EFFECTIVE OPTIMIZATION METHODS FOR SINGLE-MACHINE SCHEDULING (SURVEY)

K. V. Shakhbazyan and N. B. Lebedinskaya

The article focuses on effective single-machine scheduling algorithms. We consider the optimization of a function f defined on feasible permutations assuming that the function f induces certain job interchange relations. The interchange relations include "job insertion," interchange of symbol chains, and the mutually complementary properties of interchange and embedding. Some new nontraditional problem formulations are considered together with the corresponding methods of solution.

INTRODUCTION

Recent years have witnessed considerable progress in our understanding of many deterministic problems in scheduling theory. Major advances have been made in the investigation of complexity and in generalization of results and solution methods for problems that previously were considered to be entirely unrelated.

In this article, we will attempt to describe a fairly limited area of scheduling theory, namely single-machine scheduling, for which effective solution found in the classical books of Conway, Maxwell, and Miller [26], Tanaev and Shkurba [13], and in surveys [43, 28, 37]. The purpose of this article is to consider the current state of the theory of single-machine schedules, with special emphasis on the work of Soviet authors, as it is not covered in the available surveys. Our survey focuses only on effective solution algorithms and ignores numerous enumerative and approximate methods.

The article consists of two sections. The first deals with optimization of functions defined on feasible permutations. General methods were successfully developed for these problems and some interesting results were obtained. In this section we strive to describe a certain general approach to these problems and consider some useful techniques.

Unfortunately, there is no unifying idea for the interesting topics covered in Sec. 2. We consider these new, but unrelated problems because they are strictly nontraditional and yet admit effective solution procedures.

1. OPTIMIZATION OF FUNCTIONS DEFINED ON FEASIBLE PERMUTATIONS

A number of publications relating to optimization of functions defined on permutation sets have appeared recently and independently. In 1956, Smith [47] introduced the pairwise interchange property for adjacent jobs, which has since led to effective solution of some problems without precedence constraints. Various generalizations of this property proposed in [1, 3-5, 33, 34, 40-42] proved quite fruitful in the study of the same problems with pre-

Translated from Zapiski Nauchnykh Seminarov Leningradskogo Otdeleniya Matematicheskogo Instituta im. V. A. Steklova AN SSSR, Vol. 111, pp. 195-217, 1981.

cedence constraints. Most of these problems can be formulated in the form $1|prec|f$ (using the notation of [28]), namely:

Statement of the Problem. Let f be a function that associates a real penalty $f(\pi)$ with each permutation π of the elements of the set of jobs $\mathcal{T}=\{1,\dots,n\}$. Consider a partial order described by the directed graph $D=(\mathcal{T},\mathcal{U})$ that imposes precedence constraints on the feasible permutations. Find a minimum-penalty feasible permutation.

This problem is evidently NP-hard. It remains NP-hard even for precedence constraints in the form of chains and for "simple" functions f, such as those in the problems

$$\begin{aligned} &1\,|chain,\ p_i=1\,|\sum U_j\,,\\ &1\,|chain,\ p_i=1\,|\sum w_j T_j\,, \qquad (1)\\ &1\,|chain,\ p_i=1,\ r_i|\sum w_j C_j\,, \end{aligned}$$

which follows from the work of Lenstra and Rinnooy Kan [38]. Therefore, it is quite useless to try to devise a polynomial algorithm for the solution of the general problem. The emphasis should be on isolating subclasses in which the problem $1|prec|f$ can be solved effectively.

We now proceed to analyze three problems of the type $1|prec|f$, where the function f induces some job interchange properties (i.e., for the symbols $\mathcal{T}$), thus ensuring existence of effective algorithms. First we consider the case when f induces the complete property of "job insertion." In this case, the problem $1|prec|f$ is effectively solvable for any precedence constraints. The second problem operates with a function f that induces a complete interchange property for chains of jobs. In this case, there hardly exists an effective solution for general job precedence constraints, and an effective solution has been obtained only for precedence constraints with the so-called PC-property.

Both cases mentioned above assume completeness of the permutability relations. The last case differs from the earlier two in that it considers incomplete job interchange property reinforced by job embedding property. The NP-complete problems listed in (1) belong to this category. A certain class of effectively solvable problems has been isolated in this case among the problems without precedence constraints.

1.1. Complete Job Insertion Property

Here we consider one property of the function f which leads to an effective algorithm for the problem $1|prec|f$ with arbitrary precedence constraints.

Monma [41] considered a $1|prec|f$ sequencing problem in which f had a forward insertion property. This property implies the existence of a transitive, asymmetric, complete preference relation $\prec$ defined on the set of jobs $\mathcal{T}$ such that the following condition holds: if $i,j\in\mathcal{T}$, then the relation $j\prec i$ implies the inequality $f(sitju)\geqslant f(sjitu)$, where s,t,u are arbitrary chains of jobs.

If in the sequencing problem f has the forward insertion property, the problem $1|prec|f$ can be solved in time $O(n^2)$ with arbitrary precedence constraints.

The corresponding solution algorithm can be described as follows. In the precedence graph D find a set I of jobs without predecessors. Order the set I by the relation $\prec$. The first element in the sequence becomes the first job of the optimal permutation π.

Eliminating this job from $\mathcal{T}$, we obtain the same problem with a truncated job set.

Backward insertion property can be similarly considered.

A number of problems in which f has forward or backward insertion property are considered in [41]. In particular, these problems include $1|prec|E_{max}$, $1|prec|\max w_i C_i$, and others.

1.2. Complete Chain Interchange Relation

Let us now turn to another property of the functions f which leads to an effective sequencing algorithm with PC precedence constraints.

First we formulate the constraints on the problem $1|prec|f$ introduced in [1, 3, 4, 5, 33, 34, 40, 42]. These articles were published almost simultaneously and they contain closely related results. We will follow the notation and the terminology from [34]. All the authors assume that f is defined on the set Σ of all the permutations of the symbols from $\mathcal{T}$ and satisfies the following conditions:

Condition 1. The function f induces on the set Σ a chain interchange relation $\prec$ i.e., for any two chains β, γ and any permutations of the form $\alpha\beta\gamma\delta$ the relation $\beta \prec \gamma$ implies the inequality $f(\alpha\beta\gamma\delta) \leqslant f(\alpha\gamma\beta\delta)$.

Gordon's priority creation condition [4, 5] and Burdyuk's condition of existence of the sequencing function [1] are equivalent to the existence of the chain interchange relation [34, 42]. However, Condition 1 is not sufficiently strong for the solution of the optimization problem. All the authors stipulate an additional condition.

Condition 2. The chain interchange relation is an asymmetric, transitive, complete relation.

This condition is equivalent to regulatiry of the sequencing function in [1]. We now give some examples of problems in which f satisfies Conditions 1 and 2.

Problem 1 [1]. Given the matrix $A = \{a_{ij}\}_{i,j=1}^{n}$ find a permutation π of the set $\{1, \dots, n\}$ such that its simultaneous application to the columns and the rows of A minimizes the sum of the superdiagonal elements. Here the relation $\beta \prec \gamma$, where β and γ are chains in $\{1, \dots, n\}$, is equivalent to the inequality

$$\sum_{i\in\beta} \sum_{j\in\gamma} a_{ij} \leqslant \sum_{i\in\gamma} \sum_{j\in\beta} a_{ij}.$$

Problem 2 [1, 34, 40, 42]. The Bellman–Johnson problem considered in [24, 30]. Here the function $f(\pi)$ has the form

$$f(\pi) = \max_{1\leqslant u\leqslant n} \left\{ \sum_{i=1}^{u} a_{\pi(i)} + \sum_{i=u+1}^{n} b_{\pi(i)} \right\}.$$

The corresponding chain interchange relation $\beta \prec \gamma$ is equivalent to the inequality

$$\min\left\{f(\beta) - \sum_{i\in\beta} b_i,\ f(\gamma) - \sum_{i\in\gamma} a_i\right\} \leqslant \min\left\{f(\gamma) - \sum_{i\in\gamma} b_i,\ f(\beta) - \sum_{i\in\beta} a_i\right\}.$$

Problem 3 [34, 42]. The problem $1||\sum w_j C_j$ first considered in [47]. Here the relation $\beta \prec \gamma$ implies

$$\sum_{i\in\beta} w_i \Big/ \sum_{i\in\beta} p_i \leqslant \sum_{i\in\gamma} w_i \Big/ \sum_{i\in\gamma} p_i$$

Problem 4 [34, 42] the problem $1\|\sum w_j \exp(-\tau C_j)$ first considered in [44]. Here $\beta \prec \gamma$ for $\beta = v_1 \dots v_k$ and $\gamma = u_1 \dots u_\ell$ is equivalent to

$$\sum_{i=1}^{k} w_{v_i} e^{-\tau \sum_{s=1}^{i} p_{v_s}} + \sum_{i=1}^{\ell} w_{u_i} e^{-\tau\left[\sum_{s=1}^{k} p_{v_s} + \sum_{s=1}^{i} p_{u_s}\right]} \leqslant \sum_{i=1}^{\ell} w_{u_i} e^{-\tau \sum_{s=1}^{i} p_{u_s}} + \sum_{i=1}^{k} w_{v_i} e^{-\tau\left[\sum_{s=1}^{\ell} p_{u_s} + \sum_{s=1}^{i} p_{v_s}\right]}.$$

Now consider a diagraph $G^{\prec} = (\Sigma, U^{\prec})$, where $(\beta,\gamma) \in U^{\prec}$ if and only if $\beta \prec \gamma$. Clearly, the relation $\prec$ is asymmetric, complete, and transitive if and only if $G^{\prec}$ is an asymmetric, complete, transitive, acyclic graph.

The problem $1|\circ|f$ (without precedence constraints), where f satisfies Conditions 1 and 2, has a trivial solution and therefore for such functions f it is meaningful to consider the problem $1|prec|f$ with precedence constraints. The solution of the problem $1|prec|f$ is determined by the relationship between the graphs $G^{\prec}$ and D . If D is a subgraph of $G^{\prec}$ every permutation π, consistent with the partial order $G^{\prec}$, solves the problem. Difficulties arise when D is not a subgraph of $G^{\prec}$. For this particular case Burdyuk [1] developed an algorithm that generalizes the ideas of [3, 4, 5, 22, 44]. This algorithm successively transforms the graph D, adding arcs and merging vertices. This procedure associates with the vertices of D chains of symbols, and not individual symbols from $\mathcal{T}$. To justify these transformations of the graph D, the author derives conditions under which for given chains σ and τ there is an optimal permutation π of one of the following two forms: $\pi = \alpha\sigma\beta\tau\gamma$ or $\pi = \alpha\sigma\tau\gamma$. Here $\beta \neq \phi$, i.e., either the chain σ precedes the chain τ or it directly precedes in some optimal permutation π . In the first case, the arc (σ,τ) can be added to the graph D , and in the second case the vertices σ and τ can be merged and replaced by a single vertex with the corresponding string $\sigma\tau$.

Algorithm. Each step of the algorithm finds two vertices in the graph D , which can be either joined by an arc or merged into one, and performs the corresponding transformation of the graph D . The algorithm stops in either of the following two cases:

1. The graph D has been reduced to a single vertex. The chain corresponding to this vertex is clearly the sought optimal permutation.

2. The graph D contains no two vertices that allow further transformation. In this case, the algorithm does not produce a solution, merely lowering the dimensionality of the problem.

A class of graphs was identified in [1] for which the algorithm always solves the problem $1|prec|f$. This is the class of graphs with the recursive PC property, which may be defined as follows:

1) if D is a subgraph of $G^{\prec}$, then D has the PC property.

2) if D_1 and D_2 have the PC property, the digraph obtained by joining D_1 and D_2 in parallel or in series has the PC property.

Although the complexity of this algorithm was not investigated in [1], it is clearly polynomial.

It is easily seen that series-parallel graphs have the PC property, so that the proposed algorithm always leads to a solution for these graphs. Representing the series-parallel graphs as a decomposition tree, Monma and Sydney [42] obtained a $O(n \ln n)$ algorithm for solving the problem with series-parallel precedence constraints.

1.3. Incomplete Job Interchange Relation

The situation is much more complex if the function f induces an incomplete interchange relation. This assumption was first made by Shakhbazyan and Lebedinskaya [8, 9, 12-20]. Two mutually complementary relations are considered on the set $\mathcal{T}$, complementing each other to a complete relation. The first relation is the transitive job interchange relation. The second is the job embedding relation, which assumes that for a pair of jobs i, j linked by this relation the optimal sequence depends on their position in the permutation.

Under these assumptions, the problems $1|\circ|f$ no longer have an obvious solution, and the problems $1|prec|f$ become NP-hard even for the simplest functions f and precedence constraints as in (1). In what follows, we will describe one class of problems that admit an effective algorithm. This has necessitated the introduction of a special class of structured permutations and schedules.

We now proceed to describe this class of problems. Note that each permutation may be treated as a single-machine schedule without idle time, whose beginning is associated with some fixed time moment, say 0. Then each symbol in the permutation $\pi = i_1 \ldots i_n$ acquires a time coordinate: the symbol i_j has the coordinate $j - 1$.

We can somewhat expand the notion of schedule, by considering schedules with the beginning at an arbitrary integer point t, i.e., the pair $\langle \pi, t \rangle$. Then each symbol i_j in the permutation π acquires the coordinate $t + j - 1$. Now suppose that each job $i \in \mathcal{T}$ is of length p_i, where p_i is an integer; jobs may be preempted, but only at integer points. Consider an arbitrary pair $\langle \pi, t \rangle$, where $\pi = i_1 \ldots i_p$ contains precisely p_i symbols i $(i = 1, \ldots, n)$ so that the length of π is $p = \sum_{i=1}^{n} p_i$. Such a pair $\langle \pi, t \rangle$ may be treated as a schedule beginning at the point t, and the symbol $i_j = k$ may be interpreted as one of p_k independent unit-length operations comprising the job k which is performed at time $t + j - 1$. In this section, we only consider schedules $\langle \pi, t \rangle$ of this type without idle time and the functions f defined on such schedules. Let us now refine the definition of the function f. Let the function $H(X)$ be defined on finite numerical sequences $X = x_1, \ldots, x_m$ and consider a set of individual penalty functions $\Phi = \{\varphi_1, \ldots, \varphi_n\}$. The function f is defined for any schedule $R = \langle \pi, t \rangle$, $\pi = i_1 \ldots i_p$, by the following equality:

$$f(R) = f(\langle \pi, t \rangle) = H(\varphi_{i_1}(t), \varphi_{i_2}(t), \ldots, \varphi_{i_p}(t + p - 1)). \tag{2}$$

The problem is to minimize the function $f(R)$ at a fixed point t, i.e., to find a permutation $\pi_{opt}(t)$ such that for any permutation π

$$f(\langle \pi_{opt}, t \rangle) \leqslant f(\langle \pi, t \rangle).$$

We denote this problem by $1|pmtn, t|H, \Phi$.

We now use the important concept of a generalized succession interval (see [13]).

<u>Definition 1.</u> The interval $[t_1, t_2]$ is a generalized succession interval of type $i \longrightarrow j$

if and only if for any schedule $R=\langle\pi,t\rangle=\langle i_1\ldots i_p,t\rangle$, where $i_\sigma=i$, $i_\tau=j$ $(\sigma<\tau)$ and the coordinates i_σ and i_τ are from the interval $[t_1,t_2]$, transposition of the symbols i_σ and i_τ does not reduce the value of $f(\langle\pi,t\rangle)$.

The function f defined with the aid of the function $H,\varphi_1,\ldots,\varphi_n$ in (2) is considered in the sequel under the following assumptions.

Condition A. The function f induces two relations on the set of jobs $\mathcal{T}$.

1) the precedence relation on symbols $\prec$, such that $i\prec j$ if and only if the interval $(-\infty,+\infty)$ is a generalized succession interval of type $i\longrightarrow j$.

2) the embedding relation on symbols $\subset$, such that $i\subset j$ if and only if there is a finite point $\bar{x}(i,j)$ for which $(-\infty,\bar{x}(i,j))$ is a generalized succession interval of type and $(\bar{x}(i,j),+\infty)$ is a generalized succession interval of type $i\longrightarrow j$.

Condition B. Both relations $\prec$ and $\subset$ are asymmetric and transitive, complementing one another to a complete asymmetric relation. These relations are described by the graphs $G_\prec$ and $G_\subset$.

If the function H and the set of functions $\Phi=\{\varphi_i\}_1^n$ satisfy conditions A and B, then for convenience we identify the symbol i with the corresponding function φ_i and say that φ_i precedes φ_j , $(\varphi_i\prec\varphi_j)$, if $i\prec j$; φ_i is embedded in φ_j $(\varphi_i\subset\varphi_j)$ if $i\subset j$ and finally the set Φ is a structured function set with respect to the function H.

To solve the problem $1|pmtn,t|H,\Phi$ it is best to consider schedules of a special type — so-called structured schedules.

Definition 2. Consider the permutation π. The symbol i precedes the symbol j in π ($i\precneqq_\pi j$) if all the symbols i occur in π to the left of each symbol j . The symbol i is embedded in the symbol j in π ($i\subsetneqq_\pi j$) if all the symbols i are enclosed in π between two adjoining occurrences of the symbol j . If each pair of symbols is related either by the precedence relation or by the embedding relation in π , the permutation π and the schedule $R=\langle\pi,t\rangle$ for any t are said to be structured. The relation $\precneqq_\pi$ and $\subsetneqq_\pi$ are described by the graphs $\Gamma_\prec(R)$ and $\Gamma_\subset(R)$.

Here we have the following result.

THEOREM 1. Let $\Phi=\{\varphi_i\}_1^n$ be a structured function set with respect to the function H . Then for any integers $p_1,\ldots,p_n$ and any integer point $t\in(-\infty,+\infty)$ there is a structured schedule $R_{opt}(t)=\langle\pi_{opt}(t),t\rangle$ that solves the problem $1|pmtn,t|H,\Phi$. Moreover this schedule has the following property:

$$\begin{gathered}\Gamma_\subset(R_{opt}(t)) \text{ is a subgraph of the graph } G_\subset,\\ G_\prec \text{ is a subgraph of the graph } \Gamma_\prec(R_{opt}(t)).\end{gathered} \tag{3}$$

By Theorem 1, the graphs $G_\prec$ and $G_\subset$ impose certain restrictions on the relative position of symbols in optimal structured schedules. Indeed, by Theorem 1, with each point $t\in(-\infty,+\infty)$ is associated a set $\mathcal{R}(t)=\{\langle\pi_{opt}(t),t\rangle\}$ of structured optimal schedules and at each point t there is a schedule $R^*_{opt}\in\mathcal{R}(t)$ such that the set $\{R^*_{opt}(t)\}_{t=-\infty}^{+\infty}$ has the following properties:

if $i\prec j$, then

$$i \underset{R^*_{opt}(t)}{\prec} j \qquad \text{for all } t\in(-\infty,+\infty); \tag{4}$$

if $i \subset j$ then

1) there is an integer $T > 0$ such that

$$j \underset{R^*_{opt}(-t)}{\prec} i \text{ and } i \underset{R^*_{opt}(t)}{\prec} j \text{ for } t > T; \qquad (5)$$

2) there is a point x_{ij} such that irrespective of $p_1, \dots, p_n$ in every $R^*_{opt}(t)$ the symbol j lies to the left of the symbol i if their coordinates are not greater than x_{ij} and the symbol j lies to the right of the symbol i if their coordinates are greater than x_{ij}

Clearly, the point $\bar{x}(i,j)$ defined above (see Condition A) may be used as the point x_{ij}. It is easily seen that the converse theorem is also true.

THEOREM 2. Suppose that for some $H, \Phi = \{\varphi_i\}_1^n$ the problem $1|pmtn|H,\Phi$ for any $t \in (-\infty, +\infty)$ and any $p_1, \dots, p_n$ has the structured solution $R^*_{opt}(t)$. Also suppose that for any pair of symbols $i, j \in \mathcal{T}$, irrespective of $p_1, \dots, p_n$, either (4) or (5) holds. Then the function f defined by equality (2) satisfies conditions A and B and, moreover, (4) implies $i \prec j$ and $i \subset j$ implies (5), while the set Φ is structured with respect to H.

Now consider two special cases of the problem $1|pmtn, t|H,\Phi$ with special definition of the functions $\varphi_1, \dots, \varphi_n$ and H. Clearly, the choice of the function H imposes certain restrictions on the set Φ.

First let $H = H_1(x_1, \dots, x_m) = \sum_1^m x_i$. Then the problem $1|pmtn, t|H;\Phi$ will be denoted $1|pmtn, t|\sum \varphi_i$. We can easily obtain the following propositions.

Proposition 1. The relation $i \prec j$ (or $\varphi_i \prec \varphi_j$) is equivalent to the relation $\Delta\varphi_i(x) \geq \Delta\varphi_j(x)$ for all $x \in (-\infty, +\infty)$. (in what follows we use the notation $\Delta\varphi(x) = \varphi(x+1) - \varphi(x)$)

Proposition 2. The relation $i \subset j$ (or $\varphi_i \subset \varphi_j$) is equivalent to the difference $\Delta\varphi_i - \Delta\varphi_j$ changing its sign once from $-$ to $+$ as x increases from $-\infty$ to $+\infty$.

We thus have the following proposition.

Proposition 3. The problem $1|pmtn, t|\sum \varphi_i$ at any point t has a structured schedule $R_{opt}(t)$ satisfying (3) if for any pair of functions $\varphi_i, \varphi_j \in \Phi$ the difference $\Delta\varphi_i(x) - \Delta\varphi_j(x)$ changes its sign at most once as x increases from $-\infty$ to $+\infty$.

The problem $1|pmtn, t|\sum \varphi_i$ for an arbitrary function set $\Phi = \{\varphi_i\}_1^n$ is an assignment problem and therefore may be solved by a $O(p^3)$ algorithm. Here we are dealing with an algorithm that is applicable only to structured sets $\Phi = \{\varphi_i\}_1^n$ generating a schedule with $\leq n-1$ preempts. This is a $O(np)$ algorithm. Moreover, this algorithm may be used to minimize $f(R)$ on some interval $[T_1, T_2]$ i.e., to find the point $\bar{t} \in [T_1, T_2]$ and the permutation $\pi_{opt}(\bar{t})$ such that

$$f(\langle \pi_{opt}(\bar{t}), \bar{t} \rangle) \leq f(\langle \pi, t \rangle),$$

where $t \in [T_1, T_2]$. We denote this problem by $1|pmtn, [T_1, T_2]|H,\Phi$. This problem can be solved by a $O((p + T_2 - T_1)n)$ algorithm.

Now consider the function $H = H_2(x_1, \dots, x_m) = \max_{i=1,\dots,m} x_i$ and the problem $1|pmtn, t|H_2,\Phi$, which is henceforth denoted by $1|pmtn, t|\max \varphi_i$.

Proposition 4. The condition $i \prec j$ (or $\varphi_i \prec \varphi_j$) with respect to the function H_2 is equivalent to one of the following three conditions for the functions φ_i and φ_j :

1) φ_i is nondecreasing and $\varphi_i(x)-\varphi_j(x) \geqslant 0$ for $x \in (-\infty,+\infty)$ or

2) φ_j is nonincreasing and $\varphi_j(x)-\varphi_i(x) \geqslant 0$ for $x \in (-\infty,+\infty)$ or

3) there is a point $Z(i,j)$ such that for any $x \leqslant Z(i,j)$ the function φ_j is nonincreasing and $\varphi_j(x)-\varphi_i(x) \geqslant 0$, and for any $x > Z(i,j)$ the function φ_i is nondecreasing and $\varphi_i(x)-\varphi_j(x) \geqslant 0$.

The conditions characterizing the embedding relation $i \subset j$ (or $\varphi_i \subset \varphi_j$) with respect to the function on H_2 easily follow from the definitions.

We now proceed to describe the successive transitions algorithm that solves the problems $1|pmtn, t|H,\Phi$ and to estimate its complexity. Although this algorithm may be applied to any H and Φ, its optimality is only a matter of conjecture unless additional constraints are imposed on the function H and the set Φ. The proof of optimality of this algorithm was obtained by the authors only for the functions H_1 and H_2 defined above.

First introduce the operator O_+ which transforms the optimal structured solution $\langle \pi_{opt}(t), t \rangle$ of the problem $1|pmtn,t|H,\Phi$ with the beginning at the point t into an optimal structured solution $\langle \pi_{opt}(t+1), t+1 \rangle$ of the problem $1|pmtn,t+1|H,\Phi$ with the beginning at the point $t+1$. Thus, if the initial schedule at point t_0 is known, this operator makes it possible to move from the initial optimal structured schedule $R_0=\langle \pi_{opt}(t_0), t_0 \rangle$ $(t_0 < t)$ to the optimal structured schedule $R(t)=\langle \pi_{opt}(t), t \rangle$, solving the problem $1|pmtn,t|H,\Phi$. This transition can be effected by repeated application of the operator O_+ to R_0.

In order to gain some insight into the operator O_+, we introduce the following concepts and notation.

Let π be a structured permutation, such that each symbol i occurs in π precisely p_i times. Denote by π_i the part of the permutation π that starts with the leftmost occurrence of the symbol i and ends with the rightmost occurrence of the symbol i. Then the permutation π has the form

$$\pi = \pi_{j_1}\pi_{j_2}\ldots\pi_{j_m} \quad (j_k \in \mathcal{T})$$

for some $j_1, j_2, \ldots, j_m$. For convenience, let

$$\pi = \pi_1\pi_2\ldots\pi_m \quad (m \leqslant n).$$

Consider the set S of all the subsequences of the sequence $1,2,\ldots,m$ starting with 1. For any schedule $\langle \pi, t \rangle$ and every subsequence $s \in S$, $s=1,k_1,\ldots,k_r$, consider the schedule $\langle \Pi(\pi,s), t+1 \rangle$, where the permutation $\Pi(\pi,s)$ is obtained from the permutation π by the following rules (see Fig. 1):

1) eliminate the first occurrence of the symbol 1 from π ;

2) replace the leftmost occurrence of the symbol k_j with the symbol k_{j-1} $(j=1,\ldots,r)$, $k_0=1$;

3) place the symbol k_r at the end of the permutation $\Pi(\pi,s)$.

Clearly $\langle \Pi(\pi,s), t+1 \rangle$ is a structured schedule. We can now state the following conjecture.

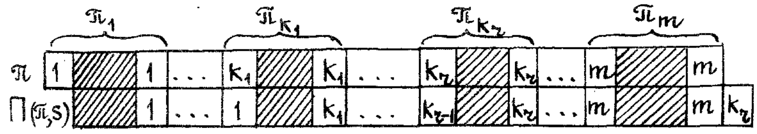

Fig. 1

Conjecture C. If the schedule $\langle \pi_{opt}, t \rangle$ is a structured solution of the problem $1|pmtn, t|H,\Phi$ beginning at the point t, then the set of structured solutions $\{\langle \Pi(\pi_{opt}, s), t+1\rangle\}_{s\in S}$ contains a schedule that solves the problem $1|pmtn, t+1|H,\Phi$, i.e., an optimal schedule beginning at the point $t+1$.

By Conjecture C, the operator O_+ selecting a minimum-penalty schedule from the set of schedules $\{\langle \Pi(\pi_{opt}, s), t+1\rangle\}_{s\in S}$ is optimal.

In [9, 15], the operator O_+ is considered for the problems $1|pmtn, t|\Sigma\varphi_i$ and $1|pmtn, t|max\,\varphi_i$. In [20], these problems were generalized and the class of problems $1|pmtn, t|H,\Phi$ was considered. Under certain assumptions concerning H and Φ. Conjecture C was proved and a O_+ algorithm was obtained for the operator $O(n)$.

Let us now see how to obtain the initial schedule R_o, i.e., the solution of the problem $1|pmtn, t_o|H,\Phi$ at some initial point $t_o, t_o < t$. Given an optimal schedule $R_o = R_{opt}(t_o)$, we can repeatedly apply the operator O_+ and so move to the point t (or to the interval $[T_1, T_2]$ if we are solving the problem $1|pmtn, [T_1,T_2]|H,\Phi$) and thus obtain a solution of the problem $1|pmtn, t|H,\Phi$ ($1|pmtn, [T_1,T_2]|H,\Phi$ respectively). The solution of the problem $1|pmtn, t|H,\Phi$ obviously is entirely determined by the values of the functions $\varphi_i\in\Phi$ on the interval $[t, t+p-1]$ (or on the interval $[T_1, T_2+p-1]$).

Thus, without loss of generality, we may claim that all the points $\bar{x}(i,j)\in[t, t+p-1]$ ($[T_1, T_2+p-1]$ respectively). Then the problem $1|pmtn, t-p|H,\Phi$ has a completely ordered function set Φ on the interval $[t-p, t]$ and therefore it has a trivial solution that sequences the jobs according to this order. We will use this solution $R_{opt} = \langle \pi_o, t-p\rangle$ as the initial solution, setting $t_o = t-p$.

Hence it follows that the time complexity of the succesive transitions algorithm for the problem $1|pmtn, t|H,\Phi$ is $O(np)$, and for the problem $1|pmtn, [T_1,T_2]|H,\Phi$ it is $O(n(p+T_2-T_1))$.

Finally, note the following point. In considering the problem $1|pmtn, t|H,\Phi$ we did not assume regular penalty functions: these are not necessarily nondecreasing functions in this case. Therefore, it is meaningful to consider the problem with idle times $1|pmtn, t, k\ times|H,\Phi$ (the number of unit idle-time "operations" clearly must be specified). The proposed method is also applicable to this problem, with complexity $O(n(p+k))$.

Recall that in all these cases there are at most $(n-1)$ preempts in the schedule.

1.4. Minimizing the Maximum Penalty

The problem $1||max\,\varphi_i$ of minimizing the maximum penalty has attracted the attention of numerous authors.

Statement of the Problem. Consider n jobs to be executed on a single machine. Each job i is characterized by its own arrival time τ_i and execution time p_i. The sequence of jobs is given. With each job i is associated a monotone nondecreasing penalty function φ_i. If the job i terminates at time C_i, it carries the penalty $\varphi_i(C_i)$. The problem involves finding a schedule that minimizes $\varphi_{max} = max\{\varphi_i(C_i)\}$ and satisfies the given job sequencing.

If all the arrival times are equal, there is no advantage to preemption. For this case Lawler's well-known algorithm [32] will produce a solution in time $O(n^2)$.

In case of arbitrary arrival times, the problem with preempting is still solved in time $O(n^2)$ by a certain modification of Lawler's algorithm [23]. Note that the nonpreemptive problem is NP-hard in the strong sense even without precedence constraints [39].

With regard to nonregular penalty functions φ_i this problem was investigated by several authors [7, 45] both in the preemptive and nonpreemptive framework.

Let us now consider the solution of our problem for one class of nonregular functions $\{\varphi_i\}$ described in [9]. Here we can apply the ideas of 1.3, but the algorithm exploits the specific features of the problem. We assume that with each job j is associated a downward quasiconvex penalty function $\varphi_j(t)$, so that there exists a point t_j such that the function $\varphi_j(t)$ is nonincreasing on the interval $(-\infty, t_j]$ and nondecreasing on the interval $[t_j, +\infty)$. We assume that the jobs are independent and accessible at any time. Consider the problem $1|pmtn, t|max\,\varphi_i$. Using the properties of the function φ_j we can calculate the penalty of job j in the schedule R using only the starting time S_j and the completion time C_j of the job j in the schedule R. Thus, $\varphi_{max}(R) = \max_{j=1,\dots,n}(\varphi_j(S_j), \varphi_j(C_j))$. Clearly, with a nonregular penalty function, preemption may have certain advantages. In what follows, we only consider preemption at integral points, as in 1.3.

If the set of downward quasiconvex penalty functions $\Phi = \{\varphi_j\}$ is a structured set, there is a $O(pn)$ algorithm solving the problem $1|pmtn, t|max\,\varphi_i$ at a fixed point t [9]. We now briefly describe this algorithm, after first introducing some definitions.

For any function $\varphi_j \in \Phi$ let $\{\varphi_{j_\ell}\}_{\ell=1}^{s_j}$ denote the set of all the functions from Φ that precede the function $\varphi_j \in \Phi$ according to the succession relation $\prec$ in the structured set and $p_{pred} = \sum_{\ell=1}^{s_j} p_{j\ell}$. Furthermore, let $\{\varphi_{j\tau}\}_{\tau=1}^{u_j}$ denote the set of all the functions from Φ that succeed the function φ_j, $p_{suc} = \sum_{\tau=1}^{u_j} p_{j\tau}$. For any job j, we define the feasible zone in the structured scheduled beginning at the point t, as the time interval $[t+p_{pred}, t+p-1-p_{suc}]$.

The embedding relation on the structured set Φ is defined by the graph G_C, whose vertices (functions from Φ or jobs from $\mathcal{T}$) are sequenced by levels in the following way. Let the first level contain all the functions from Φ that are not embedded in any function from Φ. These functions are completely ordered by the succession relation $\prec$. Sequence then in this order. To the second level assign all the functions from Φ that are embedded only in first-level functions. Sequence the second-level vertices in the order $\prec$, and so on. Thus, the last level contains all the functions in which no other function from Φ is embedded. Let the graph G_C contain k levels.

<u>Algorithm.</u> In the interval $[t, t+p-1]$, find the optimal position (minimizing the maximum penalty) of all the k-th level jobs of the graph G_C in the order $\prec$ in their feasible

zones. Then find the optimal position of the (k − 1)-th level jobs in their feasible zones in places not filled by k-th level jobs, and so on. The first level fills the empty places in the order $\prec$.

The optimality of this $O(np)$ algorithm is proved in [15].

Here it is apparently hard to eliminate P from the complexity bound of the algorithm, since even finding the minimum point for a single job with execution time p_j requires $O(p_j)$ evaluations of the function φ_j for an arbitrary downward quasiconvex penalty function φ_j.

2. NONTRADITIONAL PROBLEMS

2.1. Minimizing Job Costs

In some recent publications [35, 36, 15, 17, 18, 19], the objective function is the integrated job cost. Consider the following problem.

Statement of the Problem. Suppose that n jobs are to be executed on a single machine. For any job i, a time-dependent cost density $c_i(t)$ is given. Then the integrated cost of job i in the interval $[t-p_i, t]$, denoted by $C_i(t)$, is calculated as $C_i(t)=\int_{t-p_i}^{t} c_i(u)\,du$,

where p_i is the execution time of job i. Sequence the jobs so as to minimize the total cost of all the scheduled jobs.

In [36] this problem is considered for the case when $c_j(u)=a_j g(u)+b_j$, where $g(u)$ is a monotone nondecreasing (nonincreasing) function in the interval $[0, \sum_{i=1}^{n} p_i]$. Then the optimal sequence is obtained by sequencing the jobs in the order of nonincreasing (nondecreasing) a_j.

The objective function of the problems considered in [15, 17, 18, 19] is also the integrated cost. These problems were briefly described in 1.3. The functions $\varphi_i(t)$ $(i=1,\ldots,n)$ corresponding to the function $H=H_1=\sum x_i$ may be interpreted as the cost density of job i, and the objective function $\sum_{j=1}^{p}\varphi_{i_j}(t+j)$ as the integrated cost of the jobs in the schedule $R=\langle i_1 i_2 \ldots i_p, t\rangle$ beginning at the point t.

It is easily seen that infinite division of the interval $[t, t+p]$ replaces the cost summation of the schedule job i by the integral $\int_{\Omega_i}\varphi_i(u)\,du$, where Ω_i is a union of the nonintersecting intervals in which job i is executed.

Then the objective function takes the form

$$\sum_{i=1}^{n}\int_{\Omega_i}\varphi_i(u)\,du.$$

We can easily generalize the notion of structured schedule to this case, and the existence theorem of optimal structured schedule takes the following form.

THEOREM 3. If the set of functions $\varphi_i(t)$ is such that for each pair $i, j \in \mathcal{T}$ the difference of the derivatives $\varphi_i'(u)-\varphi_j'(u)$ changes its sign at most once on the interval $[t, t+p]$, then at point t there is an optimal structured schedule that solves the problem $1\,|\,pmtn,\ t\,|\sum_{i=1}^{n}\int_{\Omega_i}\varphi_i(u)\,du$. The number of preempts in this schedule is $\leqslant n-1$

Note that the functions $\varphi_i(u)$ are not necessarily monotone. This theorem, generalizing the results of [36], easily leads to the following.

THEOREM 4. Let the functions $\varphi_i(u)$ be such that at any point $u \in [t, t+p]$

$$\varphi'_{i_1}(u) \geqslant \varphi'_{i_2}(u) \geqslant \ldots \geqslant \varphi'_{i_n}(u).$$

Then job sequenching in the order $i_1, i_2, \ldots i_n$ is optimal for nonpreemptive problems $1|0|\sum_{i=1}^{n} \int_{\Omega_i} \varphi_i(u)\,du$ at the point t regardless of the execution times $p_1, \ldots, p_n$.

Moreover, if, for instance, the functions $\varphi_1(u)$ and $\varphi_2(u)$ are such that the difference $\varphi'_1(u) - \varphi'_2(u)$ changes its sign in the interval $[t, t+p]$, the values of the functions $\varphi_i(u)$ $(i=3,\ldots,n)$ can be easily modified so that for some values of $p_1, \ldots, p_n$ job necessarily precedes job 2 in the optimal schedule beginning at the point t, while for other values of $p_1, \ldots, p_n$ job 2 precedes job 1 at the same point t.

2.2. Processor Sharing Problems

New important problems closely related with processor sharing have been recently considered by some authors. Processor sharing usually refers to a scheduling strategy in which several (and possibly all) jobs may receive the same processing time on the processor, while the individual processing speed depends on the state of the system.

Strategies of this kind have been of considerable interest for over ten years now, since the introduction of time sharing computers. One of the simplest and best known strategies allocates the entire computing power of the system equally among all the jobs.

Some recent publications consider processor sharing strategies in which the execution speed of any job is an arbitrary positive function of the job parameters. Characteristic features of this strategy are the following: 1) time is divided into equal discrete unit intervals; 2) each unit interval is shared by jobs according to their system priority, so that each job is allocated a fraction of time proportional to its priority; 3) a discrete system of priorities is introduced, i.e., priorities may only take values from a given finite set; 4) the number of jobs executable in one unit interval is bounded.

This naturally leads to a priority assignment problem, with the priorities treated as a function of time and job parameters satisfying a certain given optimality criterion.

This problem may be considered as sequencing of jobs on a single machine which, contrary to the traditional framework, is allowed to execute more than one job at any given time.

On the other hand, the priority assignment problem is evidently reducible to the m-processor job assignment problem in which another traditional restriction is lifted, namely that no job may execute simultaneously on two processors. This traditional condition is replaced by another: a single job may not execute simultaneously on more than a fixed number of machines. Hence it follows that a job may be conveniently considered as a set of independent elementary operations, several of which may execute concurrently.

It is in this second setting that the problem was considered in [10, 11, 31].

Let us sharpen the problem following the work of Maksimenkov [10, 11].

Statement of the Problem. Suppose given m processors and n jobs. Each job i has its weight λ_i arrival time τ_i, and due date d_i, p_i is the number of unit-length independent operations comprising the job i , s_i is the maximal number of job i operations that can be executed concurrently on different processors. Find a schedule of executing the job on m processors such that the sum $\sum_{i=1}^{n} \lambda_i \omega_i$ is minimized, where ω_i is the number of job i , operations that fall outside the target interval $[r_i, d_i]$ of job i.

This problem is a generalization of the well-known problem $m|p_i=1, r_i, d_i|\sum w_i U_i$. A polynomial algorithm solving this problem was proposed in [11]. This algorithm is based on the following idea. First sequence the jobs by decreasing λ_i. The algorithm consists of sequential steps assigning the job operations to the processors in this order. If the interval $[r_i, d_i]$ of the current job i is already filled, the next operation of job i cannot be assigned without delay. An attempt is made to fit this operation in its target interval by performing a chain of permutations of the previously assigned operations within their intervals. If no such chain is found, the job i operation cannot be fitted in its target interval $[r_i, d_i]$ and the algorithm moves to the next job. The algorithm stops when all the jobs have been scanned.

Scheduling problems in a similar setting are considered in [2]. The results give sufficient conditions for the existence of schedules that satisfy the due date on a homogeneous system of machines.

The inverse problem is stated in [31]. Given the same input data, find the minimal number of processors sufficent to execute the jobs in a given time interval. It is easily seen that Maksimenkov's algorithm also solves this problem, but in [31] it is considered under additional constraints on job precedence.

The optimality criterion considered in [18, 19] makes it possible to assign job i operations to the left and to the right of the target interval $[r_i, d_i]$, but then job i carries a certain penalty if it executes outside the target interval. The problem in this case reduces to single-processor scheduling.

2.3. The Problem of Satisfying Start and Due Date

In this section we consider the following problem.

STATEMENT OF THE PROBLEM. Suppose given one machine and n jobs $\mathcal{T} = \{1, \ldots, n\}$; for each job i the ready date r_i , the due date d_i, and the execution time p_i are given $(r_i + p_i \leqslant d_i)$ The machine may not execute two jobs simultaneously. Find a schedule in which job i is executed in its target interval $[r_i, d_i]$.

We know that in case of arbitrary job durations p_i without preemption, this problem is NP-hard [39]. It has been shown that for $p_i \in \{1, 2\}$, $r_i = 0$, $d_i = D$ the existence problem of such a schedule is also NP-hard [48]. Bratley et al. [25] proposed a branch-and-bound method for minimizing the nonpreemptive schedule length subject to start and due dates $[r_i, d_i]$ if such a schedule exists. A computational experiment shows that the probability of solving problems with large n (up to 100) by this method in an acceptable time is fairly high. For nonpreemptive jobs of unit length, when r_i, d_i are arbitrary real numbers a $O(n^2 \lg n)$ polynomial algorithm for the solution of this problem was first obtained by Barbara Simons in [46], and later a $O(n \lg n)$ algorithm was proposed by Garey, Johnson, Simons, and Tarjan [27].

The problem with arbitrary durations is substantially simplified in the preemptive setting. For the single machine case, Horn [29] proposed a $O(n \lg n)$ algorithm for solving the problem with start and due date constraints, and the schedule has at most (n — 1) preempts.

If the precedence relation $\prec$ is defined on the job set, the condition $i \prec j$ implies that job j cannot start earlier than the time moment $r_i + p_i$, i.e., we may take $r_i < r_j$. Similarly $d_i < d_j$. Thus the condition $i \prec j$ induces a precedence condition on the target intervals $[r_i, d_i] \prec [r_j, d_j]$. Horn's algorithm [29] also solves the problem with start and due date constraints while preserving the given job precedence relations (if the target intervals are appropriately adjusted to take account of the job precedence relations), since it constructs a schedule sequencing the jobs in a way that preserves the interval precedence relation (and therefore the job precedence relation). The overall complexity of this algorithm is $O(n^2)$ and the number of preempts is at most (n — 1).

CONCLUSIONS

Let us now formulate some directions of future research in the areas of interest covered in this article.

In Sec. 1 we described three different interchange relations induced by the functions f on jobs or chains of jobs. Of particular interest is the existence of other interchange relations leading to effective solution algorithms.

It is important to investigate the problems $1|prec|f$, in which the function f defines two, three, or more "generalized interchange intervals," analogous to the two generalized succession intervals studied in Subsec. 2.3.

It is relevant to investigate the existence of other functions H in the problem $1|pmtn, t|H, \Phi$ in addition to $H_1 = \sum_i x_i$ and $H_2 = \max_i x_i$. Since the structured conditions are very strong, we expect that there are no other functions for which the structured set Φ exists. In any case, any definite answer to this question (whether affirmative or negative) would be of extreme interest.

With regard to the problems reviewed in Sec. 2, they arose fairly recently and they should naturally be considered for all the other known optimality criteria. We expect that natural statement of the problems will lead to effective solutions.

LITERATURE CTIED

1. V. Ya. Burdyuk and V. N. Reva, "An optimization method for functionals on permutations subject to constraints," Kibernetika, No. 1, 99-103 (1980).
2. V. G. Vizing, "Schedules observing job due dates," Kibernetika, No. 1, 128-135 (1981).
3. V. S. Gordon and E. P. Chegolina, "Single-stage scheduling problem with a tree-ordered job set," Izv. Akad. Nauk BSSR, Fiz.-Mat. Ser., No. 4, 36-40 (1977).
4. V. S. Gordon and Ya. M. Shafranskii, "Optimal sequencing for series—parallel constraints," Dokl. Akad. Nauk BSSR, No. 3, 244-247 (1978).
5. V. S. Gordon and Ya. M. Shafranskii, "Function minimization on the set of permutations of partially ordered elements," Vestsi Akad. Nauk BSSR, Fiz.-Mat. Ser., No. 2, 122-124 (1979).
6. A. M. Danil'chenko and A. V. Panishev, "Dichotomic search for a solution of one sequencing problem," Kibernetika, No. 2, 118-121 (1980).
7. V. V. Kotsubo, "Scheduling on a single machine with job start and due times," Vestsi Akad. Nauk BSSR, Fiz.-Mat. Ser., No. 3 50-55 (1970).

8. N. B. Lebedinskaya, "Minimizing the maximum deviation for preemptive jobs," Zap. Nauchn. Sem. Leningr. Otd. Mat. Inst., 80, 117-124 (1978).
9. N. B. Lebedinskaya, "Minimizing the maximum penalty for preemptive jobs," Zap. Nauchn. Sem. Leningr. Otd. Mat. Inst, 102, 61-67 (1980).
10. A. V. Maksimenkov, "Scheduling of computer resources with minimal disproportions," Avtomat. Vychisl. Tekh., No. 4, 69-75 (1978).
11. A. V. Maksimenkov, "Scheduling with constraints on resource consumption rate," Kibernetika, No. 6, 91-95 (1979).
12. V. S. Tanaev, "Optimization of functions recursively defined on permutation sets," Vestsi Akad. Nauk BSSR, Fiz.-Mat. Ser., No. 3, 27-30 (1977).
13. V. S. Tanaev and V. V. Shkurba, Introduction to the Theory of Scheduling [in Russian], Moscow (1975).
14. K. V. Shakhbazyan, "On structured solutions in scheduling problems," Zap. Nauchn. Sem. Leningr. Otd. Mat. Inst., 80, 239-248 (1978).
15. K. V. Shakhbazyan, "Sequencing a structured set of jobs to minimize the total penalty," Zap. Nauchn. Sem. Leningr. Otd. Mat. Inst. 90, 229-264 (1979).
16. K. V. Shakhbazyan, "Structured solutions in scheduling theory," Zap. Nauchn. Sem. Leningr. Otd. Mat. Inst., 102, 139-146 (1980).
17. K. V. Shakhbazyan, "On scheduling problems of type $n|1|\sum c_i(t)$," Zap. Nauchn. Sem. Leningr. Otd. Mat. Inst. 102, 147-155 (1980).
18. K. V. Shakhbazyan and N. B. Lebedinskaya, "Minimizing the total penalty in parallel sequencies of n independent jobs," Dokl. Akad. Nauk SSSR, 237, No. 4, 790-792 (1977).
19. K. V. Shakhbazyan and N. B. Lebedinskaya, "Optimal preemptive schedules for independent jobs in a N-server queuing system," Zap. Nauchn. Sem. Leningr. Otd. Mat. Inst., 70, 205-231 (1977).
20. K. V. Shakhbazyan and N. B. Lebedinskaya, "Structural solutions for scheduling problems," Preprint Leningr. Otd. Mat. Inst., E-3-81, Leningrad (1981).
21. H. M. Abdel-Wahab and T. Kameda, "Scheduling to minimizing maximum cumulative cost subject to series-parallel precedence constraints," Oper. Res., 26, 141-158 (1978).
22. D. L. Adolphson, "Single-machine job sequencing with precedence constraints," SIAM J. Comput., 6, No. 1, 40-54 (1977).
23. K. R. Baker, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan, "Preemptive scheduling of a single machine to minimize maximum cost subject to release dates and precedence constraints," Tech. Rep. 802810, Erasmus Univ., Rotterdam.
24. R. Bellman, "Mathematical aspects of scheduling theory," J. Soc. Industr. Appl. Math., 4, No. 3, 168-205 (1956).
25. P. Bratley, M. Florian, and P. Robillard, "Scheduling with earliest start and due date constraints," Nav. Res. Logist. Q., 18, 511-517 (1971).
26. R. W. Conway, W. L. Maxwell, and L. W. Miller, Theory of Scheduling, Addison-Wesley, Reading Mass. (1967).
27. M. R. Garey, D. S. Johnson, B. B. Simons, and R. E. Tarjan, "Scheduling unit-time tasks with arbitrary release times and deadlines," J. Comput. System Sci. (forthcoming).
28. R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan, "Optimization and approximation is deterministic sequencing and scheduling: a survey," Ann. Discrete Math., 5, 287-326 (1979).
29. W. Horn, "Some simple scheduling algorithms," Naval Res. Logist. Q., 21, 177-185 (1974).
30. S. M. Johnson, "Optimal two- and three-stage production schedules with setup times included," Naval Res. Logist. Q., 1, 61-68 (1954).
31. P. Kas, "On a special type job sequencing problem," Survey Math. Programming, Proc. 9th Inst. Math. Programming Symp., Budapest, Vol. 2 (1976), pp. 279-283.
32. E. L. Lawler, "Optimal sequencing of a single machine subject to precedence constraints," Manag. Sci., 19, 544-546 (1973).
33. E. L. Lawler, "Optimal sequencing of jobs subject to series-parallel precedence constraints," Math. Cent. Afd. Math. Beslisk. B.W.N., No. 54 (1975).
34. E. L. Lawler, "Sequencing problems with series-parallel precedence constraints," Proc. Conf. on Combinatorial Optimization, Urbino, Italy (1979).
35. E. L. Lawler and J. Labetoulle, "On preemptive scheduling of unrelated parallel processors by linear programming," J. ACM, 25, 612-619 (1978).
36. E. L. Lawler and B. D. Sivazlian, "Minimization of time-varying costs in single-machine scheduling," Oper. Res., 26, 263-269 (1978).
37. J. K. Lenstra, Sequencing by Enumerative Methods, Math. Centre Tracts 69, Math. Centrum, Amsterdam.

38. J. K. Lenstra and A. H. G. Rinnooy Kan, "Complexity results for scheduling chain on a single machine," Econometric Institute, Erasmum Univ., Rotterdam, Rep. 7907/0 (1979).
39. J. K. Lenstra, A. H. G. Rinnooy Kan, and P. Brucker, "Complexity of machine scheduling problems," Ann. Discrete Math., 1, 343-362 (1977).
40. C. L. Monma, "The two-machine maximum flow time problem with series-parallel precedence constraints: an algorithm and extensions," Oper. Res., 27, 792-798 (1979).
41. C. L. Monma, "Sequencing to minimize the maximum job cost," Oper. Res., 28, 942-951 (1980).
42. C. L. Monma and J. Sidney, "Sequencing with series-parallel precedence constraints," Math. Oper. Res., 4, 215-224 (1979).
43. A. H. G. Rinnooy Kan, Machine Scheduling Problems: Classification, Complexity and Computations, Nijholf, The Hague (1976).
44. M. E. Rothkopf, "Scheduling independent tasks on parallel processors," Manag. Sci., 12, 437-447 (1966).
45. J. B. Sidney, "Optimal single-machine scheduling with earliness and tardiness penalties," Oper. Res., 25, 62-69 (1977).
46. B. Simons, "A fast algorithm for single processor scheduling," 19th Annual Symp. on Foundations of Computer Sci. IEEE Computer Soc., Long Beach. Calif. (1978), pp. 246-252.
47. W. E. Smith, "Various optimizers for single-stage production," Naval Res. Logistic Quart., 3, 59-66 (1956).
48. J. D. Ullman, "NP-complete scheduling problems," J. Comput. Syst. Sci., 10, 384-393 (1975).