strained so that $r_j + p_j \leqslant d_j$. In 20-25% of the cases, the algorithms produced optimal schedules. In other cases, the error of the criterion did not exceed 10% of the optimal value. In the first problem, this error did not exceed 6%. All three algorithms solve the problems in time $O(n^2)$.

In conclusion, I would like to thank R. F. Khabibullin for suggesting the problem and for constant attention.

LITERATURE CITED

1. R. W. Conway, W. L. Maxwell, and L. W. Miller, Theory of Schedules [Russian translation], Nauka, Moscow (1975).
2. V. S. Tanaev and V. V. Shkurban, An Introduction to Scheduling Theory [in Russian], Nauka, Moscow (1975).
3. A. H. G. Rinnooy Kan, B. I. Lageweg, and J. K. Lenstra, Minimizing Total Costs in One-Machine Scheduling (1975).
4. Milan Vlach, "One-processor scheduling with nonuniform ready times," Math. Oper. Stat., Ser. Optim., 10 (1979).
5. K. R. Baker and L. E. Schrage, "Finding an optimal sequence by dynamic programming," Oper. Res., 26 (1978).
6. R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan, "Optimization and approximation in deterministic sequencing and scheduling: A survey," in: Annals of Discrete Mathematics, Vol. 5, North-Holland, Amsterdam (1979), pp. 287-326.
7. J. K. Lenstra, A. H. G. Rinnooy Kan, and P. Brucker, "Complexity of machine scheduling problems," in: Annals of Discrete Mathematics, Vol. 4, North-Holland, Amsterdam (1977), pp. 343-362.

DUAL OF THE MAXIMUM COST MINIMIZATION PROBLEM

A. A. Lazarev

The note examines a one-machine scheduling problem. A problem which in a certain sense is the dual of the original problem is formulated and solved. From the dual solution we obtain a bound on the optimal value of the original problem.

Consider one machine processing n jobs. The jobs are indexed 1 to n, and each job j is determined by its release date r_j, the processing time p_j, and some nondecreasing cost function $f_j(t)$, which characterizes the processing cost of the job j if it is completed at the moment t. The machine may process at most one job at a time; preemption is not allowed.

Let Ω be the set of all permutations of the first n natural numbers. We use ω to denote a particular permutation,

$$\omega = (j_1, j_2, \ldots, j_n),$$

where j_k, $k = 1, 2, \ldots, n$ is the index of the job which is processed k-th in the sequence. We consider only schedules without unforced idleness, i.e., idleness of the machine is not allowed if the next job is waiting to be processed. By t_{j_k} we denote the completion time

Translated from Issledovaniya po Prikladnoi Matematike, No. 10, pp. 111-113, 1984.

 0090-4104/89/4405-0642 $12.50

of job j_k in the schedule ω. For convenience, we also introduce the following notation allowing for the place of a job in the schedule ω:

$$\tau_k(\omega) = t_{j_k}; \quad \varphi_k(\omega) = f_{j_k}(t_{j_k}).$$

The maximum cost for the schedule ω is defined by

$$\max_{k=\overline{1,n}} \varphi_k(\omega) \equiv \max_{k=\overline{1,n}} f_{j_k}(t_{j_k}).$$

We seek a schedule $\omega \in \Omega$ on which the maximum cost is minimized.

The optimal value of the criterion is denoted by

$$\mu^* = \min_{\omega\in\Omega} \max_{k=\overline{1,n}} \varphi_k(\omega).$$

The following problem is the dual of the original problem in a certain sense: find

$$\nu^* = \max_{k=\overline{1,n}} \min_{\omega\in\Omega} \varphi_k(\omega). \tag{1}$$

Let

$$\nu_k = \min_{\omega\in\Omega} \varphi_k(\omega), \text{ then } \nu^* = \max_{k=\overline{1,n}} \nu_k.$$

<u>LEMMA</u>. Let all f_j be nondecreasing functions. Then for all $k = 1, 2, \dots, n-1$ we have $\nu_n \geqslant \nu_k$, i.e., e. $\nu^* = \nu_n$.

<u>Proof</u> is by contradiction. Assume that there exists k such that $\nu_k > \nu_n$. Let

$$\omega_n = \operatorname{argmin}\{\varphi_n(\omega) : \omega \in \Omega\},$$

i.e., $\nu_n = \varphi_n(\omega_n)$. Let $\omega_n = (j_1, \dots, j_n)$. Construct the schedule

$$\omega'_n = (j_{n-k+1}, \dots, j_n, j_1, \dots, j_{n-k}).$$

In this schedule, the job j_n occupies the k-th place. Since $\tau_n(\omega_n) \geqslant \tau_k(\omega'_n)$ we have $\nu_n = \varphi_n(\omega_n) \geqslant \varphi_k(\omega'_n)$. Since by assumption $\nu_k > \nu_n$, this leads to the inequality $\nu_k > \varphi_k(\omega'_n)$, which is a contradiction. Q.E.D.

The solution of the dual problem (1) thus reduces to finding ν_n. The main part of this problem is finding a schedule with minimum completion time of the (n − 1)-th job.

Let us solve the problem of finding

$$\mu_1 = \min_{\omega\in\Omega} \max_{k=\overline{1,n}} \tau_k(\omega).$$

Clearly,

$$\mu_1 = \min_{\omega\in\Omega} \tau_n(\omega). \tag{2}$$

The solution of the problem (2) is the schedule ω arranged by nondecreasing release dates: $r_{j_1} \leqslant r_{j_2}, \dots, r_{j_n}$.

The following simple algorithm solves the dual problem.

1. Reindex the jobs in the order of release dates:

$$r_1 \leqslant r_2 \leqslant \dots \leqslant r_n;$$

2. For $k = 1, 2, \dots, n$ construct the schedule

$$\omega_k = (1, 2, \dots, k-1, k+1, \dots, n, k), \; \varepsilon_k = \varphi_n(\omega_k);$$

3.
$$\nu^* = \min_{k=\overline{1,n}} \varepsilon_k.$$

THEOREM. The solution of the dual problem is a lower bound on the optimal value of the criterion of the original problem, i.e., $\mu^* \geqslant \nu^*$.

Proof. Assume the contrary, i.e., $\mu^* < \nu^*$. Let ω^* be an optimal schedule of the original problem. Then
$$\varphi_n(\omega^*) \leqslant \mu^* < \nu^*,$$
which contradicts the definition $\nu^* = \nu_n$.

This bound may be effectively utilized to construct a branch-and-bound scheme for the solution of the original problem and to estimate the error of approximate methods.

LITERATURE CITED

1. R. W. Conway, W. L. Maxwell, and L. W. Miller, Theory of Schedules [Russian translation], Nauka, Moscow (1975).
2. V. S. Tanaev and V. V. Shkurba, An Introduction to Scheduling Theory [in Russian], Nauka, Moscow (1975).

OPTIMIZING THE LOAD OF A MACHINE PROCESSING A COLLECTION OF INDEPENDENT JOBS

R. F. Khabibullin and F. R. Khalitov

In this note, we formulate the optimization problem of ensuring a uniform load for a machine processing a collection of technologically independent jobs. Two classes of approximate algorithms are proposed for the solution of the problem, based on different ideas and both having simple computer implementation.

Consider a manufacturing unit (a plant, a shop, a process) which is required to complete a certain collection of jobs during the planning period, maintaining a uniform load (the load on the manufacturing unit) during this period. The notion of uniform load may have different interpretations, and will be formalized below. We introduce the following assumptions:

1. The machine may process any number of jobs at each moment of time.

2. No preemption of jobs is allowed.

Each job $j = 1, 2, \ldots, n$ is characterized by the following parameters:

r_j – the release date of job j,

p_j – the processing time of job j,

d_j – the due date of job j.

Translated from Issledovaniya po Prikladnoi Matematike, No. 10, pp. 114-120, 1984.

0090-4104/89/4405-0644 $12.50