Computing 52, 17–30 (1994)

Computing

Scheduling Two Jobs with Fixed and Nonfixed Routes

N. V. Shakhlevich, Yu. N. Sotskov, Minsk

Received August 19, 1992

Abstract — Zusammenfassung

Scheduling Two Jobs with Fixed and Nonfixed Routes. The shop-scheduling problem with two jobs and m machines is considered under the condition that the machine order is fixed in advance for the first job and nonfixed for the second job. The problems of makespan and mean flow time minimization are proved to be NP-hard if operation preemption is forbidden. In the case of preemption allowance for any given regular criterion the $O(n_*)$ algorithm is proposed. Here, n_* is the maximum number of operations per job.

AMS Subject Classification: 90B35

Key words: Optimization, optimal makespan schedule, optimal mean flow time schedule, regular criterion, polynomial time algorithm, NP-hard problem.

Reihenfolgeprobleme bei zwei Aufträgen mit fixierten und unfixierten Routinen. Es werden Reihenfolgeprobleme mit zwei Aufträgen und m Maschinen untersucht, wobei die technologische Reihenfolge der Maschinen für den ersten Auftrag gegeben und für den zweiten Auftrag variabel ist. Es wird bewiesen, daß die Probleme "Minimierung der Gesamtbearbeitungszeit" und "Minimierung der mittleren Durchlaufzeit" NP-hard sind, wenn eine Unterbrechung der Operationen verboten ist. Für ein beliebiges reguläres Kriterium wird bei Zulassung von Unterbrechungen ein $O(n_*)$ Algorithmus entwickelt, wobei mit n_* die maximale Anzahl der Operationen für den ersten Auftrag bezeichnet wird.

1. Introduction

We consider the following shop-scheduling problem. There is a set of jobs $J = \{J_1, \ldots, J_n\}$ that have to be processed on the machine set $M = \{M_1, \ldots, M_m\}$. Each machine $M_k \in M$ processes at most one job at a time and each job $J_i \in J$ is processed on at most one machine at a time.

A shop-scheduling problem can be specified in terms of a four-field classification $\alpha|\beta|\gamma|\delta$, where α, β, γ and δ mean the number of jobs, the number of machines, the machines-jobs characteristics and the optimality criterion, respectively.

The third field γ is defined as follows. If $\gamma = J$, we have a job-shop problem, denoted by $n|m|J|F(s)$. Each job $J_i \in J$ has a route (machine order) $l^i = (l^i_1, \ldots, l^i_{n_i})$ given before. It means that job J_i is to be processed first on machine $M_{l^i_1}$, then on machine $M_{l^i_2}$ and so on up to machine $M_{l^i_{n_i}}$, $M_{l^i_q} \in M$, $q = 1, \ldots, n_i$. It should be noted that some machines from set M may be absent in route l^i and on the other hand some machines may be repeated two or more times in l^i. We assume that $\langle i, q \rangle$ denotes an operation of processing job J_i on machine $M_{l^i_q}$ in a job-shop. The processing time (duration) $p_{iq} > 0$ of each operation $\langle i, q \rangle$ is given beforehand. A special case of a job-shop is a flow-shop in which all the jobs have identical routes. In this case $\gamma = F$.

If $\gamma = O$, we have an open-shop problem denoted by $n|m|O|F(s)$. The machine order for any job is nonfixed in advance and the route of each job is to be found while constructing the schedule. Within this problem each job $J_i \in J$ is to be processed on each machine $M_k \in M$ strictly one time: $n_i = m$. For each operation $\langle i, k \rangle$ of processing job J_i on machine M_k its processing time $p_{ik} \geq 0$ is given. It should be noted that for the job-shop problem operation $\langle i, q \rangle$ is defined by job number i and by stage number q while for the open-shop problem operation $\langle i, k \rangle$ is defined by job number i and by machine number k.

The third field may include some other problem characteristics as for instance the parameter Pr indicating the preemption allowance. If preemption is forbidden, schedule s can be completely defined by starting time $t_{iq}(s) \geq 0$ of each operation $\langle i, q \rangle$ or by its completion time $c_{iq}(s) = t_{iq}(s) + p_{iq}$. In an open-shop case we use the number k of the corresponding machine M_k instead of the stage number q: $t_{ik}(s)$ and $c_{ik}(s)$.

If preemption is allowed, to determine a schedule s it is necessary to find both starting times and completion times of all operation parts at which operations $\langle i, q \rangle$, $i = 1, \ldots, n_i$, are split within schedule s.

This paper deals with the $n|m|J,O|F(s)$ and $n|m|J,O,Pr|F(s)$ problems for the so-called nonhomogeneous shop. It means that the routes of some jobs are fixed and the routes of the others are nonfixed. The criteria considered are $C_{\max}$, $\sum C_i$ and Φ. If $F(s) = C_{\max}$, the problem (it is denoted by $n|m|J,O|C_{\max}$) is to construct a schedule s^* minimizing the makespan (i.e. maximum completion time of the jobs or schedule length):

$$C_{\max}(s) = \max\{c_i(s) | J_i \in J\}.$$

Here, $c_i(s)$ is the completion time of job J_i within schedule s: $c_i(s) = c_{in_i}(s)$.

If $F(s) = \sum C_i$, the problem (it is denoted by $n|m|J,O|\sum C_i$) is to find a schedule s^* minimizing the sum of completion times of the jobs or (which is the same) minimizing mean flow-time:

$$\sum C_i(s) = \sum_{i=1}^{n} c_i(s).$$

If $F(s) = \Phi$, it is necessary to make up a schedule s^* minimizing the value of any given objective function $\Phi(c_1(s), \ldots, c_n(s))$ nondecreasing with respect to each of its arguments: if $c_i(s) \leq c_i(s')$ for all $J_i \in J$ then $\Phi(c_1(s), \ldots, c_n(s)) \leq \Phi(c_1(s'), \ldots, c_n(s'))$. Such optimality criterion is said to be a regular one [2]. It is obvious that $C_{\max}$ and $\sum C_i$ are regular criteria.

It should be recalled that the majority of results are dedicated to "pure" shops, that is, to those with either fixed or nonfixed routes for all the jobs. Combinations of such shops have been only recently investigated: Paper [8] seems to be the first one to consider the $n|m|F, O|C_{\max}$ problem. This research has been extended in [13, 14]. The main result received in [13, 14] is the $O(n \cdot \log_2 n)$ algorithm for solving $n|2|J, O|C_{\max}$ and $n|2|J, O, Pr|C_{\max}$ problems.

Our paper continues the complexity study of scheduling problems with the fixed number of jobs [1, 5, 9, 10, 11, 12, 15] in the case of a nonhomogeneous shop. In Section 2 it is proved that $2|m|J, O|C_{\max}$ and $2|m|J, O|\sum C_i$ problems are NP-hard if preemption is forbidden. In Sections 3 and 4 a linear time algorithm is given for the $2|m|J, O, Pr|C_{\max}$ problem with preemption allowance and arbitrary regular criterion.

2. Nonhomogeneous Shop without Preemption

The polynomial time algorithm for the $n|2|J, O, n_1 \leq 2|C_{\max}$ problem with $m = 2$ and arbitrary n has been proposed in [13, 14]. We prove that if vice versa $n = 2$ and the number of machines m is not restricted, then the analogous problem is (binary) NP-hard.

Theorem 1. *The* $2|m|J, O|C_{\max}$ *problem is NP-hard.*

Proof. We reduce polynomially the NP-hard PARTITION problem [3] to the following decision problem: Does there exist a schedule s^0 of processing two jobs on m machines without preemption one job having fixed route and another one having nonfixed route for which $C_{\max}(s^0) \leq y$ for a given y. The PARTITION problem can be described as follows. A set $A = \{1, \ldots, a\}$ is given and the positive integer e_i is connected with each element $i \in A$, $\sum_{i \in A} e_i = 2E$. The question is if there exists a partition of A into subsets A_1 and A_2 for which $\sum_{i \in A_1} e_i = \sum_{i \in A_2} e_i = E$. If such subsets A_1 and A_2 do exist, we say that the PARTITION problem has a solution.

Here is an example of such a $2|m|J, O|C_{\max}$ decision problem. Job J_2 is processed on machines $M = \{M_1, \ldots, M_{a+1}\}$, $m = a + 1$, its route being nonfixed, and job's J_1 route being fixed: $l^1 = (a + 1, a, a + 1)$, $n_1 = 3$. We set $y = 3E$, $p_{2,a+1} = E$, $p_{2,k} = e_k$ for $k = 1, \ldots, a$. The duration of each operation $\langle 1, 1 \rangle$, $\langle 1, 2 \rangle$ and $\langle 1, 3 \rangle$ is equal to E.

We show that schedule s^0 without preemption and satisfying $C_{\max}(s^0) \leq y$ exists for the example constructed if and only if the PARTITION problem has a solution.

Sufficiency. If the PARTITION problem has a solution $A = A_1 \cup A_2$, $\sum_{i \in A_1} e_i = \sum_{i \in A_2} e_i$, the schedule s^0 can be constructed as follows. Jobs J_1 and J_2 are processed in the closed interval $[0, 3E]$ without delays and job J_2 is processed according to the route $l^2 = (\pi_1, a+1, \pi_2)$, where π_1 is a permutation of set A_1 numbers. As $\sum_{i \in A_1} e_i = \sum_{i \in A_2} e_i = E$ the machines processing jobs J_1 and J_2 in the closed interval $[0, E]$ (and similar in the interval $(2E, 3E]$) are all different. Hence we have constructed a feasible schedule (no machine processes two jobs simultaneously) and for this schedule the following equation is true: $C_{\max}(s^0) = 3E = y$ (see Fig. 1).

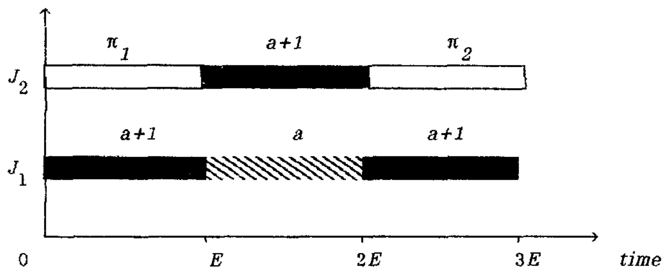

Figure 1. Schedule s^0

Necessity. Let there exist a schedule s^0 with $C_{\max}(s^0) \leq y$. Since $\sum_{k=1}^{m} p_{2k} = 3E$ and $\sum_{q=1}^{n_1} p_{1q} = 3E$ then $C_{\max}(s^0) = y$ and each job J_1 and J_2 is processed without delays in the closed interval $[0, 3E]$. Since $p_{2,a+1} + p_{1,1} + p_{1,3} = 3E$ machine M_{a+1} processes one job J_1 or J_2 at a time, i.e. this machine functions with no idle time. Job's J_1 route l^1 is fixed: $l^1 = (a+1, a, a+1)$ and so only one interval $(E, 2E]$ is suitable for processing job J_2 on the machine M_{a+1} (provided that $C_{\max}(s^0) \leq y$).

Hence, machines $\{M_1, \ldots, M_a\}$ have to process job J_2 in the closed interval $[0, E]$ and in the interval $(2E, 3E]$. Since preemption is forbidden the set of machines which process job J_2 in the closed interval $[0, E]$ gives numbers of subset A_1 elements and the set of machines which process job J_2 in the interval $(2E, 3E]$ gives numbers of the other subset A_2. So the schedule s^0 determines the solution of the PARTITION problem. Obviously, the reduction constructed is a polynomial one and so Theorem 1 is proved.

It is easy to come to the conclusion that the arguments given above are valid if $C_{\max}$ is replaced by $\sum C_i$ and the value y is set to be equal to $6E$. Such replacement is possible because of the fixed number of jobs. This case is similar to the one considered in [11, 12] for NP-hardness proof of $n|m|J|C_{\max}$ and $n|m|J|\sum C_i$ problems with $n = 3$. Thus we have the following

Theorem 2. *The problem* $2|m|J, O|\sum C_i$ *is NP-hard.*

We have shown that the $2|m|J, O|F(s)$ problem is NP-hard even for the simplest optimality criteria: $F(s) = C_{\max}$ and $F(s) = \sum C_i$. Taking into account reducibility among scheduling problems with different classical criteria (see [6], p. 9) we can conclude that the $2|m|J, O|F(s)$ problem is NP-hard for any traditional criterion: $L_{\max}$, $\sum T_i$, $\sum \omega_i T_i$, $\sum \omega_i C_i$, $\sum U_i$ and $\sum \omega_i U_i$.

Note that the stage number n_i may essentially influence the complexity of the problem. In particular, if $n_i \leq 2$ the $n|2|J, O, n_i \leq 2|C_{\max}$ problem is polynomially solvable [13, 14], but if $n_i \leq 3$ it becomes NP-hard. The latter statement follows immediately from the NP-hardness proof of the $n|2|J, n_1 = 3, n_i = 1, i \neq 1|C_{\max}$ problem proposed in [7], since this problem can be considered as a $n|2|J, O, n_i \leq 3|C_{\max}$ one.

3. Nonhomogeneous Shop with Preemptions and without Delays

We show that the problems the NP-hardness of which was established in Theorems 1 and 2, become polynomially solvable if preemption is allowed. Moreover, we shall prove that these problems are polynomially solvable for any traditional regular criterion.

We consider the $2|m|J, O, Pr|\Phi$ problem, job J_1 route being fixed: $l^1 = (l_1^1, \ldots, l_{n_1}^1)$, and job J_2 route being nonfixed. Denote $p_1 = \sum_{q=1}^{n_1} p_{1q}$ and $p_2 = \sum_{k=1}^{m} p_{2k}$.

If $p_1 \geq p_2$, one can define the stage number $\bar{q}$ for which

$$\sum_{q=1}^{\bar{q}-1} p_{1q} < p_2 \leq \sum_{q=1}^{\bar{q}} p_{1q} \tag{1}$$

Since preemption is allowed we assume that there is an equality sign in the right part of relation (1), i.e. $p_2 = \sum_{q=1}^{\bar{q}} p_{1q}$, because otherwise one may split operation $\langle 1, \bar{q} \rangle$ into two parts, the durations of the first and the second parts being equal to $p_2 - \sum_{q=1}^{\bar{q}-1} p_{1q}$ and $p_{1\bar{q}} - \left(p_2 - \sum_{q=1}^{\bar{q}-1} p_{1q}\right)$, respectively.

If $p_1 < p_2$, we set $\bar{q}$ equal to n_1.

Calculate value

$$\Theta_k = p_{2k} + \sum_{\substack{1 \leq q \leq \bar{q}, \\ l_q^1 = k}} p_{1q} \tag{2}$$

for every machine $M_k \in M$ and define number k^0 for which $\Theta_{k^0} = \max\{\Theta_k | M_k \in M\}$.

We now describe an algorithm for constructing a schedule s^0 with no delays in job J_1 (and in job J_2) processing within the closed interval $[0, p_1]$ (and $[0, p_2]$, respectively) for the case when inequality

$$\Theta_{k^0} \leq p_2 \tag{3}$$

holds. Note that job J_1 is to be processed without preemption and operation $\langle 1, 1 \rangle$ is to be fulfilled within the closed interval $[0, p_{1,1}]$, operation $\langle 1, 2 \rangle$ is to be fulfilled within the interval $(p_{1,1}, p_{1,1} + p_{1,2}]$, etc., operation $\langle 1, n_1 \rangle$ is to be fulfilled within the interval $\left(\sum_{q=1}^{n_1 - 1} p_{1,q}, \sum_{q=1}^{n_1} p_{1,q} \right]$ for the schedule s^0.

Algorithm A.

1. IF $p_1 > p_2$ THEN (*construct the schedule's s^0 finite part in the interval* $(p_2, p_1]$)
2. BEGIN
3. $t_{1,\bar{q}+1}(s^0) := p_2$; $t_{1,\bar{q}+2}(s^0) := p_2 + p_{1,\bar{q}+1}$; ...; $t_{1,n_1}(s^0) := t_{1,n_1-1}(s^0) + p_{1,n_1-1}$;
   END
4. AUXILSCHED (s')
5. PICKFRAGM $(s', l_1^1, \ldots, l_{\bar{q}}^1)$
6. ALLOCFRAGM $(s, l_1^1, \ldots, l_{\bar{q}}^1)$

Here, AUXILSCHED, PICKFRAGM and ALLOCFRAGM are the following procedures:

AUXILSCHED (s') constructs the optimal schedule s' for the auxiliary problem $2|m|O, Pr|\Phi$. This auxiliary problem differs from the given $2|m|J, O, Pr|\Phi$ problem (we call it the main problem) only by job J_1: Its route becomes nonfixed, operations of the auxiliary problem correspond to operations $\langle 1, 1 \rangle$, ..., $\langle 1, \bar{q} \rangle$ of the main problem, and the durations of the auxiliary problem operations $\langle 1, k \rangle'$ become equal to $p'_{1k} = \Theta_k - p_{2,k} = \sum_{\substack{1 \le q \le \bar{q} \\ l_q^1 = k}} p_{1q}$, $M_k \in M$.

The polynomial algorithm for solving the $2|m|O, Pr|\Phi$ problem has been suggested in [4, 9]. As it follows from condition (3) the schedule s' constructed by this algorithm has no delays within the closed interval $[0, \min\{p_1, p_2\}]$ for job J_1 processing and within the closed interval $[0, p_2]$ for job J_2 processing. Moreover, all operations (perhaps, except at most one operation) are fulfilled without preemption for the schedule s'. And if such operation with preemption exists, then it is an operation of the job the total duration of which is greater than the total duration of another one. Hence, job J_1 is processed without preemption within the schedule s'.

The optimal schedule s' is used further on to construct the initial part of the schedule s^0 in the closed interval $[0, p_2]$ (or the whole schedule s^0 if $p_1 \le p_2$).

PICKFRAGM $(s', l_1^1, \ldots, l_{\bar{q}}^1)$ picks out the schedule s' fragments according to the initial part $(l_1^1, \ldots, l_{\bar{q}}^1)$ of job J_1 route l^1 of the main problem.

If the number k of the machine M_k occurs in this part of the route one time only, then we pick out the schedule s' part of jobs J_1 and J_2 processing in the interval $(t_{1k}(s'), c_{1k}(s')]$, $k = l_q^1$, $1 \le q \le \bar{q}$, and call it a fragment.

If the number k occurs h, $h > 1$, times, i.e. $l_{i_1}^1 = \cdots = l_{i_h}^1 = k$, $1 \le i_1 \le \cdots \le i_h \le \bar{q}$, the schedule s' part in the interval $(t_{1k}(s'), c_{1k}(s')]$ should be broken into h following fragments: $(t_{1k}(s'), t_{1k}(s') + p_{1l_{i_1}^1}]$, ..., $\left(t_{1k}(s') + \sum_{j=1}^{h-1} p_{1l_{i_j}^1}, t_{1k}(s') + \sum_{j=1}^{h} p_{1l_{i_j}^1} \right]$.

ALLOCFRAGM $(s, l_1^1, \ldots, l_{\bar{q}}^1)$ allocates the fragments picked out from the schedule s' within the closed interval $[0, p_2]$ in such a way that operations $\langle 1, q\rangle$, $q = 2, \ldots, \bar{q}$, of the main problem are fulfilled within intervals $\left(\sum_{j=1}^{q-1} p_{1j}, \sum_{j=1}^{q} p_{1j}\right]$ and operation $\langle 1, 1\rangle$ is fulfilled within the closed interval $[0, p_{11}]$.

If $p_1 \leq p_2$, the schedule s^0 is constructed, otherwise it can be obtained by adding the schedule part for interval $(p_2, p_1]$ constructed at Steps 1–3 to the schedule part for the closed interval $[0, p_2]$ constructed at Steps 4–6.

The schedule constructed by Algorithm A is feasible as the schedule s' is admissible (jobs J_1 and J_2 are processed on different machines at a time); secondly, only job J_2 route is nonfixed in the main problem $2|m|J, O, Pr|\Phi$ and thirdly the operation preemption is allowed.

We evaluate the complexity of Algorithm A. The complexity of the algorithm from [4, 9] used in the AUXILSCHED procedure for solving the $2|m|O, Pr|\Phi$ problem is $O(m)$. The amount of the schedule fragments picked out at Step 5 in the PICKFRAGM procedure is $\bar{q} \leq n_1$ and this value bounds also procedure ALLOCFRAGM's complexity. Steps 1–3 also take no more than $O(n_1)$ elementary operations. So the whole complexity of Algorithm A is $O(n_*)$, where $n_* = \max\{n_1, n_2\} = \max\{n_1, m\}$.

We have described an algorithm for constructing a schedule with no delays in jobs J_1 and J_2 processing within closed intervals $[0, p_1]$ and $[0, p_2]$, respectively, if inequality (3) is true. Now we show that if vice versa there exists a schedule s^0 then inequality (3) is fulfilled.

Let condition (3) not be fulfilled and nevertheless let the schedule s^0 exist. Then machine M_{k^0} operates within the closed interval $[0, p_2]$ for Θ_{k^0} time units having no idle time since job J_2 is processed on this machine for p_{2k^0} time units (it follows from the definitions of p_2 and s^0) and job J_1 is processed on this machine for $\sum_{\substack{1 \leq q \leq \bar{q} \\ l_q^1 = k^0}} p_{1q}$ time units (it follows from the definitions of $\bar{q}$ and s^0). But this contradicts the assumption concerning the whole length of this time interval: $p_2 < \Theta_{k^0}$.

So we have proved the following

Theorem 3. *There exists a schedule s^0 for the $2|m|J, O, Pr|\Phi$ problem with no delays in job J_1 (and in job J_2) processing within the closed interval $[0, p_1]$ (and $[0, p_2]$, respectively) if and only if inequality (3) is true.*

4. Nonhomogeneous Shop with Preemptions and Delays

We investigate the $2|m|J, O, Pr|\Phi$ problem when condition (3) is not valid:

$$\Theta_{k^0} > p_2. \tag{4}$$

As it follows from Theorem 3 the delays in job J_1 processing within the closed interval $[0, p_1]$ or delays in job J_2 processing within the closed interval $[0, p_2]$ are unavoidable in this case and so the objective function value for the schedule depends on the distribution of these delays between the jobs. Note that, while in Section 3 only one schedule was optimal for all regular criteria, different schedules may be optimal in this section for different regular criteria.

Consider the auxiliary problem differing from the one given only by job J_2: Job J_2 in the auxiliary problem consists of a single operation $\langle 2, k^0 \rangle$ with duration p_{2,k^0} being the same as for the main problem. For any schedule s'' of the auxiliary problem we consider schedule s of the main problem corresponding to the schedule s''. All operations of jobs J_1 and J_2 processing, except operations $\langle 2, k \rangle$, $M_k \in M \backslash \{M_{k^0}\}$, are fulfilled within schedule s in the same time intervals as the corresponding operations in schedule s''.

We construct, at first, schedule s_1'' of the auxiliary problem. For this schedule job J_1 is processed without delays and machine M_{k^0} operates with no idle time in the closed interval $[0, c_2(s_1'')]$ processing operations $\langle 1, q_1 \rangle, \ldots, \langle 1, q_\alpha \rangle$ without preemption and operation $\langle 2, k^0 \rangle$ with preemption (see Fig. 2).

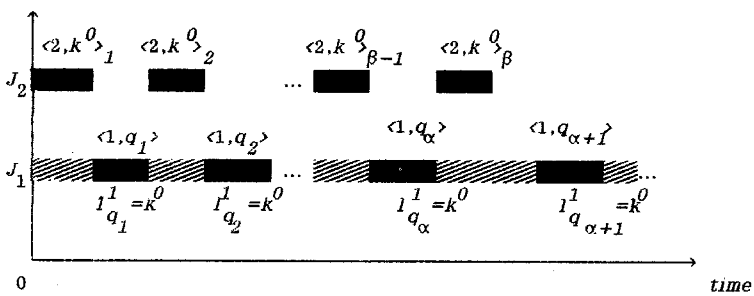

Figure 2. Schedule s_1''

Denote the operation $\langle 2, k^0 \rangle$ parts processing on machine M_{k^0} within schedule s_1'' without preemption by $\langle 2, k^0 \rangle_1, \ldots, \langle 2, k^0 \rangle_\beta$ (see Fig. 2), their durations by $p_{\langle 2,k^0 \rangle_1}, \ldots, p_{\langle 2,k^0 \rangle_\beta}$, respectively, and the sum of delays in job J_2 processing by $\delta_2(s_1'')$: $\delta_2(s_1'') = p_{1,q_1} + \cdots + p_{1,q_\alpha}$.

Note that $\delta_2(s_1'')$ is greater than the sum of durations of operations $\langle 2, k \rangle$, $M_k \in M \backslash \{M_{k^0}\}$. Otherwise, operations $\langle 1, q_1 \rangle, \langle 1, q_2 \rangle, \ldots, \langle 1, \bar{q} \rangle$ and all parts of the operation $\langle 2, k^0 \rangle$ are fulfilled up to time p_2. It means that the inequality $\sum_{\substack{1 \le q \le \bar{q} \\ l_q^1 = k^0}} p_{1,q} + p_{2,k^0} = \Theta_{k^0} \le p_2$ is valid and that contradicts inequality (4).

Thus to construct schedule s_1 for the main problem using the schedule s_1'' for the auxiliary problem one can assign operations $\langle 2, k \rangle$, $M_k \in M \backslash \{M_{k^0}\}$, fulfilling consecutively within intervals of operations $\langle 1,q_1\rangle$, $\langle 1,q_2\rangle$, ..., $\langle 1,q_\alpha\rangle$ processing: $(t_{1,q_1}(s_1''), c_{1,q_1}(s_1'')]$, $(t_{1,q_2}(s_1''), c_{1,q_2}(s_1'')]$, ..., $(t_{1,q_\alpha}(s_1''), c_{1,q_\alpha}(s_1'')]$. These operations $\langle 2,k\rangle$, $M_k \in M\backslash\{M_{k^0}\}$, may be fulfilled in any possible order with preemption. Within schedule s_1 job J_1 is processed without delays and job J_2 is processed with delays. The sum of these delays is equal to $\delta_2(s_1) = \delta_2(s_1'') - (p_2 - p_{2,k^0}) > 0$.

It is obvious that the completion time $c_2(s_1)$ may be diminished by introducing delays in job J_1 processing and reducing in such a way job J_2 delays $\delta_2(s_1)$. It is more convenient to consider schedule s_1'' of the auxiliary problem to realize such redistribution of the delays. If for some schedule s'' the equation $\delta_2(s'') = p_2 - p_{2,k^0}$ (or $\delta_2(s) = 0$ equivalently) is valid, then it is impossible to diminish job J_2 completion time and there is no sense in further diminishing the $\delta_2(s'')$ value. So one may restrict himself to considering schedules s'' of the auxiliary problem satisfying condition

$$\delta_2(s'') \geq p_2 - p_{2,k^0}. \tag{5}$$

We describe Algorithm B of constructing schedules s_2'', ..., s_v'' of the auxiliary problem satisfying inequality (5) and show that the optimal schedule $s^{*''}$ of the auxiliary problem with additional condition (5) can be selected among schedules s_1'', ..., s_v'' so that the objective function $F(s)$ should be minimal for this schedule: $F(s^{*''}) = \min\{F(s_1''),\dots,F(s_v'')\}$. The optimal schedule s^* for the main problem corresponds to the schedule $s^{*''}$ and may be constructed by allocating operations $\langle 2,k\rangle$, $M_k \in M\backslash\{M_{k^0}\}$, in intervals of delays of the job J_2 within the schedule $s^{*''}$.

Algorithm B.

1. $h := 1$; (*h is the number of the last constructed schedule s''*)
2. $\delta_2(s_h'') := \sum_{j=1}^{\alpha} p_{1,q_j}$
3. WHILE $\delta_2(s_h'') > p_2 - p_{2,k^0}$ DO
   BEGIN
4. IF $\sum_{j=1}^{\alpha-h} p_{1,q_j} \geq p_2 - p_{2,k^0}$ THEN
   INTERCHANGE1($\langle 1,q_{\alpha-h+1}\rangle, \langle 2,k^0\rangle_{\beta-h+1},\dots,\langle 2,k^0\rangle_\beta$);
   $\delta_2(s_{h+1}'') := \sum_{j=1}^{\alpha-h} p_{1,q_j}$; $h := h+1$
   ELSE
   INTERCHANGE2($\langle 1,q_{\alpha-h+1}\rangle, \langle 2,k^0\rangle_{\beta-h+1},\dots,\langle 2,k^0\rangle_\beta$);
   $\delta_2(s_{h+1}'') := p_2 - p_{2,k^0}$; $h := h+1$
   END

Here, INTERCHANGE1 and INTERCHANGE2 are procedures used to construct schedule s_{h+1}'' from schedule s_h'' by interchanging the processing order of operation $\langle 1,q_{\alpha-h+1}\rangle$ and the last parts of operation $\langle 2,k^0\rangle$ processed without delays one by one: $\langle 2,k^0\rangle_{\beta-h+1},\dots,\langle 2,k^0\rangle_\beta$.

In the case of INTERCHANGE1 operation $\langle 1,q_{\alpha-h+1}\rangle$ is fulfilled without preemption within schedule s_{h+1}'' (see Fig. 3a).

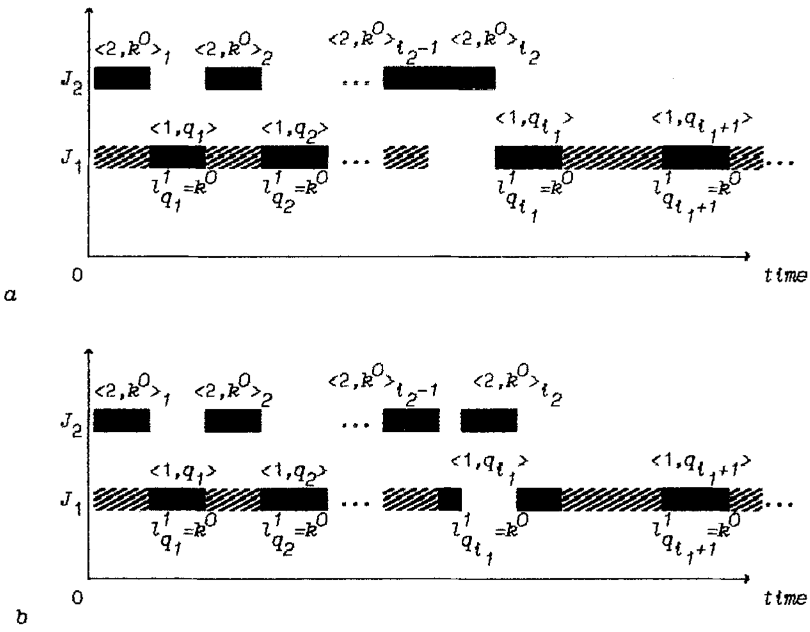

Figure 3. Schedule s_2''

In the case of INTERCHANGE2 operation $\langle 1, q_{\alpha-h+1} \rangle$ is splitted into two parts and it is processed with preemptions within schedule s_{h+1}'' (see Fig. 3b), the duration of its first part being equal to $(p_2 - p_{2,k^0}) - \sum_{j=1}^{\alpha-h} p_{1,q_j}$ and the duration of the second part being equal to $p_{1,q_{\alpha-h+1}} - \left((p_2 - p_{2,k^0}) - \sum_{j=1}^{\alpha-h} p_{1,q_j} \right) = \sum_{j=1}^{\alpha-h+1} p_{1,q_j} - (p_2 - p_{2,k^0})$ so that inequality (5) is valid for schedule s_{h+1}''.

Note that if the INTERCHANGE2 procedure is applied to schedule s_h'' then the WHILE-condition is violated for the new schedule s_{h+1}'' (it is fulfilled as equality: $\delta_2(s_h'') = p_2 - p_{2,k^0}$) and the algorithm stops. So this procedure can be applied no more than once by Algorithm B.

Now we prove the following

Lemma 1. *The optimal schedule for the auxiliary problem with additional condition* (5) *can be found among schedules* $s_1'', \ldots, s_v''$.

Proof. Consider at first class S_1'' of the schedules satisfying condition (5) and inequality $c_{2,k^0}(s'') > c_{2,q_\alpha}(s'')$. For any schedule $s'' \in S_1''$ the sum of job J_2 delays $\delta_2(s'')$ is no less than $\Delta_1^2 = \sum_{j=1}^{\alpha} p_{1,q_j}$. Schedule $s_1'' \in S_1''$ is optimal within this class since $\delta_2(s_1'') = \Delta_1^2$ for this schedule and job J_1 is processed without delays: $\delta_1(s_1'') = 0$.

Now consider class S_2'' of the schedules satisfying inequality (5) and $c_{1,q_{\alpha-1}}(s'') < c_{2,k^0}(s'') < c_{1,q_\alpha}(s'')$. For any schedule $s'' \in S_2''$ the sum of job J_2 delays $\delta_2(s'')$ is no less than $\Delta_2^2 = \max\left\{\sum_{j=1}^{\alpha-1} p_{1,q_j}, p_2 - p_{2,k^0}\right\}$ and the sum of job J_1 delays $\delta_1(s'')$ is no less than $\Delta_2^1 = p_{\langle 2,k^0\rangle_\beta}$. Schedule $s_2'' \in S_2''$ is optimal within this class since $\delta_2(s_2'') = \Delta_2^2$, $\delta_1(s_2'') = \Delta_2^1$.

It is easy to check the validity of analogous statements for the rest of the classes of schedules $S_3'', \ldots, S_v''$. Indeed, the schedules of class S_h'', $3 \le h \le v$, satisfy inequality (5) and $c_{1,q_{\alpha-h+1}}(s'') < c_{2,k^0}(s'') < c_{1,q_{\alpha-h+2}}(s'')$. For any schedule $s'' \in S_h''$ we have: $\delta_2(s'') \ge \Delta_h^2 = \max\left\{\sum_{j=1}^{\alpha-h+1} p_{1,q_j}, p_2 - p_{2,k^0}\right\}$, $\delta_1(s'') \ge \Delta_h^1 = \sum_{j=\beta-h+2}^{\beta} p_{\langle 2,k^0\rangle_j}$, and moreover these conditions are fulfilled as equations (not as inequalities) for schedule s_h''. So schedule s_h'' is optimal in the class of schedules S_h'', and the lemma is proved.

Finally, we evaluate the complexity of the optimal schedule s^* construction for the main problem. The number of schedules considered is equal to v, $v < n_1$; the construction of schedule s_1'' needs $O(n_1)$ elementary operations; construction of schedule s_{h+1}'' using schedule s_h'', $2 \le h \le v - 1$, needs $O(1)$ elementary operations. So the total complexity of constructing schedules $s_1'', \ldots, s_v''$ is restricted by $O(n_1)$.

Schedule $s^{*\prime\prime}$ is selected among schedules $s_1'', \ldots, s_v''$ and the complexity of this selection depends on the complexity of calculating the values $F(s_1''), \ldots, F(s_v'')$. Since schedule s^* is constructed on the basis of schedule $s^{*\prime\prime}$ by picking out time intervals for operations $\langle 2, k\rangle$ processing, $M_k \in M\backslash\{M_{k^0}\}$, the complexity of schedule s^* construction is $O(n_*)$, not taking into account the complexity of calculating $F(s)$.

Here is an example to illustrate the algorithm described.

Example $2|m|J,O,Pr|F(s)$. Job J_1 is processed on machine set $M = \{M_1, \ldots, M_4\}$ according to a given route $l^1 = (M_1, M_2, M_1, M_3, M_1, M_4, M_1, M_3, M_4, M_1, M_2, M_4, M_1, M_3)$; its operation durations being $p_{1,1} = 2$, $p_{1,2} = 1$, $p_{1,3} = 3$, $p_{1,4} = 1$, $p_{1,5} = 4$, $p_{1,6} = 5$, $p_{1,7} = 4$, $p_{1,8} = 3$, $p_{1,9} = 1$, $p_{1,10} = 5$, $p_{1,11} = 4$, $p_{1,12} = 5$, $p_{1,13} = 1$, $p_{1,14} = 1$. Job J_2 is processed on machine set M as well but its route is not fixed before scheduling, $p_{2,1} = 17, p_{2,2} = 2, p_{2,3} = 2, p_{2,4} = 2$. We consider four objective functions $F(s) = F_i(s)$:

$$F_1(s) = \max\{c_1(s), c_2(s)\},$$

$$F_2(s) = \max\{c_1(s) - 45, 0\} + \max\{c_2(s) - 30, 0\},$$

$$F_3(s) = c_1(s) \cdot c_2(s) + 8 \cdot c_1(s)$$

$$F_4(s) = c_1(s) \cdot c_2(s).$$

Functions $F_1(s)$ and $F_2(s)$ are traditional: Minimizing $F_1(s)$ means the $C_{\max}$ criterion, and $F_2(s)$ means minimizing the sum of tardinesses. Note that $F_3(s)$ and $F_4(s)$ are regular criteria, too.

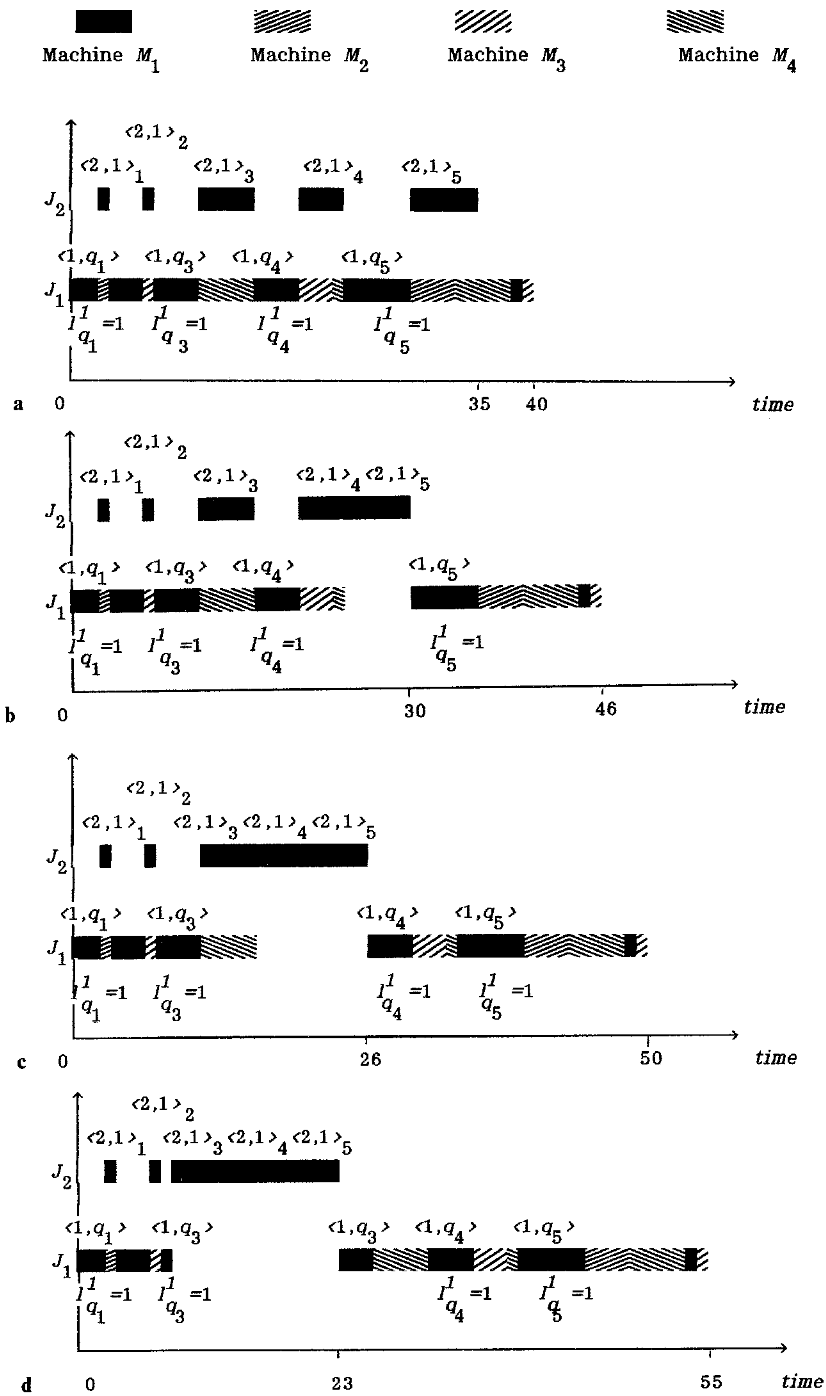

Figure 4. **a, b** Schedules s''_1, s''_2. **c, d** Schedules s''_3, s''_4

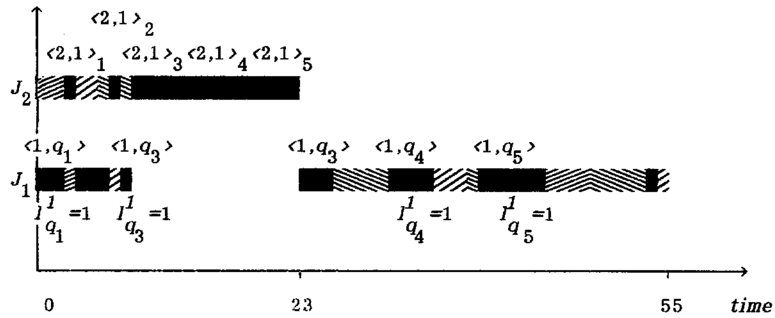

Figure 5. Schedule s^* for $F(s) = F_4(s)$

It is easy to make sure that $M_{k^0} = M_1$ and inequality (4) is valid: $\Theta_{k^0} = 17 + 2 + 3 + 4 + 4 = 30$, $p_2 = 23$ (note that $\bar{q} = 8$). According to Theorem 3 the delays are inevitable and we can apply Algorithm B to construct an optimal schedule. Schedules $s_1'', \ldots, s_4''$, $v = 4$, are represented in Fig. 4a–d. Schedule s_1'' appears to be optimal for the objective function $F_1(s)$, schedule s_2'' for $F_2(s)$, schedule s_3'' for $F_3(s)$, and schedule s_4'' for $F_4(s)$. Figure 5 represents optimal schedule s^* for the main problem with objective function $F_4(s)$. This schedule corresponds to schedule s_4'' of the auxiliary problem.

Acknowledgements

This research was partially supported by Deutshe Forschungsgemeinschaft, Project ScheMA.

The authors are grateful to anonymous referees for their suggestions, which improved the presentation of this paper.

References

[1] Brucker, P.: An efficient algorithm for the job-shop problem with two jobs. Computing *40*, 353–359 (1988).
[2] Conway, R. W., Maxwell, W. L., Miller, L. W.: Theory of scheduling. Reading: Addison-Wesley 1967.
[3] Garey, M. R., Johnson, D. S.: Computers and intractability: A guide to the theory of NP-completeness. San Francisco: Freeman 1979.
[4] Gonzalez, T., Sahni, S.: Open shop scheduling to minimize finish time. J. Assoc. Comput. Math. *23*, 665–679 (1976).
[5] Hardgrave, W. W., Nemhauser, G.: A geometric model and graphical algorithm for a sequencing problem. Oper. Res. *11*, 889–900 (1963).
[6] Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., Shmoys, D. B.: Sequencing and scheduling: algorithms and complexity. Report 8945/A, Econometric Institute, Erasmus University Rotterdam, 1989.
[7] Lenstra, J. K., Rinnooy Kan, A. H. G., Brucker, P.: Complexity of machine scheduling problems. Ann. Discr. Math. *1*, 343–362 (1977).
[8] Masuda, T., Ishii, H., Nishida, T.: The mixed shop scheduling problem. Discr. Appl. Math. *11*, 175–186 (1985).

[9] Shakhlevich, N. V., Strusevich, V. A.: Scheduling two jobs in a multi-machine open shop to minimize an arbitrary regular penalty function. Report 9125/A, Econometric Institute, Erasmus University Rotterdam, 1990.
[10] Sotskov, Yu. N.: The optimal processing of two jobs under a regular criterion. In: Automatization of design processes (Automatizacia Processov Proektirovania), pp. 86–95. Minsk: Institute of Engineering Cybernetics, 1985 (in Russian).
[11] Sotskov, Yu. N.: The complexity of shop-scheduling problems with two or three jobs. Eur. J. Oper. Res. *53*, 326–336 (1991).
[12] Sotskov, Yu. N., Shakhlevich, N. V.: NP-hardness of optimal scheduling three jobs. Izvestia Akademii Nauk BSSR, Seria Fiziko-Matematicheskikh Nauk *4*, 96–101 (1990) (in Russian).
[13] Strusevich, V. A.: On nonhomogeneous two-stage deterministic scheduling systems. Kibernetika (Kiev) *3*, 88–94 (1989) (in Russian).
[14] Strusevich, V. A.: Two machine super-shop scheduling problem. J. Opl. Res. Soc. *42*, 479–492 (1991).
[15] Szwarc, W.: Solution of the Akers-Friedman scheduling problem. Oper. Res. *8*, 782–788 (1960).

Dr. N. V. Shakhlevich
Dr. Yu. N. Sotskov
Institute of Engineering
Cybernetics
Surganov st. 6
220012 Minsk
Byelorussia