MATHEMATICAL MODEL OF MICROPROGRAM FORMATION BASED ON OPTIMIZED STRUCTURE OF THE CONTROL AUTOMATON

Yu. A. Buzunov, A. L. Kozachkov,
and S. P. Popovich

UDC 519.71.1

Synthesis of microprogramming automata is usually based on classical finite state models [1]. The real microprogramming automata (MPA), which implement dozens of different algorithms (microprograms), are complex and largely undefined [2], making it difficult to use canonical methods of finite state synthesis for their design. On the other hand, the analysis of the structures of real MPA in discrete information converters (DIC) of various purposes (and especially computers) suggest that it is convenient to describe the structure of MPA as a set of interconnected and functionally defined components. It is natural to formulate an MPA structure as a combination of two connected automata: one automaton which receives signals of algorithms (microprograms) P and logical conditions X to form signals of microcycles (micrommands) Y; and another automaton, forming from the signals P and Y the signals A, which initiate in the operating automaton (OA) of DIC the execution of microoperations. Figure 1 illustrates such an MPA structure; the first automaton is identified as microcycle former (MCF) and the second as control signal distributor (CSD). Synthesizing an MCF is relatively easy and its structure is well known [2]. Of more interest is the synthesis of CSD, which is the more complex and irregular component of MPA. In [3] it was shown that the CSD structure can be improved (simplified and made more regular) if all MPA of the microprogram are brought together into a generalizing table organized in a certain way. The number of lines in the generalizing table is equal to the number of microprograms N and the number of columns is equal to the number of microcycles m of the longest microprogram.* Table 1 is an illustration of such a table. The microoperations indicated in parentheses will be defined below. We will describe briefly the principles of generalizing table organization presented in [3].

The CSD structure is defined by the system of output functions

$$U_k = \bigvee P_i Y_j \quad k = \overline{1, s}, \quad i = \overline{1, N}, \quad j = \overline{1, m}, \tag{1}$$

where P_i, Y_j are the signals of the algorithm and the microcycle, respectively, which are coordinates of the box in the table where the microoperation A_k is posted.

The most elementary CSD structure is described by a system of minimized output functions. The functions are minimized by combining disjunctive terms with like arguments and applying the relations

$$\bigvee_{i \in R} P_i = 1, \tag{2}$$

*The design and organization of the generalizing table are considered as applicable to linear microprograms. We will demonstrate later that MPA synthesis could also be employed for nonlinear microprograms.

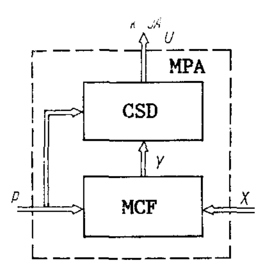

Fig. 1. MPA structure.

Translated from Kibernetika, No. 6, pp. 67-73, 79, November-December, 1986. Original article submitted April 19, 1985.

 0011-4235/86/2206-0764$12.50

TABLE 1. Generalizing Table with Nonfixed Microoperations

Algorithm signals	Microcycle signals				
	Y_1	Y_2	Y_3	Y_4	Y_5
P_1	$(A_1)\ A_2$	$(A_1)\ A_3$	$A_1\ A_3$	A_2	
P_2	$A_1\ A_3$	$A_1\ A_2$	$(A_1)\ A_3$	$(A_1)\ A_3$	
P_3	$A_1\ A_3$	A_2	$(A_1)\ A_2\ A_3$	$(A_1)\ A_3$	
P_4	$A_2\ A_3$	$(A_1)\ A_2$	$(A_1)\ A_2\ A_3$	$(A_1)\ A_3$	
P_5	$(A_1)\ A_2$	$(A_1)\ A_2$	$A_1\ A_2$	$(A_2)\ A_3$	$A_1(A_2)$

TABLE 2. Optimized Generalizing Table

Algorithm signals	Microcycle signals				
	Y_1	Y_2	Y_3	Y_4	Y_5
P_1	$A_1\ A_2$	A_3	$A_1\ A_3$	A_2	$A_1\ A_2$
P_2	$A_1\ A_3$	$A_1\ A_2$	$A_1\ A_3$	A_3	$A_1\ A_2$
P_3	$A_1\ A_3$	A_2	$A_1\ A_2\ A_3$	A_3	$A_1\ A_2$
P_4	$A_2\ A_3$	A_2	$A_1\ A_2\ A_3$	A_3	$A_1\ A_2$
P_5	$A_1\ A_2$	A_2	$A_1\ A_2$	A_3	$A_1\ A_2$

$$\bigvee_{i \in R_1} P_i = \bigwedge_{i \in R_2} \overline{P}_i, \qquad (3)$$

where R is the complete set of the signals of algorithms, and $R_1 \cup R_2 = R$, $R_1 \cap R_2 = \varnothing$.

Expression (2) corresponds to the occurrence of a microoperation in all boxes of the column labeled by the signal Y_j. If condition (2) is not fulfilled, the signal of the microcycle Y_j is gated by the permitting signals of the algorithms $\overline{P}_i$, which use this operation in the microcycle Y_j, or by the prohibiting signals of the algorithms P_i, which do not use this microoperation in the microcycle Y_j. The choice of the left-hand or right-hand part of expression (3) in the notation of a particular output function is determined by the usual minimization criterion of Boolean functions.

The system of CSD output functions corresponding to the illustration of Table 1 appears as

$$\begin{cases} U_1 = P_2 \cdot Y_1 \vee P_3 \cdot Y_1 \vee P_2 \cdot Y_2 \vee P_1 \cdot Y_3 \vee P_5 \cdot Y_3 \vee P_5 \cdot Y_5 \\ U_2 = \overline{P}_2 \cdot \overline{P}_3 \cdot Y_1 \vee \overline{P}_1 \cdot Y_2 \vee \overline{P}_1 \cdot \overline{P}_2 \cdot Y_3 \vee P_1 \cdot Y_4 \\ U_3 = \overline{P}_1 \cdot \overline{P}_5 \cdot Y_1 \vee P_1 \cdot Y_2 \vee \overline{P}_5 \cdot Y_3 \vee \overline{P}_1 \cdot Y_4. \end{cases}$$

The system of output functions can be simplified by making use of the natural structural redundancy of operating automata [3]. It is possible to introduce into the microcycles of microprograms so-called redundant microoperations, i.e., operations which are not necessary, but do not distort the information conversion algorithm. Such redundant microoperations can be entered, in particular, into the empty boxes of the table. Since MCF does not form the signal Y_j for the i-th microprogram if the element (i, j) of the table is empty, any microoperation is redundant for the i-th microprogram at the j-th microcycle. We assume that the microprograms are arranged starting from the first column, and contain no empty microcycles. Then all empty boxes in the i-th line are to the right of the column representing the last microcycle of the i-th microprogram.

One important feature of the microprograms is that they can contain so-called nonfixed microoperations, i.e., those for which several microcycles are specified with the requirement that in only one of the microcycles the microoperation in question must be executed. These sets of microcycles will be called the nonfixation interval of the microoperation; in the table, they are indicated by microoperations given in parentheses. For each microoperation within a microprogram, several nonfixation intervals are possible. Table 1 represents a case when none of the microprograms has more than one nonfixation interval with respect to A_1, A_2. Recording the microoperations in parentheses in the boxes of the table should be done in such a way that relations (2) and (3) be satisfied, yielding the most elementary CSD structure. Table 2 is an illustration of a table with redundancy and an optimal placement of nonfixed microoperations. The corresponding output functions system appears as

$$\begin{cases} U_1 = \overline{P}_4 \cdot Y_1 \vee P_2 \cdot Y_2 \vee Y_3 \vee Y_5 \\ U_2 = \overline{P}_2 \cdot \overline{P}_3 \cdot Y_1 \vee \overline{P}_1 \cdot Y_2 \vee \overline{P}_1 \cdot \overline{P}_2 \cdot Y_3 \vee P_1 \cdot Y_4 \vee Y_5 \\ U_3 = \overline{P}_1 \cdot \overline{P}_5 \cdot Y_1 \vee P_1 \cdot Y_2 \vee \overline{P}_5 \cdot Y_3 \vee \overline{P}_1 \cdot Y_4. \end{cases}$$

The recording of nonfixed microoperations simplifies the system of CSD output functions.

Redundant microoperations can generally be entered only after nonfixed microoperations are fixed. There are, however, some redundant microoperations that can be entered regardless of the presence of nonfixed microoperations in the table; for example, these are redundant microoperations of the microprograms which have no nonfixed microoperations, and the above-mentioned microoperations of empty boxes. In view of this, we will enter redundant microoperations in two steps: before and after fixing the nonfixed microoperations; by way of convention, we will identify them as redundant microoperations of the first and the second group, respectively.

The table will be optimized in the following sequence:

1) optimize the table while simultaneously placing the nonfixed redundant microoperations of the first group and (if necessary) adding such microoperations;

2) define the second group of redundant microoperations; and

3) add (if necessary) redundant microoperations of the second group.

These principles of table organization, if the table is large (real MPA implement tens and hundreds of operations consisting of tens and hundreds of microcycles), cannot be realized manually. In order to develop a formal algorithm for table organization, we assume that a line cannot have more than one nonfixation interval for each microoperation. We will show subsequently that this restriction is not essential for these algorithms. The generalizing table will be designated as the matrix A and the microoperations as elements of the matrix A.

Optimizing the structure of a microprogramming automaton, defined with the output functions U_k, $k = \overline{1, s}$ therefore becomes equal to minimizing the number of conjunctions in the notation of the output functions. The number of conjunctions S in the notation of the output functions equals exactly the number of occurrences A_k, $k = \overline{1, s}$ in the matrix A:

$$S = \sum_{j=1}^{m} S_j, \tag{4}$$

where

$$S_j = \sum_{k=1}^{s} q^k(j), \tag{5}$$

$q^k(j)$ is the number of elements A_k in the j-th column of the matrix A. By virtue of (3), expression (5) can be rewritten as

$$S_j = \sum_{k=1}^{s} \min\{q^k(j);\ N - q^k(j)\}. \tag{6}$$

Here $q^k(j)$ corresponds to the gating of the microcycle signal to the permitting signals of the algorithms P_i, and $N - q^k(j)$ by prohibiting signals of the algorithms $\overline{P}_i$.

Substituting (6) into (4), we obtain

$$S = \sum_{j=1}^{m} \sum_{k=1}^{s} \min\{q^k(j);\ N - q^k(j)\}. \tag{7}$$

Suppose that after placing nonfixed and adding redundant elements A_k, the j-th column of the matrix A contains $p^k(j)$ elements A_k. It is obvious that

$$q^k(j) \leqslant p^k(j) \leqslant q^k(j) + d^k(j) + L^k(j), \tag{8}$$

where $d^k(j)$, $L^k(j)$ is the number of places where the nonfixed and redundant element can be posted, respectively. Now the solution of the problem $\tilde{S}$ can be interpreted as

$$\tilde{S} = \min_{p^k(j)} \sum_{j=1}^{m} \sum_{k=1}^{s} \min\{p^k(j);\ N - p^k(j)\}, \tag{9}$$

where $p^k(j)$ satisfies (8).

It is obvious that the sequence of the elements A_k, $k = \overline{1, s}$, which determines the addition of redundant elements to the matrix A and the placement of nonfixed elements, does not affect optimization problem; it means that s problems have to be solved concerning the addition of redundant and optimal placement of nonfixed elements:

$$\tilde{S} = \sum_{k=1}^{s} \left(\min_{p^k(j)} \sum_{j=1}^{m} \min\{p^k(j);\ N - p^k(j)\}\right) = \sum_{k=1}^{s} \tilde{S}_k, \tag{10}$$

where

$$\tilde{S}_k = \min_{p^k(j)} \sum_{j=1}^{m} \min\{p^k(j);\ N - p^k(j)\} \tag{11}$$

(The index k will be omitted in the subsequent derivations.)

Optimization of the matrix A using only redundant elements is not difficult. It is obtained by testing the condition for each $j = \overline{1, m}$:

$$\min\{q(j);\ N - q(j)\} > \min\{q(j) + L(j);\ N - q(j) - L(j)\}. \tag{12}$$

If the condition is fulfilled, we post in the j-th column L(j) redundant elements.

We can now develop algorithms for optimization of the matrix A using only nonfixed elements, and then construct the main optimization algorithm.

Placement of Nonfixed Elements

The following notations will be used:

1) $b(i) \geq 0$, $i = \overline{1, N}$ is the number of places where nonfixed elements can be posted in the i-th row of the matrix A.

2) $D > 0$ is the number of nonfixed elements which must be posted in the matrix A

$$D = \sum_{i=1}^{N} \sigma_i, \text{ where } \sigma_i = \begin{cases} 1, & \text{if } b(i) \neq 0, \\ 0, & \text{if } b(i) = 0; \end{cases} \tag{13}$$

3) $x_{j_1 j_2} \geq 0$; $j_1, j_2 = \overline{1, m}$, $j_1 \neq j_2$ is the number of rows in which the places of the matrix A – (i, j_1), (i, j_2) are admissible for posting a nonfixed element ($x_{j_1 j_2} \equiv x_{j_2 j_1}$)

$$\sum_{\substack{j=1 \\ j \neq j_1}}^{m} x_{j_1 j} = d(j_1); \tag{14}$$

4) $S_f(j_1, \ldots, j_f)$, $f = \overline{1, m}$ is the sum corresponding to the sequence $j_1, \ldots, j_f$, defined recursively:

$$\begin{cases} S_0 = 0, \\ S_f(j_1, \ldots, j_f) = S_{f-1}(j_1, \ldots, j_{f-1}) + \min\{Q_f; \\ N - Q_f - L(f)\}, \end{cases} \tag{15}$$

where $Q_f = q(j_f) + d(j_f) - \sum_{\nu=1}^{f-1} x_{j_\nu j_f}$, [here $L(j) = 0$; $j = \overline{1, m}$].

We can now rewrite (7) as

$$S = S_m(j_1, \ldots, j_m). \tag{16}$$

Definitions. 1. We will say that the columns j_1, j_2 are connected if $x_{j_1 j_2} \neq 0$. Suppose that the places of the matrix A – (i_1, j_1), (i_1, j_2) are permissible for posting a nonfixed elements.

2. The following transformations will be called permissible in the matrix A.

Transformation 1. Post an element in the place (i_1, j_1). In this case,

$$\begin{aligned} &1)\ q(j_1) := q(j_1) + 1, \\ &2)\ d(j_1) := d(j_1) - 1, \\ &3)\ b(i_1) := 0, \\ &4)\ d(j_2) := d(j_2) - 1, \end{aligned} \tag{17}$$

i.e., $\forall j_2 \neq j_1$; the place (i_1, j_2) is not considered permissible for posting a nonfixed element.

Transformation 2. The place (i_1, j_1) is not considered permissible for posting a nonfixed element. Here

$$1)\ b(i_1) := b(i_1) - 1, \qquad 2)\ d(j_1) := d(j_1) - 1. \tag{18}$$

We will consider the optimal placement of nonfixed elements in the matrix A. Suppose that $\exists z \geq 1 : q(j_\nu) > N/2,\ \nu = \overline{1, z}$.

3. The set of columns $Z = \{j_1, \ldots, j_z\}$ will be called an **optimal set**. Suppose that an optimal arrangement has been obtained after nonfixed elements were posted first in column j_1, then in $j_2, \ldots, j_m$.

4. The sequence of columns $j_1, \ldots, j_z, \ldots, j_m$ will be called an **optimal sequence**.

THEOREM 1. For any $q(j)$ and $d(j)$, $j = \overline{1, m}$, there exists an optimal sequence, and $Z = \{j_1, \ldots, j_z\}$ is an optimal set.

In order to prove this theorem, we will consider the following lemmas.

LEMMA 1. When $d(j)$ of nonfixed elements are posted in column j, $j = \overline{1, m}$ at $q(j) \geq N/2$, we obtain

$$S := S - d(j). \tag{19}$$

Proof. Posting a nonfixed element in the column j, we obtain

$$p(j) = q(j) + 1, \tag{20}$$

then $p(j) > N/2$. This means that $\min\{p(j); N - p(j)\} = N - p(j)$. Then (6), by virtue of (20), can be written as $Sj := Sj - 1$. From (4) we have $S := S - 1$. By induction with respect to $d(j)$, we obtain (19). ■

LEMMA 2. When $d(j)$ nonfixed elements are placed in column j, $j = \overline{1, m}$, $q(j) + d(j) \leq N/2$ we obtain

$$S := S + d(j). \tag{21}$$

Proof. Posting a nonfixed element in column j, we obtain

$$p(j) = q(j) + 1. \tag{20'}$$

Then $p(j) \leq q(j) + d(j) \leq N/2$. Therefore, $\min\{p(j); N - p(j)\} = p(j)$. Now (6), taking into account (20'), can be written as $Sj := Sj + 1$. And for (4) $S := S + 1$.

By induction with respect to $d(j)$, we obtain (21). ■

Proof of Theorem 1. Suppose that

$$I_1 = \left\{ j : q(j) > \frac{N}{2} \right\},$$

$$I_2 = \left\{ j : q(j) + d(j) \leqslant \frac{N}{2} \right\},$$

$$I_3 = \left\{ j : q(j) + d(j) > \frac{N}{2} \right\}.$$

1. $\forall j = \overline{1, m}\ j \in I_1$, i.e., $z = m$. Then, by virtue of Lemma 1,

$$S = \sum_{j \in I_1} (N - q(j)) - D = m \cdot N - \sum_{j=1}^{m} q(j) - D = \text{const} = \tilde{S}, \tag{22}$$

and we can fill the matrix A with nonfixed elements in an arbitrary fashion.

2. $\forall j = \overline{1, m}\ j \in I_2$, i.e., $z = 0$. Then, by virtue of Lemma 2,

$$S = \sum_{j \in I_2} q(j) + D = \sum_{j=1}^{m} q(j) + D = \text{const} = \tilde{S}, \tag{23}$$

and we can fill the matrix A with nonfixed elements in an arbitrary fashion.

3. $\exists j \in I_2 \wedge \exists j \in I_3 (\exists j \in I_2 \wedge \exists j \in I_1)$, i.e., $0 \leq z < m$ $(1 \leq z < m)$.

Suppose that we have obtained an optimal arrangement of nonfixed elements when placing in columns j, $j = \overline{1, m}$, the column $d'(j) \leq d(j)$ of nonfixed elements. Here $\sum_{j=1}^{m} d'(j) = D$.

We will show that $\exists j_1$

$$q(j_1) + d'(j_1) > \frac{N}{2}. \tag{24}$$

Suppose that $\forall j = \overline{1, m}$ $q(j) + d'(j) \leq N/2$. Then

$$\tilde{S} = \sum_{j=1}^{m} (q(j) + d'(j)) = \sum_{j=1}^{m} q(j) + D. \tag{25}$$

We will consider the arrangement of nonfixed elements resulting from posting $d(j_1)$ elements by transformation 1 in a column $j_1 \in I_3 \wedge d'(j_1) \neq 0$, while the remaining m − 1 columns are filled, as with an optimal arrangement [taking into account (17)]. Then

$$S = \sum_{\substack{j=1 \\ j=j_1}}^{m} (q(j) + d'(j) - x_{j_1 j}) + N - q(j_1) - d(j_1) = \sum_{j=1}^{m} (q(j) + d'(j)) - d'(j_1) - q(j_1) -$$

$$\sum_{\substack{j=1 \\ j \neq j_1}}^{m} x_{j_1 j} + N - q(j_1) - d(j_1) = \sum_{j=1}^{m} q(j) + D + N - 2q(j_1) - 2d(j_1) - d'(j_1). \tag{26}$$

Since $q(j_1) + d(j_1) \geqslant \frac{N}{2} \wedge d'(j_1) > 0$, therefore

$$N - 2q(j_1) - 2d(j_1) - d'(j_1) < 0.$$

From (25)-(27) $\Rightarrow$ $S - \tilde{S} < 0$, which contradicts the choice of $\tilde{S}$. This proves (24) [if $\not\exists j \in I_2 \wedge \not\exists j \in I_1$, then (24) is evident], i.e.,

It can readily be shown that the arrangement of nonfixed elements when a column $j_1 \in Z$ contains not $q(j_1) + d'(j_1) > N/2$, but $q(j_1) + d(j_1) > N/2$ elements, will also be optimal.

Let us apply to the elements of the column j_1 transformation 1. By repeating the above reasoning for the resulting matrix, we can find the column j_2, etc. ■

THEOREM 2. If the sequence $j_1, \ldots, j_{\omega_1-1}, j_{\omega_1}, \ldots, j_z, \ldots, j_{\omega_2-1}, j_{\omega_2}, \ldots, j_m$, $\omega_1 = \overline{1, z}$, $\omega_2 = \overline{z+1, m}$, is optimal, then the sequences 1) $j_1, \ldots, j_{\omega_1}, j_{\omega_1-1}, \ldots, j_z, \ldots, j_m$ and 2) $j_1, \ldots, j_z, \ldots, j_{\omega_2}, j_{\omega_2-1}, \ldots, j_m$ are also optimal.

Proof. Suppose $j_1, \ldots, j_{\omega_1-1}, j_{\omega_1}, \ldots, j_z, \ldots, j_m$ is an optimal sequence

$$\tilde{S} = \sum_{\nu=1}^{z} \left(N - q(j_\nu) - \left(d(j_\nu) - \sum_{i=1}^{\nu-1} x_{j_i j_\nu}\right)\right) + \sum_{\nu=z+1}^{m} \left(q(j_\nu) + \left(d(j_\nu) - \sum_{i=1}^{\nu-1} x_{j_i j_\nu}\right)\right).$$

1. For the sequence $j_1, \ldots, j_{\omega_1}, j_{\omega_1-1}, \ldots, j_z, \ldots, j_m$, we have

$$S = \sum_{\substack{\nu=1 \\ \nu \neq \omega_1-1 \\ \nu \neq \omega_1}}^{z} \left(N - q(j_\nu) - \left(d(j_\nu) - \sum_{i=1}^{\nu-1} x_{j_i j_\nu}\right)\right) + \sum_{\nu=z+1}^{m} \left(q(j_\nu) + \left(d(j_\nu) - \sum_{i=1}^{\nu-1} x_{j_i j_\nu}\right)\right) +$$

$$+ N - q(j_{\omega_1}) - \left(d(j_{\omega_1}) - \sum_{i=1}^{\omega_1-2} x_{j_i j_{\omega_1}}\right) + \min\{B; N-B\},$$

where

$$B = q(j_{\omega_1-1}) + d(j_{\omega_1-1}) - \sum_{i=1}^{\omega_1-2} x_{j_i j_{\omega_1-1}} - x_{j_{\omega_1} j_{\omega_1-1}}.$$

If $B \geq N/2$, then taking into account $x_{j_{\omega_1} j_{\omega_1-1}} = x_{j_{\omega_1-1} j_{\omega_1}}$, we obtain $S = \tilde{S}$. If $B < N/2$, then by definition of the function min{*; *}, we have $S < \tilde{S}$, which is a contradiction. Hence $S = \tilde{S}$.

2. For proving that a sequence is optimal, it is sufficient to note that after nonfixed elements are posted in the columns $j_1, \ldots, j_z$, condition of item 2 of Theorem 1 will be fulfilled for the other columns. Therefore, the nonfixed elements can be posted in the columns $j_{z+1}, \ldots, j_m$ in an arbitrary order. ■

COROLLARY 1. There exist $z!(m-z)!$ optimal sequences.

COROLLARY 2. In order to obtain one of the optimal arrangements of nonfixed elements, it is sufficient to find an optimal set $Z = \{j_1, \ldots, j_z\}$.

Proof of Corollary 2. If we find $Z = \{j_1, \ldots, j_z\}$, then by virtue of Theorem 2, any sequence $j_1, \ldots, j_z, \ldots, j_m$, where $j_v \in Z$, $v = \overline{1, z}$; $j_v \notin Z$, $v = z+1, m$, is optimal. ∎

On the basis of the foregoing results, an algorithm can be constructed.

Algorithm 1 (Placement of Nonfixed Elements)

Step 1. As long as $\exists i, j: q(j) \geq N/2$ and the place (i, j) is permissible for posting a nonfixed element, apply transformation 1 to place (i, j).

Step 2. As long as $\exists i, j: q(j) + d(j) \leq N/2 \wedge b(i) \geq 2$ and the place (i, j) is permissible for posting a nonfixed element, apply transformation 2 to place (i, j).

Step 3. If $b(i) = 1$, find a column j such that the place (i, j) is permissible for posting a nonfixed element, and apply transformation 1 to the place (i, j).

Step 4. Apply step 4 to all columns for which $d(j) \neq 0$. Suppose that their number is m_1.

Step 4.f. $f = 1, 2 \ldots$

1) form $C^f_{m_1}$ different sets consisting of f columns $Z^v_f = \{j_{v_1}, \ldots, j_{v_f}\}$, $v = \overline{1, C^f_{m_1}}$;

2) apply to all $j \in Z^v_f$ transformation 1;

3) if

$$\forall j \in Z^v_f \quad q(j) > \frac{N}{2}, \tag{28}$$

$$\forall j \notin Z^v_f \quad q(j) + d(j) \leqslant \frac{N}{2}, \tag{29}$$

then admit Z_f to step 5.

Step 5. For all sets admitted to step 5, calculate $S = S_{m_1}(j_{v_1}, \ldots, j_{v_f}, \ldots, j_{m_1})$, $f = 1, 2 \ldots$; $v = \overline{1, C^f_{m_1}}$; find $S(j_1, \ldots, j_z, \ldots, j_{m_1}) = \min\limits_{f,v} S_{m_1}(j_{v_1}, \ldots, j_{v_f}, \ldots, j_{m_1})$.

Step 6. Apply transformation 1 successively to all elements of column j_1, then $j_2, \ldots, j_z, \ldots, j_{m_1}$.

Substantiation of Algorithm 1

Step 1. Follows directly from item 1 of Theorem 1.

Step 2. Follows directly from item 2 of Theorem 1.

Step 3. Follows from definition of a nonfixed element.

Now for all columns the following condition holds

$$1)\ d(j) = 0 \ \text{ or } \ 2)\ d(j) \neq 0 \wedge q(j) < \frac{N}{2} \ \wedge q(j) + d(j) > \frac{N}{2}. \tag{30}$$

Step 4. Suppose that $j_1, \ldots, j_z, \ldots, j_{m_1}$ is an optimal sequence. After applying to it successively transformation 1, (28) and (29) wil be fulfilled. Since condition (28) is the necessary condition of optimality of the set at step 4.z (i.e., at $f = z$), the step $\{j_1, \ldots, j_1\}$ will be admitted to step 5.

Step 5. Since $j_1, \ldots, j_z, \ldots, j_{m_1}$ is an optimal sequence, therefore any sequence $j_{i_1}, \ldots, j_{i_z}, \ldots, j_{m_1}$ (its first z terms are formed of the set Z) will also be optimal, according to Theorem 2.

Then $S_{m_1}(j_{i_1}, \ldots, j_{i_z}, \ldots, j_{m_1}) = S_{m_1}(j_1, \ldots, j_{m_1})$, i.e., sequence $j_{i_1}, \ldots, j_{i_z}, \ldots, j_{m_1}$, obtained at step 5 is also optimal, and at step 6 we can apply to it transformation 1, so as to obtain one of the solutions of problem 2. ∎

Examples of optimal sequences relative to nonfixed element A_1 are the following column sequences – 1, 3, 2, 4, 5 and 3, 1, 4, 2, 5 of the matrix A defined by Table 1. The set {1, 3} is optimal.

This algorithm is, obviously, exponential.

Instead of algorithm 1, in certain instances, direct scanning of the fillings of the matrix A with nonfixed elements can be preferable. Indeed, suppose that after steps 1-3 the row i contains b(i) places permissible for posting a nonfixed element.

There are $\prod_{i=1}^{N} b(i)$ ways of filling the matrix A with nonfixed elements. Therefore, if after steps 1-3 we have $\prod_{i=1}^{N} b(i) < 2^{m_1}$, then instead of steps 4-6 we can compare $\prod_{i=1}^{N} b(i)$ alternatives of filling of the matrix A with nonfixed elements.

The dimension of the problem can be reduced greatly by separating groups of connected columns after steps 1-3. Then 4-6 steps will be applied to each group. Determining the groups of connected columns is not difficult; it is similar to finding the connectivity components in a nonoriented graph (where columns are nodes, and edges link the nodes corresponding to connected columns).

Another highly effective way of reducing the amount of calculations is to stop at step 4 if all f-element sets have been admitted to step 5, or if all sets have this property: after transformation 1 is applied to elements of the sets, each set contains columns for which $q(j) \le N/2$.

Despite the potential improvements of algorithm 1 in real large-dimensionality problems, a polynomial approximation of the algorithm is more efficient (algorithm 2). For constructing algorithm 2, it is necessary to alter step 4. We introduce the notation α_j^f, $f = \overline{1, m_1}$, $j = \overline{1, m_1 - f + 1}$, or changing the sum S at the step 4.f of the approximate algorithm if nonfixed elements are posted in the column j. The approximate solution will be found when $q(j) < N/2 \wedge q(j) + d(j) > N/2$ in the form $\tilde{S}_{appr.} = S + \sum_{f=1}^{m_1} \min\{\alpha_j^f\}$, $j = \overline{1, m_1 - f + 1}$.

<u>THEOREM 3.</u> When d(j) nonfixed elements are posted in the column j, subject to the conditions

$$\begin{cases} q(j) < \dfrac{N}{2}, \\ q(j) + d(j) > \dfrac{N}{2} \end{cases} \tag{31}$$

we have

$$S_1 = S + N - 2q(j) - d(j). \tag{32}$$

<u>Proof.</u> Let S' be sum (6), corresponding to the case when the elements of the column j are not considered, i.e., $S = S' + \min\{q(j), N - q(j)\}$. By virtue of (31),

$$S = S' + q(j). \tag{33}$$

Posting in the j-th column d(j) nonfixed elements, we obtain $S_1 = S' + \min\{q(j) + d(j), N - (q(j) + d(j))\}$. Taking into account (31),

$$S_1 = S' + N - q(j) - d(j). \tag{34}$$

From (33), (34), we obtain (32). ■

On the basis of these results, we can construct algorithm 2.

<u>Algorithm 2</u>

<u>Step 1.</u> Same as step 1 of algorithm 1.

<u>Step 2.</u> Same as step 2 of algorithm 1.

<u>Step 3.</u> Same as step 3 of algorithm 1.

<u>Step 4.</u> To be applied to all columns for which $d(j) \ne 0$. Suppose that there are m_1 such columns.

Step 4.f. $f = \overline{1, m_1}$:

1) find the column u:

$$u = \arg\max \{2q(j) + d(j)\}_{j=\overline{1, m_1 - f + 1}}; \qquad (35)$$

2) apply transformation 1 to the elements of the column u;

3) if $\exists j: q(j) + d(j) \le N/2$, go to step.

Substantiation of Algorithm 2

Step 4. Post at step 4.f d(j) nonfixed elements in a certain column j, $j = \overline{1, f - m_1 + 1}$; by virtue of Theorem 3 we have $S_1 = S + N - 2q(j) - d(j)$. Therefore $\alpha_j^f = N - 2q(j) - d(j)$. Then

$$\min \{\alpha_j^f\}_{j=\overline{1, f-m_1+1}} = N - \max \{2q(j) + d(j)\}_{j=\overline{1; f-m_1-1}}.$$

Item 3 in step 3 is introduced, because after transformation 1 is applied to column u, a column j can be formed, such that $q(j) + d(j) \le N/2$ (i.e., a column for which the condition of Theorem 3 does not hold). We apply step 2 to such columns. ■

Note that the above-mentioned constraint of the uniqueness of nonfixation interval was used only in defining b(i); in order to generalize these results, we must modify definition of b(i) and transformations 1, 2, as well as the steps in the algorithms associated with these definitions.

Algorithm of Combined Optimization

The tasks of entering redundant elements and posting nonfixed elements are interconnected. This can be seen from Table 1. If a place for posting a nonfixed element A_2 is sought disregarding the possibility of supplementing the matrix with redundant elements A_2 (in this case, redundant elements of empty boxes), then boxes (5, 4) and (5, 5) will not differ [since $q^2(4) + d^2(4) = 2 < N/2$ and $q^2(5) + d^2(5) = 1 < N/2$].

If a nonfixed element A_2 is posted in the fourth column, then no redundant elements of A_2 will be posted in the fifth column in this matrix. Considering the interdependences of redundant and nonfixed elements of A_2, we obtain the optimized matrix given in Table 2.

The interdependency of nonfixed and redundant elements of the first group is taken into account in the following approximate algorithm of joint optimization.

Algorithm 3

Step 1. Same as step 1 of algorithm 1.

Step 2. Same as step 2 of algorithm 1. The condition $q(j) + d(j) \le N/2$ is replaced by the condition $q^k(j) + d^k(j) + L^k(j) \le N/2$.

Step 3. Same as step 3 of algorithm 1.

Step 4. Same as step 4 of algorithm 2. Here (35) is replaced by

$$u = \arg\max \{2q^k(j) + d^k(j) + L^k(j)\}_{j=\overline{1, m_1 - f + 1}}. \qquad (35')$$

Algorithm 3 is carried out for each $k = \overline{1, s}$. The substantiation of algorithm 3 is similar to that of algorithm 2.

Now, after placing nonfixed elements, it becomes possible to determine the redundant elements of the second group, and use them for optimization [as mentioned, we have only to test conditions (12)].

Note that at step 4 of algorithm 3 we can use a modified step 4 of algorithm 1 to obtain an exact algorithm. This is inefficient, however, because the use of algorithm 3 assumes that, subsequently, redundant elements of the second group will be considered. Since it is in principle impossible to determine these elements before placing nonfixed elements, no reasonable exact algorithm of joint optimization exists that could take into account the redundant elements of the second group.

These algorithms have been developed for an MPA implementing a set of linear microprograms, but they also can be used for nonlinear microprograms. This requires a modification of MCF. In turn, branching and cyclic microprograms in the generalizing table may require adjustments of nonfixation intervals. Indeed, the nonfixation interval of a microoperation should be limited to the branch of the microprogram to which it belongs.

The algorithms have been implemented in PL-1 language of ES operating system, and can be used in computer-aided design of digital information converters.

LITERATURE CITED

1. V. M. Glushkov, Synthesis of Digital Automata [in Russian], Fizmatgiz, Moscow (1962).
2. Yu. A. Buzunov and E. N. Vavilov, Principles of Computer Design [in Russian], Tekhnika, Kiev (1972).
3. Yu. A. Buzunov, E. N. Vavilov, and T. P. Kachanov, "Tabular method of construction of the control signal distributor of a microprogramming automaton," in: Technical Cybernetics [in Russian], Naukova Dumka, Kiev (1971), pp. 184-197.
4. M. Gary and D. Johnson, Computers and Difficult Problems [Russian translation], Mir, Moscow (1982).
5. V. S. Mikhalevich and A. I. Kuksa, Methods of Sequential Optimization of Discrete Network Problems of Optimal Resource Allocation [in Russian], Nauka, Moscow (1983).

BOUNDING ALGORITHM FOR THE ROUTING PROBLEM WITH ARBITRARY PATHS AND ALTERNATIVE SERVERS

S. V. Sevast'yanov

UDC 519.854.2

A nontrivial property of optimal solutions of the Bellman–Johnson problem was discovered independently in [1 and 2] in 1974. For any individual problem with activity length matrix $A=(a_{ij})_{j=\overline{1,n}}^{i=\overline{1,m}}$, where m is the number of machines and n the number of parts, the length of the optimal schedule $T(P_{opt})$ was found to lie in the interval $\mathbf{E}=[H(A), H(A)+\varphi(m)h(A)]$, where $H(A)=\max_{i=\overline{1,m}}\sum_{j=1}^{n}a_{ij}$, $h(A)=\max_{ij} a_{ij}$, $\varphi(m)$ is a function of m.

Thus, without enumerating the entire feasible set of schedules, we can effectively find the optimum up to a value independent of the number of parts (which is important for problems where $n \gg m$). Effective algorithms to find approximate schedules P' with lengths T(P') from the interval $\mathbb{E}$ were also proposed in [1, 2], with absolute and relative accuracy estimates

$$T(P')-T(P_{opt})\leqslant \varphi(m)h(A), \tag{1}$$

$$T(P')/T(P_{opt})\leqslant 1+\varphi(m)h(A)/H(A), \tag{2}$$

if $H(A) > 0$. From (2) it follows, in particular, that if $m \leq$ const and $h(A)/H(A) \to 0$, then the solution P' is asymptotically optimal. It is significant that solutions P' with accuracy bounds (1), (2) were obtained in the class of permutation schedules with continuous activities, which justifies the intuitive attempt of many researchers working with the Johnson problem to restrict the analysis to this particular subclass of schedules.

The Akers–Friedman problem is more complex by an order of magnitude: here instead of a single technological path of the parts through the machines (1, 2,...,m) there may be up to m! different paths, which are permutations of the numbers 1, 2,...,m. Analysis of this model is further complicated by the absence of analytical representation of the objective function T(P). Nevertheless, the authors of [3-6] successfully established nontrivial properties of optimal solutions for the case $a_{ij} \equiv 1$. In particular, the bound

$$T(P_{opt})\leqslant H(A)\sqrt{m}+f(m)h(A)$$

was obtained in [3]. This bound immediately suggests the following natural question: is it possible to replace $\sqrt{m}$ with 1 [and f(m) with some function $\varphi'(m)$] so as to bound the optimum inside the interval $[H(A), H(A)+\varphi'(m)h(A)]$? First results suggesting an affirmative answer

Translated from Kibernetika, No. 6, pp. 74-79, November-December, 1986. Original article submitted December 12, 1983.