

Article

Approximation of the Objective Function of Single-Machine Scheduling Problem

Alexander Lazarev ¹, Nikolay Pravdivets ^{1,2,*} and Egor Barashov ¹

¹ V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, 65 Profsoyuznaya Street, 117997 Moscow, Russia; jobmath@mail.ru (A.L.); barashov.eb@gmail.com (E.B.)

² Department of Mathematics, Faculty of Economic Sciences, HSE University, 11 Pokrovsky Boulevard, 109028 Moscow, Russia

* Correspondence: pravdivets@ipu.ru

Abstract: The problem of the approximation of the coefficients of the objective function of a scheduling problem for a single machine is considered. It is necessary to minimize the total weighted completion times of jobs with unknown weight coefficients when a set of problem instances with known optimal schedules is given. It is shown that the approximation problem can be reduced to finding a solution to a system of linear inequalities for weight coefficients. For the case of simultaneous job release times, a method for solving the corresponding system of inequalities has been developed. Based on it, a polynomial algorithm for finding values of weight coefficients that satisfy the given optimal schedules was constructed. The complexity of the algorithm is $O(n^2(N + n))$ operations, where n is the number of jobs and N is the number of given instances with known optimal schedules. The accuracy of the algorithm is estimated by experimentally measuring the function $\varepsilon(N, n) = \frac{1}{n} \sum_{j=1}^n \frac{|w_j - w_j^0|}{w_j^0}$, which is an indicator of the average modulus of the relative deviation of the found values w_j from the true values w_j^0 . An analysis of the results shows a high correlation between the dependence $\varepsilon(N, n)$ and a function of the form $\alpha(n)/N$, where $\alpha(n)$ is a decreasing function of n .

Keywords: scheduling theory; single machine scheduling; total weighted completion times; approximation

MSC: 90B35



Citation: Lazarev, A.; Pravdivets, N.; Barashov, E. Approximation of the Objective Function of Single-Machine Scheduling Problem. *Mathematics* **2024**, *12*, 699. <https://doi.org/10.3390/math12050699>

Academic Editor: Javier Alcaraz

Received: 31 January 2024

Revised: 21 February 2024

Accepted: 23 February 2024

Published: 28 February 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Scheduling theory is a branch of discrete optimization devoted to planning operations over time. The problems of scheduling theory are diverse, including, but not limited to, production scheduling, including human-robot collaborations [1], classical Resource-Constrained Project Scheduling Problem (RCPSP) [2], creating smart planning eco-systems [3], distributed scheduling problems [4], and dynamic systems [5]. However, classical problems for a single machine remain relevant, still capture the attention of the scientific community [6] and are also used in modern approaches of Industry 4.0 [7].

In the classical single-machine scheduling problem, there is a set of jobs with given release times, processing times and due dates. The goal is to schedule the jobs to proceed on the machine, minimizing some objective functions. A wide variety of studies of this problem can be found for such objective functions as the total or maximum lateness [8,9], the weighted number of tardy jobs [10], the total (weighted) completion time [11,12] or any arbitrary non-decreasing function of the completion time [13]. The idea is that the objective function is known and should be maximized or minimized.

In practice, the quality criterion and, therefore, the objective function are not defined for some applied scheduling problems. Let some optimal schedules be pre-given. The goal is to construct an optimal schedule for the new input data and to estimate unknown objective function. To apply the methods of scheduling theory to this problem, it is necessary

to determine the objective function. The task of the objective function approximation arises as a sub-task in the scheduling automation process, so the approximation algorithm must be fast enough compared to the required frequency of scheduling. An indicator of the quality of the approximation algorithm is the quality of the schedule obtained for the approximated objective function.

A linear approximation for a single-machine scheduling problem is considered: it is assumed that there exists an objective function that is linear in job-weighted completion times, and the previously constructed schedules are optimal with respect to this objective function. The initial problem is reduced to solving a system of linear inequalities with respect to the unknown values of the weight coefficients of the objective function. A detailed review of the results of the algebraic theory of linear inequalities was given by [14], which also includes an algorithm for finding a general formula for non-negative solutions of [15], based on the principle of boundary solutions. General results on this topic were also presented by [16,17], who also used some properties of convex cones in the solution space. Refs. [18,19] obtained results for a set of solutions (redundancy and dimension) of a certain class of systems of inequalities—normal systems.

The main contribution of this paper is an algorithm for approximating the weight coefficients for the case of simultaneous job release times. The system of linear inequalities in this case has a sparse matrix (most of the elements are zero), which contains a significant number of dependent inequalities. The solution method is primarily based on the exclusion of dependent inequalities from the system and using some general properties of linear inequality systems.

The structure of this paper is organized as follows. In Section 2 the mathematical formulation of the problem $1 \parallel \sum w_j C_j$ will be given and thereupon mathematical statement of the problem of approximation of objective function weight coefficients is formulated; in Section 3 a method for solving the problem is proposed; in Section 4 a numerical study of the constructed approximation algorithm is carried out and its final complexity is calculated. Section 5 contains a brief conclusion.

2. Mathematical Problem Formulation

The problem studied in this article is in some way the inverse of one of the classic problems in scheduling theory, the problem of minimizing the total weighted completion time $1 \mid r_j \mid \sum w_j C_j$. To describe the main problem, problem $1 \mid r_j \mid \sum w_j C_j$ should be introduced first. There is a single machine and a set $J = \{1, 2, \dots, n\}$ of n jobs that need to be processed on the machine. For each job $j \in J$, the release time r_j and processing time p_j are given. There are no precedence relations of jobs, and interruptions of the job processing are prohibited. The order in which the jobs $(j_1, \dots, j_n)$ are processed is called a schedule. In problem $1 \mid r_j \mid \sum w_j C_j$, it is necessary to find a schedule π^0 minimizing the total weighted completion time $\sum w_j C_j$, where C_j is the completion time of job j , and $w_j > 0$ is a weight coefficient of the completion time of the corresponding job j . Completion times of jobs under schedule $\pi = (j_1, \dots, j_n)$ are defined as follows:

$$C_{j_1}(\pi) = r_{j_1} + p_{j_1};$$

$$C_{j_k}(\pi) = \max\{r_{j_k}, C_{j_{k-1}}(\pi)\} + p_{j_k}, k = 2, \dots, n.$$

Remark 1. The case of problem $1 \mid r_j \mid \sum w_j C_j$ with zero weight coefficients can be solved trivially: all corresponding jobs are processed last and are excluded from the consideration. The case with negative weight coefficients does not make sense from a practical point of view. In this regard, it is assumed in this paper that all weight coefficients are positive.

Remark 2. Note, that the problem $1 \mid r_j \mid \sum w_j L_j$ about the total weighted lateness $L_j = C_j - d_j$, where d_j is the due date for job $j \in J$, is equivalent to problem $1 \mid r_j \mid \sum w_j C_j$ and the optimal schedule π^0 in problem $1 \mid r_j \mid \sum w_j L_j$ does not depend on the values of due dates d_j . Further,

the problem $1 \mid r_j \mid \sum w_j C_j$ will be considered, implying that all the results obtained can be applied to the problem $1 \mid r_j \mid \sum w_j L_j$.

Definition 1. A set of job release times and processing times $I = \{r_1, \dots, r_n, p_1, \dots, p_n\}$ is called a problem instance I of problem $1 \mid r_j \mid \sum w_j C_j$.

The general case of problem $1 \mid r_j \mid \sum w_j C_j$ with weight coefficients w_j is NP-hard in the strong sense [20]. An analysis of the approaches and methods for solving single-machine scheduling problems can be found in [21]. According to generalized Smith theorem [22], a polynomially solvable special case of problem $1 \mid r_j \mid \sum w_j C_j$ is problem $1 \parallel \sum w_j C_j$, in which, despite the general case, a simultaneous release times $r_1 = \dots = r_n = r$ are assumed. An optimal schedule in this case will be a schedule constructed in order of non-decreasing values $p_j/w_j, j \in J$.

Definition 2. A set of job processing times $I = \{p_1, \dots, p_n\}$ is called a problem instance I of problem $1 \parallel \sum w_j C_j$.

Now, the problem of approximation of objective function weight coefficients can be formulated as follows.

Problem 1. N instances $I_k, k = 1, \dots, N$, of the problem $1 \parallel \sum w_j C_j$ (or the problem $1 \mid r_j \mid \sum w_j C_j$ in the general case) are given, and corresponding optimal schedules π_k^0 are known. The goal is to approximate values of unknown weight coefficients $w_j, j \in J$.

Thus, the problem considered in this paper is inverse to the problem $1 \parallel \sum w_j C_j$. A linear approximation is considered for the problem for a single machine: it is assumed that there is an objective function that is linear with respect to the completion times, the optimal schedule is known, but the weight coefficients of the objective function are unknown. This problem may arise, for example, at the initial stages of automating individual processes at enterprises where there is a long experience in scheduling “manually”, but modeling the entire production process and the analytical formulation of the optimality criterion is, for some reason, impossible or not obvious.

3. Approximation Problem Solving Method

The method for solving the problem of approximating weight coefficients (both for general and for particular cases) is based on determining the optimality of schedules $\pi_k^0, k = 1, 2, \dots, N$:

$$\sum_{j=1}^n C_j^k(\pi) w_j \geq \sum_{j=1}^n C_j^k(\pi_k^0) w_j, \quad \forall \pi.$$

Therefore,

$$\sum_{j=1}^n \left(C_j^k(\pi) - C_j^k(\pi_k^0) \right) w_j \geq 0, \quad \forall \pi, k = 1, 2, \dots, N. \quad (1)$$

Thus, the values of w_j are generally determined by a system of $N(n! - 1)$ inequalities of the form (1), i.e., the dependence of the number of inequalities in the system on the number of jobs is not polynomial, because for any instance there are $n!$ possible schedules. The non-polynomial complexity of the approximation algorithm can be avoided if among the $N(n! - 1)$ inequalities there is a polynomial number m of independent ones, while the remaining inequalities are consequences of these m inequalities.

In case $N = 1$ the following approximation problem can be considered: one instance I of problem $1 \mid r_j \mid \sum w_j C_j$ of dimension n is given, i.e., for n jobs, the release times r_j and

processing times p_j are defined, and the optimal schedule π^0 is also known. Then, for an arbitrary acceptable schedule π the following inequality holds:

$$\sum_{j=1}^n (C_j(\pi) - C_j(\pi^0)) w_j \geq 0. \quad (2)$$

Repeating the arguments about the non-polynomial number of inequalities, the following proposition can be formulated. The proposition is sufficient for the polynomial approximation algorithm.

Proposition 1. For an arbitrary instance I of problem $1 \mid r_j \mid \sum w_j C_j$ in a system of $(n! - 1)$ inequalities of the form (2) there is a subsystem of m inequalities, the solution of which coincides with the solution of the initial system, and the number m depends polynomially on n .

Each of the strict inequalities of the form (2) corresponds to some non-optimal schedule π . In this case, Proposition 1 means that among $(n! - 1)$ non-optimal schedules there are m schedules $\Pi = \{\pi_1, \dots, \pi_m\}$ such that the solution of this system of inequalities of the form (2) corresponding to the schedules π_i , $i = 1, \dots, m$, coincides with the solution of the initial system, and the number m depends polynomially on n . In other words, for any schedule $\pi \notin \Pi$, an inequality of the form (2) corresponding to schedule π is a consequence of the system of inequalities corresponding to schedules π_i , $i = 1, \dots, m$. Finding proof (or inconsistency) of the Proposition 1 is the key to solving the approximation problem for the $1 \mid r_j \mid \sum w_j C_j$.

3.1. The Initial and Efficient System of Inequalities for the Problem $1 \mid \sum w_j C_j$

Consider approximating the weight coefficients of the problem $1 \mid \sum w_j C_j$. According to the generalized Smith theorem [22], for problem instance $I = \{p_1, \dots, p_n\}$ of $1 \mid \sum w_j C_j$ there exists an optimal schedule $\pi^* = (j_1, j_2, \dots, j_n)$, for which

$$\frac{p_{j_1}}{w_{j_1}} \leq \frac{p_{j_2}}{w_{j_2}} \leq \dots \leq \frac{p_{j_n}}{w_{j_n}}. \quad (3)$$

Remark 3. If one or more adjacent non-strict inequalities of set (3) turn into equality, that is, the ratios $\frac{p_j}{w_j}$ are equal for jobs $j \in \{j_k, \dots, j_{k+l}\}$, all those schedules $\tilde{\pi}^*$ are also optimal, in which the jobs $j_k, \dots, j_{k+l}$ are placed in any other arbitrary order $\sigma(j_k, \dots, j_{k+l})$:

$$\begin{aligned} \pi^* &= (j_1, \dots, j_{k-1}, j_k, \dots, j_{k+l}, j_{k+l+1}, \dots, j_n), \\ \tilde{\pi}^* &= (j_1, \dots, j_{k-1}, \sigma(j_k, \dots, j_{k+l}), j_{k+l+1}, \dots, j_n), \\ \sum w_j C_j(\tilde{\pi}^*) &= \sum w_j C_j(\pi^*) = \min_{\pi} \sum w_j C_j(\pi). \end{aligned}$$

An inequality of the form (3) is valid for all N given instances $I_k = \{p_1^k, \dots, p_n^k\}$ of the problem and the corresponding optimal schedules $\pi_k^0 = (j_1^k, j_2^k, \dots, j_n^k)$, $k = 1, \dots, N$.

Definition 3. The system of inequalities for weight coefficients w_j

$$\frac{p_{j_1}^k}{w_{j_1^k}} \leq \frac{p_{j_2}^k}{w_{j_2^k}} \leq \dots \leq \frac{p_{j_n}^k}{w_{j_n^k}}, \quad k = 1, \dots, N, \quad (4)$$

we will call the **initial system** of inequalities of the problem of approximation of weight coefficients for the case $r_1 = \dots = r_n$.

The initial system (4) contains $N(n-1)$ inequalities. To reconstruct the system to a more convenient form for solving, the following notation will be used.

Let $K = \{1, \dots, N\}$ be the set of indices $k \in K$ corresponding to the given pairs (I_k, π_k^0) of problem instances and their optimal schedules, and let $\tilde{K} \subset K$ be some subset of the set K . Further, the record of the form

$$\min_{k \in \tilde{K}} \text{ (or } \max_{k \in \tilde{K}})$$

will mean the minimum (maximum) for all possible pairs (I_k, π_k^0) such that $k \in \tilde{K}$.

For an arbitrary pair of jobs $i, j \in \{1, \dots, n\}$, $i \neq j$, the set K can be divided into two subsets $K_{i,j}$ and $K_{j,i}$ depending on the relative position of the jobs i, j in the schedule π_k^0 :

$$K_{i,j} = \{k \in K : \pi_k^0 = (\dots, i, \dots, j, \dots)\},$$

$$K_{j,i} = \{k \in K : \pi_k^0 = (\dots, j, \dots, i, \dots)\}.$$

Then, from the inequalities (4) of the initial system for the corresponding weight coefficients w_i, w_j the following inequalities can be constructed:

$$\frac{p_i^k}{w_i} \leq \frac{p_j^k}{w_j}, \quad k \in K_{i,j},$$

$$\frac{p_j^k}{w_j} \leq \frac{p_i^k}{w_i}, \quad k \in K_{j,i},$$

or, more conveniently,

$$\frac{w_j}{w_i} \leq \frac{p_j^k}{p_i^k}, \quad k \in K_{i,j}, \quad (5a)$$

$$\frac{w_j}{w_i} \geq \frac{p_j^k}{p_i^k}, \quad k \in K_{j,i}. \quad (5b)$$

Remark 4. Obviously, $K_{i,j} \cap K_{j,i} = \emptyset$, $K_{i,j} \cup K_{j,i} = K$ for all $i, j \in J$, $i \neq j$. Consequently, in inequalities (5a) and (5b) all inequalities of the initial system related to w_i, w_j occurred.

Let

$$Y(i, j) = \min_{k \in K_{i,j}} \left(\frac{p_j^k}{p_i^k} \right),$$

$$X(i, j) = \max_{k \in K_{j,i}} \left(\frac{p_j^k}{p_i^k} \right),$$

then the system of inequalities (5a) and (5b) for the chosen i, j is equivalent to the double inequality:

$$X(i, j) \leq \frac{w_j}{w_i} \leq Y(i, j). \quad (6)$$

Remark 5. Consider the case when one of the sets $K_{i,j}, K_{j,i}$ is empty. Let, for example, $K_{i,j} = \emptyset$, that is, there were no inequalities of the form (5a) in the initial system. In this case, it will be assumed that $Y(i, j) = \infty$ for the uniformity of the algorithm, and the inequality (6) will have form

$$X(i, j) \leq \frac{w_j}{w_i} < \infty.$$

Similarly, if $K_{j,i} = \emptyset$, then it will be assumed that $X(i, j) = 0$ and the inequality (6) will have form

$$0 \leq \frac{w_j}{w_i} \leq Y(i, j).$$

Jobs i, j were chosen arbitrarily; therefore, the inequality (6) can be written for any pair of different jobs $i, j \in J$.

Definition 4. A system of inequalities

$$X(i, j) \leq \frac{w_j}{w_i} \leq Y(i, j), \quad (7)$$

where:

$$X(i, j) = \max_{k \in K_{j,i}} \left(\frac{p_j^k}{p_i^k} \right),$$

$$Y(i, j) = \min_{k \in K_{i,j}} \left(\frac{p_j^k}{p_i^k} \right),$$

$$K_{i,j} = \{k \in K : \pi_k^0 = (\dots, i, \dots, j, \dots)\},$$

$$K_{j,i} = \{k \in K : \pi_k^0 = (\dots, j, \dots, i, \dots)\},$$

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

we call the **efficient system** of inequalities of the weight coefficient approximation problem for the case $r_1 = \dots = r_n$.

Lemma 1. The initial and efficient systems of inequalities (4) and (7) are equivalent.

Proof. The efficient system is the result of equivalent transformations of the inequalities of the initial system and, taking into account the Remark 4, includes all the inequalities of the initial system related to w_i, w_j for all possible pairs $i, j \in J, i \neq j$, that is, it contains all the inequalities of the initial system. Therefore, the efficient system of inequalities is equivalent to the initial system. $\square$

Thus, the solution sets of the efficient and initial systems will coincide. Lemma 1 allows us to turn to the efficient system consisting of $n(n-1)$ inequalities of the form (6) from the initial system consisting of N inequalities of the form (4). To find solutions to an efficient system, the following two lemmas are necessary.

Lemma 2. The set of solutions of the initial (and efficient) system is a convex polyhedral cone in n -dimensional space.

Proof. The initial system for the case $r_j = r, j \in J$, consists of inequalities of the form (4):

$$\frac{p_{j_1}^k}{w_{j_1}^k} \leq \frac{p_{j_2}^k}{w_{j_2}^k} \leq \dots \leq \frac{p_{j_n}^k}{w_{j_n}^k}, \quad k = 1, \dots, N.$$

Thus, the initial system of inequalities is a system of $N(n-1)$ linear inequalities. The general solution of this system, as well as a solution of any system of linear inequalities, is a convex cone with a finite number of faces. $\square$

Corollary 1. Any plane section of the set of solutions is also a convex set as the intersection of two convex sets: the solution of the initial system and the cutting plane.

Lemma 3. Let $\mathcal{P}$ be a rectangular parallelepiped in n -dimensional space $\mathbb{R}^n$, $n \in \mathbb{N}$. Let a convex set $\mathbf{M} \subset \mathbb{R}^n$ of hyperspace points touch each face of $\mathcal{P}$. Then, the center O of the parallelepiped $\mathcal{P}$ is an interior point of the set $\mathbf{M}$.

Proof. Obviously, the proof must be based on the property of convexity of the set $\mathbf{M}$. If the assertion of Lemma 3 for the subset $\mathbf{M}'$ of the set $\mathbf{M}$, which is limited only by the tangency points of the hyperspace by the initial set will be proved, then it will also be proved for the whole set $\mathbf{M}$. Next, consider arbitrary n tangency points of the hyperspace by the set $\mathbf{M}'$. The hyperplane drawn through these points will always “cut off” part of the hyperspace without its center. Thus, no matter what points are chosen, the center of the hyperspace will always be in the “non-cut-off” part, and therefore, will be an internal point of the set $\mathbf{M}'$. Figure 1 shows an illustration for the lemma in space $\mathbb{R}^3$. The statement of the lemma in this case takes the following form: the center of the parallelepiped, inside which there is a convex figure touching each of the sides of the parallelepiped, is an interior point of this figure. As can be seen from the figure, the points of contact of the convex set K, L and M form a plane that “cuts” the initial parallelepiped into two parts, one of which contains its center O . Since the convex set is tangent to each side of the parallelepiped, the point O will always be interior. $\square$

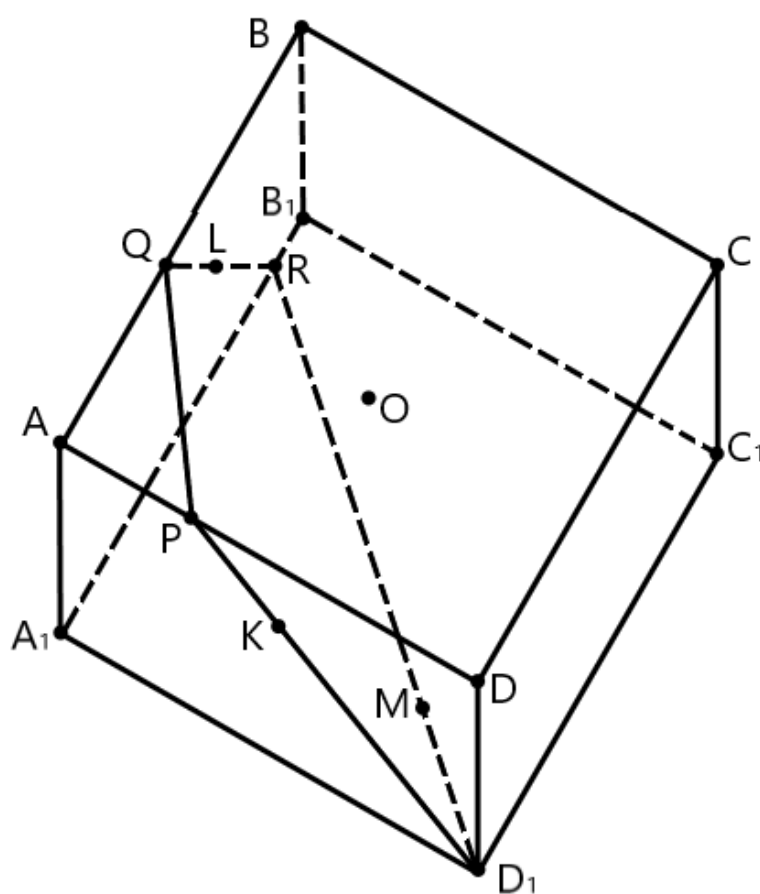


Figure 1. Illustration for Lemma 3 in space $\mathbb{R}^3$.

3.2. Method for Solving the Efficient System of Inequalities

Details of the efficient system of inequalities solution method are described in this section. Input is a system $\frac{n(n-1)}{2}$ of double inequalities of the form

$$X(i, j) \leq \frac{w_j}{w_i} \leq Y(i, j), \quad i, j \in J, \quad i \neq j. \quad (8)$$

This system can also be written as $n(n-1)$ simple linear inequalities:

$$\begin{cases} w_j - Y(i, j)w_i \leq 0, \\ -w_j + X(i, j)w_i \leq 0, \end{cases} \quad i, j \in J, \quad i \neq j,$$

Thus, the approximation problem has been reduced to a system of linear homogeneous inequalities. For such systems, there are many algorithms for finding a system of generating vectors sufficient to write a general formula for non-negative solutions. However, the resulting system has a special form, significantly different from the general one. In addition, the goal of the approximation problem is not to find all possible solutions to the system. Thus, a fast algorithm for finding a particular solution to the system of inequalities of the obtained form will be described.

The inequality (8) for three pairs formed from the jobs $\{f, g, h\}$ will take the following form:

$$X(f, g) \leq \frac{w_g}{w_f} \leq Y(f, g), \quad (9a)$$

$$X(g, h) \leq \frac{w_h}{w_g} \leq Y(g, h), \quad (9b)$$

$$X(h, f) \leq \frac{w_f}{w_h} \leq Y(h, f). \quad (9c)$$

The idea of the method is that one or more of the inequalities from (9a)–(9c) can be strengthened using the others. For example, multiplying the inequalities (9a) and (9b) gives:

$$X(f, g)X(g, h) \leq \frac{w_h}{w_f} \leq Y(f, g)Y(g, h)$$

or

$$\frac{1}{Y(f, g)Y(g, h)} \leq \frac{w_f}{w_h} \leq \frac{1}{X(f, g)X(g, h)}. \quad (10)$$

So, the inequality (10), which is a consequence of the inequalities (9a) and (9b), can improve the estimate of the ratio w_f/w_h in the inequality (9c). Indeed, if the values $X(h, f)$, $Y(f, g)$, $Y(g, h)$ are such, that

$$X(h, f) \leq \frac{1}{Y(f, g)Y(g, h)},$$

then the inequality (9c) can be transformed by setting

$$X(h, f) = \frac{1}{Y(f, g)Y(g, h)}.$$

We use the definition of matrices from the efficient system of inequalities:

$$\frac{1}{X(i, j)} = \frac{1}{\max_{k \in K_{j,i}} \left(\frac{p_j^k}{p_i^k} \right)} = \min_{k \in K_{j,i}} \left(\frac{p_i^k}{p_j^k} \right) = Y(j, i). \quad (11)$$

If $X(h, f) \leq X(h, g)X(g, f)$, then, taking into account the ratio (11), the inequality (9c) can be strengthened, setting $X(h, f) = X(h, g)X(g, f)$, as a result of which the set of solutions of the system (9) and the efficient system of inequalities remain unchanged.

Similarly, all the inequalities of the efficient system can be strengthened:

$$X(i, j) := \max \left\{ X(i, j); \max_{l=1, \dots, n, l \neq i, l \neq j} \{X(i, l)X(l, j)\} \right\}, \quad i, j \in J, i \neq j. \quad (12)$$

The procedure (12) must be repeated for all possible pairs of jobs i, j . On some steps after the change of $X(i, j)$ it can appear that some of $X(k, j)$ or $X(i, k)$ can also be updated, and the inequality with $X(i, j)$ can be strengthened again, where $k = 1, \dots, n$, $i \neq k \neq j$. Therefore, it is necessary to repeat the procedure (12) until none of the inequalities of the efficient system can be strengthened during the whole iteration. As will be shown in Section 4.3.3, this procedure will need to be repeated not more than $O(n^3)$ times. After that, the values of $Y(j, i)$ must also be

updated according to the relation (11). The matrices X, Y obtained after a sufficient number of repetitions of the procedure (12) will be denoted by $\tilde{X}, \tilde{Y}$.

Thus, if $w_j/w_i = z \in [\tilde{X}(i, j); \tilde{Y}(i, j)]$ for any pair $i, j \in J$, $i \neq j$, then among the inequalities of the system remaining after the corresponding simplifications, there are no pairs that contradict each other. Therefore, for each $z \in [\tilde{X}(i, j); \tilde{Y}(i, j)]$ there exists a solution to the initial system (4) of inequalities such that $w_j/w_i = z$.

The solution of the system, as shown in Lemma 2, is a convex cone. Therefore, the weight coefficients can be scaled: problems $1 \parallel \sum w_j C_j$ and $1 \parallel \sum (\gamma w_j) C_j$, where $\gamma > 0$, are equivalent, and if the set of coefficients $w = \{w_1, \dots, w_n\}$ is a solution of a system of inequalities (initial or efficient), then the coefficients $\gamma w = \{\gamma w_1, \dots, \gamma w_n\}$ are also a solution to this system. Due to this fact, one of the weight coefficients can be chosen arbitrarily, then the system can be solved for the remaining weight coefficients, and the resulting solution can be scaled. The resulting solution will correspond to the solution of the initial system of inequalities.

Let, for example, $w_1 = 1$. By Corollary 1 the cross section of the set of solutions of the initial system by the plane $w_1 = 1$ is also a convex set. Then, for arbitrary $j \in \{2, \dots, n\}$ a parallelepiped in a hyperspace of dimension $(n - 1)$ can be described:

$$\tilde{X}(1, j) \leq w_j \leq \tilde{Y}(1, j). \quad (13)$$

Moreover, as was proved above, for all w_j and $z \in [\tilde{X}(1, j); \tilde{Y}(1, j)]$ there exists a solution to the system such that $w_j/w_1 = z$ or $w_j = z$. In particular, for $w_1 = 1$ there exists a solution such that $w_j = \tilde{X}(1, j)$ and such that $w_j = \tilde{Y}(1, j)$. In other words, the section of the set of solutions by the plane $w_1 = 1$ lies inside the parallelepiped described by the inequalities (13) and has at least one common point with each of its faces. Therefore, by Lemma 3 the center of this parallelepiped is an interior point of the set of solutions to the initial system of inequalities and is a solution to the initial system.

The discussion given above can be summarized as the following theorem.

Theorem 1. A vector $w = (w_1, \dots, w_j, \dots, w_n)$, where

$$w_j = \begin{cases} 1, & j = l; \\ (\tilde{X}(l, j) + \tilde{Y}(l, j))/2, & j \neq l, \end{cases}$$

is a solution of the efficient system (7) of inequalities (index l can be chosen arbitrarily).

Any value within the interval $[\tilde{X}(l, j); \tilde{Y}(l, j)]$ can be taken as w_j , $j \in J$. We used the value in the middle of the interval $w_j = (\tilde{X}(l, j) + \tilde{Y}(l, j))/2$.

3.3. Algorithm for Solving the Approximation Problem

This section describes the algorithm for solving the problem $1 \parallel \sum w_j C_j$, $j \in J$, with unknown weight coefficients w_j . N instances of this problem are given: $I_k = \{p_1^k, \dots, p_n^k\}$, $k \in N$, and for each instance the optimal schedule $\pi_k^0 = (j_1^k, j_2^k, \dots, j_n^k)$ is known. It is necessary to approximate unknown values of the weight coefficients w_j , $j \in J$.

The algorithm for approximating the weight coefficients of the objective function is based on solving the efficient system of inequalities:

$$X(i, j) \leq \frac{w_j}{w_i} \leq Y(i, j), \quad (14)$$

where:

$$X(i, j) = \max_{k \in K_{j,i}} \left(\frac{p_j^k}{p_i^k} \right), \quad (15)$$

$$Y(i, j) = \min_{k \in K_{i,j}} \left(\frac{p_j^k}{p_i^k} \right), \quad (16)$$

$$K_{i,j} = \{k \in K : \pi_k^0 = (\dots, i, \dots, j, \dots)\}, \quad (17)$$

$$K_{j,i} = \{k \in K : \pi_k^0 = (\dots, j, \dots, i, \dots)\}, \quad (18)$$

$$i, j \in J, \quad i \neq j.$$

To approximate the coefficients w_j it is necessary:

1. construct sets $K_{i,j}$, $K_{j,i}$ according to Formulas (17) and (18);
2. construct matrices X and Y according to Formulas (15) and (16);
3. calculate matrices $\tilde{X}$ and $\tilde{Y}$ by repeating procedure (12) as it is described in Section 3.2;
4. calculate $w_j = \begin{cases} 1, & j = l \\ (\tilde{X}(l, j) + \tilde{Y}(l, j))/2, & j \neq l \end{cases}$, where index l is chosen arbitrarily.

4. Numerical Study

4.1. Description of Numerical Experiment

To study the efficiency of the constructed method for approximating the objective function of problem 1 $\| \sum w_j C_j$, the solution algorithm described in Section 3.3 was programmed in a *Python* environment and computational experiments were carried out for various numbers of jobs n and numbers of given instances N with known optimal schedules. For each experiment, $n \in \{10, 50, 100, 150, 200, 250\}$ weight coefficients $w_j^0, j \in J$ and $N \in [5, 100]$ (with a step of 5) problem instances $I_k = \{p_1^k, \dots, p_n^k\}, k = 1, \dots, N$, (values p_j^k and w_j^0 have a uniform distribution on the interval $[0; 1]$) were generated, and the constructed approximation algorithm was executed. The algorithm output is a set of the weight coefficients $w_j, j \in J$. To compare the found values of w_j and the true weight coefficients w_j^0 , both sets are normalized (scaling is allowed due to the linearity of the objective function):

$$\| w \| = \sqrt{\sum_{j=1}^n w_j^2},$$

$$w_j := \frac{w_j}{\| w \|}, \quad w_j^0 := \frac{w_j^0}{\| w^0 \|}.$$

Measure of efficiency $\varepsilon(N, n)$ of the algorithm is the modulus of the relative deviation of the found normalized values w_j (averaged over $j \in J$) from their true normalized values w_j^0 :

$$\varepsilon(N, n) = \frac{1}{n} \sum_{j=1}^n \frac{|w_j - w_j^0|}{w_j^0}. \quad (19)$$

The results of a series of experiments for the dependence of the approximation efficiency measure $\varepsilon(N, n)$ on N for different values of n are shown in Figures 2–5.

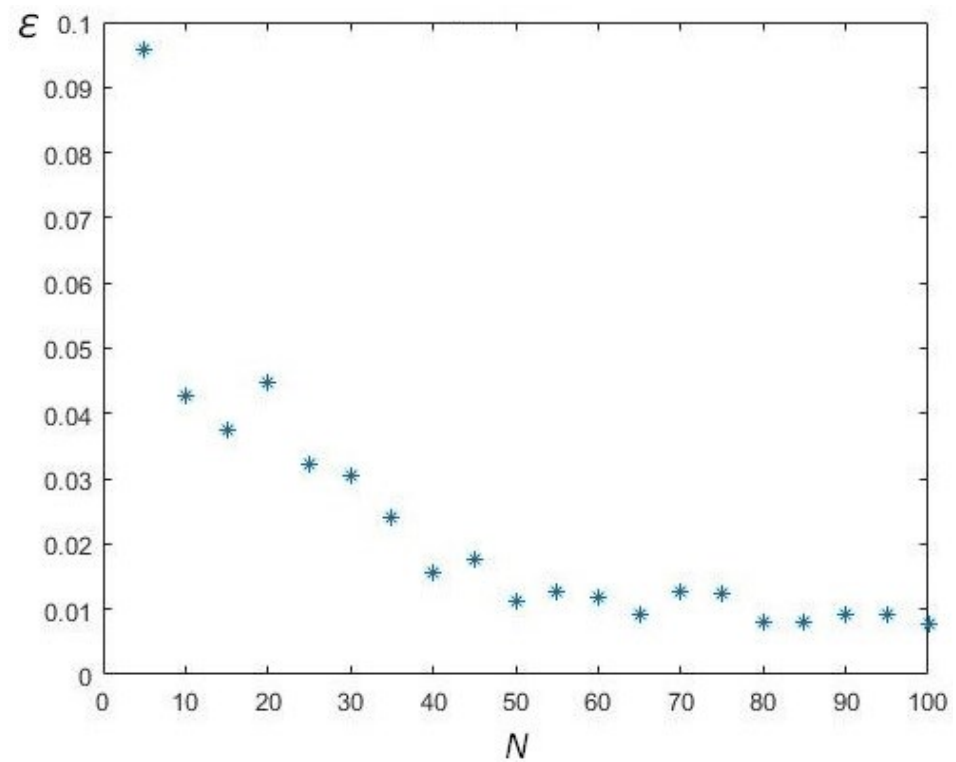


Figure 2. Modelling results, $n = 10$.

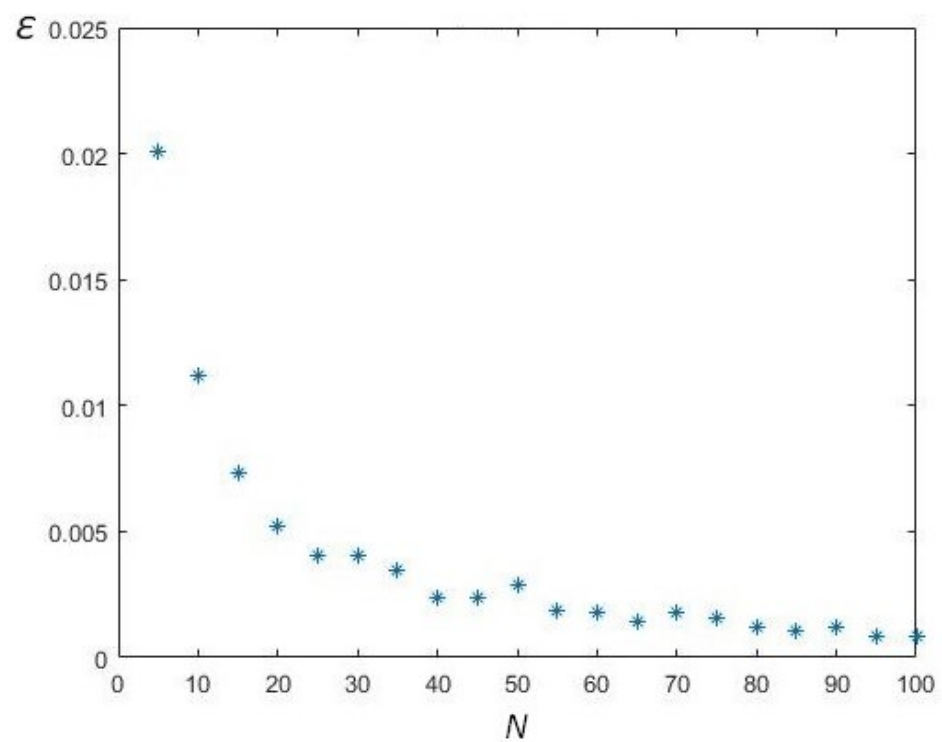


Figure 3. Modelling results, $n = 50$.

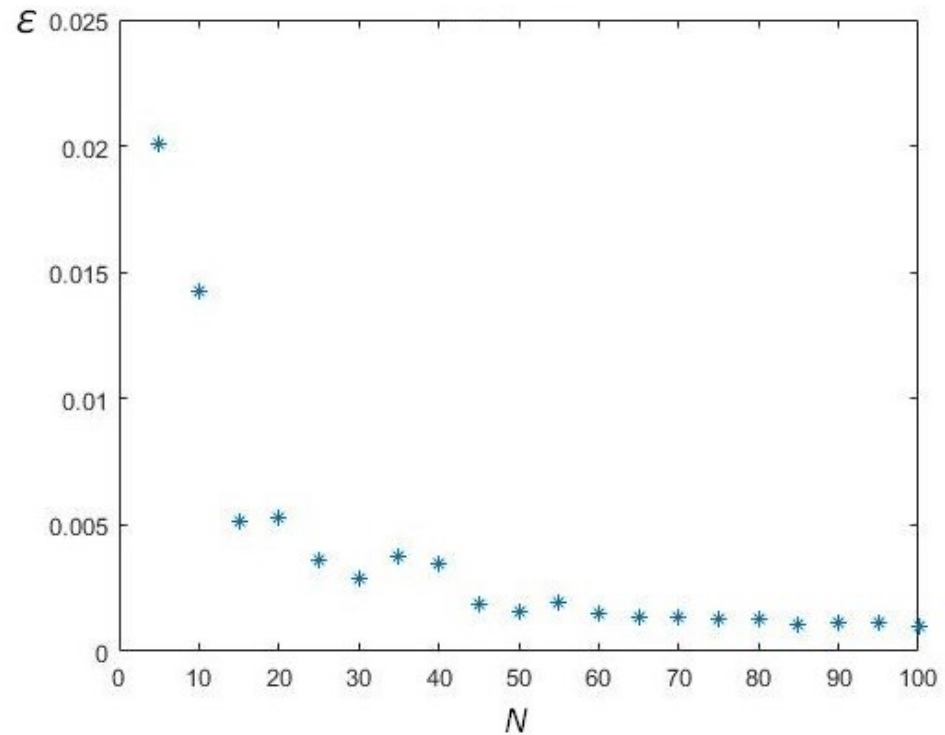


Figure 4. Modelling results, $n = 100$.

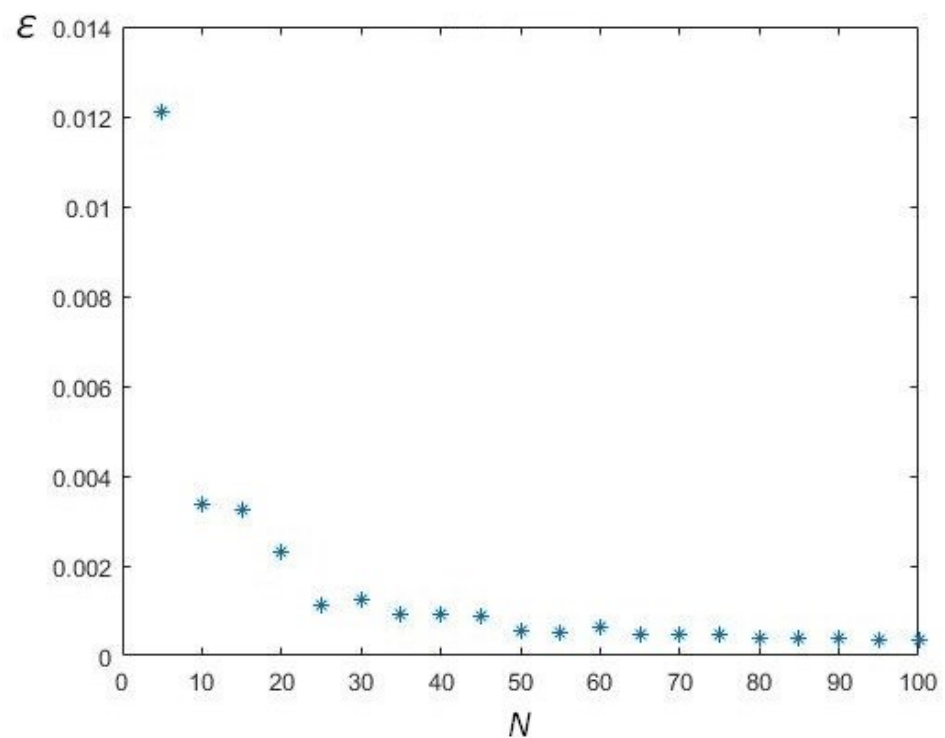


Figure 5. Modelling results, $n = 250$.

4.2. Analysis of Experiment Results

Function $\varepsilon(N, n)$ decreases as the number of known optimal schedules N increases. Moreover, the least squares (LSM) approximation of the dependence of $1/\varepsilon$ on N by the line $y = ax$ in all cases with a sufficiently large number of repetitions of the experiment, gives a high linear correlation coefficient $r > 0.9$. In Figure 6 a graphical representation of the result of a linear approximation with a linear correlation coefficient of $r = 0.93$ is presented.

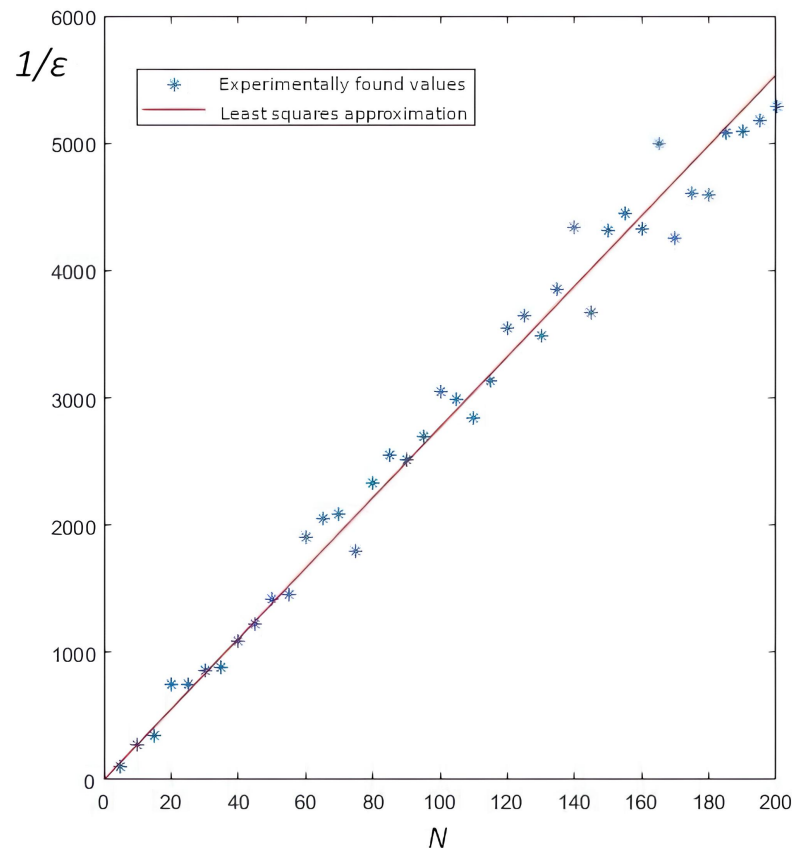


Figure 6. Modelling results, $n = 200$.

Due to the described correlation, the following hypothesis about the type of dependence $\varepsilon(N, n)$ is made:

$$\varepsilon(N, n) = \frac{a(n)}{N}. \quad (20)$$

This hypothesis possibly depends on the distribution of the input parameters with the growth of the number n of jobs and number N of given instances with known optimal schedules.

Moreover, as can be seen from the diagrams above, with a greater number of jobs n , more accurate results of the algorithm can be obtained. That is, the function $a(n)$ tends to be decreasing. For accurate statistical verification of the hypothesis, it is necessary to conduct a large number of computational experiments.

4.3. Computational Complexity Estimation

The approximation algorithm consists of four sequential procedures:

1. construction of sets $K_{i,j}$, $K_{j,i}$;
2. calculation of matrices X and Y ;
3. calculation of matrices $\tilde{X}$ and $\tilde{Y}$;
4. calculation of w_j .

The computational complexity of each of these procedures will be as follows.

4.3.1. Construction of Sets $K_{i,j}$, $K_{j,i}$

As it was described in Section 3.1, sets $K_{i,j}$ and $K_{j,i}$ are defined as follows:

$$K_{i,j} = \{k \in K : \pi_k^0 = (\dots, i, \dots, j, \dots)\},$$

$$K_{j,i} = \{k \in K : \pi_k^0 = (\dots, j, \dots, i, \dots)\}.$$

To construct these sets, it is necessary to determine the position of each requirement $j \in J$ in each schedule $\pi_k^0, k = 1, \dots, N$. Then, for each pair of different jobs $i, j \in J$ it is necessary to compare their positions in schedule π_k^0 . Therefore, the construction of sets $K_{i,j}, K_{j,i}$ requires the following number of operations:

$$N \cdot \left(O(n) + O\left(\frac{n(n-1)}{2}\right) \right) = O(n^2N),$$

where n is a number of jobs, N is a number of schedules.

4.3.2. Calculation of Matrices X and Y

The $n \times n$ matrices X and Y are calculated as follows:

$$X(i, j) = \max_{k \in K_{j,i}} \left(\frac{p_j^k}{p_i^k} \right), \quad Y(i, j) = \min_{k \in K_{i,j}} \left(\frac{p_j^k}{p_i^k} \right), \quad i, j \in J, \quad i \neq j.$$

For each instance I_k of the problem and all possible pairs $(i, j), i \neq j$, the ratio p_j^k/p_i^k is calculated, which requires $O(Nn(n-1))$ operations. Then, for each set $K_{j,i}$, the largest value is chosen, which is placed in $X(i, j)$, which requires $O(n(n-1) \cdot O(N))$ operations. After all, finding $Y(i, j)$ taking into account the relation (11) requires $O(n(n-1))$ operations. Therefore, to calculate the matrices X and Y , it is required to perform the following number of operations:

$$O(Nn(n-1)) + O(n(n-1) \cdot O(N)) + O(n(n-1)) = O(n^2N).$$

4.3.3. Calculation of Matrices $\tilde{X}$ and $\tilde{Y}$

Calculation of the matrices $\tilde{X}$ and $\tilde{Y}$ is the most time-consuming part of the algorithm in terms of computational complexity. Indeed, a procedure of the form (12):

$$X(i, j) := \max \left\{ X(i, j); \max_{l \in J, i \neq l \neq j} \{X(i, l)X(l, j)\} \right\}, \quad i, j \in J, \quad i \neq j, \quad (21)$$

for each pair i, j involves $(n-2)$ computing the product $X(i, l)X(l, j)$ and finding the maximum of $(n-1)$ values. Thus, the number of operations required to perform the procedure (12) once is

$$O((n-2) + (n-1)) = O(n).$$

To find the matrices $\tilde{X}$ and $\tilde{Y}$, it is necessary to repeat the procedure (12) in a loop for all possible pairs of jobs i, j until there will be no such pair i, j so that the element $X(i, j)$ can be increased. However, after each repetition of the procedure, the element $X(i, j)$ either is increased or unchanged, so the number of repetitions of the procedure can be significantly reduced. This can be conducted by choosing a special order in which the pairs $i, j \in J$ are considered such that the procedure (12) will be executed for each pair of jobs no more than two times. In this case, the number of operations required to calculate the matrices $\tilde{X}$ and $\tilde{Y}$, is

$$O\left(2 \cdot \frac{n(n-1)}{2} \cdot n\right) = O(n^3).$$

4.3.4. Calculation of w_j

Computational complexity of finding all values

$$w_j = \begin{cases} 1, & j = l; \\ (\tilde{X}(l, j) + \tilde{Y}(l, j))/2, & j \neq l, \end{cases} \quad j \in J,$$

is $O(n)$ operations.

4.3.5. Resulting Complexity

Thus, the computational complexity of the algorithm can be estimated with the following number of operations:

$$O(n^2N) + O(n^2N) + O(n^3) + O(n) = O(n^2N + n^3) = O(n^2(N + n)).$$

5. Conclusions

The result of the work is an algorithm for approximating the values of the weight coefficients of the problem $1 \parallel \sum w_j C_j$ using N given instances of the problem with known optimal schedules. The result of the algorithm is a set of weight coefficients w_j , $j \in J$ such that for each of the N given instances, the optimal schedule found for the approximate values of the weights either is equal to the given optimal schedule corresponding to the unknown true set of weights w_j^0 , or has the same objective function value with it. The results described in this paper are also relevant to the problem $1 \parallel \sum w_j L_j$.

The computational complexity of the algorithm is limited to $O(n^2(N + n))$ operations, where n is the number of jobs and N is a number of initial instances with known optimal schedules.

A numerical experiment was carried out to study the efficiency of the method. All shown examples have $N \in [5, 100]$ given instances of the problem $1 \parallel \sum w_j C_j$, where the number of jobs n is the same for all instances in each set and were given from the interval $[10, 250]$ to check the correlation. The accuracy of the algorithm is estimated by experimentally measuring the function $\varepsilon(N, n) = \frac{1}{n} \sum_{j=1}^n \frac{|w_j - w_j^0|}{w_j^0}$, which is an indicator of the average modulus of the relative deviation of the found values w_j from the true values w_j^0 . An analysis of the results shows a high correlation between the dependence $\varepsilon(N, n)$ and a function of the form $\alpha(n)/N$, where $\alpha(n)$ is a decreasing function of n . So, based on the result of the experiment, it is clear that a greater number of jobs n , more accurate results of the algorithm can be obtained. In Section 4.2 it is shown that the dependence of $1/\varepsilon$ on N is approximated by the line $y = ax$; the least squares method shows a high linear correlation coefficient $r > 0.9$ in all cases with a sufficiently large number of repetitions of the experiment.

Based on the current results, further work is planned in the following areas:

- searching for a formal proof of the hypothesis about the form of dependence $\varepsilon(N, n)$ from Section 4.2 when $r_j = \text{const}$, $j \in J$;
- continue studying the general case $1 \mid r_j \mid \sum w_j C_j$, where jobs can have different release times; it is necessary to find either a subsystem of inequalities with a polynomial number of inequalities, equivalent to the original system, or the strongest subsystem with a polynomial number of inequalities with an estimate of the approximation error;
- trying to adapt the results to solve the problem of approximating more complex objective functions.

Author Contributions: Conceptualization, A.L.; methodology, A.L. and N.P.; software, E.B.; validation, A.L. and N.P.; formal analysis, A.L. and N.P.; investigation, E.B. and N.P.; data curation, E.B.; writing—original draft preparation, E.B. and N.P.; writing—review and editing, A.L. and N.P.; visualization, E.B.; supervision, A.L.; project administration, A.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The paper describes a theoretical research. No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: Authors gratefully thank Nikolay A. Loginov, who took an active part at the initial stage of the work.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Vahedi-Nouri, B.; Tavakkoli-Moghaddam, R.; Hanzálek, Z.; Dolgui, A. Production scheduling in a reconfigurable manufacturing system benefiting from human-robot collaboration. *Int. J. Prod. Res.* **2024**, *62*, 767–783. [\[CrossRef\]](#)
2. Lazarev, A.A.; Nekrasov, I.; Pravdivets, N. Evaluating typical algorithms of combinatorial optimization to solve continuous-time based scheduling problem. *Algorithms* **2018**, *11*, 50. [\[CrossRef\]](#)
3. Rzevski, G.; Skobelev, P.; Zhilyaev, A. Emergent Intelligence in Smart Ecosystems: Conflicts Resolution by Reaching Consensus in Resource Management. *Mathematics* **2022**, *10*, 1923. [\[CrossRef\]](#)
4. Lei, D.; Liu, M. An artificial bee colony with division for distributed unrelated parallel machine scheduling with preventive maintenance. *Comput. Ind. Eng.* **2020**, *141*, 106320. [\[CrossRef\]](#)
5. Zhao, X.; Liu, H.; Wu, Y.; Qiu, Q. Joint optimization of mission abort and system structure considering dynamic tasks. *Reliab. Eng. Syst. Saf.* **2023**, *234*, 109128. [\[CrossRef\]](#)
6. Koulamas, C.; Kyparisis, G.J. A classification of dynamic programming formulations for offline deterministic single-machine scheduling problems. *Eur. J. Oper. Res.* **2023**, *305*, 999–1017. [\[CrossRef\]](#)
7. Martinelli, R.; Mariano, F.C.M.Q.; Martins, C.B. Single machine scheduling in make to order environments: A systematic review. *Comput. Ind. Eng.* **2022**, *169*, 108190. [\[CrossRef\]](#)
8. Tanaka, K.; Vlach, M. Minimizing maximum absolute lateness and range of lateness under generalized due dates on a single machine. *Ann. Oper. Res.* **1999**, *86*, 507–526. [\[CrossRef\]](#)
9. Mosheiov, G.; Oron, D.; Shabtay, D. Minimizing total late work on a single machine with generalized due-dates. *Eur. J. Oper. Res.* **2021**, *293*, 837–846. [\[CrossRef\]](#)
10. Hermelin, D.; Karhi, S.; Pinedo, M.; Shabtay, D. New algorithms for minimizing the weighted number of tardy jobs on a single machine. *Ann. Oper. Res.* **2021**, *298*, 271–287. [\[CrossRef\]](#)
11. Janiak, A.; Kovalyov, M. Single machine group scheduling with ordered criteria. *Ann. Oper. Res.* **1995**, *57*, 191–201. [\[CrossRef\]](#)
12. Rudek, R. The single machine total weighted completion time scheduling problem with the sum-of-processing time based models: Strongly NP-hard. *Appl. Math. Mod.* **2017**, *50*, 314–332. [\[CrossRef\]](#)
13. Lazarev, A.; Pravdivets, N.; Werner, F. On the dual and inverse problems of scheduling jobs to minimize the maximum penalty. *Mathematics* **2020**, *8*, 1131. [\[CrossRef\]](#)
14. Chernikov, S. Linear Inequalities. In *Itogi Nauki i Tekhniki. Series 'Algebra. Topology. Geometry'*; VINITI: Moscow, Russia, 1968. (In Russian)
15. Chernikova, N.V. Algorithm for finding a general formula for the non-negative solutions of a system of linear inequalities. *USSR Comput. Math. Math. Phys.* **1965**, *5*, 228–233. [\[CrossRef\]](#)
16. Zhu, Y. Generalizations of some fundamental theorems on linear inequalities. *Acta Math. Sin.* **1966**, *16*, 25–40.
17. Fan, K. On infinite systems of linear inequalities. *J. Math. Anal. Appl.* **1968**, *21*, 475–478. [\[CrossRef\]](#)
18. Eckhardt, U. Theorems on the dimension of convex sets. *Linear Algebra Appl.* **1975**, *12*, 63–76. [\[CrossRef\]](#)
19. Eckhardt, U. Representation of Convex Sets. In *Extremal Methods and Systems Analysis*; Springer: New York, NY, USA, 1980.
20. Lenstra, J.; Rinnooy Kan, A.; Brucker, P. Complexity of Machine Scheduling Problems. *Ann. Discret. Math.* **1977**, *1*, 343–362.
21. Lazarev, A. *Scheduling Theory: Methods and Algorithms*; ICS RAS: Moscow, Russia, 2019. (In Russian)
22. Smith, W. Various optimizers for single-stage production. *Nav. Res. Logist. Q.* **1956**, *3*, 59–66. [\[CrossRef\]](#)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.