MATHEMATICS OF OPERATIONS RESEARCH
Vol. 6, No. 3, August 1981
Printed in U.S.A.

A VECTOR-SUM THEOREM AND ITS APPLICATION TO IMPROVING FLOW SHOP GUARANTEES*

IMRE BÁRÁNY

Hungarian Academy of Sciences

We prove that if a closed polygonal path in R^d consists of a finite number of line segments of at most unit length, then it is possible to transpose the segments in such a way that the new polygonal path is contained in a ball of radius $[\frac{3}{2}d]$. Using this result we give a near optimal algorithm for the *NP*-complete flow shop problem. The error of the algorithm cannot exceed a constant depending on the maximal execution time and the number of machines but not depending on the number of jobs. Our theorems improve earlier results of the same type by Belov and Stolin.

Introduction. The m-machine n-job flow shop problem can be stated as follows. The shop is an ordered set of m different machines indexed by $1, \ldots, m$. There are n jobs to be executed. Each job consists of m operation indexed by $1, \ldots, m$ and the jth operation of a job precedes its $(j+1)$th operation for $j = 1, \ldots, m-1$. Further, the jth operation of any job has to be carried out on the jth machine. An operation cannot be interrupted once it has begun execution. The execution time of the jth operation of the ith job is given. We must give the order of the executions of the operations on the m machines so that finish time be minimal.

This problem is solved effectively only when $m = 2$. The solution was given by S. M. Johnson [6]. His algorithm works in time at most proportional to $n \log n$ where n is the number of jobs. For $m > 2$ there has been a pronounced absence of efficient schedule optimization algorithm for flow shops [5]. This lack can be explained by the fact that for $m > 2$ the flow shop problem belongs to the class of *NP*-complete problems [5]. For *NP*-complete problems it is a natural approach to search for efficient algorithms that are not optimal, only near optimal.

In 1974 Belov and Stolin [2] gave an algorithm for the flow shop problem yielding a permutation schedule, i.e., a schedule for which the order of the execution of the operations is the same for each machine. This permutation schedule is near optimal in the sense that its error cannot exceed a constant $C(m, K)$ independent of the number of jobs, depending only on the number of machines m and on the maximal execution time K. They use a theorem from [7] which is known in the Russian literature as the Steinitz lemma. It says that there is a constant $C(d)$ depending only on d such that if a closed polygonal path in R^d consists of a finite number of line segments each of which is of at most unit length, then it is possible to transpose the segments in such a way that the new (closed) polygonal path is contained in a ball of radius $C(d)$. In fact, Kadec [7] gave an algorithm for this end yielding $C(d) = \sqrt{\frac{1}{3}(4^d - 1)} = O(2^d)$ with complexity $O(n^d)$ where n is the number of segments. From this fact Belov and Stolin proved $C(m, K) = (m-1)(\sqrt{m-1}\, C(m-1) + 1)K = O(m^{3/2}2^m K)$, their algorithm is of complexity $O(n^{m-1})$.

*Received February 16, 1979; revised September 15, 1980.
AMS 1980 subject classification. Primary 90B35.
IAOR 1973 subject classification. Main: Scheduling. Cross reference: Computational analysis.
OR/MS Index 1978 subject classification. Primary 583 Production/scheduling, flow shop.
Key words. Flow shop scheduling, computational complexity, Steinitz lemma.

0364-765X/81/0603/0445$01.25

The aim of this paper is to present better bounds for $C(d)$ and for $C(m, K)$ with better complexity results. We shall give an algorithm proving $C(d) \leqslant [\frac{3}{2}d]$ with complexity $O(n^2d^3 + nd^4)$. This, in turn, yields an algorithm of complexity $O(n^2m^3 + nm^4)$ for the flow shop problem with bound $C(m, K) = (m-1)[\frac{1}{2}(3m-1)]K = O(m^2K)$.

In 1977, independently of Belov and Stolin's results, Tibor Fiala [4] rediscovered the connection between the flow shop problem and the Steinitz lemma and proved the latter for $d = 2$. It was his result that made me think of proving the above statement on polygonal paths. My thanks are due to him not only for that but for the many discussions we had and for the help in preparing this paper as well. I am also indebted to E. G. Bajmóczy for valuable suggestions and conversations.

After having written this paper I learned that Sevast'yanov [8], too, had found an algorithm yielding $C(d) \leqslant d$ in $O(d2^dn^2)$ steps. From this he gets

$$C(m, K) = (m-1)^{5/2}K$$

with an algorithm of complexity $O(m2^mn^2)$.

Comparing Sevast'yanov's algorithm with our algorithm, we first mention that the algorithm presented here runs in time polynomial both in m and n. Previous algorithms were exponential in m. Secondly, the translation between the Steinitz lemma and the performance bound has been tightened saving a factor of m in our algorithm. But Sevast'yanov's algorithm if combined with this improved translation yields a slightly better performance guarantee (m^2K vs. $\frac{3}{2}m^2K$). However, Sevast'yanov's algorithm is much less efficient ($O(m2^mn^2)$ vs. $O(n^2m^3 + nm^4)$).

Notations and theorems. The m machines will be indexed by $j = 1, 2, \ldots, m$ according to their order of priority. There are n jobs, the ith job is given by an ordered set of nonnegative real numbers $t_{i,1}, \ldots, t_{i,m}$ where $t_{i,j}$ is the execution time of the jth operation of the ith job ($1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant m$). Let us introduce the following notations:

$$K = \max\{t_{i,j}\}, \qquad M_j = \sum_{i=1}^{n} t_{i,j}, \qquad \text{and}$$

$$M = \max_j M_j.$$

A permutation of the index set $1, 2, \ldots, n$ will be denoted by $i_1, i_2, \ldots, i_n$. Given a vector $x = (x_1, \ldots, x_d) \in R^d$ we write $\|x\|$ for the maximum norm of x, i.e., $\|x\| = \max|x_j|$. For a finite set H we denote the number of elements by $|H|$. Finally we mention that speaking of a set $V \subseteq R^d$ we usually mean a multiset, i.e., an indexed family of elements where the same element may occur with different indices. This will not cause any confusion.

Now we give our theorems.

THEOREM 1. *For a finite set* $V = \{v_1, \ldots, v_n\} \subseteq R^d$ *with*

$$\sum_{i=1}^{n} v_i = 0 \qquad \text{and} \quad \|v_i\| \leqslant 1 \qquad (i = 1, \ldots, n),$$

there is a permutation $i_1, \ldots, i_n$ *such that*

$$\max_{1 \leqslant k \leqslant n} \left\| \sum_{j=1}^{k} v_{i_j} \right\| \leqslant \left[\frac{3}{2} d \right].$$

This permutation can be given in $O(n^2d^3 + nd^4)$ *steps.*

THEOREM 2. *There is a permutation schedule for which finish time T satisfies the following inequalities*

$$M \leqslant T \leqslant M + (m-1)\left\lfloor \frac{3m-1}{2} \right\rfloor K.$$

This schedule can be given in $O(n^2m^3 + nm^4)$ steps.

PROOF OF THEOREM 2 (using Theorem 1). First we construct fictive execution times $t'_{i,j}$ such that $t_{i,j} \leqslant t'_{i,j} \leqslant K$ and $\sum_{i=1}^{n} t'_{i,j} = M$ for all $j = 1, \ldots, m$. This can be done quite easily (in $O(nm)$ steps) because

$$\sum_{i=1}^{n} (K - t_{i,j}) = nK - M_j \geqslant M - M_j.$$

Thus if $M_j < M$, then we can divide the value $M - M_j$ into n parts so that the ith part does not exceed $K - t_{i,j}$. Adding these numbers to the $t_{i,j}$ values we get the fictive $t'_{i,j}$.

It is known [9] and can be checked easily that if a permutation schedule is given by the $i_1, \ldots, i_n$ permutation of the jobs, then the finish time T can be expressed as

$$T = \max_{1 = k_0 < k_1 < \ldots < k_m = n} \left\{ \sum_{j=0}^{m-1} \sum_{s=k_j}^{k_{j+1}} t_{i_s, j+1} \right\}.$$

Applying this formula to the fictive execution times we get

$$T' = \max_{1 = k_0 < \ldots k_m = n} \left\{ \sum_{j=0}^{m-1} \sum_{s=k_j}^{k_{j+1}} t'_{i_s, j+1} \right\}.$$

Clearly $M \leqslant T \leqslant T'$. Further, rewriting the above expression

$$T' = M + \max\left(\sum_{j=1}^{m-1} t'_{i_{k_j}, j+1} + \sum_{j=1}^{m-1} \sum_{s=1}^{k_j} \left(t'_{i_s, j} - t'_{i_s, j+1} \right) \right)$$

$$\leqslant M + (m-1)K + \max \sum_{j=1}^{m-1} \sum_{s=1}^{k_j} \left(t'_{i_s, j} - t'_{i_s, j+1} \right)$$

where max is taken over $1 = k_0 \leqslant k_1 \leqslant \ldots k_m = n$. In order to estimate the double sum put

$$v_i = (t'_{i,1} - t'_{i,2}, t'_{i,2} - t'_{i,3}, \ldots, t'_{i,m-1} - t'_{i,m}) \in R^{m-1}.$$

Clearly $\sum_{i=1}^{n} v_i = 0$ and $\|v_i\| \leqslant K$, further,

$$\max \sum_{j=1}^{m-1} \sum_{s=1}^{k_j} \left(t'_{i_s, j} - t'_{i_s, j+1} \right)$$

$$\leqslant (m-1) \max_{1 \leqslant k \leqslant n} \left\| \sum_{s=1}^{k} v_{i_s} \right\|.$$

Now Theorem 1 (with $d = m - 1$) gives a permutation $i_1, \ldots, i_n$ for which

$$\max_{1 \leqslant k \leqslant n} \left\| \sum_{s=1}^{k} v_{i_s} \right\| \leqslant \left\lfloor \frac{3(m-1)}{2} \right\rfloor K.$$

This implies

$$M \leqslant T \leqslant T' \leqslant M + (m-1)\left\lfloor \frac{3m-1}{2} \right\rfloor K$$

as desired. The complexity of the algorithm is the same as in Theorem 1 with $d = m - 1$. ∎

PROOF OF THEOREM 1. We are going to give an algorithm to produce the desired permutation. We say that a map $\gamma : V \to R$ is a linear dependence (for V) if $\sum_{v \in V} \gamma(v) v = 0$. For any $c \in R$ put $V(\gamma > c) = \{v \in V : \gamma(v) > c\}$. $V(\gamma = c)$ and $V(\gamma < c)$ are defined analogously. The algorithm will produce a finite sequence of linear dependences $\gamma_0, \gamma_1, \ldots, \gamma_p$. Put now for $i = 0, \ldots, p$

$$A_i = V(\gamma_i = 1), \qquad B_i = V(\gamma_i > 0) \cap V(\gamma_i < 1), \qquad \text{and}$$

$$C_i = V(\gamma_j = 0).$$

The linear dependences will satisfy the following conditions:

(a) $0 \leqslant \gamma_i(v) \leqslant 1$ for $v \in V$,
(b) $\sum_{v \in V} \gamma_i(v) v = 0$,
(c) $|B_i| \leqslant d$,
(d) $A_{i+1} \supseteq A_i$ and $A_{i+1} \neq A_i$,
(e) $A_p = V$, and
(f) $|B_{i+1} \cup A_{i+1} \backslash A_i| \leqslant 2d$.

Because of (a) A_i, B_i and C_i are pairwise disjoint and their union is V. The sum in (b) is a relation between vectors from A_i and B_i. Condition (c) says that there are only a few vectors in this relation with coefficients different from 1 and so the sum of the vectors from A_i is near to zero. Condition (d) shows that the $A_i - s$ form a strictly increasing sequence of sets. In view of (f), this sequence cannot increase very fast, so the sum of all vectors from A_i and a few vectors from $A_{i+1} \backslash A_i$ is also near to zero.

For the construction of this sequence we shall make use of the following lemma which is a suitable modification of the well-known Carathéodory's theorem.

LEMMA. *Let $V \subset R^d$, $|V| = n$ and let $\lambda : V \to [0, \infty)$ be a nontrivial linear dependence and v^* be an arbitrary vector from $V(\lambda > 0)$. Then one can find in $O(nd^3)$ steps an $\alpha : V \to [0, \infty)$ nontrivial linear dependence such that for $D = V(\alpha > 0)$*

$$v^* \in D \qquad \text{and} \qquad |D| \leqslant d + 1.$$

PROOF OF THE LEMMA. If $|V(\lambda > 0)| \leqslant d + 1$, then put $\alpha = \lambda$. If not, then let $G \subseteq V(\lambda > 0) \backslash \{v^*\}$ be an arbitrary set with $|G| = d + 1$ and let us solve the following linear system by the well-known Gauss algorithm:

$$\sum_{v \in V} \mu(v) v = 0. \tag{1}$$

We get a nontrivial solution because there are d equations and $d + 1$ variables. Putting $\mu(v) = 0$ for $v \notin G$ we get a $\mu : V \to R$ nontrivial linear dependence. Now, if all $\mu(v)$ are nonnegative, then write

$$t_0 = \max \left\{ - \frac{\lambda(v)}{\mu(v)} : \mu(v) > 0 \right\},$$

and if $\mu(v) < 0$ for some $v \in G$, then put

$$t_0 = \min \left\{ - \frac{\lambda(v)}{\mu(v)} : \mu(v) < 0 \right\}.$$

Now define $\lambda' = \lambda + t_0 \mu$. Because of the definition of t_0 $\lambda' : V \to [0, \infty)$ is again a linear dependence, moreover,

$$|V(\lambda' > 0)| < |V(\lambda > 0)| \qquad \text{and}$$

$$v^* \in V(\lambda' > 0). \tag{2}$$

If $|V(\lambda' > 0)| \leqslant d + 1$, then put $\alpha = \lambda'$; if not, we replace λ by λ' and repeat the above procedure.

In view of (2), the linear system (1) has to be solved at most n times. As the complexity of the Gauss algorithm is $O(d^3)$ we can get α in $O(nd^3)$ steps.

Having finished the proof of the lemma we give the construction of the sequence γ_s for $s = 0, 1, \ldots, p$.

For $s = 0$ let $\gamma_0 = 0$ for each $v \in V$, i.e., we start with the trivial linear dependence. Clearly A_0 and B_0 equal the empty set and $C_0 = V$, so for $i = 0$ conditions (a), (b) and (c) hold. Using induction on s we suppose that γ_s is defined for some $s \geqslant 0$ so that for $i = s$ (a), (b) and (c) hold. Now to construct γ_{s+1} we consider two cases.

Case 1. $|V \backslash A_s| > d$.

Let us consider the following nontrivial linear dependence:

$$\lambda(v) = 1 - \gamma_s(v) \qquad \text{for} \quad v \in V.$$

Let v^* be an arbitrary element from B_s (or, if it is empty, from C_s) and apply the lemma. We get an $\alpha : V \to [0, \infty)$ nontrivial linear dependence and a set $D = V(\alpha > 0)$ with $|D| \leqslant d + 1$ and $v^* \in D$. For this set D

$$|B_s \cup D| \leqslant 2d \tag{3}$$

because $|B_s| \leqslant d$ by induction, $|D| \leqslant d + 1$ and if B_s is not void, then v^* is a common element of B_s and D. Now we determine the maximal value of t for which

$$\gamma_s(v) + t\alpha(v) \leqslant 1 \qquad \text{for all} \quad v \in V.$$

This value is given by

$$t_0 = \min\left\{ \frac{1 - \gamma_s(v)}{\alpha(v)} : \alpha(v) > 0 \right\}.$$

Put now $\beta(v) = \gamma_s(v) + t_0\alpha(v)$. Clearly $\beta : V \to [0, 1]$ is again a linear dependence and, by the choice of t_0,

$$V(\beta = 1) \supset A_s, \qquad V(\beta = 1) \neq A_s \qquad \text{and} \tag{4}$$

$$V(0 < \beta < 1) = B_s \cup D \backslash V(\beta = 1). \tag{5}$$

At this point we split Case 1 into two further subcases.

Case 1(a). $|V(0 < \beta < 1)| \leqslant d$. In this case put $\gamma_{s+1} = \beta$. Now (4) implies that (d) holds for $i = s$. Condition (f) is a consequence of (5). It is obvious that (a), (b) and (c) hold true in this case.

Case 1(b). $|V(0 < \beta < 1)| > d$. In this case we decrease $|V(0 < \beta < 1)|$ in the same manner as in the lemma.

Let us choose a set $G \subseteq V(0 < \beta < 1)$ with $|G| = d + 1$ and find a nontrivial solution to the following system

$$\sum_{v \in G} \mu(v) v = 0. \tag{6}$$

Defining $\mu(v) = 0$ for $v \notin G$ we determine the maximal number t_0 for which

$$0 \leqslant \beta(v) + t_0\mu(v) \leqslant 1$$

holds for all $v \in V$. Then for $\beta' = \beta + t_0\mu$ we have by the definition of t_0

$$|V(0 < \beta' < 1)| < |V(0 < \beta < 1)|. \tag{7}$$

If $|V(0 < \beta' < 1)| > d$, then we replace β by β' and repeat the above procedure from

the choice of G to the construction of β'. At the end we get the linear dependence β' with

$$|V(0 < \beta' < 1)| \leqslant d, \tag{8}$$

$$V(\beta' = 1) \supset A_s, \qquad V(\beta' = 1) \neq A_s \qquad \text{and} \tag{9}$$

$$V(\beta' > 0) \backslash A_s \subseteq B_s \cup D. \tag{10}$$

Putting $\gamma_{s+1} = \beta'$ we get the following element of the sequence $\{\gamma_s\}$. Now condition (d) holds for $i = s$ because of (9), (f) is a consequence of (10) and (3) and, in view of (8), condition (c) holds true for $i = s + 1$.

Case 2. $|V \backslash A_s| \leqslant d$. If $|V \backslash A_s| = 0$, we are done; γ_s is the last element of the sequence and $p = s$. Otherwise define $\gamma_{s+1}(v) = 1$ for all $v \in V$ and $p = s + 1$, so γ_{s+1} is the last element of the sequence. Clearly $A_p = V$, B_p and C_p are empty, thus conditions (a), (b) and (c) hold for $i = p$ and (d) and (f) hold for $i = p - 1 = s$.

Having finished the construction of sets A_i we define the order of vectors as the order of first appearance in some A_i. More precisely, we define a total order $>_i$ on each of the sets $A_i \backslash A_{i-1}$ arbitrarily. Secondly we define a total order on V: for $v \in A_i \backslash A_{i-1}$ and $v' \in A_j \backslash A_{j-1}$ let $v > v'$ iff either $i = j$ and $v >_i v$, or $i < j$. Clearly, this is a total order on V. Now let i_1 be the index of the greatest element of V for this total order, i_2 the second greatest and so on. We claim that $i_1, i_2, \ldots, i_n$ is the desired permutation.

We have to show that the estimation of the theorem holds. To do so first remark that by condition (b)

$$\sum_{v \in A_i} v = -\sum_{v \in B_i} \gamma_i(v) v,$$

so by (c) we have for $i = 0, 1, \ldots, p$

$$\left\| \sum_{v \in A_i} v \right\| \leqslant d.$$

Now let $F = \{v_{i_1}, \ldots, v_{i_k}\}$ and suppose that $A_i \subseteq F \subseteq A_{i+1}$. In this case

$$\left\| \sum_{t=1}^{k} v_{i_t} \right\| \leqslant \left\| \sum_{v \in A_i} v \right\| + \left\| \sum_{v \in F \backslash A_i} v \right\| \leqslant d + |F \backslash A_i|.$$

Similarly

$$\sum_{t=1}^{k} v_{i_t} = \sum_{v \in A_{i+1}} v - \sum_{v \in A_{i+1} \backslash F} v$$

$$= -\sum_{v \in B_{i+1}} \gamma_{i+1}(v) v - \sum_{v \in A_{i+1} \backslash F} v,$$

consequently

$$\left\| \sum_{t=1}^{k} v_{i_t} \right\| \leqslant |B_{i+1} \cup A_{i+1} \backslash F|.$$

Because of condition (f) the sum of these two bounds for $\|\sum_{t=1}^{k} v_{i_t}\|$ cannot exceed $3d$. It follows from this that the minimum of these bounds cannot exceed $[3d/2]$.

To estimate the complexity of the algorithm we remark that condition (d) implies $p \leqslant n$, so we apply the lemma at most n times. This gives complexity $O(n^2 d^3)$. (5), (7) and (3) imply that for the construction of γ_{s+1} from γ_s we must solve the equation (6)

at most d times in Case 1(b). This gives complexity $O(nd^4)$, so the desired permutation can be given in $O(n^2d^3 + nd^4)$ steps. ∎

Numerical example. Consider the following three machine examples with $4n$ jobs (n is arbitrary). The jobs are of three types:

$$A \text{ type jobs: } t_{i1} = 2, t_{i2} = 1, t_{i3} = 3 \quad (i = 1, \ldots, 2n),$$
$$B \text{ type jobs: } t_{i1} = 2, t_{i2} = 2, t_{i2} = 0 \quad (i = 2n + 1, \ldots, 3n),$$
$$C \text{ type jobs: } t_{i1} = 1, t_{i2} = 3, t_{i3} = 1 \quad (i = 3n + 1, \ldots, 4n).$$

In this case $M = M_1 = M_2 = M_3 = 7n$ and $K = 3$. It is easy to see that the optimal order of the jobs is

$$ACAB\ ACAB\ ACAB\ \ldots\ \ldots\ ACAB$$

when the machines work without idle time and $T = T_{\min} = 7n + 3$.

There is some freedom in implementing the algorithm of Theorem 2 (for instance, the choice of v^* or the ordering of $A_{i+1} \setminus A_i$). For this example the algorithm gives several near-optimal schedules (including the optimal one) for all of which, of course,

$$7n + 3 \leqslant T \leqslant 7n + 24.$$

We mention that other standard heuristics could have errors proportional to n for this example. For instance, the method of Palmer [9] gives the permutation schedule

$$AAA \ldots A\ CCC \ldots C\ BBB \ldots B$$

with finish time $T = 9n + 1$. The method of Cambell, Dudek and Smith [9] gives the same schedule. This is explained by the fact that these heuristics schedule the jobs in such a way that if an A type job precedes a B type one than *all* A type jobs precede *all* B type jobs. This is not so with the algorithm of Theorem 2.

REMARKS. Theorem 2 says that if we fix the maximal execution time K and the number of machines, then letting $n \to \infty$, the finish time guaranteed by our algorithm is asymptotically equal to the optimal. However, we remark that Theorem 2 is not sharp even for $m = 2$. Applying Johnson's rule, $T_{\min} \leqslant M + K$ is trivially a sharp upper bound in this case. For $m = 3$ $T_{\min} \leqslant M + 5K$ is known [4].

Concerning Theorem 1 we mention that the estimation given in it is fairly good. Firstly, using Hadamard matrices one can give vectors $v_1, \ldots, v_n \in R^d$ with $\sum_{i=1}^n v_i = 0$ and $\|v_i\| \leqslant 1$ (for $i = 1, \ldots, n$) such that for any permutation $i_1, \ldots, i_n$ $\max_{1 \leqslant k \leqslant n} \|\sum_{t=1}^k v_{i_t}\| \geqslant \sqrt{d}/2$. Secondly, as it can be seen from the proof, Theorem 1 holds true for any norm. Now for the l_1 norm we can give another example $v_1, \ldots, v_n \in R^d$ with $\sum_{i=1}^n v_i = 0$ and $\|v_i\|_{l_1} \leqslant 1$ such that for any permutation $\max_{1 \leqslant k \leqslant n} \|\sum_{t=1}^k v_{i_t}\|_{l_1} \geqslant d/2$.

Finally we mention that the running time of the algorithm of Theorem 1 (and, consequently, of Theorem 2) can be improved to $O(n^2 + nd)M(d)$ where $M(d)$ is the current best bound for d by d matrix multiplication (see [1]). The current value of $M(d)$ is somewhere about $O(d^{2.6})$, so this would be an asymptotic improvement although quite impractical on reasonably sized problems. One can even wonder whether the *repeated* nature of the Gaussian elimination can be used to yield further savings in the running time of the algorithm. Probably Sevast'yanov's better bound can be obtained in time polynomial both in n and d.

Note added in proof. The answer to the last question is in the affirmative as it was shown recently by V. S. Grinberg and S. V. Sevast'yanov. Their algorithm will appear soon.

References

[1] Aho, A., Hopcroft, J. and Ullmann, J. (1974). *The Design and Analysis of Computer Algorithms.* Addison-Wesley, p. 242.

[2] Belov, I. S. and Stolin, J. I. (1974). An Algorithm for the Flow Shop Problem. In *Mathematical Economics and Functional Analysis.* Nauka, Moscow (in Russian).

[3] Coffman, E. G., Jr. (1976). *Computer Job-Shop Scheduling Theory.* John Wiley and Sons, New York.

[4] Fiala, T. (1977). Közelitő Algoritmus a Három Gép Problémára. *Alkalmazott Matematikai Lapok* **3** 389–398 (in Hungarian).

[5] Garey, M. R., Johnson, D. S. and Sethi, R. (1975). The Complexity of Flow Shop and Job Shop Scheduling. Technical Report No. 198, Computer Science Dept., Pennsylvania State University.

[6] Johnson, S. M. (1954). Optimal Two- and Three-Stage Production Schedules. *Naval Res. Logist. Quart.* **1** 1.

[7] Kadec, M. I. (1953). On a Property of Polygonal Paths in n-Dimensional Space. *Uspehi Mat. Nauk* 139–143.

[8] Sevast'yanov, S. V. (1978). On Approximate Solutions of Scheduling Problems. *Metody Discretnogo Analiza.* **32** 66–75 (in Russian).

[9] Tanaev, V. S. and Shkurba, V. V. (1975). *Introduction to the Theory of Scheduling.* Nauka, Moscow (in Russian).

MATHEMATICAL INSTITUTE, HUNGARIAN ACADEMY OF SCIENCES, 1053, BUDAPEST, REÁLTANODA 13-15, HUNGARY