ISSN 0965-5425, Computational Mathematics and Mathematical Physics, 2009, Vol. 49, No. 2, pp. 373–386. © Pleiades Publishing, Ltd., 2009.
Original Russian Text © A.A. Lazarev, 2009, published in Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki, 2009, Vol. 49, No. 2, pp. 382–396.

Estimates of the Absolute Error and a Scheme for an Approximate Solution to Scheduling Problems

A. A. Lazarev

Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, ul. Profsoyuznaya 65, Moscow, 117997 Russia

e-mail: alexandr.Lazarev@mail.ru

Received December 12, 2007;
in final form, May 26, 2008

Abstract—An approach is proposed for estimating absolute errors and finding approximate solutions to classical NP-hard scheduling problems of minimizing the maximum lateness for one or many machines and makespan is minimized. The concept of a metric (distance) between instances of the problem is introduced. The idea behind the approach is, given the problem instance, to construct another instance for which an optimal or approximate solution can be found at the minimum distance from the initial instance in the metric introduced. Instead of solving the original problem (instance), a set of approximating polynomially/pseudopolynomially solvable problems (instances) are considered, an instance at the minimum distance from the given one is chosen, and the resulting schedule is then applied to the original instance.

DOI: 10.1134/S0965542509020158

Keywords: scheduling theory, minimization of maximum lateness, absolute error estimate, approximate solution.

1. INTRODUCTION

We consider the classical NP-hard scheduling problems $\{1, P, R, Q\}|\text{prec}, r_j|\{L_{\max}, C_{\max}\}$ in the notation [1]. These problems have been intensively studied since the early 1950s. For example, the "simplest" of these problems, $1|r_j|L_{\max}$, was proved to be NP-hard in the strong sense (see [2]). The most effective method for solving these problems is a reduced exhaustive search of instances, specifically, branch-and-bound algorithms. Therefore, an important task is to find approximate solutions and estimate the objective function error.

For the problem $1|r_j|L_{\max}$, a number of polynomially solvable instances were found starting with Jackson's early result [3], where all the jobs are available for processing in one time. The solution is a schedule where the jobs are arranged in nondecreasing order of their due dates (according to the EDD rule). In [4], an iterative version of Jackson's extended rule (IJ) [3] was presented and it was proved that $L_{\max}(\text{IJ})/L^*_{\max} \le 3/2$. In [5], the iterative version was modified and an algorithm (MIJ) was developed that guarantees $L_{\max}(\text{MIJ})/L^*_{\max} \le 4/3$. Moreover, two approximating schemes were suggested in [5] that guarantee that an ε-approximate solution can be determined in $O(n\log n + n(1/\varepsilon)^{O(1/\varepsilon^2)})$ and $O((n/\varepsilon)^{O(1/\varepsilon)})$ operations. An improved approximating scheme running in $O(n + (1/\varepsilon)^{O(1/\varepsilon)})$ time was proposed in [6]. Special cases of the problem $1|\text{prec}; r_j|C_{\max}$, $1|\text{prec}; p_j = p; r_j|L_{\max}$ and $1|\text{prec}; r_j; pmtn|L_{\max}$ with precedence constraints satisfied in the processing of the jobs were discussed in [7–9]. A polynomial-time algorithm (with $O(n^2\log n)$ operations) was proposed in [10] for the special case when the job parameters (the minimum possible start time r_j, the processing time $p_j > 0$, and the due date d_j, $j \in N$) satisfy the constraints $d_j - p_j - A \le r_j \le d_j - A$, where A is a constant $\forall\, j \in N$. A pseudopolynomial algorithm for the NP-hard case when the start times and the due dates are arranged in reverse order ($r_1 \ge \ldots \ge r_n$ and $d_1 \le \ldots \le d_n$) was developed in [11, 12]. The complexity of the algorithm is $O(nP(n + p_{\max}))$ operations, where

$$P = \sum_{j \in N} p_j \quad \text{and} \quad p_{\max} = \max_{j \in N} p_j.$$

In this paper, a two-step method is proposed for finding approximate solutions to the NP-hard scheduling problems $\{1, P, R, Q\}|\text{prec}, r_j|\{L_{\max}, C_{\max}\}$. At the first step, a linear programming problem is solved and the parameters of the initial instance A (the start times, processing times, and due dates of the jobs) are changed so that the resulting instance B (with the precedence graph G being the same for both instances) is a polynomially/pseudopolynomially solvable special case of the original problem. An optimal schedule for the instance B is found at the second step. The optimal schedule π^B for A has the minimum estimate for the absolute error of the objective function (maximum lateness) for all the instances of the solvable class.

It should be noted that a similar approach to finding an absolute error estimate was applied to the one-dimensional knapsack problem in [13]. Specifically, instead of solving the initial instance, an instance with unit weights was constructed and an approximate solution to the original problem was then found by a "greedy" algorithm (in polynomial time).

2. PROBLEMS FOR SEVERAL MACHINES. DEFINITIONS

The task is to process n jobs j, where $j \in N = \{1, 2, \dots, n\}$, on m machines $M = \{1, 2, \dots, m\}$. Interruptions in processing the jobs are not allowed. At every moment of time, each machine can process at most one job. For each job $j \in N$, a possible start time r_j, a processing time $0 \le p_{ji} \le +\infty$ (on the ith machine)[1], and a due date d_j are given. Precedence relations between the jobs are specified as an acyclic directed graph $G \subset N \times N$.

By π_i, we denote a permutation (schedule) of the elements of N processed on the machine M_i, where $i = 1, 2, \dots, m$. The set of jobs in a schedule π_i is denoted by $\{\pi_i\} \subseteq N$, and a schedule for the entire set of jobs $\pi = \bigcup_{i=1}^{m} \pi_i$ is defined as $\{\pi_\alpha\} \cap \{\pi_\beta\} = \varnothing$ if $\alpha \ne \beta$. Naturally, we consider only *admissible* schedules without artificial idle times, i.e., those satisfying the precedence relations. For any acyclic graph G, the set of admissible schedules is not empty. The completion time of a job j on the machine M_i under a schedule π is defined by

$$C_j(\pi) = \max\left\{r_j, \max_{k \in N_j} C_k(\pi), \max_{(k \to j)_{\pi_i}} C_k(\pi_i)\right\} + p_j, \tag{1}$$

where N_j is the set of jobs preceding j in the graph G and $(k \longrightarrow j)_{\pi_i}$ are the jobs processed by M_i before $j \in \{\pi_i\}$ according to the schedule π_i.

By a schedule, we mean the set $S = \{s_j | j \in N_i, i \in M\}$ of start times of the jobs, where $\{N_i | i \in M\}$ form a partition of N[2]. The completion time of $j \in N$ under the schedule S is $C_j(S) = s_j(S) + p_{ji}$, $j \in N_i$. A schedule S is called *admissible* if $r_j \le s_j(S)$, $C_j(S) < \infty$ $\forall j \in N$, $C_j(S) \le s_k(S)$ $\forall (j, k) \in G$.

The problem $P|\text{prec}; r_j|L_{\max}$ under study generalizes a number of NP-hard problems, specifically, $P|\text{intree}, r_j, p_j = 1|C_{\max}$; $P|\text{outtree}, p_j = 1|L_{\max}$ [14]; $P2|\text{chains}|C_{\max}$ [15]; $P \,||\, C_{\max}$ [16]; $1|r_j|L_{\max}$, $P2 \,||\, C_{\max}$ [2]; and $P|\text{prec}, p_j = 1|C_{\max}$ [17], which have polynomially solvable special cases.

In this section, we give the basic notation and definitions used in what follows.

Definition 1. An instance A of the problem is a set $\{G^A, (r_j^A, p_j^A, d_j^A) | j \in N\}$. A schedule minimizing the maximum lateness (an optimal schedule) for this instance is denoted by π^A, i.e.,

$$L_{\max}^A(\pi^A) = \min_{\pi} \max_{j \in N} L_j^A(\pi),$$

where the minimum is over the set of all admissible schedules π. By a schedule, we mean the collection of permutations $\pi = \bigcup_{i=1}^{m} \pi_i$ and the vector of start times $S = \{s_j | j \in N_i, i \in M\}$. Given an instance A, an optimal vector of permutations and an optimal schedule are denoted by π^A and S^A, respectively.

In an early schedule, the processing of each job $j \in N$ begins at the earliest admissible time: at the start time r_j^A, immediately after the machine completes the processing of the previous job, or immediately after the processing of the job preceding the given job j (according to G) is completed.

[1] If $p_{ji} = +\infty$, then the job j cannot be processed by the ith machine.
[2] We assume that $\bigcup_{i \in M} N_i = N$ and $N_i \cap N_l = \varnothing$ for $i \ne l$.

Definition 2. Given an instance A and a schedule π, the extended precedence graph G_π^A is obtained from G^A by adding edges according to the permutations π_i, $i = 1, 2, \ldots, m$.

A schedule π is admissible for an instance A if and only if the extended precedence graph G is acyclic.

Definition 3. Any chain that is a subgraph of G_π^A is called a precedence chain ($\sigma \subset G_\pi^A$).

If $\sigma = (j_1, \ldots, j_k) \subset G_\pi^A$ is a precedence chain, then it follows from (1) that

$$C_{j_k}^A(\pi) \ge r_{j_1}^A + \sum_{j \in \{\sigma\}} p_j^A. \tag{2}$$

Definition 4. A precedence chain $\sigma = \{j_1, \ldots, j_k\}$ is called a delaying chain for the job j_k, which is denoted by $\sigma^*(j_k)$, if the processing of the job j_1 begins at the time r_{j_1} or, for $l = 1, 2, \ldots, k - 1$, the processing of the job j_{l+1} begins at the completion time of the preceding job j_l.

In fact, a delaying chain is a precedence chain for which inequality (2) becomes an equality:

$$C_{j_k}^A(\pi) = r_{j_1}^A + \sum_{j \in \{\sigma^*(j_k)\}} p_j^A. \tag{3}$$

A delaying chain exists for each job, since we consider only schedules without artificial idle times.

Definition 5. An instance $A = \{G^A, (r_j^A, p_j^A, d_j^A) | j \in N\}$ is said to be inverse to an instance $B = \{G^B, (r_j^B, p_j^B, d_j^B) | j \in N\}$ if

$$r_j^A = -d_j^B, \quad p_j^A = p_j^B, \quad d_j^A = -r_j^B \quad \forall j \in N;$$

moreover, $(k \longrightarrow j) \in G^A$ if and only if $(j \longrightarrow k) \in G^B$, i.e., the orientation of the edges in B is opposite: $\overleftarrow{G}^B = \overrightarrow{G}^A$. A permutation $\pi_i' = (j_{n_i}, j_{n_i - 1}, \ldots, j_1)$ is called inverse to the permutation $\pi_i = (j_1, \ldots, j_{n_i})$ for processing on the machine M_i. A permutation vector π' is said to be inverse to π if all its schedules over the corresponding machines are inverse to the corresponding schedules in π. The inverse of a precedence relation G is denoted by $\overleftarrow{G}$. A schedule $S' = \{s_j' | j \in N_i, i \in M\}$ is called inverse to $S = \{s_j | j \in N_i, i \in M\}$ if $s_j' = -s_j - p_j$ $\forall j \in N_i$, $\forall i \in M$, where N_i is the set of jobs processed by M_i and s_j is the start time of the job j.

Definition 6. Given an instance $V = \{G^V, (r_j^V, p_j^V, d_j^V) | j \in N\}$, a schedule S admissible for V is called completely admissible if each job $j \in N$ is processed in its due interval $[r_j^V, d_j^V]$.

By $\Delta = \Delta(V, S)$, we denote a minimum value (possibly negative) that has to be added to all the due dates of the jobs in S so that the admissible schedule S becomes completely admissible for the resulting instance $V(\Delta)$. Obviously, $\Delta(V, S) = L_{\max}^V(S)$.

Definition 7. For any two instances A and B of the problems $\{P, Q, R\}|\text{prec}, r_j|L_{\max}$, we define the functions

$$\begin{aligned}
\rho_d(A, B) &= \max_{j \in N}\{d_j^A - d_j^B\} - \min_{j \in N}\{d_j^A - d_j^B\}, \\
\rho_r(A, B) &= \max_{j \in N}\{r_j^A - r_j^B\} - \min_{j \in N}\{r_j^A - r_j^B\}, \\
\rho_p(A, B) &= \sum_{j \in N}(\max_{i \in M}(p_{ji}^A - p_{ji}^B)_+ + \max_{i \in M}(p_{ji}^A - p_{ji}^B)_-), \\
\rho(A, B) &= \rho_d(A, B) + \rho_r(A, B) + \rho_p(A, B),
\end{aligned} \tag{4}$$

where

$$(x)_+ = \begin{cases} x, & x > 0, \\ 0, & x \le 0; \end{cases} \quad (x)_- = \begin{cases} -x, & x < 0, \\ 0, & x \ge 0; \end{cases} \quad |x| = (x)_+ + (x)_-.$$

Remark 1. The function $\rho_p(A, B)$ can be written in a different manner:

$$\rho_p(A, B) = \sum_{j \in N} (\max_{i \in M}\{(p_{ji}^A - p_{ji}^B), 0\} - \min_{i \in M}\{(p_{ji}^A - p_{ji}^B), 0\}).$$

Remark 2. If $p_{ji} \in \{p_j, +\infty\}$ $\forall j \in N$, $\forall i \in M$ (i.e., it is independent of the index of the machine to be processed on), then

$$\rho_p(A, B) = \sum_{j \in N} \left|p_j^A - p_j^B\right|. \tag{5}$$

Definition 8. Given an instance A on a set of jobs N (with a precedence relation G), an instance B on the same set of jobs is said to inherit a parameter x from A if $x_j^B = x_j^A$ $\forall j \in N$.

3. ESTIMATION OF THE ABSOLUTE ERROR

Lemma 1. *Suppose that an instance B inherits from A all the parameters, except for* $\{d_j | j \in N\}$. *Then any admissible* (*for both instances*) *schedule* π *satisfies*

$$L_{\max}^B(\pi) - L_{\max}^A(\pi) \le \max_{j \in N}\{d_j^A - d_j^B\}.$$

Proof. For any $i \in N$, we have $L_{\max}^A(\pi) + \max_{j \in N}\{d_j^A - d_j^B\} \ge C_k(\pi) - d_k^A + d_k^A - d_k^B = C_k(\pi) - d_k^B$. Therefore, $L_{\max}^A(\pi) + \max_{j \in N}\{d_j^A - d_j^B\} \ge \max_{k \in N}\{C_k(\pi) - d_k^B\} = L_{\max}^B(\pi)$. Lemma 1 is proved.

Let π^B be an optimal solution to B. Therefore, any schedule π satisfies

$$L_{\max}^B(\pi^B) \le L_{\max}^B(\pi) \quad \forall \pi.$$

Lemma 2. *Suppose that an instance B inherits from A all the parameters, except for* $\{d_j | j \in N\}$, *and let a* $\tilde{\pi}^B$ *be an approximate solution to B that satisfies the condition*[3]

$$0 \le L_{\max}^B(\tilde{\pi}^B) - L_{\max}^B(\pi^B) \le \delta_B,$$

where π^B *is an optimal solution. Then*

$$0 \le L_{\max}^A(\tilde{\pi}^B) - L_{\max}^A(\pi^A) \le \rho_d(\mathrm{A}, \mathrm{B}) + \delta_B.$$

Proof. Indeed,

$$\begin{aligned} L_{\max}^A(\tilde{\pi}^B) - L_{\max}^A(\pi^A) &\le L_{\max}^B(\tilde{\pi}^B) - L_{\max}^A(\pi^A) + \max_{j \in N}\{d_j^B - d_j^A\} \\ &\le \delta_\mathrm{B} + L_{\max}^B(\pi^B) + \max_{j \in N}\{d_j^B - d_j^A\} - L_{\max}^B(\pi^A) + \max_{j \in N}\{d_j^A - d_j^B\} \\ &\le \delta_B + \rho_d(A, B). \end{aligned}$$

Lemma 2 is proved.

Remark 3. There are instances for which an equality holds in the upper bound:

$$0 \le L_{\max}^B(\pi^\mathrm{A}) - L_{\max}^B(\pi^B) = \rho_d(A, B).$$

[3] If $\delta_B = 0$, then $\tilde{\pi}^B$ is an optimal solution.

Proof. For $n = 3$, $m = 1$, and $G = \varnothing$, the parameters of A and B are

$$r_j^A = 0, \quad p_j^A = 1, \quad j = 1, 2, 3, \quad d_1^A = 1, \quad d_2^A = 2, \quad d_3^A = 3,$$

$$r_j^B = r_j^A, \quad p_j^B = p_j^A, \quad d_j^B = 2, \quad j = 1, 2, 3,$$

respectively. It is easy to see that

$$\pi^A = (1, 2, 3), \quad L_{\max}^A(\pi^A) = 0,$$

$$\pi^B = (3, 2, 1), \quad L_{\max}^A(\pi^B) = 2,$$

$$\max_{j \in N}\{d_j^A - d_j^B\} = \max_{j \in N}\{d_j^B - d_j^A\} = 1,$$

$$\rho_d(A, B) = 2.$$

Lemma 3. *Let V and W be inverse instances with a set of jobs N, and let π and π' be mutually inverse permutations. Then $L_{\max}^V(S_\pi^V) = L_{\max}^W(S_{\pi'}^W)$.*

Proof. Let $\Delta = \Delta(V, S_\pi^V)$. Since the schedule S_π^V is completely admissible for $V(\Delta)$, the inverse S' of S_π^V is completely admissible for the inverse W' of $V(\Delta)$. This means that $L_{\max}^{W'}(S') \le 0$. Note that the instance W' differs from W (which is inverse to V) in that all $\{r_j\}$ are decreased by Δ. If S' is shifted to the right by Δ, then the resulting schedule (S'') is admissible for W, and $L_{\max}^W(S'') \le \Delta = L_{\max}^V(S_\pi^V)$. Since the jobs in S'' are processed with the permutation vector π', we have

$$L_{\max}^W(S_{\pi'}^W) \le L_{\max}^W(S'') \le L_{\max}^V(S_\pi^V).$$

Interchanging V with W and π with π', we obtain the reverse inequality $L_{\max}^V(S_\pi^V) \le L_{\max}^W(S_{\pi'}^W)$, which implies $L_{\max}^V(S_\pi^V) = L_{\max}^W(S_{\pi'}^W)$, as required. Lemma 3 is proved.

Corollary 1. If π is an optimal solution for V, then the inverse solution π' is optimal for the inverse of V.

Lemma 4. *Suppose that an instance C inherits from B all the parameters, except for $\{r_j | j \in N\}$ and let $\tilde{\pi}^C$ be an approximate solution to C that satisfies the condition*

$$0 \le L_{\max}^C(\tilde{\pi}^C) - L_{\max}^C(\pi^C) \le \delta_C.$$

Then

$$0 \le L_{\max}^B(\tilde{\pi}^C) - L_{\max}^B(\pi^B) \le \rho_r(B, C) + \delta_C.$$

Proof. Let E and F be the inverses of B and C, respectively, with the job parameters $r_j^E = -d_j^B$, $p_j^E = p_j^B$, $d_j^E = -r_j^B$ and $r_j^F = -d_j^C$, $p_j^F = p_j^C$, $d_j^F = -r_j^C$, $G^E = G^F = \overline{G}^B = \overline{G}^C$. Let π^E, π^F, and $\tilde{\pi}^F$ be the inverses of the solutions π^B, π^C, and $\tilde{\pi}^C$, respectively. By Corollary 1, π^E and π^F are optimal solutions for E and F, respectively. By Lemma 2,

$$\delta_C + \rho_d(E, F) \ge L_{\max}^E(\tilde{\pi}^F) - L_{\max}^E(\pi^E) \ge 0.$$

By Lemma 3, $L_{\max}^B(\pi^B) = L_{\max}^E(\pi^E)$ and $L_{\max}^B(\tilde{\pi}^C) = L_{\max}^E(\tilde{\pi}^F)$.

Therefore,

$$\delta_C + \rho_d(E, F) \ge L_{\max}^B(\tilde{\pi}^C) - L_{\max}^B(\pi^B) \ge 0.$$

We have $\rho_d(E, F) = \max_{j \in N}\{d_j^E - d_j^F\} + \max_{j \in N}\{d_j^F - d_j^E\} = \max_{j \in N}\{r_j^C - r_j^B\} + \max_{j \in N}\{r_j^B - r_j^C\} = \rho_r(B, C)$). Combining this with (10), we conclude the proof of Lemma 4.

Lemma 5. *Suppose that an instance D inherits from C all the parameters, except for* $\{p_{ji} < \infty | j \in N, i \in M\}$. *Then any admissible schedule* π *satisfies*

$$L_{\max}^D(\pi) - L_{\max}^C(\pi) \le \sum_{j \in N} \max_{i \in M}(p_{ji}^D - p_{ji}^C)_+.$$

Proof. We have $L_{\max}^D(\pi) = \max_{j \in N}\{C_j^D(\pi) - d_j^D\}$. Let j^* be the job for which this maximum is reached (on the machine i^*), i.e., $L_{\max}^D(\pi) = C_{j^*}^D(\pi) - d_{j^*}^D$. Moreover, $L_{\max}^C(\pi) \ge C_{j^*}^C(\pi) - d_{j^*}^C = s_{j^*}(\pi) + p_{j^*i^*}^C - d_{j^*}^C$. The processing of the next job cannot begin earlier than the previous one is completed. Therefore, $s_j \ge C_k^C(\pi)$ if the job k precedes j (according to the precedence relation G or in a permutation on the same machine). Therefore,

$$L_{\max}^C(\pi) \ge s_{j_1} + \sum_{j_k \in \{\sigma^*(j^*)\}} p_{j_k i_k}^C - d_{j^*}^C,$$

where j_1 is the first job in the delaying chain $\sigma^*(j^*) = (j_1, \ldots, j^*)$. Furthermore,

$$L_{\max}^C(\pi) \ge s_{j_1} + \sum_{j_k \in \{\sigma^*(j^*)\}} p_{j_k i_k}^D - d_{j^*}^D - \sum_{j_k \in \{\sigma^*(j^*)\}} (p_{j_k i_k}^D - p_{j_k i_k}^C)$$

$$\ge L_{\max}^D(\pi) - \sum_{j_k \in \{\sigma^*(j^*)\}} (p_{j_k i_k}^D - p_{j_k i_k}^C)_+ \ge L_{\max}^D(\pi) - \sum_{j \in N} (p_{ji}^D - p_{ji}^C)_+,$$

which implies the assertion of Lemma 5.

Lemma 6. *Suppose that an instance D inherits from C all the parameters, except for* $\{p_{ij} | j \in N, i \in M\}$; *and let* $\tilde{\pi}^D$ *be an approximate solution to D that satisfies the condition*

$$L_{\max}^D(\tilde{\pi}^D) - L_{\max}^D(\pi^D) \le \delta_D. \tag{6}$$

Then

$$0 \le L_{\max}^C(\tilde{\pi}^D) - L_{\max}^C(\pi^C) \le \rho_p(C, D) + \delta_D.$$

Proof. The proof sketch is similar to that of Lemma 2. For this reason, we give only the derivation

$$L_{\max}^C(\tilde{\pi}^D) - L_{\max}^C(\pi^C) \le L_{\max}^D(\tilde{\pi}^D) - L_{\max}^C(\pi^C) + \sum_{j \in N} \max_{i \in M}(p_{ji}^C - p_{ji}^D)_+$$

$$\le \delta_D + L_{\max}^D(\pi^D) + \sum_{j \in N} \max_{i \in M}(p_{ji}^C - p_{ji}^D)_+ - L_{\max}^D(\pi^C) + \sum_{j \in N} \max_{i \in M}(p_{ji}^D - p_{ji}^C)_+$$

$$\le \delta_D + \sum_{j \in N} (\max_{i \in M}(p_{ji}^C - p_{ji}^D)_+ + \max_{i \in M}(p_{ji}^C - p_{ji}^D)_-) = \delta_D + \rho_p(C, D).$$

Remark 4. The estimate in Lemma 6 is sharp.

Proof. For $n = 1$ and $m = 2$, the instance parameters are

$$r_1^A = d_1^A = 0; \quad p_{11}^A = 1, \quad p_{12}^A = 3 \quad \text{for } A,$$

$$r_1^B = d_1^B = 0; \quad p_{11}^B = p_{12}^B = 2 \quad \text{for } B.$$

It is easy to see that

$$\pi^A = \{(1), (\varnothing)\}; \quad L_{\max}^A(\pi^A) = 1,$$

where $\pi = \{\pi_1, \pi_2\}$ means that π_1 and π_2 are schedules for the first and second machines, respectively;

$$\pi^B = \{(\varnothing), (1)\}; \quad L^A_{\max}(\pi^B) = 3,$$

$$\max_{i \in M}(p^A_{1i} - p^B_{1i})_+ = \max_{i \in M}(p^A_{1i} - p^B_{1i})_- = 1,$$

$$\rho_p(A, B) = 2.$$

Theorem 1. *Suppose that an instance D inherits from A all the parameters, except for* $\{d_j, r_j, p_{ji} | j \in N, i \in M\}$, *and let* $\tilde{\pi}^D$ *be an approximate solution to D that satisfies condition* (6). *Then*

$$0 \le L^A_{\max}(\tilde{\pi}^D) - L^A_{\max}(\pi^A) \le \rho(A, D) + \delta_D.$$

Proof. The inequality $0 \le L^A_{\max}(\tilde{\pi}^D) - L^A_{\max}(\pi^A)$ follows from the fact that the solution π^A is optimal for A. Let us prove the inequality $L^A_{\max}(\tilde{\pi}^D) - L^A_{\max}(\pi^A) \le \rho(A, D) + \delta_D$. To this end, we use the additive property of the estimates obtained in Lemmas 2, 4, and 6. Consider auxiliary instances B and C with the job parameters

$$d^A_j, \quad d^B_j = d^C_j = d^D_j;$$

$$r^A_j = r^B_j, \quad r^C_j = r^D_j;$$

$$p^A_{ji} = p^B_{ji} = p^C_{ji}, \quad p^D_{ji}.$$

Then $\rho(A, B) = \rho_d(A, B)$, $\rho(B, C) = \rho_r(B, C)$, and $\rho(C, D) = \rho_p(C, D)$.

Sequentially applying Lemmas 6, 4, and 2 and setting $\tilde{\pi}^D = \tilde{\pi}^C$, $\delta_C = \delta_D + \rho_p(C, D)$, $\tilde{\pi}^C = \tilde{\pi}^B$, and $\delta_B = \delta_C + \rho_r(B, C)$, we obtain

$$\begin{aligned} 0 \le L^A_{\max}(\tilde{\pi}^D) - L^A_{\max}(\pi^A) &\le \rho_d(A, B) + \rho_r(B, C) + \rho_p(C, D) + \delta_D \\ &= \rho_d(A, D) + \rho_r(A, D) + \rho_p(A, D) + \delta_D = \rho(A, D) + \delta_D. \end{aligned}$$

Theorem 1 is proved.

Remark 5. The estimate in Theorem 1 is sharp.

Proof. For $n = 4$ and $m = 1$, the instance parameters are

$$r^A_1 = r^A_3 = r^A_4 = 0, \; r^A_2 = 1, \; p^A_j = 1, \; j = 1, 2, 3; \; d^A_j = j, \; j = 1, 2, 3, 4, \quad \text{for } A;$$

$$r^B_1 = 1, \; r^B_j = r^A_j, \; j = 2, 3, 4, \; p^B_j = p^A_j, \; j = 1, 2, 3, \; p^B_4 = 0, \; d^B_1 = 2, \; d^B_j = d^A_j, \; j = 2, 3, 4, \quad \text{for } B.$$

It is easy to see that

$$\pi^A = (1, 2, 3, 4), \quad L^A_{\max}(\pi^A) = 0, \quad \pi^B = (4, 3, 2, 1), \quad L^A_{\max}(\pi^B) = 3,$$

$$\rho_d(A, B) = \rho_r(A, B) = \rho_p(A, B) = 1, \quad \rho(A, B) = 3.$$

4. SCHEME FOR FINDING AN APPROXIMATE SOLUTION TO THE PROBLEM $P|\text{prec}, r_j|L_{\max}$

An approximate solution to the NP-hard problem under study is found in two steps. At the first step, while solving a linear programming problem, we change the job parameters $\{(r^A_j, p^A_j, d^A_j) | j \in N\}$ of the initial instance $A = \{G, (r^A_j, p^A_j, d^A_j) | j \in N\}$ so that the resulting instance $B = \{G, (r^B_j, p^B_j, d^B_j) | j \in N\}$ (with the same precedence graph G for both instances) is a polynomially/pseudopolynomially solvable special case of the original problem. An optimal schedule for B is found at the next step. According to Theorem 1, the optimal schedule π^B for A has a minimum estimate for the absolute error of the objective function $0 \le L^A_{\max}(\pi^B) - L^A_{\max}(\pi^A) \le \rho(A, B)$ for all the instances of the solvable class.

Consider a polynomially solvable instance of the original problem when the job parameters satisfy k linearly independent inequalities

$$XR + YP + ZD \leq H \tag{7}$$

(under the natural constraint $p_j \geq 0 \ \forall j \in N$), where $R = (r_1, \ldots, r_n)^T$; $P = (p_1, \ldots, p_n)^T$; $D = (d_1, \ldots, d_n)^T$; X, Y, and Z are $k \times n$ matrices; and $H = (h_1, \ldots, h_k)^T$ is a k-dimensional vector (here, the superscript ^{T} denotes the transpose). Then, in the class of instances (7), an instance B with the minimum "distance" $\rho(A, B)$ from the original instance A is found by solving the linear programming problem

$$\begin{gathered}(x^d - y^d + x^r - y^r) + \sum_{j \in N} x_j^p \longrightarrow \min, \\ y^d \leq d_j^A - d_j^B \leq x^d \quad \forall j \in N, \quad y^r \leq r_j^A - r_j^B \leq x^r \quad \forall j \in N, \\ -x_j^p \leq p_j^A - p_j^B \leq x_j^p \quad \forall j \in N, \quad 0 \leq x_j^p \quad \forall j \in N, \\ XR^B + YP^B + ZD^B \leq H.\end{gathered} \tag{8}$$

Problem (8) with $3n + 4 + n$ variables (r_j^B, p_j^B, and d_j^B for $j = 1, 2, \ldots, n$ and x_d, y_d, x_r, y_r and x_j^p for $j = 1, 2, \ldots, n$) and with $7n + k$ inequalities can sometimes be solved in polynomial time (in n and k) if the specific nature of the constraints in problem (8) is taken into account. For example, the problem $1|r_j|L_{\max}$, which is NP-hard in the strong sense, has two polynomially solvable nontrivial instances:

$$d_1 \leq \ldots \leq d_n, \quad d_1 - r_1 - p_1 \geq \ldots \geq d_n - r_n - p_n, \tag{9}$$

$$\max_{k \in N}\{d_k - r_k - p_k\} \leq d_j - r_j \quad \forall j \in N. \tag{10}$$

For case (9), an optimal solution to the problem $1|r_j|L_{\max}$ can be found in $O(n^3 \log n)$ operations [32, 33]. Problem (8) can be solved in $O(n \log n)$ operations [18, 34].

For case (10), an optimal schedule is found in $O(n^2 \log n)$ operations [10]. As in case (9), the minimum of the absolute error in the maximum lateness is found in polynomial time, specifically, in $O(n)$ operations (see [18]).

When the original problem has no polynomially solvable instances (generally, qualitatively new absolute-error estimates cannot be found in trivial cases) or when the distance $\rho(A, C)$ from any polynomially solvable instance C is "too large", but the absolute error estimate for the maximum lateness of the approximate schedule $\tilde{\pi}$ is "acceptable" for some instance $B = \{G, (r_j^B, p_j^B, d_j^B) | j \in N\}$, then the approximate schedule $\tilde{\pi}$ for the initial instance $A = \{G, (r_j^A, p_j^A, d_j^A) | j \in N\}$ has a guaranteed error of the optimal objective function value that does not exceed $\delta^B(\tilde{\pi}) + \rho(A, B)$ according to Theorem 1. Sometimes, $\delta^B(\tilde{\pi}) + \rho(A, B)$ is considerably less than $\rho(A, C)$ for any polynomially or pseudopolynomially of solvable instance C.

5. NORMED SPACE OF INSTANCES

Consider the set of instances from the class $P|\text{prec}, r_i|L_{\max}$ with n jobs, m machines, and a precedence graph G. This set of instances forms a $3n$-dimensional linear space ($3n$ parameters: r_j, p_j, and d_j for $j = \overline{1, n}$).

Definition 9. Instances $A = \{G, (r_j^A, p_j^A, d_j^A) | j \in N\}$ and $B = \{G, (r_j^B, p_j^B, d_j^B) | j \in N\}$ are said to be equivalent if there exist two scalar parameters d and r such that

$$d_j^A = d_j^B + d, \quad r_j^A = r_j^B + r, \quad p_j^A = p_j^B \quad \forall j \in N.$$

This definition generates a partition of the set of problem instances into equivalence classes. For convenience, a representative of the class is chosen to be an instance such that $r_1 = 0$ and $d_1 = 0$. The resulting family of equivalence classes is a $(3n - 2)$-dimensional linear space, which is denoted by $\mathcal{L}_n$. An instance A is said to be in $\mathcal{L}_n$ if

$$r_1 = d_1 = 0.$$

Remark 6. Equivalent instances have the same sets of optimal schedules (permutations). On the space of equivalence classes of instances, define the functional

$$\varphi(A) = \max_{j \in N} r_j^A - \min_{j \in N} r_j^A + \max_{j \in N} d_j^A - \min_{j \in N} d_j^A + \sum_{j \in N} |p_j^A| \geq 0 \quad \forall A \in \mathcal{L}_n.$$

It satisfies the norm properties:

$$\varphi(A) = 0 \Leftrightarrow A \equiv \varnothing,$$
$$\varphi(\alpha A) = \alpha\varphi(A),$$
$$\varphi(A + B) \leq \varphi(A) + \varphi(B),$$

$A \equiv \varnothing$ if $r_j = p_j = d_j = 0\ \forall j \in N$. The first property follows from the definition of $\varphi(A)$, the second property is directly verified, and the third one follows from the fact that the maximum of a sum is no greater than the sum of maxima and the modulus of a sum is no greater than the sum of moduli.

Thus, $\mathcal{L}_n$ is a $(3n - 2)$-dimensional linear normed space with the norm $\|A\| = \varphi(A)$. Note that this norm naturally generates the metric defined in (4) and (5): $\rho(A, B) = \|A - B\|$.

Consider some properties of the functional $\varphi(A) = \|A\|$. To specify the topology of the space, it suffices to define a system of neighborhoods of zero. In this case, they are balls centered at zero. Now, we describe the unit ball in the space of instances of the problem with varying parameters r_i and d_i. The unit ball is the set of points (instances) A satisfying the condition

$$\|A\| \leq 1. \tag{11}$$

Let $n = 2$. Condition (11) is then rewritten in the following form (see Fig. 1):

$$|r_2| + |d_2| \leq 1.$$

Let $n = 3$. Condition (11) can be rewritten as

$$r + d \leq 1,$$
$$r \geq 0,$$
$$d \geq 0,$$
$$-d \leq d_2 \leq d,$$
$$-d \leq d_3 \leq d,$$
$$-d \leq d_3 - d_2 \leq d,$$
$$-r \leq r_2 \leq r,$$
$$-r \leq r_3 \leq r,$$
$$-r \leq r_3 - r_2 \leq r.$$

The projection of the unit ball onto a plane parallel to the plane (d_2, d_3) is shown in Fig. 2.

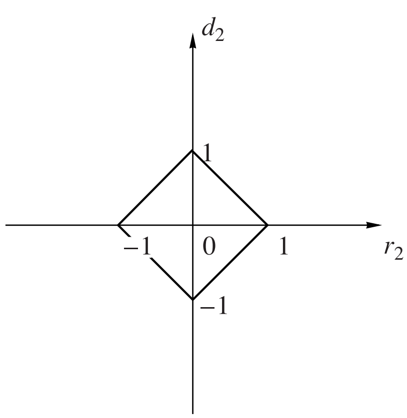

Fig. 1.

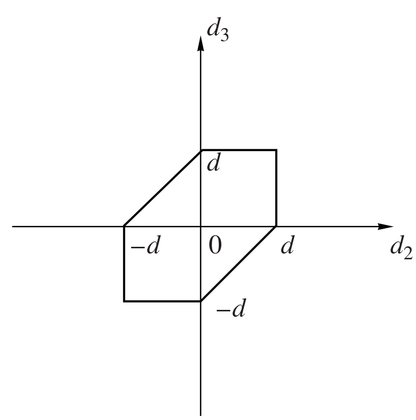

Fig 2.

For arbitrary n, condition (11) is equivalent to the system of inequalities

$$r + d \le 1,$$

$$r \ge 0,$$

$$d \ge 0,$$

$$-d \le d_i \le d, \quad 2 \le i \le n,$$

$$-d \le d_i - d_j \le d, \quad 2 \le i < j \le n,$$

$$-r \le r_i \le r, \quad 2 \le i \le n,$$

$$-r \le r_i - r_j \le r, \quad 2 \le i < j \le n.$$

6. SCHEMES FOR FINDING AN APPROXIMATE SOLUTION

Let us show how the approach described above can be used to find approximate solutions to scheduling problems with an absolute-error estimate. Suppose that we want to solve an instance A of the problem $\alpha|\beta|L_{\max}$, which is denoted by $\alpha^A|\beta^A|L_{\max}$. The machine and job parameters α^A and β^A in A and possibly the objective function $L_{\max}$ on $C_{\max}$ have to be changed to obtain an instance B of the problem $\alpha^B|\beta^B|\{L_{\max}, C_{\max}\}$ for which we can find an approximate (or exact) solution $\tilde{\pi}^B$ and then apply it to the initial instance A of the problem $\alpha^A|\beta^A|L_{\max}$.

6.1. Reduction Scheme for the Problems $\alpha|\beta|L_{\max} \longrightarrow \alpha|\beta|C_{\max}$ *and* $\alpha|\beta, r_j|L_{\max} \longrightarrow \alpha|\beta, r_j = 0|L_{\max}$

Suppose that we are given an instance A of the NP-hard problem $\alpha^A|\beta^A|L_{\max}$, and let an exact or approximate solution $\tilde{\pi}^B$ (obtained in polynomial/pseudopolynomial time) of the problem instance B: $\alpha^A|\beta^A|C_{\max}$ be known with an absolute error not exceeding $\delta_B \ge 0$.

For instance B, we have $d_j^B = 0 \ \forall j \in N$. Therefore, Lemma 2 yields the estimate

$$0 \le L^A_{\max}(\tilde{\pi}^B) - L^A_{\max}(\pi^A) \le \rho_d(A, B) + \delta_B = \max_{j \in N} d_j^A - \min_{j \in N} d_j^A + \delta_B.$$

An approximate solution to the problem $\alpha|\beta, r_j|L_{\max}$ can be found using the solution to the problem $\alpha|\beta, r_j = 0|L_{\max}$. In this case, Lemma 4 gives the following estimate for the absolute error of the approximate solution $\tilde{\pi}^B$:

$$0 \le L^A_{\max}(\tilde{\pi}^B) - L^A_{\max}(\pi^A) \le \max_{j \in N} r_j^A - \min_{j \in N} r_j^A + \delta_B.$$

6.2. Reduction Scheme for the Problems $\alpha|\beta, p_j|L_{\max} \longrightarrow \alpha|\beta, p_j = p|L_{\max}$ *and* $\alpha|\beta, p_j|L_{\max} \longrightarrow \alpha|\beta, p_j = 1|L_{\max}$

We want to solve an instance A of the NP-hard problem $\alpha^A|\beta^A|L_{\max}$. Let an exact or approximate solution $\tilde{\pi}^B$ instance of the problem instance $\alpha^A|\beta^A, p_j = p|L_{\max}$ be known with an absolute error not exceeding $\delta_B \ge 0$.

For the absolute error of the resulting solution to be minimal, we have to find an optimal parameter value p for which the distance $\rho_p(A, B)$ between the instances is minimal:

$$\rho_p(A, B) = \sum_{j=1}^{n} \left| p_j^A - p \right| \longrightarrow \min_p. \tag{12}$$

This is a continuous, convex, and piecewise linear function. If n is odd, then the minimum is reached for the "mean" p_j^A, i.e., if all p_j^A are arranged in nondecreasing order, then the solution to problem (12) is $p^* = p^A_{\frac{n+1}{2}}$.

If n is even, then the minimum is reached for two means p_j^A and also for any value between them, i.e., $p^* \in [p_{n/2}^A; p_{n/2+1}^A]$. Thus, $p^* = p_{[(n+1)/2]}^A$. By Lemma 6, we have

$$0 \le L_{\max}^A(\tilde{\pi}^B) - L_{\max}^A(\pi^A) \le \sum_{j=1}^{n} \left| p_j^A - p_{\left[\frac{n+1}{2}\right]}^A \right| + \delta_B.$$

6.3. Reduction Scheme for $\alpha|\beta, p_j|L_{max} \longrightarrow \alpha|\beta, p_j = 1|L_{\max}$

Suppose that we are given an instance A of the NP-hard problem $\alpha^A|\beta^A|L_{\max}$, and let an exact or approximate solution $\tilde{\pi}^B$ to the problem $\alpha^A|\beta^A, p_j = p|L_{\max}$ be known with an absolute error not exceeding $\delta_B \ge 0$. The condition $p_j = 1$ means that $p_j = p$ and all the parameters r_j^A and d_j^A are multiples of p. As p, we can use $p = p_{\left[\frac{n+1}{2}\right]}^A$. For all r_j^A and d_j^A to be multiples of p, the remainder of the division of them by p has to be subtracted from them. Then $\rho_r(A, B) \le p$ and $\rho_d(A, B) \le p$. For the instance B, we have $p_j^B = p = p_{\left[\frac{n+1}{2}\right]}^A$, $r_j^B = r_j^A - (r_j^A \bmod p)$, $d_j^B = d_j^A - (d_j^A \bmod p)$ $\forall j \in N$, and the estimate of the absolute error is

$$0 \le L_{\max}^A(\tilde{\pi}^B) - L_{\max}^A(\pi^A) \le \sum_{j \in N} \left| p_j^A - p_{\left[\frac{n+1}{2}\right]}^A \right| + 2p_{\left[\frac{n+1}{2}\right]}^A.$$

6.4. Reduction Scheme for the Problems $\{R, Q\}|\beta|L_{\max} \longrightarrow P|\beta|L_{\max}$

In contrast to $\{R, Q\}|\beta|L_{\max}$, the machines in the problems $P|\beta|L_{\max}$ are not identical, i.e., the processing times of the job $j \in N$ can differ for different machines: $p_{ji} \ne p_{jk}$ for $i \ne k$ and $i, k \in M$.

Suppose that we are given an instance A of the NP-hard problem $R|\beta^A|L_{\max}$, and let an exact or approximate solution $\tilde{\pi}^B$ of the problem $P|\beta^A, p_{ij} \in \{p_j, \infty\}|L_{\max}$ be known with an absolute error not exceeding $\delta_B \ge 0$.

For the absolute error of $\tilde{\pi}^B$ to be minimal, we have to find optimal parameter values p_j^B with the minimal distance between A and B:

$$\rho_p(A, B) = \sum_{j \in N} \left(\max_{i \in M}\{(p_{ji}^A - p_j^B), 0\} - \min_{i \in M}\{(p_{ji}^A - p_j^B), 0\}\right) \longrightarrow \min_{p_j^B\ \forall j \in N}.$$

It is well known that

$$\rho_p(A, B) = \sum_{j \in N} \left(\max_{i \in M}\{p_{ji}^A, p_j^B\} - \min_{i \in M}\{p_{ji}^A, p_j^B\}\right) = \sum_{j \in N} \left(\max_{i \in M} p_{ji}^A - \min_{i \in M} p_{ji}^A\right)$$

if $p_j^B \in [\max_{i \in M} p_{ji}^A, \min_{i \in M} p_{ji}^A]$ $\forall j \in N$. Therefore,

$$0 \le L_{\max}^A(\tilde{\pi}^B) - L_{\max}^A(\pi^A) \le \sum_{j \in N} \left(\max_{i \in M} p_{ji}^A - \min_{i \in M} p_{ji}^A\right).$$

6.5. Reduction Scheme for the Problem $R|\beta|L_{\max} \longrightarrow Q|\beta|L_{\max}$

Suppose that we are given an instance A of the NP-hard problem $R|\beta|L_{\max}$ ($p_{ji} < \infty$), and let an exact or approximate solution $\tilde{\pi}^B$ to the problem $Q|\beta|L_{\max}$ be known with an absolute error not exceeding $\delta_B \ge 0$. In the problem $Q|\beta|L_{\max}$, the processing times are calculated by the formula $p_{ji} = \sigma_i p_j\ \forall j \in N, \forall i \in M$, where σ_i is the efficiency of the ith machine.

To estimate the absolute error of $\tilde{\pi}^B$, we have to solve the problem

$$\rho_p(A, B) = \sum_{j \in N}(\max_{i \in M}\{(p_{ji}^A - p_j^B\sigma_i^B), 0\} - \min_{i \in M}\{(p_{ji}^A - p_j^B\sigma_i^B), 0\}) \longrightarrow \min_{p_j^B, \sigma_i^B},$$

which can be written in the form of the linear programming problem

$$\sum_{j \in N}(\alpha_j - \beta_j) \longrightarrow \min_{\alpha_j, \beta_j, p_j^B, \sigma_i^B},$$

$$\beta_j \le p_{ji}^A - p_j^B\sigma_i^B \le \alpha_j, \quad j \in N, \quad i \in M,$$

$$\beta_j \le 0 \le \alpha_j, \quad j \in N.$$

6.6. Reduction Scheme for the Problem $\alpha|\beta|L_{\max} \longrightarrow \alpha|\beta, p_j \in \{p_1, \ldots, p_k\}|C_{\max}$

Suppose that we are given an instance A of the NP-hard problem $\alpha^A|\beta^A|L_{\max}$, and let an exact or approximate solution $\tilde{\pi}^B$ to the problem B: $\alpha^A|\beta^A, p_j \in \{p_1^B, \ldots, p_k^B\}|C_{\max}$ be known with an absolute error not exceeding $\delta_B \ge 0$. Then the absolute error is estimated as

$$0 \le L_{\max}^A(\tilde{\pi}^B) - L_{\max}^A(\pi^A) \le \rho_p(A, B) + \rho_d(A, B) + \delta_B,$$

where $\rho_d(A, B) = \max_{j \in N} d_j^A - \min_{j \in N} d_j^A$. To find $\rho_p(A, B)$, we have to solve the problem

$$\sum_{j=1}^{n} \min_{1 \le l \le k}\left|p_j^A - p_l^B\right| \longrightarrow \min_{p_1^B, \ldots, p_k^B}.$$

Consider several applications of the schemes described above. They can be used separately or in aggregate.

6.7. Single Machine

The problem $1|r_j|L_{\max}$ is NP-hard in the strong sense (see [2]). Schemes for finding approximate solutions were described in [18]. The more complicated problem $1|\text{prec}, r_j|L_{\max}$ is also NP-hard in the strong sense. It has two polynomially solvable instances: $1|\text{prec}, r_j|C_{\max}$ [7] and $1|\text{prec}, p_j = p, r_j|L_{\max}$ [8]. This problem can be approximately solved by applying one of the two algorithms, depending on where the absolute error is smaller.

6.8. Two Parallel Machines

The problem $P2|\text{chains}|C_{\max}$ is NP-hard in the strong sense (see [15]). There are polynomial-time algorithms for solving “similar” problems, for example, $P2|p_j = p, \text{prec}|L_{\max}$ [19]; $P2|p_j = 1, r_j|L_{\max}$ [20]; and $Q2|p_j = p, \text{chanis}|C_{\max}$ [21].

6.9. An Arbitrary Number of Parallel Machines

The following problems are NP-hard in the strong sense:

$P\,||\,C_{\max}$ [16], $P|p_j = 1, \text{intree}, r_j|C_{\max}$, $P|p_j = 1, \text{outtree}|L_{\max}$ [14],

$P|p_j = 1, \text{prec}|C_{\max}$ [17], $Q|p_j = p, \text{chains}|C_{\max}$ [22].

They can be solved using the polynomially solvable instances

$P|p_j = p, r_j|L_{\max}$ [23], $P|p_j = p, \text{tree}|C_{\max}$ [24], [25], $P|p_j = p, \text{outtree}, r_j|C_{\max}$ [14],

$P|p_j = p, \text{intree}|L_{\max}$ [14], $P|p_j = 1, \text{chains}, r_j|L_{\max}$ [26].

For problems with an arbitrary precedence relation, there are no exact polynomially solvable instances. In our view, an important task is to “modify” the precedence graph so as to derive error estimates.

6.10. Flow-Shop Problems

This approach can be used to find an approximate solution and to estimate the absolute error for flow-shop problems.

The following problems are NP-hard in the strong sense:

$$F2 \,||\, L_{\max},\quad F2|r_j|C_{\max},\quad F2|\text{chains}|C_{\max}\ [2],\quad F3 \,||\, C_{\max}\ [16],\ F|p_{ji}=1, \text{prec}|C_{\max}\ [27], [28],$$
$$F|p_{ji}=1, \text{intree}, r_j|C_{\max},\quad F|p_{ji}=1, \text{outtree}|L_{\max}\ [29].$$

The polynomially solvable instances are $F2||\, C_{\max}$ [30], $F|p_{ji} = 1$, outtree, $r_j|C_{\max}$, $F2|p_{ji} = 1$, prec, $r_j|L_{\max}$, $F|p_{ji} = 1$, tree$|C_{\max}$, $F|p_{ji} = 1$, intree$|L_{\max}$ [31].

7. CONCLUSIONS

Thus, the parameter-variation method with constructing an approximate solution can be applied to most NP-hard scheduling problems with polynomially/pseudopolynomially solvable instances.

Studies are now being performed on how this approach can be applied to other classical NP-hard combinatorial optimization problems, such as partition, knapsack, traveling salesperson, etc. In the author’s view, interesting issues are those concerning variations in the structure of the precedence graph and the derivation of error estimates for scheduling problems and graph theory.

ACKNOWLEDGMENTS

The author is grateful to the reviewer for his remarks and suggestions that helped to improve this paper. This work was supported by the program “Leading Scientific Schools,” project no. NSh-5833.2006.1.

REFERENCES

1. R. L. Graham, E. L. Lawler, J. K. Lenstra, et al., “Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey,” Ann. Discrete Math., No. 5, 287–326 (1979).
2. J. K. Lenstra, A. H. G. Rinnooy Kan, and P. Brucker, “Complexity of Machine Scheduling Problems,” Eur. J. Operat. Res. **4**, 270–275 (1980).
3. J. R. Jackson, “Scheduling a Production Line to Minimize Maximum Tardiness,” Manag. Sci. Res. Project, UCLA, 43 (1955).
4. C. N. Potts, “Analysis of a Heuristic for One Machine Sequencing with Release Dates and Delivery Times,” Operat. Res. **28**, 1436–1441 (1980).
5. L. A. Hall and D. B. Shmoys, “Jackson’s Rule for One-Machine Scheduling: Making a Good Heuristic Better,” Math. Operat. Res. **17**, 22–35 (1992).
6. M. Mastrolilli, “Efficient Approximation Schemes for Scheduling Problems with Release Dates and Delivery Times,” J. Scheduling, No. 6, 521–531 (2003).
7. E. L. Lawler, “Optimal Sequencing of a Single Machine Subject to Precedence Constraints,” Manag. Sci. **19** (5), 544–546 (1973).
8. B. B. Simons, “A Fast Algorithm for Single Processor Scheduling,” *Proceedings of 19th IEEE Annual Symposium on Foundations of Computer Science* (Ann. Arbor. Mich., New York, 1978), pp. 246–252.
9. 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,” Operat. Res. **31**, 381–386 (1983).
10. J. A. Hoogeveen, “Minimizing Maximum Promptness and Maximum Lateness on a Single Machine,” Math. Operat. Res. **21**, 100–114 (1996).
11. A. A. Lazarev and O. N. Shul’gina, “Pseudopolynomial-Time Algorithm for the NR-Hard Minimizing Maximum Lateness Problem,” *Proceedings of 11th International Baikal School-Seminar on Optimization Methods and Applications, July 5–12, 1998* (Irkutsk, 1998), pp. 163–167.
12. A. A. Lazarev and O. N. Shul’gina, “Polynomial-Time Solvable Special Cases of the Minimizing Maximum Lateness Problem,” Available from VINITI, No. 3019-B00 (Kazan, 2000).
13. Yu. Yu. Finkel’shtein, *Approximate Methods and Applications in Discrete Programming* (Nauka, Moscow, 1976) [in Russian].

14. P. Brucker, M. R. Garey, and D. S. Johnson, "Scheduling Equal-Length Tasks under Treelike Precedence Constraints to Minimize Maximum Lateness," Math. Operat. Res. **2** (3), 275–284 (1977).
15. J. Du and J. Y.-T. Leung, and G. H. Young, "Scheduling Chain-Structured Tasks to Minimize Makespan and Mean Flow Time," Inf. Comput. **92** (2), 219–236 (1991).
16. M. R. Garey and D. S. Johnson, "'Strong' NP-Completeness Results: Motivation, Instances, and Implications," J. Assoc. Comput. Machinery **25** (3), 499–508 (1978).
17. J. D. Ullman, "NP-Complete Scheduling Problems," J. Comput. Syst. Sci. **10**, 384–393 (1975).
18. A. A. Lazarev, R. R. Sadykov, and S. V. Sevastianov, "Scheme for Approximate Solution of the Problem $1|r_j|L_{\max}$," Diskret. Anal. Issled. Operats., Ser. 2. **13** (1), 57–76 (2006).
19. M. R. Garey and D. S. Johnson, "Scheduling Tasks with Nonuniform Deadlines on Two Processors," J. Assoc. Comput. Machinery **23**, 461–467 (1976).
20. M. R. Garey and D. S. Johnson, "Two-Processor Scheduling with Start-Times and Deadlines," SIAM J. Comput. **6** (3), 416–426 (1977).
21. P. Brucker, J. Hurink, and W. Kubiak, "Scheduling Identical Jobs with Chain Precedence Constraints on Two Uniform Machines," Math. Methods Operat. Res. **49** (2), 211–219 (1999).
22. W. Kubiak, "Exact and Approximate Algorithms for Scheduling Unit Time Tasks with Tree-Like Precedence Constraints," *Abstracts of EURO IX-TIMS XXVIII* (Paris, 1988), p. 195.
23. B. Simons, "Multiprocessor Scheduling of Unit-Time Jobs with Arbitrary Release Times and Deadlines," SIAM J. Comput. **2**, 294–299 (1983).
24. G. I. Davida and D. J. Linton, "A New Algorithm for the Scheduling of Tree Structured Tasks," *Proceedings of Conference on Information Sciences and Systems, Baltimore, MD* (John Hopkins Univ., 1976), pp. 543–548.
25. T. C. Hu, "Parallel Sequencing and Assembly Line Problems," Operat. Res. No. 9, 841–848 (1961).
26. P. Baptiste, P. Brucker, S. Knust, and V. Timkovsky, "Ten Notes on Equal-Execution-Time Scheduling," Quart. J. Belgian French Italian Operat. Res. Soc., No. 2, 111–127 (2004).
27. J. Y.-T. Leung, O. Vornberger, J. D. Witthoff, "On Some Variants of the Bandwidth Minimization Problem," SIAM J. Comput. **13** (3), 650–667 (1984).
28. V. G. Timkovsky, "Identical Parallel Machines vs. Unit-Time Shops and Preemptions vs. Chains in Scheduling Complexity," Eur. J. Operat. Res. **149** (2), 355–376 (2003).
29. P. Brucker and S. Knust, "Complexity Results for Single-Machine Problems with Positive Finish-Start Time-Lags," Computing **63**, 299–316 (1999).
30. S. M. Johnson, "Optimal Two- and Three-Stage Production Schedules with Setup Times Included," Naval Res. Logistics Quarterly **1**, 61–68 (1954).
31. J. Bruno, J. W. Jones III, and K. So, "Deterministic Scheduling with Pipelined Processors," IEEE Trans. Comput. **29** (4), 308–316 (1980).
32. A. A. Lazarev, "Pareto Optimal Set NR-Hard Problem of Minimization of the Maximum Lateness for a Single Machine," J. Comput. Syst. Sci. Int. **45**, 943–949 (2006).
33. A. A. Lazarev, "Estimation of Absolute Error in Scheduling Problems of Minimizing the Maximum Lateness," Dokl. Akad. Nauk **415** (4), 446–449 (2007) [Dokl. Math. **76** (1), 572–574 (2007)].
34. A. A. Lazarev, Candidate's Dissertation in Mathematics and Physics (Kazan. Gos. Univ., Kazan, 1989).