we can redistribute the tasks by slightly decreasing their shares on maximally loaded machines and increasing the shares on underloaded machines. This reduces the maximum load, which shows that the nonuniform distributions are not optimal. If a distribution involves a reducing cycle, then we can reduce the load of any processor in this cycle without changing the loads of all other processors by means of a simple ε-transmission. In this way, we can construct a nonuniform distribution with a load no heavier than initial. Since the nonuniform distributions are nonoptimal, the initial distribution is also nonoptimal.

Conversely, let us prove that any uniform distribution δ without reducing cycles is optimal. First, consider the case when the distribution δ is free.

Suppose that there is a distribution δ' with a lighter load. We say that a task J is *outgoing* for a processor P if $\delta'(P,J) < \delta(P,J)$, and it is *ingoing* if $\delta'(P,J) > \delta(P,J)$. Note that any processor has at least one outgoing task because its load under δ is heavier than that under δ'.

A chain of processors $P_1, \dots, P_k$ is said to be *outgoing* if, for each $i < k$, there is a task J_i that is outgoing for P_i and ingoing for P_{i+1}.

Let us show that, under the conditions of the theorem, there is always an outgoing cycle. The construction can be started with taking any processor as P_1. If an outgoing chain $P_1, \dots, P_k$ is already constructed, then we consider any task J outgoing from P_k and take any processor for which J is ingoing as P_{k+1} (such a processor always exists). We terminate the construction as soon as there occurs a repetition in the chain. The piece of the chain between the coinciding processors is an outgoing cycle.

Since no outgoing task is simultaneously ingoing, the cycle constructed is nondegenerate. It has transmission coefficient greater than 1 because it is free. But then, the transmission coefficient of the inverse cycle in δ' is smaller than 1, which contradicts the optimality of δ'.

Now, consider an arbitrary uniform distribution δ without reducing cycles. Assuming it to be nonoptimal, consider an optimal distribution δ' with a lighter load. Let us modify the distribution δ without changing the machine loads in such a way that the modified distribution has no nondegenerate signature cycles. The modified distribution still has no reducing cycles and remains uniform and nonoptimal. We denote it by the same symbol δ. Next, we “free” the processor form in accordance with Lemma 2. This does not affect the load of δ, while the load of δ' may somewhat change. We perturb the processor form so slightly that the load of δ' remains lighter than that of δ. With the new processor form being uniform without reducing cycles and free, the distribution δ is not optimal, in contradiction with what is proved above. □

5. THE GEOMETRY OF TRANSMISSIONS

A *direct* transmission from a machine P to a machine Q is a change in distribution such that it transfers some part of the task $J(P,Q)$ from P to Q; in other words, a direct transmission is a simple ε-transmission with some ε along the chain P, Q.

A transmission is called *admissible* if the arising distribution has no reducing cycles.

By $[P..Q]$, we denote the minimum transmission coefficient over all possible chains connecting P to Q.

Lemma 3 (admissibility criterion)**.** *A direct transmission from P to Q is admissible in a distribution δ if and only if $[P,Q]_\delta = [P..Q]_\delta$.*

Proof. If $[P,Q]$ is not minimal, then $[P,P_1,\dots,P_k,Q] < [P,Q]$ for some chain $P,P_1,\dots,P_k,Q$. The cycle $P,P_1,\dots,P_k,Q,P$ is reducing in the modified distribution. Indeed, in the modified distribution, Q is assigned a task $I = J(P,Q)$ and, therefore, $[Q,P]_{\delta'} \le \frac{P(I)}{Q(I)} = \frac{1}{[P,Q]}$, whence

$$[P,P_1,\dots,P_k,Q,P] \le \frac{[P,P_1,\dots,P_k,Q]}{[P,Q]} \le 1.$$

This proves the necessity of the minimality condition for the admissibility.

Now, suppose that $[P,Q]$ is the shortest transmission. The signature of the distribution arising as a result of the transmission may differ from the signature of the initial distribution only on the machines P and Q. Moreover, it can become only poorer on the machine P (in the case where the entire $I = J(P, Q)$ has departed P) and only richer on the machine Q (because of the arrival of I at this machine). Let us consider an arbitrary processor cycle and evaluate its transmission coefficient. If the cycle does not contain Q, then all transmission coefficients of its links in the new signature are no smaller than those in the initial signature, and none of such cycles is reducing because the initial signature has no reducing cycles. The same applies to the cycles containing Q and such that $J(Q, Q_1) \neq I$, where Q_1 is the processor following Q in the cycle. It remains to consider the cycles of the form $Q, Q_1, \ldots, Q_k, Q$ for which $J(Q, Q_1) = I$. The minimality condition for $[P, Q]$ gives $[P, Q] \leq [P, Q_1, \ldots, Q_k, Q]$. Next, $[P, Q_1] \leq \frac{Q_1(I)}{P(I)}$. These two inequalities imply the third inequality

$$\frac{Q_1(I)}{P(I)}[Q_1, \ldots, Q_k, Q] \geq [P, Q],$$

whence

$$[Q_1, \ldots, Q_k, Q] \geq [P, Q]\frac{P(I)}{Q_1(I)} = \frac{Q(I)}{P(I)}\frac{P(I)}{Q_1(I)} = \frac{Q(I)}{Q_1(I)}.$$

Therefore, $\frac{Q_1(I)}{Q(I)}[Q_1, \ldots, Q] \geq 1$. Finally, $\frac{Q_1(I)}{Q(I)} = [Q, Q_1]$ implies that the cycle $[Q, Q_1, \ldots, Q_k, Q]$ has transmission coefficient greater than or equal to 1. $\square$

Two admissible transmissions are called *consistent* if each of them remains admissible after the other is performed.

Consider a set of processors M and a processor P. We define $[M..P]$ to be the minimum of transmission coefficients over the paths connecting processors from M with P. A path from M to P is said to be *shortest* if its transmission coefficient equals $[M..P]$. An admissible pair P, Q is called M-radial if there is a shortest path from M to Q passing through P.

Lemma 4. *If processor pairs P, Q and P', Q' are radial with respect to a set M, then they are consistent.*

Proof. Let $I = J(P, Q)$. Suppose that transferring a part of the task I from P to Q makes the direct transmission $P' \to Q'$ inadmissible, i.e., such that the transmission coefficient of a certain path from P' to Q' is smaller than $[P', Q']$. Then, this path necessarily passes through Q and, if R is the processor following Q in this path, $[Q, R] = [Q, R]^I$. In this case, we have

$$[P'..Q][Q, R]^I[R..Q'] < [P', Q']. \tag{4}$$

The condition that P, Q is M-radial implies

$$[M..P][P, Q]^I \leq [M..P'][P', Q]. \tag{5}$$

Multiplying (4) by (5) and cancel ling $[P', Q]$, we obtain

$$[M..P][P, Q]^I[Q, R]^I[R..Q'] \leq [M..P'][P', Q']. \tag{6}$$

Since $[P, Q]^I[Q, R]^I = [P, R]^I$, the left-hand side of the inequality obtained is itself the transmission coefficient of a path from M to Q', and this path is shorter than the shortest path from M to Q' (the transmission coefficient of the latter equals the right-hand side of (6) because P', Q' is M-radial). $\square$

6. AN OPTIMIZATION ALGORITHM

Initialization. First, we construct a distribution assigning each task to the machine that accomplishes it faster. If there are few machines accomplishing some task at a maximum speed, then the task is assigned to the machine with the least number.

Generally, the distribution constructed is this way has no transmission coefficients smaller than one; therefore, it has no reducing cycles. If the distribution obtained is uniform, then the problem is solved. Otherwise, we will *equalize* this distribution by means of admissible transmissions. The property of the absence of reducing cycles will be retained during the entire process and, when we equalize the loads of all machines, we will have an optimal distribution by virtue of Theorem 1.

Determination of shortest transmissions. For a given distribution without reducing cycles, we determine all shortest transmission paths by the Floyd–Warshall algorithm [1]; thus, we determine the values $[P..Q]$ for any P and Q.

If $\mathcal{M}$ is a set of processors and P is a processor, then $[\mathcal{M}..P]$ denotes the minimum of $[Q..P]$ over $Q \in \mathcal{M}$. The value $[P..\mathcal{M}]$ is defined similarly.

Construction of a sink filtration. First, we determine the set $\mathcal{M}_0$ of maximally loaded machines.

Then, we construct step by step an increasing sequence of sets of machines $\mathcal{M}_0 \subset \mathcal{M}_1 \subset \mathcal{M}_2 \subset \ldots \subset \mathcal{M}_n$ whose last term is the set of all machines: $\mathcal{M}_n = \mathcal{M}$.

The set $\mathcal{M}_{k+1} \backslash \mathcal{M}_k$ is defined as the set of machines P such that

$$[\mathcal{M}_0..P^-] + [P^-, P] = [\mathcal{M}_0..P] \tag{7}$$

for a certain $P^- \in M_k$, which is called the *source*[2] of the processor P.

Construction of a source filtration. If $\mathcal{M}_0$ contains processors that are sources for no processors in $\mathcal{M}_1$, then we construct a sequence of sets of processors $\mathcal{M}_0 \supset \mathcal{M}_{-1} \supset \ldots \supset \mathcal{M}_{-m} = \varnothing$ according to the following rule. The set $\mathcal{M}_0 \backslash \mathcal{M}_{-1}$ coincides with the set of sources for $\mathcal{M}_1$. The set $\mathcal{M}_{-i} \backslash \mathcal{M}_{-i-1}$ is defined as the set of those P for which there is a P^+ in $\mathcal{M}_{-i+1} \backslash \mathcal{M}_{-i}$ satisfying the relation

$$[P, P^+] + [P^+..\mathcal{M}_0 \backslash \mathcal{M}_{-1}] = [P..\mathcal{M}_0]. \tag{8}$$

We call P^+ the *sink*[3] of P.

Transmission scheme. We define the *sink index* of a processor P to be the number k for which $P \in \mathcal{M}_k \backslash \mathcal{M}_{k-1}$ and denote it by $\operatorname{ind} P$.

The source of a processor P with nonpositive index is defined as the set of processors for which P is a sink; it is denoted by P^- just as for processors of positive index.

Similarly, the sink of a processor P of positive index is defined as the set of processors for which P is a source and is denoted by P^+.

By $|P^\pm|$, we denote the number of elements in the set $P^\pm$. Note that $|P^-| = 1$ if $\operatorname{ind} P > 0$ and $|P^+| = 1$ if $\operatorname{ind} P < 1$.

Note also that if $Q \in P^-$ or $P \in Q^+$, then $\operatorname{ind} P = \operatorname{ind} Q + 1$

We write $P \to Q$ if $P \in Q^-$ or, equivalently, $Q \in P^+$, and write $J\colon P \to Q$ if $P \to Q$ and $J(P, Q) = J$.

A *transmission scheme* is an oriented marked graph whose vertices correspond to processors. The vertices corresponding to processors P and Q are joined with an edge if and only if $P \to Q$; here, the edge is oriented from source to sink and is marked by $J(P, Q)$.

[2] If there are few such processors, then the source is the processor with the least number.

[3] If there are few such P', then the sink is that with the largest number.

Sink coefficients. For each processor P, we define the *sink coefficient* $[P]$ as follows. For the *origins*, that is, processors of minimum sink index, this coefficient is equal to 1. Suppose that the sink coefficients are defined for all processors of index $<k$. For a processor P of nonpositive sink index k, we set

$$[P] = 1 + \sum_{Q \in P^-} [Q][Q, P]. \tag{9}$$

In this case, the sink coefficient determines the unloading rate of the processor in order that all processors of nonpositive index be unloaded at the same rate.

For any processor P of positive sink index k, we set

$$[P] = \frac{[P^-][P^-, P]}{|(P^-)^+|}. \tag{10}$$

In this case, the sink coefficient expresses the rate of loading the processor by tasks transmitted to it. The division by $|(P^-)^+|$ is performed because the equalizing transmission uniformly distributes the tasks outgoing from P^- between all processors from $(P^-)^+$.

The equalizing transmission. A task J is called *ingoing* for Q and *outgoing* for P if $J\colon P \to Q$. In the process of equalizing transmission, ingoing tasks arrive at each processor and outgoing tasks depart. The other tasks remain intact.

Let ε be a small number. We define a distribution $\delta(\varepsilon)$ obtained from a distribution δ by the ε-equalizing transmission as follows.

If $J\colon Q \to P$ and $\operatorname{ind} P \leq 0$, then

$$P^{\delta(\varepsilon)}(J) = P^{\delta}(J) + [Q][Q, P]\varepsilon. \tag{11}$$

If $J\colon Q \to P$ and $\operatorname{ind} P > 0$, then

$$P^{\delta(\varepsilon)}(J) = P^{\delta}(J) + \frac{[Q][Q, P]}{|Q^+|}\varepsilon. \tag{12}$$

If $\operatorname{ind} P \leq 0$ and J is a task outgoing from P, then its share on P after the ε-transmission is given by

$$P^{\delta(\varepsilon)}(J) = P^{\delta}(J) - [P]\varepsilon. \tag{13}$$

On a processor P of positive index, the share of a task J outgoing from P to k processors is expressed as

$$P^{\delta(\varepsilon)}(J) = P^{\delta}(J) - \frac{k}{|P^+|}[P]\varepsilon. \tag{14}$$

The sink coefficients are chosen in such a way that the equalizing ε-transmission reduces the loads of all processors with nonpositive index by ε, does not affect the loads of the processors of positive index with nonempty sinks, and increases the load of each processor P with empty sink by $[P]\varepsilon$.

Definition of transmission capacity. Now, let us define the value of the maximum possible equalizing ε-transmission, that is, the transmission capacity of a distribution.

We start with defining the *transmission capacity* $\varepsilon(P, I)$ of a processor P with respect to an outgoing task I. Let $k(I)$ denote the number of processors to which I departs from P. Then,

$$\varepsilon(P, I) = \frac{k(I)}{|J(P^+)|[P]}. \tag{15}$$

We define the transmission capacity of the *transmission scheme* constructed as the minimum of $\varepsilon(P, I)$, where P and I range over all pairs for which I is a task outgoing from P.

The transmission capacity expresses the greatest lower bound of those ε for which the equalizing ε-transmissions do not affect distribution signatures. The equalizing ε-transmission with ε equal to the transmission capacity of the scheme changes the distribution signature. Namely, it removes the tasks I for which $\varepsilon(P, I)$ is minimal from the processor P, which results in the disappearance of the edges outgoing from P and marked by the tasks I.

An equalizing algorithm. For a given distribution without reducing cycles (in particular, for that obtained after initialization), we determine a transmission scheme.

The determination of a transmission scheme includes calculating shortest transmissions and constructing sink and source filtrations.

Then, we perform the maximal equalizing transmission; for this purpose, we first calculate the sink coefficients and the transmission capacities.

If the maximal equalizing transmission results in a load of some processors with positive sink index heavier than that of processors with nonpositive index (i.e., if an *excess* occurs), then we perform the *correction* procedure, i.e., decrease the transmission value so that the load of one sink processor is equal precisely to the load of the origins and the load of all other sink processors is no larger. If $e(P)$ is the value by which the load of the processor P exceeds the load of the origins, then, as is easy to verify, a decrease in transmission by $\frac{e(P)}{1+[P]}$ equates the load of P and that of the origins. Thus, we must perform a transmission which is less than the maximal transmission by $\min_P \frac{e(P)}{1+[P]}$. In this case, we recalculate the transmission scheme on the basis of the increased set of machines with maximum load.

If none of the processors with nonpositive index loses a sink and no excess occurs after the maximal transmission, then we perform transmission according to the residual scheme, i.e., the scheme from which the edges $J\colon P \to Q$ such that the maximal transmission entirely transfers J to Q are removed. In this case, we only need to recalculate the transmission capacities. We continue to perform equalizing transmissions by residual schemes until either the number of maximally loaded machines increases or one of the sources breaks. In the latter case, we must recalculate the entire transmission scheme.

7. CONCLUDING REMARKS

The minimum load time over all processors is called the *minimum load* of a distribution. Note that equalizing transmissions can only increase the minimum load of a distribution. The equalizing algorithm constructs an optimal distribution from an arbitrary distribution without reducing cycles. Thus, the following theorem is valid.

Theorem 2. *For an arbitrary multiprocessor distribution without reducing cycles, the minimum load does not exceed the load of the optimal distribution.*

This theorem helps to construct distributions close to optimal. We say that the *load margin* of a distribution is the difference between the load times of the maximally and minimally loaded machines. If the distribution has no reducing cycles, then, by Theorem 2, its load is heavier than the optimal load by a value no greater than the load margin. Therefore, if we need to construct a distribution ε-close to an optimal distribution, we can stop the equalizing algorithm at the moment when the load margin of the current distribution is less than ε.

On the complexity of the algorithm. The initialization requires time $O(nm)$. The outer cycle of the algorithm has length m. Each step of the outer cycle increases the number of maximally loaded machines by one. At each step of the outer cycle, we must construct a transmission scheme at least once. The search for the shortest paths by the Floyd–Warshall algorithm takes time $O(m^3)$; the same time is required for constructing the equalizing transmission scheme. It is only unclear how many breaks of sources can occur under an equalizing transmission for a fixed $\mathcal{M}_0$. Every such break is caused by a transfer of some task from one processor of nonpositive index to another. We conclude the paper with the following question.

Is the algorithm suggested above polynomial?

ACKNOWLEDGMENTS

This work was partly supported by the Russian Foundation for Basic Research, project no. 02-01-00014.

REFERENCES

1. Papadimitriou, Ch. and Steiglitz, K., *Combinatorial Optimization: Algorithms and Complexity*, Englewood Cliffs: Prentice-Hall, 1982. Translated under the title *Kombinatornaya optimizatsiya. Algoritmy i slozhnost'*, Moscow: Mir, 1985.
2. Potts, C.N., Analysis of a Linear Programming Heuristic for Scheduling Unrelated Parallel Machines, *Discrete Appl. Math.*, 1985, vol. 10, pp. 155–164.
3. Shchepin, E.V. and Vakhania, N., Task Distributions on Multiprocessor Systems, *Lect. Notes Comput. Sci.*, 2000, vol. 187, pp. 112–125.
4. Lenstra, J.K., Shmoys, D.B., and Tardos, E., Approximation Algorithms for Scheduling Unrelated Parallel Machines, *Math. Program. A*, 1990, vol. 46, pp. 259–271.
5. Plotkin, S.A., Shmoys, D.B., and Tardos, E., Fast Approximation Algorithms for Fractional Packing and Covering Problems, *Math. Oper. Res.*, 1995, vol. 20, no. 2, pp. 257–301.

Translated by O. Sipacheva