Applied Mathematical Modelling 89 (2021) 1163–1176

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

A metric approach for scheduling problems with minimizing the maximum penalty

Alexander A. Lazarev[a], Darya V. Lemtyuzhnikova[a,b], Frank Werner[c,*]

[a] Institute of Control Sciences, 65 Profsoyuznaya street, Moscow 117997, Russia
[b] Moscow Aviation Institute, 4 Volokolamsk highway, 125993 Moscow, Russia
[c] Fakultät für Mathematik, Otto-von-Guericke-Universität Magdeburg, PSF 4120, 39016 Magdeburg, Russia

ARTICLE INFO

Article history:
Received 21 November 2019
Revised 8 July 2020
Accepted 12 July 2020
Available online 15 August 2020

Keywords:
Scheduling
Minimization of maximum penalty
Maximum lateness
Metric approach
Approximation algorithm

ABSTRACT

NP-hard scheduling problems with the criterion of minimizing the maximum penalty, e.g. maximum lateness, are considered. For such problems, a metric which delivers an upper bound on the absolute error of the objective function value is introduced. Taking the given instance of some problem and using the introduced metric, the nearest instance is determined for which a polynomial or pseudo-polynomial algorithm is known. A schedule is constructed for this determined instance which is then applied to the original instance. It is shown how this approach can be applied to different scheduling problems.

1. Introduction

A class of *NP*–hard scheduling problems with minimizing the maximum penalty is considered. Even the special case of the single machine problem of minimizing maximum lateness subject to given release dates [1] belongs already to this class. For these problems, the existence of a polynomial algorithm is unlikely. There exist two types of methods for solving such problems: exact and approximate ones [2]. The first group includes integer linear programming [3], dynamic programming [4], the branch and bound method [5], the local elimination algorithm [6], and so on. In this case, the optimal objective function value is calculated without any error, but exact algorithms require large computation times and also a huge memory. Approximate methods such as genetic algorithms [7], ant colony algorithms [8], bee colony algorithms [9], tabu search [10], evolutionary algorithms [11] and many others obtain much faster a heuristic solution but there are usually no estimates of the deviation of the objective function value from the optimal one [12].

This paper deals with sub-cases of scheduling problems with a single and several machines. The focus is on the use of these polynomially solvable sub-cases for the solution of a the more general problem. We describe an approximation approach which is denoted as metric one [13]. The core part consists in constructing a solution with a specified maximal absolute error for scheduling problems on parallel machines with the criterion of minimizing maximum lateness. The absolute error of the approximate solution is bounded by a metric function $\rho(A, B)$. The idea of the metric approach is as follows. An instance A of a problem is characterized by a point of the input data. For this problem, we consider all known polynomial and pseudo-polynomial algorithms. Then we introduce some metric. By means of this metric, we find that polynomially or

* Corresponding author.
E-mail addresses: jobmath@mail.ru (A.A. Lazarev), darabbt@gmail.com (D.V. Lemtyuzhnikova), frank.werner@ovgu.de (F. Werner).

https://doi.org/10.1016/j.apm.2020.07.048
0307-904X/

pseudo-polynomially solvable instance with the smallest distance from the given instance. To do this, we compile a system of linear inequalities derived from the initial data. In other words, we construct the projection of the initial point onto a particular instance of a suitable sub-space by the introduced metric.

Up to now, the following results have been obtained for this metric approach. In [14], a comparison of feasible regions is carried out and the concept of a polynomial unsolvability measure for the single machine scheduling problem of minimizing maximum lateness subject to given release dates are given. It can be possibly reduced further by finding new sub-spaces of polynomially solvable instances. In [15,16], appropriate metrics for scheduling problems with total tardiness function are derived.

Polynomial and pseudo–polynomial algorithms are widely used to solve practical scheduling problems. For example, in [4], a supply chain scheduling problem is considered. The objective is to minimize the sum of the weighted number of tardy jobs as well as the due date assignment, resource allocation, and batch delivery costs. The authors propose a pseudo-polynomial dynamic programming algorithm for this problem. In addition, polynomial time approximation schemes are often used. In [17], a coordinated scheduling problem of production and transportation in which each job is transported to a single batching machine for further processing is studied. For a special case of this problem, where the job assignment to the vehicles is predetermined, the authors provide a polynomial time algorithm.

In this paper, the main focus is on several types of parallel machine problems. Even the special case of two identical parallel machines has important practical applications. In [18], the authors investigate the problem of wafer fabrication with two parallel machines. Molds are presented as resource constraints. Two or more jobs with the same mold requirement cannot be processed on the same or different machines at the same time. Three simple fast and efficient heuristics with a worst performance ratio of 3/2 are proposed to solve the problem. The results of the calculations show that the proposed heuristic allows to get good solutions for large-size problems. In [19], the problem of scheduling a given set of n jobs on two identical parallel machines with a single server is considered. As the authors mention, such problems have their applications in several manufacturing systems, e.g. in the semiconductor manufacturing systems, in robotic cells or in automated material handling systems. Each job must be processed on one of the machines. For the makespan minimization problem, fast polynomial algorithms are given which are tested for instances with up to 10,000 jobs. The algorithms have an excellent performance and obtain function values very close to the lower bounds or even optimal solutions.

The remainder of this paper is as follows. In Section 2, we describe the problem under consideration in more detail. Some necessary preliminaries and the notion of an inverse instance are given in Section 3. Some fundamental lemmas and theorems necessary for the subsequent approximation approach are given in Section 4. The approach for solving scheduling problems approximately with an error estimate is described in Section 5 for the problem of minimizing maximum lateness on identical parallel machines subject to given release dates and precedence constraints. Section 6 introduces a normalized space of the instances. In Section 7, the application of the approach to several other scheduling problems is briefly discussed. In Section 8, the metric approach for solving the scheduling problem with a single machine is tested and compared with other algorithms. Some concluding remarks complete the paper.

2. Formulation of the problem

In this paper, we will use the classical 3-parameter notation $\alpha|\beta|\gamma$ for scheduling problems [20]. Our main focus is on different types of parallel machine problems, which can be described as follows. A set of n jobs $j, j \in N = \{1, \dots, n\}$, has to be processed on a set of m machines $i, i \in M = \{1, \dots, m\}$. Preemptions of a job are not allowed. At any given time, any of the machines can process no more than one job.

For each job $j \in N$, a release date r_j, a due date d_j and the processing times p_{ij} for processing job j on machine $i \in M$ are given with $0 \le p_{ij} \le +\infty$. If $p_{ij} = +\infty$, then job j cannot be processed on machine i. In the case of identical parallel machines $(\alpha = P)$, we use p_j for the processing time of job j on any machine. In addition, precedence relations between jobs may be given by an directed acyclic graph $G \subset N \times N$.

In addition to identical machines ($\alpha = P$), we also consider problems with uniform ($\alpha = Q$) or unrelated ($\alpha = R$) machines. A schedule for such a problem is obtained by partitioning the set N of jobs into subsets N_i of jobs processed on machine $i, i = 1, \dots, m$. For each set N_i, one has to find the job sequence π_i processed on machine i. Inserted idle times between the processing of jobs are not allowed. This means that, if a job j is assigned to a free machine i and the release time r_j allows to start job i, then it must be started.

Since we consider only regular optimization criteria, the assignment of the jobs to the machines (i.e., the specification of the sets $N_1, \dots, N_m$) and the job sequences $\pi_1, \dots, \pi_m$ describe completely a schedule by the set of job sequences: $\pi = \{\pi_1, \dots, \pi_m\}$. In a semi-active schedule, each job $j \in N$ starts its processing at the earliest possible time: either at the release date r_j, or immediately after the completion of the previous job on this machine, or immediately after the end of a job preceding it according to the graph G. Instead of the job sequences, one can equivalently give the starting times s_j of all jobs $j \in N_i, i = 1, \dots, m$. We denote the set of starting times by $S = \bigcup_{j \in N} s_j$.

Let $Pred(j)$ be the set of all jobs which are a predecessor of job j in the precedence graph G and $(k \to j)_{\pi_i}$ be the jobs scheduled on machine i before job j according to the sequence π_i. Then the starting time of a job $j \in N_i$, $i = 1, \dots, m$, in the schedule π is given by

$$s_j(\pi) = \max\left\{r_j, \max_{k\in Pred(j)} (s_k(\pi) + p_{ik}), \max_{(k\to j)_{\pi_i}} (s_k(\pi_i) + p_{ik})\right\}. \quad (1)$$

Since preemptions of jobs are not allowed, the completion time of job $j \in N_i$ in a schedule π is given by

$$C_j(\pi) = s_j(\pi) + p_{ij}, \ j \in N_i.$$

The schedule π is called *feasible*, if $r_j \le s_j(\pi)$ and $C_j(\pi) \le s_k(\pi)$ for all arcs $(j, k) \in G$.

Remark 1. If a schedule π is known, the starting times S can be uniquely determined and vice versa, if all starting times S (together with the sets $N_1, \dots, N_m$) are known, this uniquely identifies the resulting schedule π.

The optimization criterion is to minimize the maximum lateness:

$$\min_{\pi} \max_{j\in N} \left\{C_j(\pi) - d_j\right\}.$$

Note that, if $d_j = 0$ for all jobs $j \in N$, the objective turns into the makespan criterion.

Problems with $\alpha = P$ and $\alpha = Q$ are special cases of a problem with $\alpha = R$ (if the other parameters in the 3-parameter are identical). A problem with $\alpha = P$ assumes that $p_{ij} = p_j$ for all machines i. If $\alpha = Q$, it is assumed that $p_{ij} = p_i/v_i$, where v_j is the speed of machine i.

3. Preliminaries

In this section, we give some preliminaries necessary for the subsequent considerations.

Definition 1. An instance A of the problem under consideration is defined by $\{G^A, (r_j^A, p_j^A, d_j^A) | \, j \in N\}$, i.e., by the precedence graph and the job parameters. An optimal schedule with minimum maximum lateness value is denoted by π^A, i.e.,

$$L^A_{\max}(\pi^A) = \min_{\pi} \max_{j\in N} L^A_j(\pi),$$

where the minimum is taken over the set of all feasible schedules π.

Definition 2. Let A be an instance and π be some schedule for this instance, satisfying (1). Any chain that is a subgraph of the graph G^A is called a precedence chain and is denoted by $\sigma \subset G^A$.

If $\sigma = (j_1, \dots, j_k) \subset G^A$ is a precedence chain, then by means of formula (1), we have:

$$C^A_{j_k}(\pi) \ge r^A_{j_1} + \sum_{j\in\sigma} p^A_j, \quad (2)$$

where $C^A_j(\pi)$ is the completion time of job $j \in N$ for the data of instance A under the schedule π.

Definition 3. A delaying chain is the precedence chain for which the inequality (2) becomes an equality:

$$C^A_{j_k}(\pi) = r^A_{j_1} + \sum_{j\in\sigma^*(j_k)} p^A_j. \quad (3)$$

It can be noted that a delaying chain exists for any job since we consider only schedules without inserted idle times.

Next, we use an extended definition of the scheduling problems, which have negative values of the release dates and the due dates. This is necessary to consider inverse instances. Then we define inverse instances for the scheduling problems under consideration.

Definition 4. The graphs G_1 and G_2 are called **inverse** if they have the same set of vertices and edges, but the orientation of all edges is opposite. The **inverse precedence relation** to G is denoted by $\overleftarrow{G}$.

Next, we consider the case of identical parallel machines. Recall that for this case, the processing times are $p_{ij} = p_j$. We assume the following relationships between the dates for the instances A and B for all jobs $j \in N$:

$$r^A_j = -d^B_j, \quad p^A_j = p^B_j, \quad d^A_j = -r^B_j. \quad (4)$$

One can see that the values of the release dates and the due dates (4) can be negative. This is because when solving a system of linear equations, solutions can be obtained that are out of the domain of the variables r_j and d_j. In order to find a feasible instance, we construct an inversion to the given one as described below. This is described in more detail further. Now we continue with the definition of inverse instances.

Definition 5. If for an instance $A = \{G^A, \ (r^A_j, p^A_j, d^A_j) | \, j \in N\}$, properties (4) hold and G^A is inverse to G^B, then the instances A and B are called **inverse**.

Permutations and schedules can also be inverse. The **permutation**$\pi_i' = (j_{n_i}, j_{n_i-1}, \dots, j_1)$ being processed on machine i is called **inverse** to the permutation $\pi_i = (j_1, \dots, j_{n_i})$. The **schedule** $\pi' = (\pi_1', \dots, \pi_m')$ with the starting times $\{s_j' \mid j \in N_i, i \in M\}$ is inverse to the schedule $\pi = (\pi_1, \dots, \pi_m)$ with the starting times $\{s_j \mid j \in N_i, i \in M\}$, if all permutations π_i' are inverse to π_i for $i = 1, \dots m$ and $s_j' = -s_j - p_j$ for all jobs $j \in N_i$ and $i \in M$.

Definition 6. For two arbitrary instances A and B of the problems $\{P, Q, R\}|prec, r_j|L_{\max}$, we define the following functions:

$$\begin{cases} \rho_d(A, B) = \max_{j \in N}\{d_j^A - d_j^B\} - \min_{j \in N}\{d_j^A - d_j^B\}; \\ \rho_r(A, B) = \max_{j \in N}\{r_j^A - r_j^B\} - \min_{j \in N}\{r_j^A - r_j^B\}; \\ \rho_p(A, B) = \sum_{j \in N} \left(\max_{i \in M}\{(p_{ij}^A - p_{ij}^B), 0\} - \min_{i \in M}\{(p_{ij}^A - p_{ij}^B), 0\}\right); \\ \rho(A, B) = \rho_d(A, B) + \rho_r(A, B) + \rho_p(A, B). \end{cases} \tag{5}$$

It is necessary to note one important property of the function $\rho(A, B)$: it is separable with respect to the parameters d, r and p.

Definition 7. Let A be an instance with the set of jobs N and the precedence relation G. We say that instance B with the same set of jobs inherits the parameter x from the instance A if $x_j^B = x_j^A$ for all jobs $j \in N$.

Next, we will change the due date of the jobs and invert the graph of the precedence relations G.

4. Basic results

In this section, we present some lemmas and theorems which are required for the subsequent approximation approach.

Lemma 1. *Let the instance B inherit all parameters from the instance A except the due dates $\{d_j | j \in N\}$. Then for any feasible schedule π (for both instances), the following inequality holds:*

$$L_{\max}^B(\pi) - L_{\max}^A(\pi) \le \max_{j \in N}\{d_j^A - d_j^B\}. \tag{6}$$

Proof. Indeed, for any job $j \in N$ and any schedule π, we have:

$$L_{\max}^A(\pi) + \max_{j \in N}\{d_j^A - d_j^B\} \ge C_j(\pi) - d_j^A + d_j^A - d_j^B = C_j(\pi) - d_j^B.$$

Thus,

$$L_{\max}^A(\pi) + \max_{j \in N}\{d_j^A - d_j^B\} \ge \max_{j \in N}\left\{(C_j(\pi) - d_j^B)\right\} = L_{\max}^B(\pi).$$

□

Lemma 2. *Assume that the instance B inherits all parameters from the instance A except the due dates $\{d_j | j \in N\}$, and let $\tilde{\pi}^B$ be an approximate solution for the instance B satisfying the condition*

$$0 \le L_{\max}^B(\tilde{\pi}^B) - L_{\max}^B(\pi^B) \le \delta_B, \tag{7}$$

where π^B is an optimal solution, i.e., for all schedules π it satisfies the condition

$$L_{\max}^B(\pi^B) \le L_{\max}^B(\pi). \tag{8}$$

Then for any optimal schedule π^A of the initial instance A, we obtain

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

Proof. For the schedules π^A and $\tilde{\pi}^B$, we obtain the following estimation:

$$\begin{aligned} 0 &\le L_{\max}^A(\tilde{\pi}^B) - L_{\max}^A(\pi^A) \le \\ &\le L_{\max}^B(\tilde{\pi}^B) - L_{\max}^A(\pi^A) + \max_{j \in N}(d_j^B - d_j^A) \le \\ &\le \delta_B + L_{\max}^B(\pi^B) + \max_{j \in N}(d_j^B - d_j^A) - L_{\max}^B(\pi^A) + \max_{j \in N}(d_j^A - d_j^B) \le \\ &\le \delta_B + \rho_d(A, B). \end{aligned}$$

The second inequality above holds due to (6), the next inequality holds due to (6) and (7), and the latter inequality holds due to (8). The lemma has been proved. □

Definition 8. A feasible schedule π for the instance V is called **early**, if $C_j(\pi) \le d_j^V$ for all jobs $j \in N$. The value $\Delta = \Delta(V, \pi)$ is the minimal value that has to be added to the due dates of all jobs, i.e., $d_j = d_j^V + \Delta$, $j = 1, 2, \dots, n$, so that the schedule π will become early.

Lemma 3. *Let V and W be mutually inverse instances with the job set N and π and π' be mutually inverse permutations. Then we have*

$$L^V_{\max}(\pi) = L^W_{\max}(\pi').$$

Proof. Let $\Delta = \Delta(V, \pi)$. The feasible schedule π is early for the instance $V(\Delta)$ with the due dates $d_j = d^V_j + \Delta$ for all jobs $j = 1, \ldots, n$. Thus, the feasible schedule π' inverse to the schedule π is early for the instance W', inverse to the instance $V(\Delta)$. This means that $L^{W'}_{\max}(\pi') \le 0$. Note that the instance W' is different from the instance W (inverse to V) by the fact that all release dates are reduced by the value Δ, i.e., $r^{W'}_j = r^W_j - \Delta,\ j = 1, \ldots, n$. If the schedule π' is shifted to the right by Δ, then the resulting schedule π'' will be feasible for the instance W, and we get

$$L^W_{\max}(\pi'') \le \Delta = L^V_{\max}(\pi).$$

Since the jobs in the schedule π'' are executed in the order π', we obtain

$$L^W_{\max}(\pi') \le L^W_{\max}(\pi'') \le L^V_{\max}(\pi).$$

Interchanging the instances V and W as well as the sequences π and π', we get the opposite inequality:

$$L^V_{\max}(\pi) \le L^W_{\max}(\pi'),$$

from which the required equality follows:

$$L^V_{\max}(\pi) = L^W_{\max}(\pi').$$

□

Remark 2. If π is an optimal schedule for the instance V, then the inverse schedule π' is an optimal schedule for the inverse schedule W.

Lemma 4. *Let the instance C inherit all parameters from the instance B except the release dates $\{r_j | j \in N\}$, and $\tilde{\pi}^C$ be an approximate solution for the instance C satisfying the condition*

$$0 \le L^C_{\max}(\tilde{\pi}^C) - L^C_{\max}(\pi^C) \le \delta_C. \quad (9)$$

Then

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

Proof. Consider two instances B and C inverse to the instances E and F, respectively, with the job parameters

$$r^E_j = -d^B_j,\ \ p^E_j = p^B_j,\quad d^E_j = -r^B_j,\quad j \in N$$

and

$$r^F_j = -d^C_j,\quad p^F_j = p^C_j,\quad d^F_j = -r^C_j,\quad j \in N.$$

Let τ^E and τ^F be inverse permutations to the permutations π^B and π^C, respectively. According to Remark 2, the permutations τ^E, τ^F are optimal permutations for the instances E and F, respectively. Then, by Lemma 2, for the approximate solution $\tilde{\tau}^F$, we get

$$\delta_C + \rho_d(E, F) \ge L^E_{\max}(\tilde{\tau}^F) - L^E_{\max}(\tau^E) \ge 0.$$

According to Lemma 3, we get

$$L^B_{\max}(\pi^B) = L^E_{\max}(\tau^E) \quad \text{and} \quad L^B_{\max}(\tilde{\pi}^C) = L^E_{\max}(\tilde{\tau}^F).$$

In fact, we use the approximate value $L^B_{\max}(\tilde{\pi}^C)$. Therefore, we obtain the inequality

$$\delta_C + \rho_d(E, F) \ge L^B_{\max}(\tilde{\pi}^C) - L^B_{\max}(\pi^B) \ge 0. \quad (10)$$

We have

$$\begin{aligned} \rho_d(E, F) &= \max_{j \in N}\{d^E_j - d^F_j\} + \max_{j \in N}\{d^F_j - d^E_j\} \\ &= \max_{j \in N}\{r^C_j - r^B_j\} + \max_{j \in N}\{r^B_j - r^C_j\} \\ &= \rho_r(B, C). \end{aligned}$$

From the latter equality and (10), Lemma 4 follows. □

Lemma 5. *Let the instance D inherit all parameters from the instance C except the processing times $\{p_{ij} | i \in M, j \in N\}$, and let $\tilde{\pi}^D$ be an approximate solution of the instance D satisfying the condition*

$$L^D_{\max}(\tilde{\pi}^D) - L^D_{\max}(\pi^D) \le \delta_D. \quad (11)$$

Then

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

Proof. The scheme of the proof is similar to the scheme of Lemma 2, and so we give here only the calculations:

$$\begin{aligned}
&L^C_{\max}(\tilde{\pi}^D) - L^C_{\max}(\pi^C) \le \\
&\le L^D_{\max}(\tilde{\pi}^D) - L^C_{\max}(\pi^C) + \sum_{j\in N} \max_{i\in M}(p^C_{ij} - p^D_{ij})_+ \le \\
&\le \delta_D + L^D_{\max}(\pi^D) + \sum_{j\in N} \max_{i\in M}(p^C_{ij} - p^D_{ij})_+ - L^D_{\max}(\pi^C) + \\
&+ \sum_{j\in N} \max_{i\in M}(p^D_{ij} - p^C_{ij})_+ \le \\
&\le \delta_D + \sum_{j\in N}\left(\max_{i\in M}(p^C_{ij} - p^D_{ij})_+ + \max_{i\in M}(p^C_{ij} - p^D_{ij})_-\right) = \delta_D + \rho_p(C, D).
\end{aligned}$$

□

Theorem 1. *Let the instance D inherit all parameters from the instance A except the values* $\{d_j, r_j, p_{ij} | j \in N, i \in M\}$, *and let* $\tilde{\pi}^D$ *be an approximate solution of the instance D satisfying the condition (11). Then*

$$0 \le L^A_{\max}(\tilde{\pi}^D) - L^A_{\max}(\pi^A) \le \rho(A, D) + \delta_D. \tag{12}$$

Proof. The inequality

$$0 \le L^A_{\max}(\tilde{\pi}^D) - L^A_{\max}(\pi^A)$$

follows from the optimality of the solution π^A for the instance A. Next, we prove the inequality

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

We use the additive property of the estimates obtained in Lemmas 2, 4 and 5. Consider the auxiliary instances B and C with the following job parameters:

$$\begin{array}{ccccccc}
d^A_j, & & d^B_j & = & d^C_j & = & d^D_j; \\
r^A_j & = & r^B_j, & & r^C_j & = & r^D_j; \\
p^A_{ij} & = & p^B_{ij} & = & p^C_{ij}, & & p^D_{ij}.
\end{array}$$

Then we obtain:

$$\rho(A, B) = \rho_d(A, B); \quad \rho(B, C) = \rho_r(B, C); \quad \rho(C, D) = \rho_p(C, D).$$

Applying successively Lemmas 5, 4 and 2, and using

$$\tilde{\pi}^D = \tilde{\pi}^C, \quad \delta_C = \delta_D + \rho_p(C, D), \quad \tilde{\pi}^C = \tilde{\pi}^B, \quad \delta_B = \delta_C + \rho_r(B, C),$$

we get

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

Theorem 1 has been proved. □

5. A metric approach for the problem $P|prec, r_j|L_{\max}$

In this section, we suggest a metric approach for the *NP*-hard problem $P|prec, r_j|L_{\max}$. To do this, we introduce a few basic definitions. Let us define a P-point.

Definition 9. Let $\mathfrak{A}$ be the space, where each point represents the data of an instance of the problem $P|prec, r_j|L_{\max}$. The sub-space $\tilde{\mathfrak{A}} \subset \mathfrak{A}$ is called **P-cone**, if all instances represented by points of this sub-space can be solved by a polynomial or pseudo-polynomial algorithm. These points in $\tilde{\mathfrak{A}}$ are called **P-points**.

There exist several studies devoted to finding P-points for various problems. A rather complete overview of the results about P-points has been given in [21]. An analysis of the graphs related to the problems is also provided, which is not covered in this paper. During the past three years, P-points have been found for the following problems: the classical multi-machine problem [22], the scheduling problem with an agreement graph on two identical machines [23] and with preparation constraints [24], open shops subject to the constraint that some conflicting jobs cannot be processed simultaneously on different machines [25] and job shop problems with tooling constraints [26], as well as some practical applications, for example the multi-league sport scheduling problem [27]. We can conclude that for many problems, there exists a space in which P-points are located. Subsequently, we consider the subspace of P-points. The third part of the monograph in Theoretical Computer Science is devoted to polynomially solvable time-dependent scheduling problems [28]. Three types of P-points for polynomially solvable problems are described for polynomially solvable problems: single machine problems, parallel machine problems, and dedicated machine problems.

Definition 10. Let there be a point (instance) $A \notin \tilde{\mathfrak{A}}$. Using some metric ρ, we can construct a projection onto the space $\tilde{\mathfrak{A}}$ with respect to A. The resulting point (instance) $B \in \tilde{\mathfrak{A}}$ is called the projection of A by the metric ρ.

Definition 11. The sub-space $\tilde{\mathfrak{A}}^{\epsilon}_{\rho}(A) \in \tilde{\mathfrak{A}}$ is called an ϵ–projection of A by the metric ρ if for each of its points $x \in \tilde{\mathfrak{A}}$, the following inequality is satisfied:

$$L^A_{max}(\pi^x) - L^A_{max}(\pi^A) \le \epsilon.$$

The metric approach consists of two steps. In the first step, we change the parameters $\{(r^A_j, p^A_j, d^A_j) | j \in N\}$ of the original instance $A = \{G, (r^A_j, p^A_j, d^A_j)\}$, where $j \in N, A \notin \tilde{\mathfrak{A}}$, so that the projection of A by the metric ρ gives an instance $B = \{G, (r^B_j, p^B_j, d^B_j) | j \in N\}$ in the P-cone. In the next step, we find an optimal schedule π^B for the instance B. According to Theorem 1, we apply the schedule π^B to the initial instance A. As a result, we obtain the following estimate of the absolute error:

$$0 \le L^A_{\max}(\pi^B) - L^A_{\max}(\pi^A) \le \rho(A, B).$$

We consider the P-cone when the parameters of the jobs satisfy the following k linearly independent inequalities:

$$XR + YP + ZD \le H, \tag{13}$$

where $R = (r_1, \dots, r_n)^T$, $P = (p_1, \dots, p_n)^T$ (as usual, we assume $p_j \ge 0$ for all jobs $j \in N$), $D = (d_1, \dots, d_n)^T$ and X, Y, Z are matrices of the dimension $k \times n$, $H = (h_1, \dots, h_k)^T$ is a k-dimensional vector (the upper index ^{T} denotes the transpose operation). Then in the class of instances (13), we determine an instance B with the minimal distance $\rho(A, B)$ to the original instance A by solving the following problem:

$$\begin{cases} (x^d - y^d + x^r - y^r) + \sum_{j \in N} x^p_j \longrightarrow \min \\ y^d \le d^A_j - d^B_j \le x^d, \\ y^r \le r^A_j - r^B_j \le x^r, \\ -x^p_j \le p^A_j - p^B_j \le x^p_j, \\ 0 \le x^p_j, \\ XR^B + YP^B + ZD^B \le H. \end{cases} \quad \forall\ j \in N. \tag{14}$$

The linear programming problem (14) with $3n + 4 + n$ variables (r^B_j, p^B_j, d^B_j, x_d, y_d, x_r, y_r, and x^p_j, $j = 1, \dots, n$) and $7n + k$ inequalities can sometimes be solved with a polynomial number (in n and k) of operations, given the specificity of the constraints of the problem (14). For example, for the problem $1|r_j|L_{\max}$, which is NP–hard in the strong sense, there are two types of non-trivial P-points:

$$\begin{cases} d_1 \le \dots \le d_n, \\ d_1 - r_1 - p_1 \ge \dots \ge d_n - r_n - p_n, \end{cases} \tag{15}$$

and

$$\max_{k \in N}\{d_k - r_k - p_k\} \le d_j - r_j, \tag{16}$$

for all $j \in N$.

In the case of (15), an optimal solution of the problem $1|r_j|L_{\max}$ can be found in $O(n^3 \log n)$ operations. The linear programming problem (14) can be solved in $O(n \log n)$ operations. In the case of (16), an optimal schedule can be found in $O(n^2 \log n)$ operations [29]. As in the case of (15), the minimum absolute error of the maximum lateness can be found in polynomial time, in this case with $O(n)$ operations. In addition, we can present also two new types of P-points for this NP-hard problem:

$$r_i \le r_j \Rightarrow d_i \ge d_j, ;$$

$$d_j - p_j \le d_{\min}(N),$$

and

$$r_i \le r_j \Rightarrow d_i - p_i \ge d_j.$$

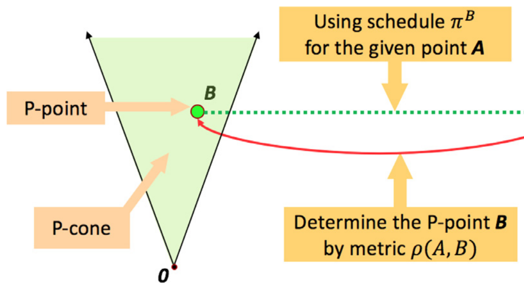

Fig. 1. Geometric illustration of the metric approach.

$i, j \in N, \quad i \neq j$

In the first case, a solution algorithm with the complexity of $O(n^2)$ operations can be presented, in the second case one can give an algorithm with $O(n\log n)$ operations [13].

In the case when the original problem is not projected onto a P-point, the selected objectives (we can note that trivial P-points usually do not give qualitatively new estimates of the absolute error) or the 'distance' $\rho(A, C)$ to any polynomially solvable instance C is not appropriate. However, if for some instance $B = \{G, (r_j^B, p_j^B, d_j^B) | \ j \in N\}$, the estimate of the absolute error of the maximum lateness of the approximate schedule $\tilde{\pi}$ is 'acceptable', then the approximate schedule $\tilde{\pi}$ for the original instance $A = \{G, (r_j^A, p_j^A, d_j^A) | \, j \in N\}$ has a guaranteed absolute error of $\delta^B(\tilde{\pi}) + \rho(A, B)$ from the optimal value of the objective function according to Theorem 1. The value of $\delta^B(\tilde{\pi}) + \rho(A, B)$ is sometimes significantly less than $\rho(A, C)$ for any polynomially or pseudo-polynomially solvable instance C.

It is an upper bound on the absolute error of the objective function value. In Fig. 1, we show the idea of the metric approach. Let us consider the P-cone which includes all known P-points bounded by a system of linear inequalities (note that, since we consider this sub-space with respect to point zero, it is a cone). In this P-cone, we can find a point B, which corresponds to an instance B with the obtained schedule π^B. Thus, for the instance B, there exists a polynomially or pseudo-polynomially solution algorithm.

Now we want to find a polynomially or pseudo-polynomially algorithm for an instance A, which does not belong to the P-cone. Then we construct a metric ρ, which characterizes the difference of the two elements. Here the elements are some functions whose parameters allow finding easily schedules for the instances B and A. Thus, the metric looks like $\rho = L^A_{max}(\pi^B) - L^A_{max}(\pi^A)$. That is, the projection of the initial point A from the $3n$-dimension space onto a point of a feasible sub-space is determined. For any instance (i.e., a point in the $3n$-dimensional space), we know the projection of the initial point A onto the feasible sub-space.

6. The normalized space of instances

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

Definition 12. Two 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 called equivalent, if

$$\exists \; d, r: \quad d_j^A = d_j^B + d, \quad r_j^A = r_j^B + r, \quad p_j^A = p_j^B,$$

for all $j \in N$.

This definition generates a partition of the sets of instances of the problem into equivalence classes. As a representative of each class, we choose the instance with $r_1 = 0$ and $d_1 = 0$. The resulting class is a $(3n-2)$-dimensional linear space. Let us denote this space by $\mathcal{L}_n$. We say that the instance A belongs to the class $\mathcal{L}_n$ if the condition

$$r_1 = d_1 = 0 \tag{17}$$

is satisfied.

Remark 3. Equivalent instances have the same set of optimal schedules (permutations).

Next, we consider the following functionals on the space of equivalence classes of the instances:

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

for all $\ A \in \mathcal{L}_n$.

This functional satisfies the following three properties of a norm:

$$\begin{cases} \varphi(A) = 0 \Longleftrightarrow A \equiv 0; \\ \varphi(\alpha A) = \alpha\varphi(A); \\ \varphi(A+B) \le \varphi(A) + \varphi(B). \end{cases} \tag{18}$$

We have $A = \emptyset$ if $r_j = p_j = d_j = 0$ for all $j \in N$. The first property follows from the definition of the functional $\varphi(A)$. The second one can be checked directly. The third one describes the triangle inequality. The metric functions are separable. This means that any metric function f is representable as the sum of some functions $\phi(a)$ and $\psi(b)$, depending on the initial parameters a and b of function f:

$$f(a, b) = \phi(a) + \psi(b).$$

Thus, $\mathcal{L}_n$ is a $(3n-2)$-dimensional linear normalized space with the norm

$$||A|| = \varphi(A).$$

It should be noted that this rule leads to the metric defined in (5)):

$$\rho(A, B) = ||A - B||.$$

7. Some schemes for finding an approximate solution

Next, we investigate how to use polynomially solvable sub-cases for the solution of more general problems. We show how one can use the above approach for finding an approximate solution of certain scheduling problems with an estimate of the absolute error. For an instance A of a problem $\alpha|\beta|L_{\max}$, we use the notation $\alpha^A|\beta^A|L_{\max}$. It is necessary to change the parameters of the machines and the jobs of the instance A with α^A, β^A and possibly the objective function $L_{\max}$ to $C_{\max}$ to result in an instance B of a problem $\alpha^B|\beta^B|\{L_{\max}, C_{\max}\}$, for which we can find an approximate solution $\tilde{\pi}^B$ (or even an optimal one π^B) and then to apply it to the original instance A of the problem $\alpha^A|\beta^A|L_{\max}$.

In [30–34], a list of subproblems of the problem $P|prec, r_j|L_{\max}$ is given to which various $\alpha = R$ and $\alpha = Q$ can be reduced.

7.1. Reduction scheme $\alpha|\beta|L_{max} \to \alpha|\beta|C_{max}$

Let there be some instance A of a problem $\alpha^A|\beta^A|L_{\max}$ belonging to the class of NP-hard problems and a known approximate schedule $\tilde{\pi}^B$ (or even an optimal one π^B) for the instance B of the problem $\alpha^A|\beta^A|C_{\max}$ with an absolute error not exceeding $\delta_B \ge 0$. In the instance B, we have $d_j^B = 0$ for all jobs $j \in N$ and thus, from Lemma 2, we obtain the following bound:

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

In fact, the obtained estimate allows to estimate the transition from the objective function $L_{\max}$ to the makespan $C_{\max}$.

For an approximate solution of the problem $\alpha|\beta, r_j|L_{\max}$, one can use a solution of the problem $\alpha \mid \beta, r_j = 0 \mid L_{\max}$. In this case, we obtain an estimate of the absolute error of the approximate solution $\tilde{\pi}^B$ according to Lemma 4:

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

This estimate establishes a connection between the problems of minimizing $L_{\max}$ with simultaneous and non-simultaneous release dates of the jobs.

7.2. Reduction scheme $\alpha \mid \beta, p_j \mid L_{\max} \to \alpha \mid \beta, p_j = p \mid L_{\max}$

Let us solve an instance A of the problem $\alpha^A|\beta^A|L_{\max}$ belonging to the class of NP-hard problems, and assume that an approximate schedule $\tilde{\pi}^B$ (or even an optimal one π^B) is known for the instance B of the problem $\alpha^A|\beta^A, p_j = p|L_{\max}$ with an absolute error not exceeding $\delta_B \ge 0$.

In order to guarantee that the absolute error of the obtained solution is as small as possible, it is necessary to find the optimal value of the parameter p, for which the distance $\rho_p(A, B)$ between the instances is minimal:

$$\rho_p(A, B) = \min_p \left\{ \sum_{j=1}^{n} |p_j^A - p| \right\}. \tag{19}$$

This function is a continuous, convex, piecewise linear function with respect to p. If n is odd, then the minimum is reached on the value p_j^A, i.e., if all processing times p_j^A are ordered according to non-decreasing values, then the solution of the problem (19) is $p^* = p^A_{\frac{n+1}{2}}$. If n is even, then the minimum is achieved on the two median values p^A_j as well as on any value between them, i.e., $p^* \in [p^A_{\frac{n}{2}}; p^A_{\frac{n}{2}+1}]$. Thus, $p^* = p^A_{\lfloor\frac{n+1}{2}\rfloor}$ and by Lemma 5, we have

$$0 \le L^A_{\max}(\tilde{\pi}^B) - L^A_{\max}(\pi^A) \le \sum_{j=1}^{n} \left| p^A_j - p^A_{\lfloor\frac{n+1}{2}\rfloor} \right| + \delta_B.$$

Let us consider an instance A of the problem $\alpha^A|\beta^A|L_{\max}$, belonging to the class of NP-hard problems, and assume that an approximate schedule $\tilde{\pi}^B$ (or even an optimal one π^B) is known for the instance B of the problem $\alpha^A|\beta^A, p_j = p|L_{\max}$ with an absolute error not larger than $\delta_B \ge 0$. The condition $p_j = 1$ includes the case of $p_j = p$ when all parameters r^A_j and d^A_j are multiples of p. As p, we can take $p = p^A_{\lfloor\frac{n+1}{2}\rfloor}$, and in order that all values r^A_j and d^A_j are multiples of the number p, we must subtract from them the remainder of the division of these parameters by p. Then $\rho_r(A, B) \le p$ and $\rho_d(A, B) \le p$. For the instance B, we have for all jobs $j \in N$:

$$p^B_j = p = p^A_{\lfloor\frac{n+1}{2}\rfloor}, r^B_j = r^A_j - (r^A_j \ mod \ p), d^B_j = d^A_j - (d^A_j \ mod \ p),$$

and the estimate of the absolute error is given by

$$0 \le L^A_{\max}(\tilde{\pi}^B) - L^A_{\max}(\pi^A) \le \sum_{j\in N} \left| p^A_j - p^A_{\lfloor\frac{n+1}{2}\rfloor} \right| + 2p^A_{\lfloor\frac{n+1}{2}\rfloor}.$$

7.3. Problems with allowed preemptions of jobs

In the problems $\{R, Q\}|\beta|L_{\max}$, in contrast to the problem $P|\beta|L_{\max}$, the processing times of the jobs $j \in N$ on different machines may vary: $p_{ij} \ne p_{kj}$, $i \ne k$, where $i, k \in M$.

Let A be an instance of the problem $R|\beta^A|L_{\max}$ from the class of NP-hard problems and assume that an approximate schedule $\tilde{\pi}^B$ (or even an optimal one π^B) is known for the instance B of the problem $P|\beta^A$, $p_{ij} \in \{p_j, \infty\}|L_{\max}$ with an absolute error not larger than $\delta_B \ge 0$.

In order to guarantee that the absolute error of the obtained solution $\tilde{\pi}^B$ is the smallest one, we need to find the values of the parameters p^B_j which would minimize the distance between the instances A and B:

$$\rho_p(A, B) = \min_{p^B_j} \left\{ \sum_{j\in N} \left(\max_{i\in M}\{(p^A_{ij} - p^B_j), 0\} - \min_{i\in M}\{(p^A_{ij} - p^B_j), 0\} \right) \right\}.$$

We have

$$\rho_p(A, B) = \sum_{j\in N} \left(\max_{i\in M}\{p^A_{ij}, p^B_j\} - \min_{i\in M}\{p^A_{ij}, p^B_j\} \right) = \sum_{j\in N} \left(\max_{i\in M}\{p^A_{ij}\} - \min_{i\in M}\{p^A_{ij}\} \right)$$

provided that

$$p^B_j \in [\min_{i\in M}\{p^A_{ij}\}, \max_{i\in M}\{p^A_{ij}\}]$$

for all $j \in N$. Therefore, we get

$$0 \le L^A_{\max}(\tilde{\pi}^B) - L^A_{\max}(\pi^A) \le \sum_{j\in N} \left(\max_{i\in M}\{p^A_{ij}\} - \min_{i\in M}\{p^A_{ij}\} \right).$$

7.4. Reduction scheme $R|\beta|L_{max} \to Q|\beta|L_{max}$

Let us consider an instance A of the problem $R|\beta|L_{\max}$ ($p_{ij} < \infty$), belonging to the class of NP–hard problems and assume that we know an approximate schedule $\tilde{\pi}^B$ (or even an optimal one π^B) for the instance B of the problem $Q|\beta|L_{\max}$ with an absolute error not exceeding $\delta_B \ge 0$. For the instance B, the processing times of the jobs are calculated by the formula $p_{ij} = \sigma_i p_j$ for all jobs $j \in N$ and $i \in M$, where σ_i represents the speed of machine i.

Now, to find an estimate of the absolute error of the solution $\tilde{\pi}^B$, one needs to solve the problem:

$$\rho_p(A, B) = \min_{p^B_j, \sigma^B_i} \left\{ \sum_{j\in N} \left(\max_{i\in M}\{(p^A_{ij} - p^B_j\sigma^B_i), 0\} - \min_{i\in M}\{(p^A_{ij} - p^B_j\sigma^B_i), 0\} \right) \right\}.$$

This problem can be written as a linear programming problem:

$$\begin{cases} \min\limits_{\alpha_j, \beta_j, p^B_j, \sigma^B_i} \left\{ \sum\limits_{j\in N} (\alpha_j - \beta_j) \right\} \\ \beta_j \le p^A_{ij} - \sigma^B_i p^B_j \le \alpha_j, \qquad i \in M, j \in N; \\ \beta_j \le 0 \le \alpha_j, \qquad j \in N. \end{cases} \tag{20}$$

7.5. Reduction scheme $\alpha \mid \beta \mid L_{\max} \to \alpha|\beta, p_j \in \{p_1, \ldots, p_k\}|C_{\max}$

First, let us consider the problem with a single machine. Assume that we have an instance A of the problem $\alpha^A|\beta^A|L_{\max}$ from the class of NP-hard problems, and that an approximate schedule $\tilde{\pi}^B$ (or even an optimal one π^B) for the instance B is known for the problem $\alpha^A|\beta^A, p_j \in \{p_1^B, \ldots, p_k^B\}|C_{\max}$ with an absolute error not larger than $\delta_B \geq 0$. In this case, we get the following estimate of the absolute error:

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

where

$$\rho_d(A, B) = \max_{j \in N}\{d_j^A\} - \min_{j \in N}\{d_j^A\}$$

and to find $\rho_p(A, B)$, it is necessary to solve the following problem:

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

Recall that the problem $1|r_j|L_{\max}$ is NP-hard in the strong sense [1]. An approach for finding approximate solutions has been given in [35]. The more complex problem $1|prec, r_j|L_{\max}$ is also NP-hard in the strong sence. Polynomially solvable subcases of this problem are given in [36]. The problem under consideration can be approximately solved by applying one of the above two algorithms, depending on for which the absolute error is less.

Let us now consider the scheduling problems with two parallel machines. The problem $P2|chains|C_{\max}$ is NP-hard in the strong sense [31]. There exist polynomial solution algorithms for similar problems as given in [37]. For an arbitrary number of parallel machines, NP-hard problems in the strong sense have been given in [30,32,34]. On the other hand, polynomially solvable cases can be found in [30,38]. There are no exact polynomially solvable cases for problems with arbitrary precedence relations. In our opinion, the question of modifying the precedence graph and obtaining error estimates as a result of such a modification is important.

7.6. Flow-shop problems

Finally, we only briefly note that this approach can also be applied to construct an approximate solution and to obtain an estimate of the absolute error for some flow-shop problems [39]. In a flow-shop problem, each of the jobs has to be processed on the machines in the same order from machine 1 up to machine m. It is well known that for a problem with up to three machines and a regular optimization criterion, there exists an optimal permutation schedule with the same job sequence π. For a general flow-shop problem, one has to consider also different job sequences on the machines when looking for an optimal solution. For applying the approach presented, one has to use the known polynomially or pseudo-polynomially solvable problems. In particular, we note that NP–hard problems in the strong sense are given in [32,33,40,41]. On the other hand, polynomially solvable cases can be found in [42].

8. Computational results for the problem $1|r_j|L_{\max}$

In this section, we present some computational results for three polynomial algorithms: the algorithms by Schrage [43], by Hoogeveen [29] and the metric approach [12,44] for solving randomly generated instances of the problem $1|r_j|L_{max}$ using the following three indicators. The parameter μ determines the percentage of instances for which the algorithm has found an optimal solution. It is calculated by the formula $\mu = \frac{K^* \cdot 100\%}{K}$, where K is the total number of generated instances, and K^* is the number instances optimally solved by the corresponding algorithm.

The parameters β_{av} and β_{max} determine the average and maximum relative error, respectively, of the value of the objective function of the schedule obtained by the corresponding algorithm. The parameters β_{av} and β_{max} are calculated by the following formulas:

$$\beta_{av} = \sum_{i=1}^{\bar{K}} \frac{L_{max}(\pi_i) - L_{max}(\pi^*)}{L_{max}(\pi^*)} \cdot 100\%;$$

$$\beta_{max} = \max_{i=1,\ldots,\bar{K}} \left\{ \frac{L_{max}(\pi_i) - L_{max}(\pi^*)}{L_{max}(\pi^*)} \cdot 100\% \right\},$$

where $\bar{K}$ is the number of generated instances for which the solution obtained by the corresponding algorithm was not optimal ($\bar{K} = K - K^*$), π_i is the schedule obtained by the corresponding approximation algorithm and π_i^* is an optimal schedule for the ith generated instance which was not solved optimally. In our study, we experimentally determined the parameters μ, β_{av} and β_{max} for the three polynomial algorithms mentioned above.

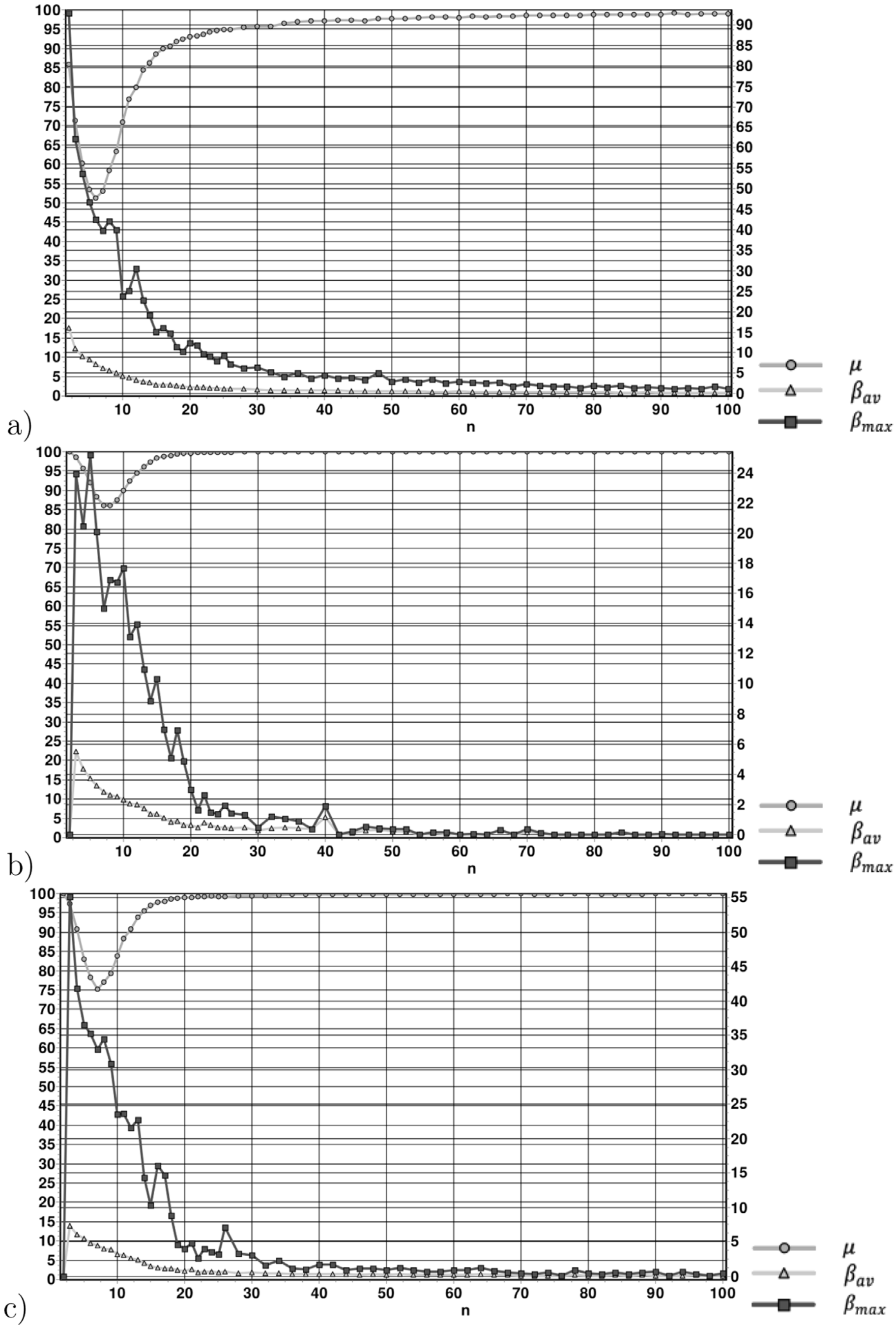

Fig. 2. (a) Schrage algorithm; (b) Hoogeveen algorithm; (c) metric approach.

We generated 10,000 instances for each number n of jobs considered in the range from 2 to 100 according to an $N(0, 1)$ normal distribution on the $3n$-dimensional unit sphere. In the experiments, we changed only the release dates and the due dates. The processing times remained unchanged for all instances.

Fig. 2 shows the dependence of the investigated parameters on the number of jobs n for the Schrage algorithm (Fig. 2a), for the Hoogeveen algorithm (Fig. 2b) and for the metric approach (Fig. 2c). The algorithms by Schrage and Hoogeveen are classical approximation algorithms with guaranteed accuracy estimation.

It turns out that the parameter μ has a similar dependence for all three algorithms. For a number of jobs $n \le 5$, the parameter μ decreases with an increase in the number of jobs. However, starting with $n \in \{6, 7\}$, it begins to increase and tends to 100%.

For example, for numbers n greater than 30, the metric approach finds an optimal solution for more than 99.7% of the instances, and the algorithm by Hoogeveen finds an optimal solution for more than 99.9% of the instances. Of course, there are classes of instances that cannot be optimally solved by these polynomial algorithms. However, if we take into account the whole set of instances of large dimension, the probability of finding a 'hard' instance is very small. If we consider the parameters β_{av} and β_{max}, then we can see that the average and maximum relative error of the solutions obtained by the algorithms decreases with an increase in the number of jobs. Again, the algorithm by Hoogeveen shows the best results.

As can be seen from the graphs the relative error did not exceed some threshold level of the dimension of instances.

The average relative error for this algorithm does not exceed 1% for $n \geq 17$ jobs, and the maximum relative error did not exceed 2% in our experiments for instances with $n \geq 23$. For the metric approach, this holds 1% for $n \geq 19$ and 2% for $n \geq 36$, respectively. As can be seen from Fig. 2, the relative error did not exceed some threshold value of the dimension of the instances.

Based on the results of our experimental study, we conjecture that the percentage of 'hard' instances in relation to the entire set of instances of the problem $1|r_j|L_{max}$ decreases with increasing dimension. This may explain the fact that in practice, exact non-polynomial algorithms can quickly find an optimal solution for large instances despite the *NP*-hardness of the problem [45,46].

According to our experiments, the metric approach appears to be effective for most of the tested instances of the problem $1|r_j|L_{max}$. We conjecture that the metric approach will be also effective for various types of multi-machine problems. This investigation will be a subject of future work.

9. Concluding remarks

In this paper, we presented an approach for solving particular scheduling problems with minimizing the maximum penalty approximately which also gives an estimate for the maximum absolute error. This approach is based on introducing an appropriate metric and using polynomially or pseudo-polynomially solvable sub-cases of the problems. We described this approach in detail for the problem with identical parallel machines, given release dates and minimizing maximum lateness and sketched the application to other scheduling problems briefly.

In this paper, a new concept of metrics was introduced and algorithmically used for the class of problems under consideration. For the first time, an approach of obtaining a (pseudo-) polynomial solution of the problem considered using a projection from the original instance to a set of solvable instances by means of a metric was constructed. New schemes for reducing scheduling problems from more complex one to less complex ones have been formalized. In this way, the metric approach allowed us to obtain an approximate value of the objective function together with an estimation of the absolute error for an NP-hard problem in (pseudo-) polynomial time.

In addition, we also conducted experiments for the single machine problem $1|r_j|L_{\max}$. It turned out that the metric approach obtained promising results. For future work, we plan to make detailed computational tests for several parallel machine problems.

To apply the suggested approach to a broader class of problems, it is important to find new polynomially solvable special cases of *NP*-hard problems. The obtained estimates can be used for solving the problems by such methods as branch and bound and its variations (constraint programming, branch and cut, branch and price, etc.). We note that it is possible to construct metrics for any scheduling problem with a separable objective function.

Currently, the authors are working on the use of the metric approach for some practical problems. Among them, we mention the problem of railway planning for a single-track way, the problem of scheduling the agricultural machinery to combat insects, the problem of monitoring operating rooms in hospitals, and a problem of data transfer between processors. Unfortunately, all these results are protected by a Non-disclosure agreement. Preliminary experiments have shown that the metric approach can be used to reduce the number of iterations in branch and bound algorithms and the domains in constraint programming.

Acknowledgments

This research was partially supported by RFBR (20-58-S52006) and the Basic Research Program of the National Research University Higher School of Economics.

References

[1] J. Lenstra, A. Rinnooy Kan, P. Brucker, Complexity of machine scheduling problems, Ann. Discr. Math., 1 (1977) 343–362.
[2] P. Brucker, S. Knust, Complex scheduling Springer-Verlag Berlin Heidelberg, Eur. J. Oper. Res. 169 (2) (2011) 638–653.
[3] S. Umetani, Y. Fukushima, H. Morita, A linear programming based heuristic algorithm for charge and discharge scheduling of electric vehicles in a building energy management system, Omega, 67 (2017) 115–122.
[4] M. Rasti-Barzoki, S. Hejazi, Pseudo-polynomial dynamic programming for an integrated due date assignment, resource allocation, production, and distribution scheduling model in supply chain scheduling, Appl. Math. Model. 39 (12) (2015) 3280–3289.
[5] P. Fattahi, S. Hosseini, F. Jolai, R. Tavakkoli-Moghaddam, A branch and bound algorithm for hybrid flow shop scheduling problem with setup time and assembly operations, Appl. Math. Model. 38 (1) (2014) 119–134.
[6] D. Lemtyuzhnikova, V. Leonov, Large–scale problems with quasi–block matrices, J. Comput. Syst. Sci. Int. 58 (4) (2019) 571–578.
[7] F. Werner, A survey of genetic algorithms for shop scheduling problems, in: Heuristics: Theory and Applications, Nova Science Publishers, 2013, p. 161-222.

[8] L. Zuo, L. Shu, S. Dong, C. Zhu, T. Hara, A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing, IEEE Access, 3 (2015) 2687–2699.
[9] Q. Pan, An effective co-evolutionary artificial bee colony algorithm for steelmaking-continuous casting scheduling, Eur. J. Oper. Res. 250 (3) (2016) 702–714.
[10] V. Sels, J. Coelho, A. Dias, M. Vanhoucke, Hybrid tabu search and a truncated branch-and-bound for the unrelated parallel machine scheduling problem, Comput. Oper. Res. 53 (2015) 107–117.
[11] Z. Jia, L. Gao, X. Zhang, A new history-guided multi-objective evolutionary algorithm based on decomposition for batching scheduling, Expert Syst. Appl. 141 (2020) 112920.
[12] A. Lazarev, Estimation of absolute error in scheduling problems of minimizing the maximum lateness, Dokl. Math., 76 (2007) 572–574.
[13] A. Lazarev, Scheduling theory, in: Methods and Algorithms (in Russian), Institue of Control Sciences of the Russian Academy of Sciences, Moscow / Russia, 2019, pp. 1–408.
[14] A. Lazarev, D. Arkhipov, Estimation of the absolute error and polynomial solvability for a classical *NP*–hard scheduling problem, Dokl. Math. 97 (3) (2018) 262–265.
[15] A. Lazarev, P. Korenev, A. Sologub, A metric for total tardiness minimization, Autom. Remote Control 78 (4) (2017) 732–740.
[16] A. Lazarev, A. Kvaratskheliya, Metrics in scheduling problems, Dokl. Math. 81 (3) (2010) 497–499.
[17] L. Tang, H. Gong, The coordination of transportation and batching scheduling, Appl. Math. Model. 33 (10) (2009) 3854–3862.
[18] T. Chung, J. Gupta, H. Zhao, F. Werner, Minimizing the makespan on two identical parallel machines with mold constraints, Comput. Oper. Res. 105 (2019) 141–155.
[19] K. Hasani, S. Kravchenko, F. Werner, Minimizing the makespan for the two-machine scheduling problem with a single server: two algorithms for very large instances, Eng. Optim. 48 (1) (2016) 173–183.
[20] R. Graham, E. Lawler, J. Lenstra, A. Kan, Optimization and approximation in deterministic sequencing and scheduling: a survey, Ann. Discr. Math. 5 (1979) 287–326.
[21] P. Damien, O. Bellenguez-Morineau, A survey on how the structure of precedence constraints may change the complexity class of scheduling problems, J. Sched. 21 (1) (2018) 3–16.
[22] X. Yue, J. Gao, Z. Chen, A polynomial time algorithm for scheduling on processing time constraints, in: Proceedings of the 2019 the Ninth International Conference on Communication and Network Security, 2019.
[23] A. Mohabeddine, M. Boudhar, New results in two identical machines scheduling with agreement graphs, Theor. Comput. Sci., 779 (2019) 37–46.
[24] W. Labbi, B. Mourad, O. Ammar, Scheduling two identical parallel machines with preparation constraints, Int. J. Prod. Res. 55 (6) (2017) 1531–1548.
[25] N. Tellache, M. Boudhar, Open shop scheduling problems with conflict graphs, Discr. Appl. Math. 227 (2017) 103–120.
[26] L. Libralesso, V. Jost, K.H. Salem, F. Fontan, F. Maffray, Study on partial flexible job-shop scheduling problem under tooling constraints: complexity and related problems, HAL-02183503 (see also https://hal.archives-ouvertes.fr/hal-02183503/document, 2019.
[27] M. Davari, J. Belien, D. Goossens, F. Spieksma, Polynomial algorithms for the multi-league sport scheduling problem, in: Proceedings of the 33rd Annual conference of the Belgian Operations Research Society (ORBEL33), HasseltBelgium, 2019, pp. 101–103.
[28] S. Gawiejnowicz, Models and Algorithms of Time-Dependent Scheduling, Springer, Berlin, 2019.
[29] J. Hoogeveen, Minimizing maximum promptness and maximum lateness on a single machine, Math. Oper. Res., 21 (1996) 100–114.
[30] P. Brucker, M. Garey, D. Johnson, Scheduling equal-length tasks under treelike precedence constraints to minimize maximum lateness, Math. Oper. Res. 2 (3) (1977) 275–284.
[31] J. Du, J. Leung, G. Young, Scheduling chain-structured tasks to minimize makespan and mean flow time, Inform. Comput. 92 (2) (1991) 219–236.
[32] M. Garey, D. Johnson, "strong" *NP*–completeness results:motivation, examples, and implications, J. Assoc. Comput. Mach. 25 (3) (1978) 499–508.
[33] J. Lenstra, A. Rinnooy Kan, P. Brucker, Complexity of machine scheduling problems, Eur. J. Oper. Res. 4 (1980) 270–275.
[34] J. Ullman, *NP*–complete scheduling problems, J. Comput. Syst. Sci. 10 (1975) 384–393.
[35] A. Lazarev, R. Sadikov, S. Sevastyanov, Approximate solution scheme of $1|r_j|l_{\max}$ problem, Discr. Anal. Oper. Res. 13 (1) (2006) 57–76.
[36] E. Lawler, Optimal sequencing of a single machine subject to precedence constraints, Manag. Sci. 19 (5) (1973) 544–546.
[37] P. Brucker, J. Hurink, W. Kubiak, Scheduling identical jobs with chain precedence constraints on two uniform machines, Math. Methods Oper. Res. 49 (2) (1999) 211–219.
[38] B. Simons, Multiprocessor scheduling of unit-time jobs with arbitrary release times and deadlines, SIAM J. Comput. 12 (2) (1983) 294–299.
[39] E. Birgin, J. Ferreira, D. Ronconi, A filtered beam search method for the m-machine permutation flowshop scheduling problem minimizing the earliness and tardiness penalties and the waiting time of the jobs, Comput. Oper. Res. 114 (2020) 104824.
[40] J. Leung, O. Vornberger, J. Witthoff, On some variants of the bandwidth minimization problem, SIAM J. Comput. 13 (3) (1984) 650–667.
[41] V. Timkovsky, Identical parallel machines vs. unit-time shops and preemptions vs. chains in scheduling complexity, Eur. J. Oper. Res. 149 (2) (2003) 355–376.
[42] J. Bruno, J. Jones, K. So, Deterministic scheduling with pipelined processors, IEEE Trans. Comput. 29 (4) (1980) 308–316.
[43] L. Schrage, Solving resource-constrained network problems by implicit enumeration: non preemptive case, Oper. Res. 18 (1970) 263–278.
[44] A. Lazarev, The pareto-optimal set of the NP-hard problem of minimization of the maximum lateness for a single machine, J. Comput. Syst. Sci. Int. 45 (6) (2006) 943–949.
[45] J. Carlier, The one-machine sequencing problem, Eur. J. Oper. Res. 11 (1) (1982) 42–47.
[46] B. Lageweg, J. Lenstra, A. Kan, Minimizing maximum lateness on one machine: computational experience and applications, Statistica Neerlandica 30 (1976) 25–41.