

Article

The Optimality Region for a Single-Machine Scheduling Problem with Bounded Durations of the Jobs and the Total Completion Time Objective

Yuri N. Sotskov * and Natalja G. Egorova

United Institute of Informatics Problems, National Academy of Sciences of Belarus, Surganova Street 6, 220012 Minsk, Belarus; NataMog@yandex.ru

* Correspondence: sotskov48@mail.ru; Tel.: +375-17-284-2120

Received: 28 February 2019 ; Accepted: 19 April 2019; Published: 26 April 2019



Abstract: We study a single-machine scheduling problem to minimize the total completion time of the given set of jobs, which have to be processed without job preemptions. The lower and upper bounds on the job duration is the only information that is available before scheduling. Exact values of the job durations remain unknown until the completion of the jobs. We use the optimality region for the job permutation as an optimality measure of the optimal schedule. We investigate properties of the optimality region and derive $O(n)$ -algorithm for calculating a quasi-perimeter of the optimality set (i.e., the sum of lengths of the optimality segments for n given jobs). We develop a fast algorithm for finding a job permutation having the largest quasi-perimeter of the optimality set. The computational results in constructing such permutations show that they are close to the optimal ones, which can be constructed for the factual durations of all given jobs.

Keywords: single-machine scheduling; uncertain job durations; total completion time objective; optimality region

1. Introduction

A lot of real-life scheduling problems involve different forms of uncertainties. For dealing with uncertain scheduling problems, several approaches have been developed in the literature. In a stochastic approach, job durations are assumed to be random variables with the specific probability distributions known before scheduling [1,2]. If there is no sufficient information to determine the probability distribution for each random duration of the given job, other approaches have to be used [3–5]. In the approach of seeking a robust schedule [3,6], a decision-maker prefers a schedule that hedges against the worst-case scenario. A fuzzy approach [7–9] allows a scheduler to find best schedules with respect to fuzzy durations of the given jobs. A stability approach [10] is based on the stability analysis of the optimal schedules to possible variations of the job durations. In this paper, we apply the stability approach to the single-machine scheduling problem with interval durations of the given jobs.

In Section 2, we present settings of the uncertain scheduling problems, the related literature and closed results. In Section 3, we investigate properties of the optimality region for the job permutation, which is used for processing given jobs. Efficient algorithms for calculating a quasi-perimeter of the optimality region are derived in Section 4. In Section 5, we show how to find a job permutation with the largest quasi-perimeter of the optimality region and develop algorithm for finding an approximate solution for the uncertain scheduling problem. In Section 6, we report on the computational results for finding solutions for the tested instances. The paper is concluded in Section 7.

2. Problem Descriptions, The Related Literature and Closed Results

There is given a set of jobs $\mathcal{J} = \{J_1, J_2, \dots, J_n\}$ to be processed on a single machine. The duration p_i of the job $J_i \in \mathcal{J}$ can take any real value from the given segment $[p_i^L, p_i^U]$, where the inequalities $p_i^U \geq p_i^L > 0$ hold. The exact value $p_i \in [p_i^L, p_i^U]$ of the job duration remains unknown until the completion time of the job $J_i \in \mathcal{J}$.

Let R_+^n denote a set of all non-negative n -dimensional real vectors, $R_+^n \subseteq R^n$, where R^n is space of n -dimensional real vectors. The set of all vectors $(p_1, p_2, \dots, p_n) = p \in R_+^n$ of the feasible durations is presented as the Cartesian product of the segments $[p_i^L, p_i^U]$:

$$T = [p_1^L, p_1^U] \times [p_2^L, p_2^U] \times \dots \times [p_n^L, p_n^U] = \{p : p \in R_+^n, p_i^L \leq p_i \leq p_i^U, i \in \{1, 2, \dots, n\}\}.$$

A vector $p \in T$ of the job durations is called a scenario. Let $S = \{\pi_1, \pi_2, \dots, \pi_n!\}$ denote a set of all permutations $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n})$ of the given jobs \mathcal{J} .

Given a scenario $p \in T$ and a permutation $\pi_k \in S$, let $C_i = C_i(\pi_k, p)$ denote the completion time of the job J_i in the schedule determined by the permutation π_k . The criterion $\sum C_i$ denotes the minimization of the following sum of the completion times:

$$\sum_{J_i \in \mathcal{J}} C_i(\pi_t, p) = \min_{\pi_k \in S} \left\{ \sum_{J_i \in \mathcal{J}} C_i(\pi_k, p) \right\}, \tag{1}$$

where the permutation $\pi_t = (J_{t_1}, J_{t_2}, \dots, J_{t_n}) \in S$ is optimal. From the equality (1), it follows that only semi-active schedule [11] may be optimal. Each permutation $\pi_k \in S$ determines exactly one semi-active schedule.

The above uncertain scheduling problem is denoted as $1|p_i^L \leq p_i \leq p_i^U|\sum C_i$ using the three-field notation $\alpha|\beta|\gamma$ [12], where α denotes the processing system, β characterizes conditions for processing the jobs and γ determines the criterion.

2.1. The Related Literature

If a scenario $p \in T$ is fixed before scheduling (i.e., the equality $[p_i^L, p_i^U] = [p_i, p_i]$ holds for each job $J_i \in \mathcal{J}$), then the uncertain problem $1|p_i^L \leq p_i \leq p_i^U|\sum C_i$ is turned into the deterministic one $1||\sum C_i$. In what follows, we use the notation $1|p|\sum C_i$ to indicate an instance of the deterministic problem $1||\sum C_i$ with scenario $p \in T$. Any instance $1|p|\sum C_i$ is solvable in $O(n \log n)$ time [13] due to the following necessary and sufficient condition for the optimality of the job permutation $\pi_k \in S$.

Theorem 1. *The job permutation $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n}) \in S$ is optimal for the instance $1|p|\sum C_i$ if and only if the following inequalities hold:*

$$p_{k_1} \leq p_{k_2} \leq \dots \leq p_{k_n}. \tag{2}$$

If the strict inequality $p_{k_u} < p_{k_v}$ holds, then the job J_{k_u} precedes the job J_{k_v} in any optimal job permutation π_k .

Since the scenario $p \in T$ is not fixed in the uncertain problem $1|p_i^L \leq p_i \leq p_i^U|\sum C_i$, the completion time C_i of the job $J_i \in \mathcal{J}$ cannot be determined for the permutation $\pi_k \in S$ before completing the job J_i . Thus, the value of the objective function $\sum_{J_i \in \mathcal{J}} C_i(\pi_t, p)$ for the permutation π_k remains uncertain until all jobs \mathcal{J} have been completed. Since for the uncertain problem $\alpha|p_i^L \leq p_i \leq p_i^U|\gamma$, there does not usually exist an optimal schedule for all feasible scenarios p from the set T , an additional objective or some agreements are used in the literature.

A robust schedule minimizing the worst-case regret has been developed in [3,8,14–18]. For any permutation $\pi_k \in S$ and any feasible scenario $p \in T$, the difference $\gamma_p^k - \gamma_p^t =: r(\pi_k, p)$ is called the regret for the permutation π_k . In the above notation $r(\pi_k, p)$, the objective function γ is equal to γ_p^k for the permutation π_k under scenario p and the optimal value of the objective function γ is equal to

γ_p^t for the optimal permutation π_k under scenario p . The value of $Z(\pi_k) = \max\{r(\pi_k, p) : p \in T\}$ is called the worst-case absolute regret. The value of $Z^*(\pi_k) = \max\{\frac{r(\pi_k, p)}{\gamma_p^t} : p \in T\}$ is called the worst-case relative regret.

While the deterministic problem $1|\sum C_i$ is polynomially solvable [13], finding a permutation $\pi_t \in S$ minimizing the worst-case absolute regret $Z(\pi_k)$ or the relative regret Z^* for the problem $1|p_i^L \leq p_i \leq p_i^U|\sum C_i$ are binary NP-hard even for two possible scenarios $\{p_1, p_2\}$ [3,17,19]. Discrete sets $\{p_1, p_2, \dots, p_m\}$ of the uncertain scenarios have been investigated in [3,17,19].

The complexity of minimizing the total flow time with continues data T is characterized in [20], where it is proven that finding a permutation $\pi_t \in S$ minimizing the worst-case absolute regret $Z(\pi_k)$ for the problem $1|p_i^L \leq p_i \leq p_i^U|\sum C_i$ is binary NP-hard. For a special case of this problem, where all intervals of uncertainty have the same center, it is shown that this problem can be solved in $O(n \log n)$ time if the number of jobs is even, and remains NP-hard if the number of jobs is odd [20]. In [6], a branch-and-bound algorithm was developed for finding a permutation π_k minimizing the absolute regret for the problem $1|p_i^L \leq p_i \leq p_i^U|\sum w_i C_i$, where the jobs $J_i \in \mathcal{J}$ have different weights $w_i > 0$. The computational experiments showed that the developed algorithm is able to find such a permutation π_k for the instances with up to 40 jobs.

The fuzzy scheduling technique was used in [7–9,21] to develop a fuzzy analogue of dispatching rules or to solve mathematical programming problems to determine a schedule that minimizes a fuzzy-valued objective function.

In [22], several heuristics were developed for the problem $1|p_i^L \leq p_i \leq p_i^U|\sum w_i C_i$. The computational experiments including different probability distributions of the job durations showed that there was at least one heuristic among all performing heuristics with the error 1.1% of the optimal objective function value $\sum w_i C_i$ obtained after completing the given jobs when their factual durations became known.

In Sections 3–6, we adopt the stability approach [5,10,23–25] to the uncertain problem $1|p_i^L \leq p_i \leq p_i^U|\sum C_i$ with the additional criterion of maximizing a quasi-perimeter of the optimality region introduced in Section 3.

2.2. The Stability Approach to Single-Machine Scheduling Problems

Let M denote a subset of the set $N = \{1, 2, \dots, n\}$. In [23,24], an optimality box for the job permutation $\pi_k \in S$ for the uncertain problem $1|p_i^L \leq p_i \leq p_i^U|\sum C_i$ is defined as follows.

Definition 1. *The maximal rectangular box $\mathcal{OB}(\pi_k, T) = \times_{k_i \in M} [l_{k_i}^*, u_{k_i}^*] \subseteq T$ is called an optimality box for the permutation $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n}) \in S$ (with respect to T), if the permutation π_k being optimal for the instance $1|p|\sum C_i$ with the scenario $p = (p_1, p_2, \dots, p_n) \in T$ remains optimal for the instance $1|p'|\sum C_i$ with any scenario $p' \in \mathcal{OB}(\pi_k, T) \cup \{\times_{k_j \in N \setminus M} [p_{k_j}, p_{k_j}]\}$. If there does not exist a scenario $p \in T$ such that the permutation π_k is optimal for the instance $1|p|\sum C_i$, it is assumed that $\mathcal{OB}(\pi_k, T) = \emptyset$.*

In Section 3, we use the following remark for the definition of the optimality segment for the job $J_{k_i} \in \mathcal{J}$ in the permutation $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n}) \in S$.

Remark 1. *Any variation p'_{k_i} of the duration p_{k_i} of the job $J_{k_i} \in \mathcal{J}$ within the maximal segment $[l_{k_i}^*, u_{k_i}^*]$ indicated in Definition 1 cannot violate the optimality of the permutation $\pi_k \in S$ provided that the inclusion $p'_{k_i} \in [l_{k_i}^*, u_{k_i}^*]$ holds. The non-empty maximal segment $[l_{k_i}^*, u_{k_i}^*]$ indicated in Definition 1 with the inequality $l_{k_i}^* \leq u_{k_i}^*$ and the length $u_{k_i}^* - l_{k_i}^* \geq 0$ is called an optimality segment for the job $J_{k_i} \in \mathcal{J}$ in the permutation π_k . We denote the optimality segment as follows: $[l_{k_i}^{opt}, u_{k_i}^{opt}]$.*

If the maximal segment $[l_{k_i}^*, u_{k_i}^*]$ indicated in Definition 1 is empty for the job $J_{k_i} \in \mathcal{J}$, we say that this job has no optimality segment in the permutation π_k . It is clear that if the job J_{k_i} has no optimality segment in the permutation π_k , then the strict inequality $l_{k_i}^* > u_{k_i}^*$ holds. In [23,24], it is shown that

for calculating the optimality box $OB(\pi_k, T)$, one can calculate the optimality box for the modified instance $1|\widehat{p}_i^L \leq p_i \leq \widehat{p}_i^U|\sum w_i C_i$, where the segments $[p_i^L, p_i^U]$ for the possible job durations p_i are reduced, $[\widehat{p}_i^L, \widehat{p}_i^U] \subseteq [p_i^L, p_i^U]$, based on the following formulas:

$$\frac{w_i}{\widehat{p}_i^L} = \min_{1 \leq j \leq i \leq n} \left\{ \frac{w_j}{p_j^L} \right\}, \quad \frac{w_i}{\widehat{p}_i^U} = \max_{1 \leq i \leq j \leq n} \left\{ \frac{w_j}{p_j^U} \right\}. \tag{3}$$

The following theorem has been proven in [24].

Theorem 2. *The optimality box for the permutation $\pi_k \in S$ for the problem $1|p_i^L \leq p_i \leq p_i^U|\sum C_i$ is equal to the optimality box for the same permutation π_k for the problem $1|\widehat{p}_i^L \leq p_i \leq \widehat{p}_i^U|\sum C_i$ with the feasible segments $[\widehat{p}_i^L, \widehat{p}_i^U], J_i \in \mathcal{J}$, determined in (3).*

In [23], it is shown that Theorem 2 remains correct for the problem $1|p_i^L \leq p_i \leq p_i^U|\sum w_i C_i$ with the different weights $w_i > 0$ prescribed to the jobs $J_i \in \mathcal{J}$.

Following [23,24], the notion of a block for the jobs \mathcal{J} is determined for the problem $1|p_i^L \leq p_i \leq p_i^U|\sum C_i$ as follows.

Definition 2. *A maximal subset $B_r = \{J_{r_1}, J_{r_2}, \dots, J_{r_{|B_r|}}\}$ of the set \mathcal{J} , for which the inequality $\max_{J_{r_i} \in B_r} p_{r_i}^L \leq \min_{J_{r_i} \in B_r} p_{r_i}^U$ holds, is called a block. The segment $[b_r^L, b_r^U]$, where equalities $b_r^L = \max_{J_{r_i} \in B_r} p_{r_i}^L$ and $b_r^U = \min_{J_{r_i} \in B_r} p_{r_i}^U$ hold, is called a core of the block B_r .*

The following claim was proven in [23].

Lemma 1. *For the problem $1|p_i^L \leq p_i \leq p_i^U|\sum w_i C_i$, the set $B = \{B_1, B_2, \dots, B_m\}$ of all blocks can be uniquely determined in $O(n \log n)$ time.*

If the job $J_i \in \mathcal{J}$ belongs to a single block, we say that the job J_i is fixed in this block. We say that the block B_v is virtual, if there is no fixed job in this block. We say that the job J_k is non-fixed, if the job $J_k \in \mathcal{J}$ belongs to two or more blocks and at least one of these blocks is not virtual.

Remark 2. *Any permutation $\pi_k \in S$ determines a distribution of all non-fixed jobs to the blocks. Due to such fixings of the positions of the non-fixed jobs, some virtual blocks from the set B may be destroyed for the fixed permutation π_k . Furthermore, the cores of some non-virtual blocks may be increased in the permutation π_k .*

Each block in the set B has the following properties proven in [23].

Lemma 2. *At most two jobs in the block $B_r \in B$ may have optimality segments in the permutation $\pi_k \in S$.*

Lemma 3. *If $OB(\pi_k, T) \neq \emptyset$, then any two jobs $J_v \in B_r$ and $J_w \in B_s$, which are fixed in different blocks, $r < s$, must be ordered in the permutation $\pi_k \in S$ with decreasing left bounds (and right bounds as well) of the cores of their blocks, i.e., the permutation π_k looks as follows: $\pi_k = \{\dots, J_v, \dots, J_w, \dots\}$, where the inequality $b_r^L < b_s^L$ holds.*

In what follows, we assume that all blocks in the set $B = \{B_1, B_2, \dots, B_m\}$ are numbered according to decreasing left bounds of their cores, i.e., the strict inequality $b_v^L < b_u^L$ implies the strict inequality $v < u$. Due to Definition 2, each block $B_r = \{J_{r_1}, J_{r_2}, \dots, J_{r_{|B_r|}}\}$ may include jobs of the four types as follows.

If $p_{r_i}^L = b_r^L$ and $p_{r_i}^U = b_r^U$, we say that job J_{r_i} is a core job in the block B_r . If $p_{r_i}^L < b_r^L$, we say that job J_{r_i} is a left job in the block B_r . If $p_{r_i}^U > b_r^U$, we say that job J_{r_i} is a right job in the block B_r . Let B_r^* denote the set of all core jobs. The set B_r^- (the set B_r^+) is the set of all left (right) jobs in the

block B_r . Some jobs $J_{r_i} \in B_r$ may be left-right jobs in the block B_r , since it is possible that condition $B \setminus \{B_r^* \cup B_r^- \cup B_r^+\} \neq \emptyset$ holds.

The jobs J_v and J_w are identical if both equalities $p_v^L = p_w^L$ and $p_v^U = p_w^U$ hold. If the set $B_r \in B$ is a singleton, $|B_r| = 1$, then the equality $B_r = B_r^*$ holds. The following theorem was proven in [24].

Theorem 3. For the problem $1|p_i^L \leq p_i \leq p_i^U| \sum C_i$, any permutation $\pi_k \in S$ has an empty optimality box, $\mathcal{OB}(\pi_k, T) = \emptyset$, if and only if for each block $B_r \in B$, either condition $|B_r| = |B_r^*| \geq 2$ holds or condition $B_r = B_r^- \cup B_r^+$ holds and all jobs in the set B_r^- (in the set B_r^+) are identical and both inequalities $|B_r^-| \geq 2$ and $|B_r^+| \geq 2$ hold.

The following criterion was proven in [23].

Theorem 4. Let all jobs from the set \mathcal{J} be fixed in their blocks from the set B . Then the permutation π_k with the largest optimality box $\mathcal{OB}(\pi_k, T)$ may be constructed in $O(n \log n)$ time.

The rest of this paper is devoted to an optimality set (it is called an optimality region), which is a superset of the optimality box $\mathcal{OB}(\pi_k, T)$ for the same permutation π_k .

3. The Optimality Region

For the permutation $\pi_k \in S$, we formally define the optimality region $\mathcal{OR}(\pi_k, T)$ such that the inclusion $\mathcal{OB}(\pi_k, T) \subseteq \mathcal{OR}(\pi_k, T)$ holds.

Definition 3. The maximal closed subset $\mathcal{OR}(\pi_k, T) \subseteq T$ of the set R_+^n is called the optimality region for the permutation $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n}) \in S$ (with respect to T) if the permutation π_k is optimal for the instance $1|p| \sum C_i$ with any scenario $p = (p_1, p_2, \dots, p_n) \in \mathcal{OR}(\pi_k, T)$. If there does not exist a scenario $p \in T$ such that the permutation π_k is optimal for the instance $1|p| \sum C_i$, it is assumed that $\mathcal{OR}(\pi_k, T) = \emptyset$.

We demonstrate the above definitions and notions on the instance of the problem $1|p_i^L \leq p_i \leq p_i^U| \sum C_i$ with $n = 8$ jobs. The segments $[p_i^L, p_i^U]$ defining all possible durations of the jobs $J_i \in \mathcal{J} = \{J_1, J_2, \dots, J_8\}$ are given in Table 1. The segments $[p_i^L, p_i^U]$ of the job durations are also represented in a coordinate system in Figure 1 for the permutation $\pi_1 = (J_1, J_2, \dots, J_8) \in S$. The abscissa axis indicates the segments $[p_i^L, p_i^U]$ determining durations of the jobs. The ordinate axis indicates all jobs \mathcal{J} . There are three blocks in this instance. The jobs J_1, J_2, J_3, J_4, J_5 and J_7 belong to the block B_1 . The segment $[6, 7]$ is a core of the block B_1 . The jobs J_2, J_3, J_4, J_5, J_6 and J_7 belong to the block B_2 . The one-point segment $[8, 8]$ is a core of the block B_2 . The jobs J_4, J_5, J_6, J_7 and J_8 belong to the block B_3 . The segment $[10, 11]$ is a core of the block B_3 . The jobs J_2, J_3, J_4, J_5, J_6 and J_7 are non-fixed jobs. The jobs J_1 and J_8 are fixed in their blocks.

Due to the optimality criterion (2) for the permutation $\pi_k \in S$ given in Theorem 1, one can distinguish three types of segments for each job $J_{k_r} \in \mathcal{J}$, which characterize a possibility for the permutation $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n})$ to be optimal, namely:

- the segment of optimality $[l_{k_r}^{opt}, u_{k_r}^{opt}] \subseteq [p_{k_r}^L, p_{k_r}^U]$;
- the segment of conditional optimality $[l_{k_r}^{cpt}, u_{k_r}^{cpt}] \subseteq [p_{k_r}^L, p_{k_r}^U]$;
- and the segment of non-optimality $[l_{k_r}^{non}, u_{k_r}^{non}] \subseteq [p_{k_r}^L, p_{k_r}^U]$.

The segment of optimality $[l_{k_r}^{opt}, u_{k_r}^{opt}]$ for the job J_{k_r} in the permutation π_k is formally determined in Definition 1 and Remark 1.

Table 1. Input data for the instance of the problem $1|p_i^L \leq p_i \leq p_i^U|\sum C_i$.

i	1	2	3	4	5	6	7	8
p_i^L	5	5	6	4	2	8	3	10
p_i^U	7	8	9	11	14	11	17	12

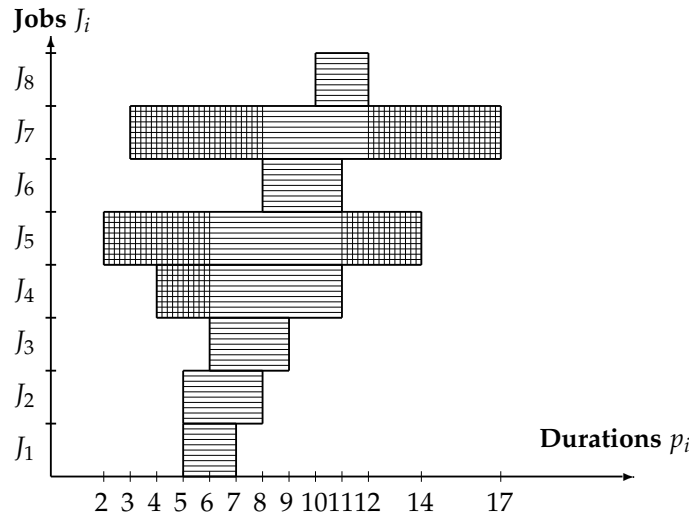


Figure 1. The segments of non-optimality (double-shaded) and the segments of conditional optimality (shaded) for the jobs $J_i \in \mathcal{J}$ in the permutation $\pi_1 = (J_1, J_2, \dots, J_8) \in S$.

The segment of non-optimality for the job J_{k_r} in the permutation $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n}) \in S$ is a maximal (with respect to the inclusion) segment $[l_{k_r}^{non}, u_{k_r}^{non}] \subseteq [p_{k_r}^L, p_{k_r}^U]$ such that for any point $p_{k_r}^* \in (l_{k_r}^{non}, u_{k_r}^{non})$, the permutation $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n})$ cannot be optimal for an instance $1|p^*|\sum C_i$ with any scenario $p^* = (\dots, p_{k_r}^*, \dots) \in T$. Thus, due to the necessary and sufficient condition (2) for the permutation $\pi_k \in S$ to be optimal for the instance $1|p|\sum C_i$, we conclude that either there exists a job $J_{k_v} \in \mathcal{J}$ such that the inequality $r < v$ holds along with the following condition:

$$p_{k_v}^U = l_{k_r}^{non} < p_{k_r}^U = u_{k_r}^{non} \tag{4}$$

or there exists a job $J_{k_w} \in \mathcal{J}$ such that $w < r$ and the following condition holds:

$$l_{k_r}^{non} = p_{k_r}^U < u_{k_r}^{non} = p_{k_w}^L. \tag{5}$$

Furthermore, due to Definition 1, the segment $[l_{k_r}^{non}, u_{k_r}^{non}]$ of non-optimality for the job J_{k_r} in the permutation $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n})$ has no common point with the open interval $(l_{k_r}^{opt}, u_{k_r}^{opt})$ of optimality for the job J_{k_r} :

$$[l_{k_r}^{non}, u_{k_r}^{non}] \cap (l_{k_r}^{opt}, u_{k_r}^{opt}) = \emptyset. \tag{6}$$

In Figure 1, the segments of non-optimality for all jobs $J_i \in \mathcal{J}$ in the permutation $\pi_1 = (J_1, J_2, \dots, J_8)$ are double-shaded.

The segment of conditional optimality for the job J_{k_r} in the permutation $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n}) \in S$ is a maximal (with respect to the inclusion) segment $[l_{k_r}^{opt}, u_{k_r}^{opt}] \subseteq [p_{k_r}^L, p_{k_r}^U]$ such that each point $p_{k_r}^* \in [l_{k_r}^{opt}, u_{k_r}^{opt}]$ does not belong to the open interval of non-optimality, $p_{k_r}^* \notin (l_{k_r}^{non}, u_{k_r}^{non})$, and there exists a job $J_{k_d} \in \mathcal{J}, d \neq r$, with the following inclusion: $p_{k_r}^* \in [p_{k_d}^L, p_{k_d}^U]$. Thus, one can conclude that for some points $p_{k_u} \in [l_{k_r}^{opt}, u_{k_r}^{opt}]$, the permutation $\pi_k \in S$ is optimal for the instance $1|p'|\sum C_i$, where $p' = (\dots, p_{k_u}, \dots) \in T$, while for other points $p_{k_v} \in [l_{k_r}^{opt}, u_{k_r}^{opt}]$, the permutation $\pi_k \in S$ cannot be optimal for the instance $1|p''|\sum C_i$, where $p'' = (\dots, p_{k_v}, \dots) \in T$.

The segment $[l_{k_r}^{copt}, u_{k_r}^{copt}]$ of conditional optimality for the job J_{k_r} in the permutation $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n}) \in S$ has no common point with the open interval of optimality $(l_{k_r}^{opt}, u_{k_r}^{opt})$ and no common point with the open interval of non-optimality $(l_{k_r}^{non}, u_{k_r}^{non})$:

$$[l_{k_r}^{copt}, u_{k_r}^{copt}] \cap (l_{k_r}^{opt}, u_{k_r}^{opt}) = \emptyset; \tag{7}$$

$$[l_{k_r}^{copt}, u_{k_r}^{copt}] \cap (l_{k_r}^{non}, u_{k_r}^{non}) = \emptyset. \tag{8}$$

If the segment $[l_{k_r}^{copt}, u_{k_r}^{copt}]$ of conditional optimality is empty for the job $J_{k_r} \in \mathcal{J}$, we say that this job J_{k_r} has no conditional optimality in the permutation π_k .

In Figure 1, all segments of conditional optimality for the jobs $J_i \in \mathcal{J}$ in the permutation $\pi_1 = (J_1, J_2, \dots, J_8)$ are shaded.

Remark 3. Due to Theorem 1, for each job $J_i \in \mathcal{J}$ in the permutation $\pi_k \in S$, there may exist at most one segment of optimality, at most two segments of conditional optimality and at most two segments of non-optimality.

In Figure 1, job J_4 has one segment $[4, 6]$ of non-optimality and one segment of conditional optimality $[6, 11]$. Job J_5 has two segments of non-optimality $[2, 6]$ and $[11, 14]$ and one segment of conditional optimality $[6, 11]$.

The following claim is based on Remark 3 and the above definitions of the segments of optimality, non-optimality and conditional optimality.

Lemma 4. Each segment $[p_{k_r}^L, p_{k_r}^U]$ of possible durations of the job $J_{k_r} \in \mathcal{J}$ is the union of the segments of optimality, non-optimality and conditional optimality for the job J_{k_r} in the permutation $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n}) \in S$.

We next show that for constructing the optimality region $\mathcal{OR}(\pi_k, T)$ for the permutation $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n}) \in S$, it is sufficient to construct the optimality region for the instance $1|\hat{p}_i^L \leq p_i \leq \hat{p}_i^U| \sum C_i$ with the reduced segments of job durations: $[\hat{p}_i^L, \hat{p}_i^U] \subseteq [p_i^L, p_i^U]$. To construct the reduced segments for the permutation $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n}) \in S$, we use the equalities (9) for all jobs $J_{k_r} \in \{J_{k_1}, J_{k_2}, \dots, J_{k_n}\} = \mathcal{J}$:

$$\hat{p}_{k_r}^L = \max_{1 \leq j \leq r \leq n} p_{k_j}^L, \hat{p}_{k_r}^U = \min_{1 \leq r \leq j \leq n} p_{k_j}^U. \tag{9}$$

We denote $\hat{T} = [\hat{p}_1^L, \hat{p}_1^U] \times [\hat{p}_2^L, \hat{p}_2^U] \times \dots \times [\hat{p}_n^L, \hat{p}_n^U]$. One can prove the following claim similarly to the proof of Theorem 2 proven in [24].

Theorem 5. The optimality region for the permutation $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n}) \in S$ for the instance $1|p_i^L \leq p_i \leq p_i^U| \sum C_i$ is equal to the optimality region for the same permutation for the instance $1|\hat{p}_i^L \leq p_i \leq \hat{p}_i^U| \sum C_i$ with the reduced segments $[\hat{p}_i^L, \hat{p}_i^U]$ of the possible durations of jobs $J_i \in \mathcal{J}$ determined in (9).

Figure 2 represents the segments of non-optimality and conditional optimality for jobs $J_i \in \mathcal{J}$ in the permutation $\pi_2 = (J_1, J_2, J_4, J_5, J_6, J_8, J_7, J_3) \in S$ for the instance $1|p_i^L \leq p_i \leq p_i^U| \sum C_i$ with the input data T given in Table 1.

From Definition 3 and Theorem 5, one can directly derive the following claim.

Lemma 5. For the instance $1|\hat{p}_i^L \leq p_i \leq \hat{p}_i^U| \sum C_i$ with the reduced segments $[\hat{p}_i^L, \hat{p}_i^U]$, $J_i \in \mathcal{J}$, of the job durations determined in (9), the open interval of optimality $(l_{k_r}^{opt}, u_{k_r}^{opt})$ for the job J_{k_r} in the permutation $\pi_k \in S$ has no common point with the segment $[p_{k_d}^L, p_{k_d}^U]$ of possible durations of any job $J_{k_d} \in \mathcal{J}$, $d \neq r$, i.e., the following equality holds:

$$(l_{k_r}^{opt}, u_{k_r}^{opt}) \cap [p_{k_d}^L, p_{k_d}^U] = \emptyset. \tag{10}$$

We next prove a criterion for the extreme case when the equality $\mathcal{OR}(\pi_k, T) = \emptyset$ holds.

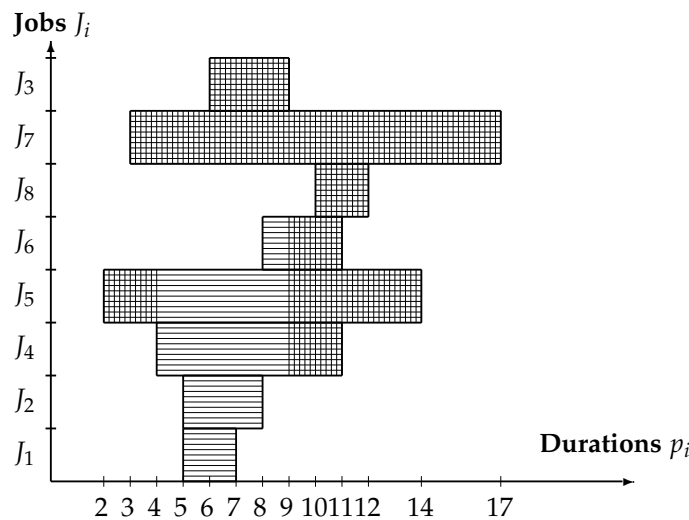


Figure 2. The segments of non-optimality (double-shaded) and the segments of conditional optimality (shaded) for the jobs $J_i \in \mathcal{J}$ in the permutation $\pi_2 = (J_1, J_2, J_4, J_5, J_6, J_8, J_7, J_3) \in S$.

Theorem 6. *The optimality region $\mathcal{OR}(\pi_k, T)$ for the permutation $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n}) \in S$ is empty, if and only if there exists at least one job $J_{k_r} \in \mathcal{J}$ with the inequality $p_{k_r}^L < p_{k_r}^U$ in the permutation π_k , which has no segment of optimality and no conditional optimality.*

Proof. Sufficiency. Let there exist a job $J_{k_r} \in \mathcal{J}$ in the permutation $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n})$, which has no segment of optimality and no conditional optimality. Due to the inequality $p_{k_r}^L < p_{k_r}^U$ and Lemma 4, the relations $[u_{k_r}^{non}, u_{k_r}^{non}] = [p_{k_r}^L, p_{k_r}^U] \neq \emptyset$ hold, and either there exists a job $J_{k_v} \in \mathcal{J}$ such that $r < v$ and the condition (4) holds or there exists a job $J_{k_w} \in \mathcal{J}$ such that $w < r$ and the condition (5) holds.

In the former case, the inequality $p_{k_v} < p_{k_r}$ holds for each duration $p_{k_r} \in [p_{k_r}^L, p_{k_r}^U]$ of the job J_{k_r} and for each duration $p_{k_v} \in [p_{k_v}^L, p_{k_v}^U]$ of the job J_{k_v} . In the latter case, the inequality $p_{k_w} > p_{k_r}$ holds for each duration $p_{k_r} \in [p_{k_r}^L, p_{k_r}^U]$ of the job J_{k_r} and for each duration $p_{k_w} \in [p_{k_w}^L, p_{k_w}^U]$ of the job J_{k_w} .

Due to Theorem 1, in both cases the permutation π_k cannot be optimal for the instance $1|p|\sum C_i$ with any scenario $p \in T$. Hence, the optimality region for the permutation $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n}) \in S$ is empty: $\mathcal{OR}(\pi_k, T) = \emptyset$. Sufficiency is proven.

Necessity. We prove necessity by a contradiction. Let the equality $\mathcal{OR}(\pi_k, T) = \emptyset$ hold. However, we assume that there is no job $J_{k_r} \in \mathcal{J}$ with the inequality $p_{k_r}^L < p_{k_r}^U$ in the permutation $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n}) \in S$, which has no segment of optimality and no conditional optimality.

Due to Definition 3, the equality $\mathcal{OR}(\pi_k, T) = \emptyset$ means that there is no scenario $p \in T$ such that the permutation π_k is optimal for the instance $1|p|\sum C_i$ with the scenario p .

However, we show next how to construct a scenario $p^* \in \hat{T}$ with the inclusion $p^* \in \mathcal{OR}(\pi_k, \hat{T})$. If the segment $[l_{k_i}^{opt}, u_{k_i}^{opt}]$ of optimality of the job J_{k_i} in the permutation π_k is not empty, then there exists a point $p_{k_i}^* \in [l_{k_i}^{opt}, u_{k_i}^{opt}]$. We choose the value of $p_{k_i}^*$ as the duration of the job J_{k_i} .

If the segment $[l_{k_j}^{opt}, u_{k_j}^{opt}]$ of optimality of the job J_{k_j} in the permutation π_k is empty, then due to the above assumption, the segment $[l_{k_j}^{cop}, u_{k_j}^{cop}]$ of conditional optimality for the job J_{k_j} in the permutation π_k is not empty. We choose the value of $l_{k_j}^{cop}$ as the duration of the job J_{k_j} , i.e., $p_{k_j}^* = l_{k_j}^{cop}$. Thus, we determine the scenario $p^* = (p_{k_1}^*, p_{k_2}^*, \dots, p_{k_n}^*)$. From the equalities (7) and (8) and Lemma 5 with the equality (10), it follows that the permutation π_k is optimal for the instance $1|p^*|\sum C_i$ with the scenario p^* . Thus, $p^* \in \mathcal{OR}(\pi_k, \hat{T})$ and the relations $\emptyset \neq \mathcal{OR}(\pi_k, \hat{T}) = \mathcal{OR}(\pi_k, T)$ hold contradicting to our assumption that $\mathcal{OR}(\pi_k, T) = \emptyset$. The proof of Theorem 6 is completed. \square

From Theorem 6, one can directly derive the following claim.

Corollary 1. *If the condition $\mathcal{OR}(\pi_k, T) \neq \emptyset$ holds, then the dimension of the optimality region $\mathcal{OR}(\pi_k, T)$ is equal to n .*

In Figure 1, there is no job $J_i \in \mathcal{J}$ in the permutation $\pi_1 = (J_1, J_2, \dots, J_8) \in S$, which has no segment of optimality and no conditional optimality. Thus, due to Theorem 6, the optimality region for the permutation $\pi_1 \in S$ is not empty, i.e., $\mathcal{OR}(\pi_1, T) \neq \emptyset$.

In Figure 2, for the segment $[l_3^{non}, u_3^{non}] = [6, 9]$ of non-optimality for the job J_3 in the permutation $\pi_2 = (J_1, J_2, J_4, J_5, J_6, J_8, J_7, J_3)$, the following equalities $[l_3^{non}, u_3^{non}] = [6, 9] = [p_3^L, p_3^U]$ hold. Thus, there exists a job $J_3 = J_{k_r} \in \mathcal{J}$ in the permutation $\pi_2 = \pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n}) \in S$, which has no segment of optimality and no conditional optimality. Due to Theorem 6, one can conclude that the optimality region for the permutation $\pi_2 = (J_1, J_2, J_4, J_5, J_6, J_8, J_7, J_3) \in S$ is empty, i.e., $\mathcal{OR}(\pi_2, T) = \emptyset$.

We next prove a criterion for another extreme case for the optimality region $\mathcal{OR}(\pi_k, T)$, namely, we prove the necessary and sufficient condition for the equality $\mathcal{OR}(\pi_k, T) = T$ when the optimality region is maximally possible.

Theorem 7. *The optimality region for the permutation $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n}) \in S$ is maximally possible (i.e., the equality $\mathcal{OR}(\pi_k, T) = T$ holds), if and only if for each job $J_{k_r} \in \mathcal{J}$ in the permutation π_k the following equality holds:*

$$[l_{k_r}^{opt}, u_{k_r}^{opt}] = [p_{k_r}^L, p_{k_r}^U]. \tag{11}$$

Proof. Sufficiency. Let the equality (11) hold for each job $J_{k_r} \in \mathcal{J}$ in the permutation $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n})$.

Due to Definition 1 and Remark 1, the following equalities hold: $\mathcal{OB}(\pi_k, T) = \times_{k_r \in M} [l_{k_r}^*, u_{k_r}^*] = \times_{k_r \in M} [l_{k_r}^{opt}, u_{k_r}^{opt}] = \times_{k_r \in M} [p_{k_r}^L, p_{k_r}^U] = T$, where $M = \{1, 2, \dots, n\}$. From Definition 1, it follows that the permutation π_k is optimal for the instance $1|p'| \sum C_i$ with any scenario $p' \in \mathcal{OB}(\pi_k, T) = T$. Thus, due to Definition 3, we obtain the desired equality $\mathcal{OR}(\pi_k, T) = T$. Sufficiency is proven.

Necessity. Let the equality $\mathcal{OR}(\pi_k, T) = T$ hold. However, we assume that there is a job $J_{k_r} \in \mathcal{J}$ in the permutation $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n}) \in \Pi$ such that the equality (11) does not hold.

Due to Lemma 4, either there exists a segment of non-optimality $[l_{k_r}^{non}, u_{k_r}^{non}] \neq \emptyset$ or a segment of conditional optimality $[l_{k_r}^{cop}, u_{k_r}^{cop}] \neq \emptyset$ for job $J_{k_r} \in \mathcal{J}$ in the permutation $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n})$. In the former case, the equality (6) holds. In the latter case, the equality (8) holds.

Thus, in both cases, there exists a scenario $p^* = (\dots, p_{k_r}^*, \dots) \in T$, where $p_{k_r}^* \in (l_{k_r}^{non}, u_{k_r}^{non}) \cup (l_{k_r}^{cop}, u_{k_r}^{cop}) \neq \emptyset$, such that the permutation π_k is not optimal for the instance $1|p^*| \sum C_i$ with the scenario $p^* \in T$. Hence, due to Definition 3 we obtain a contradiction $\mathcal{OR}(\pi_k, T) \neq T$ with the above assumption. This contradiction completes the proof of Theorem 7. \square

In the rest of this paper, we show how to use the above results for solving the uncertain scheduling problem $1|p_i^L \leq p_i \leq p_i^U| \sum C_i$ approximately.

4. Algorithms for Calculating a Quasi-Perimeter of the Optimality Region for the Fixed Permutation

We next present Algorithm 1 for testing the equality $\mathcal{OR}(\pi_k, T) = \emptyset$. If it appears that the optimality region is not empty for the permutation π_k , i.e., $\mathcal{OR}(\pi_k, T) \neq \emptyset$, then Algorithm 1 constructs an instance $1|\hat{p}_i^L \leq p_i \leq \hat{p}_i^U| \sum C_i$ with the reduced segments \hat{T} of possible durations for the jobs \mathcal{J} .

Algorithm 1: Construction of the instance with the reduced segments of possible durations

Input: The segments $[p_i^L, p_i^U]$ for all jobs $J_i \in \mathcal{J}$;
the permutation $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n}) \in S$.
Output: The reduced segments $[\hat{p}_i^L, \hat{p}_i^U]$ for all jobs $J_i \in \mathcal{J}$ if $\mathcal{OR}(\pi_k, T) \neq \emptyset$.

Step 1: Set $\hat{p}_{k_1}^L = p_{k_1}^L, t_L = p_{k_1}^L, r = 2$;
Step 2: **IF** $p_{k_r}^U \geq t_L$ **THEN GOTO** step 3 **ELSE** $[l_{k_r}^{non}, u_{k_r}^{non}] = [p_{k_r}^L, p_{k_r}^U]$;
GOTO step 6;
Step 3: **IF** $p_{k_r}^L > t_L$ **THEN** Set $t_L = p_{k_r}^L, \hat{p}_{k_r}^L = t_L, r := r + 1$;
ELSE Set $\hat{p}_{k_r}^L = t_L, r := r + 1$;
Step 4: **IF** $r \leq n$ **THEN GOTO** step 2 **ELSE** Set $\hat{p}_{k_n}^U = p_{k_n}^U, t_U = p_{k_n}^U$;
Step 5: **FOR** $r = n - 1$ **to** 1 **STEP -1 DO**
IF $p_{k_r}^U < t_U$ **THEN** Set $t_U = p_{k_r}^U, \hat{p}_{k_r}^U = t_U$ **ELSE** Set $\hat{p}_{k_r}^U = t_U$;
END FOR STOP
Step 6: $\mathcal{OR}(\pi_k, T) = \emptyset$ **STOP**.

In steps 1, 2, 3 and 6 of Algorithm 1, the equality $\mathcal{OR}(\pi_k, T) = \emptyset$ is tested. If $\mathcal{OR}(\pi_k, T) \neq \emptyset$, then the problem $1 | \hat{p}_i^L \leq p_i \leq \hat{p}_i^U | \sum C_i$ with the reduced segments \hat{T} of the feasible durations of the jobs \mathcal{J} is constructed in steps 2–5. It takes $O(n)$ time to realize Algorithm 1.

4.1. A Quasi-Perimeter of the Optimality Region $\mathcal{OR}(\pi_k, T)$

Due to Theorem 5, the optimality region for the permutation $\pi_k \in S$ for the problem $1 | p_i^L \leq p_i \leq p_i^U | \sum C_i$ coincides with the optimality region for the same permutation π_k for the problem $1 | \hat{p}_i^L \leq p_i \leq \hat{p}_i^U | \sum C_i$ with the reduced segments \hat{T} of the feasible durations of the jobs \mathcal{J} . Therefore, in what follows, we consider the problem $1 | \hat{p}_i^L \leq p_i \leq \hat{p}_i^U | \sum C_i$ instead of the problem $1 | p_i^L \leq p_i \leq p_i^U | \sum C_i$.

Definition 4. A maximal permutation $s_v^{\pi_k} = (J_{k_v}, J_{k_{v+1}}, \dots, J_{k_{m_v}})$, where $1 \leq v \leq m_v \leq n$ and $|s_v^{\pi_k}| = m_v$, is called a section of the permutation $\pi_k \in S$, if for each real number $d \in (\hat{p}_{k_v}^L, \hat{p}_{k_{m_v}}^U)$, there exists a job J_{k_i} with $i \in \{v, v + 1, \dots, m_v\}$ such that the inclusion $d \in (\hat{p}_{k_i}^L, \hat{p}_{k_i}^U)$ holds. The segment $[\hat{p}_{k_v}^L, \hat{p}_{k_{m_v}}^U]$ is called a scope of the section $s_v^{\pi_k}$. The section $s_v^{\pi_k}$ consisting of a single job, $s_v^{\pi_k} = (J_{k_v})$, is called a trivial section.

It is clear that for each fixed permutation $\pi_k \in S$, a set of all sections $\{s_1^{\pi_k}, s_{m_1+1}^{\pi_k}, \dots, s_w^{\pi_k}\} =: S(\pi_k)$ is uniquely determined. Note that index w in the last section $s_w^{\pi_k}$ coincides with index w of the job J_{k_w} , which is the first job in the section $s_w^{\pi_k} = (J_{k_w}, J_{k_{w+1}}, \dots, J_{k_{m_w}})$ of the permutation π_k .

Remark 4. Definition 4 implies that each job $J_{k_i} \in \mathcal{J}$ either belongs to a single section of the permutation π_k or this job does not belong to any section of the permutation π_k . Furthermore, if there exists at least one job $J_{k_i} \in \mathcal{J}$, which does not belong to any section from the set $S(\pi_k)$, then this job has no segment of optimality and no conditional optimality and, due to Theorem 6, the equality $\mathcal{OR}(\pi_k, T) = \emptyset$ holds.

From Remark 4 and Theorem 6, one can derive the following claim.

Corollary 2. The condition $\mathcal{OR}(\pi_k, T) \neq \emptyset$ holds if and only if $\pi_k = (s_1^{\pi_k}, s_{m_1+1}^{\pi_k}, \dots, s_w^{\pi_k})$.

If the section $s_v^{\pi_k} \in S(\pi_k)$ is trivial, i.e., the equality $s_v^{\pi_k} = (J_{k_v})$ holds, then $[\hat{p}_{k_v}^L, \hat{p}_{k_v}^U]$ is a scope of the section $s_v^{\pi_k}$. If the section $s_j^{\pi_k} \in S(\pi_k)$ is not trivial and $\mathcal{OR}(\pi_k, T) \neq \emptyset$, then we partition the scope

$[\widehat{p}_{k_j}^L, \widehat{p}_{k_{m_j}}^U]$ of the section $s_j^{\pi_k} \in S(\pi_k)$ into the maximal (with respect to the inclusion) semi-intervals of the optimality and conditional optimality:

$$[\widehat{p}_{k_j}^L, \widehat{p}_{k_{m_j}}^U] = [l_1^j(s_j^{\pi_k}), u_1^j(s_j^{\pi_k})] \cup [l_2^j(s_j^{\pi_k}), u_2^j(s_j^{\pi_k})] \cup \dots \cup [l_{n(j)}^j(s_j^{\pi_k}), u_{n(j)}^j(s_j^{\pi_k})]. \tag{12}$$

In the equality (12), the semi-intervals $[l_i^j(s_j^{\pi_k}), u_i^j(s_j^{\pi_k})]$ differ one from another in a way such that for different subsets $\mathcal{J}_i^j = \{J_{k_i}, J_{k_{i+1}}, \dots, J_{k_{|\mathcal{J}_i^j|}}\}$ of the set of jobs $\{J_{k_j}, J_{k_{j+1}}, \dots, J_{k_{m_j}}\}$, where $j \leq i \leq m_j$, the inclusion $[l_i^j(s_j^{\pi_k}), u_i^j(s_j^{\pi_k})] \subseteq [\widehat{p}_{k_r}^L, \widehat{p}_{k_r}^U]$ holds for each job $J_{k_r} \in \mathcal{J}_i^j$. Let $\widehat{J}_i^j = (J_{k_i}, J_{k_{i+1}}, \dots, J_{k_{|\mathcal{J}_i^j|}})$ denote a permutation of all jobs from the set $\mathcal{J}_i^j = \{J_{k_i}, J_{k_{i+1}}, \dots, J_{k_{|\mathcal{J}_i^j|}}\}$. Note that the permutation \widehat{J}_i^j is a part of the permutation π_k , where index i in the permutation \widehat{J}_i^j coincides with index i of the job J_{k_i} , which is the first job in the permutation \widehat{J}_i^j .

Let the condition $\mathcal{OR}(\pi_k, T) \neq \emptyset$ hold. We can calculate a quasi-perimeter of the optimality region $\mathcal{OR}(\pi_k, T)$. We define a quasi-perimeter $Per(\pi_k, T)$ of the optimality region $\mathcal{OR}(\pi_k, T)$ as a sum of all lengths (cardinalities) of the segments (sets) $OS(J_{k_r}, \pi_k)$, where jobs $J_{k_r} \in \mathcal{J}$ occupy optimal positions in the permutation π_k :

$$Per(\pi_k, T) = \sum_{r=1}^n OS(J_{k_r}, \pi_k). \tag{13}$$

A sum of the lengths (of the cardinalities) $OS(J_{k_r}, \pi_k)$ for the job $J_{k_r} \in \mathcal{J}$ in the permutation π_k may be calculated as follows:

$$OS(J_{k_r}, \pi_k) = (u_{k_r}^{opt} - l_{k_r}^{opt}) + OS_{k_r}^{copt}, \tag{14}$$

where the value of $(u_{k_r}^{opt} - l_{k_r}^{opt})$ is a length of the segment $[l_{k_r}^{opt}, u_{k_r}^{opt}]$ of the optimality for the job J_{k_r} in the permutation π_k . In the equality (14), the value of $OS_{k_r}^{copt}$ determines a cardinality of the optimality subset, which is based on a single segment $[l_{k_r}^{copt}, u_{k_r}^{copt}]$ or on both segments $[l_{k_r}^{copt}, u_{k_r}^{copt}]$ and $[l_{k_r}^{(copt)}, u_{k_r}^{(copt)}]$ (see Remark 3) of the conditional optimality of the job J_{k_r} in the permutation π_k .

To calculate the value of $OS(J_{k_r}, \pi_k) = (u_{k_r}^{opt} - l_{k_r}^{opt}) + OS_{k_r}^{copt}$, we use the partition (12) of the scope $[\widehat{p}_{k_j}^L, \widehat{p}_{k_{m_j}}^U]$ of the section $s_j^{\pi_k} \in S(\pi_k)$ such that the job J_{k_r} belongs to the set $\mathcal{J}_i^j = \{J_{k_i}, J_{k_{i+1}}, \dots, J_{k_{|\mathcal{J}_i^j|}}\}$ determined for the section $s_j^{\pi_k}$, i.e., the inequalities $j \leq r \leq m_j$ hold. Because of the condition $\mathcal{OR}(\pi_k, T) \neq \emptyset$, one can conclude (due to remark 4) that there exists a single section $s_j^{\pi_k}$ containing the job J_{k_r} . The value of $OS(J_{k_r}, \pi_k)$ may be calculated as follows:

$$OS(J_{k_r}, \pi_k) = (u_{k_r}^{opt} - l_{k_r}^{opt}) + OS_{k_r}^{copt} = \sum_{[l_i^j(s_j^{\pi_k}), u_i^j(s_j^{\pi_k})] \subseteq [\widehat{p}_{k_r}^L, \widehat{p}_{k_r}^U]} \frac{u_i^j(s_j^{\pi_k}) - l_i^j(s_j^{\pi_k})}{|\mathcal{J}_i^j|}. \tag{15}$$

The correctness of the equality (15) follows from the fact that the cardinality $OS(J_{k_r}, [l_i^j(s_j^{\pi_k}), u_i^j(s_j^{\pi_k})])$ of the optimality subset for the job J_{k_r} , which is based on the semi-interval $[l_i^j(s_j^{\pi_k}), u_i^j(s_j^{\pi_k})]$ of the optimality or conditional optimality for the job J_{k_r} , is equal to the following fraction:

$$OS(J_{k_r}, [l_i^j(s_j^{\pi_k}), u_i^j(s_j^{\pi_k})]) = \frac{u_i^j(s_j^{\pi_k}) - l_i^j(s_j^{\pi_k})}{|\mathcal{J}_i^j|}, \tag{16}$$

since, due to Theorem 1, a position of the job J_{k_r} may be optimal in the permutation $\pi_k \in S$, only if the following $|\mathcal{J}_i^j|$ inequalities (17) hold:

$$\begin{cases} p_{k_r} \leq p_{k_v}, \text{ if } r < v, J_{k_v} \in \mathcal{J}_i^j; \\ p_{k_r} \geq p_{k_d}, \text{ if } d < r, J_{k_d} \in \mathcal{J}_i^j; \\ l_i^j(s_j^{\pi_k}) \leq p_{k_r} \leq u_i^j(s_j^{\pi_k}). \end{cases} \tag{17}$$

4.2. How to Calculate a Quasi-Perimeter for the Fixed Permutation

We next demonstrate the above notations and formulas on the calculation of the quasi-perimeter $Per(\pi_k, T)$ for the permutation $\pi_k = \pi_3 = (J_5, J_1, J_2, J_3, J_4, J_6, J_8, J_7) = (J_{k_1}, J_{k_2}, \dots, J_{k_8})$ presented in Figure 3.

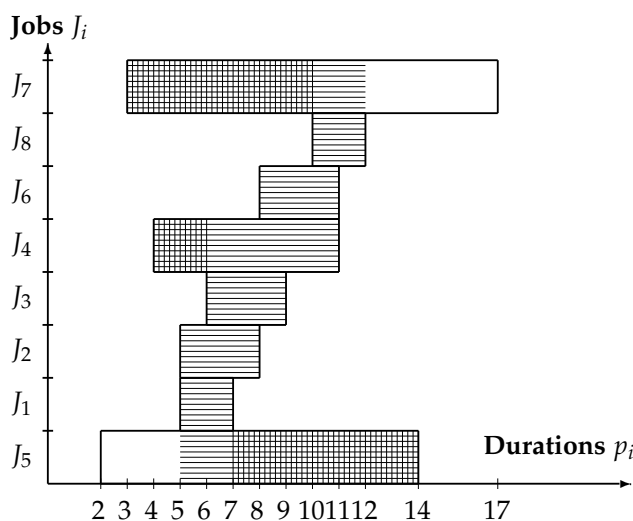


Figure 3. The segments of optimality, non-optimality (double-shaded) and conditional optimality (shaded) for the jobs $J_i \in \mathcal{J}$ in the permutation $\pi_3 = (J_5, J_1, J_2, J_3, J_4, J_6, J_8, J_7) \in S$.

For the permutation $\pi_k = \pi_3$, there exists a single section $s_1^{\pi_k} = s_1^{\pi_3} = (J_5, J_1, J_2, J_3, J_4, J_6, J_8, J_7) = \pi_3$, $S(\pi_3) = \{s_1^{\pi_3}\} = \{\pi_3\}$, with the scope $[\hat{p}_{k_1}^L, \hat{p}_{k_8}^U] = [\hat{p}_5^L, \hat{p}_7^U] = [2, 17]$. We obtain the following partition (12) of the scope $[\hat{p}_5^L, \hat{p}_7^U]$:

$$[\hat{p}_{k_1}^L, \hat{p}_{k_8}^U] = [2, 17] = [2, 5) \cup [5, 6) \cup [6, 7) \cup [7, 8) \cup [8, 9) \cup [9, 10) \cup [10, 11) \cup [11, 12) \cup [12, 17].$$

For the obtained nine semi-intervals $I_i^1(s_1^{\pi_3})$ in the above partition of the scope $[\hat{p}_{k_1}^L, \hat{p}_{k_8}^U]$, the following equalities hold: $\hat{J}_1^1 = (J_5)$, $\hat{J}_2^1 = (J_5, J_1, J_2)$, $\hat{J}_3^1 = (J_5, J_1, J_2, J_3, J_4)$, $\hat{J}_4^1 = (J_2, J_3, J_4)$, $\hat{J}_5^1 = (J_3, J_4, J_6)$, $\hat{J}_6^1 = (J_4, J_6)$, $\hat{J}_7^1 = (J_4, J_6, J_8, J_7)$, $\hat{J}_8^1 = (J_8, J_7)$, $\hat{J}_9^1 = (J_7)$. Using the equality (14), we calculate the optimality set $OS(J_{k_1}, \pi_k) = OS(J_5, \pi_3)$ for the job $J_{k_1} = J_5$ in the permutation $\pi_k = \pi_3$ as follows:

$$OS(J_{k_1}, \pi_k) = OS(J_5, \pi_3) = (u_5^{opt} - l_5^{opt}) + OS_5^{copt} = (5 - 2) + OS_5^{copt} = 3 + \frac{1}{3} + \frac{1}{5} = 3\frac{8}{15},$$

where the value of OS_5^{copt} is calculated based on the equality (15), namely: $OS_{k_1}^{copt} = \sum_{[l_i^1(s_1^{\pi_3}), u_i^1(s_1^{\pi_3})] \subseteq [\hat{p}_{k_1}^L, \hat{p}_{k_1}^U]} \frac{u_i^1(s_1^{\pi_3}) - l_i^1(s_1^{\pi_3})}{|\mathcal{J}_i^1|} = \frac{6-5}{3} + \frac{7-6}{5} = \frac{8}{15}$.

A cardinality of the optimality set OS_5^{copt} is illustrated on Figure 4, where the pyramid $\mathcal{PCO}\{J_{k_1}[l_{k_1}^1, u_{k_1}^1]\hat{J}_1^1\} = \mathcal{PCO}\{J_5[5, 6](J_5, J_1, J_2)\}$ of the optimality subset OS_5^{copt} for the job J_5 , which is based on the semi-interval $[5, 6)$ for the permutation $\hat{J}_1^1 = (J_5, J_1, J_2)$ that is a part of the permutation

π_3 . The volume of the pyramid $\mathcal{PCO}\{J_5[5,6](J_5, J_1, J_2)\}$ is determined by the system of inequalities (17), where $\pi_k = \pi_3, J_{k_r} = J_{3_1} = J_5, \mathcal{J}_i^j = \mathcal{J}_1^1$.

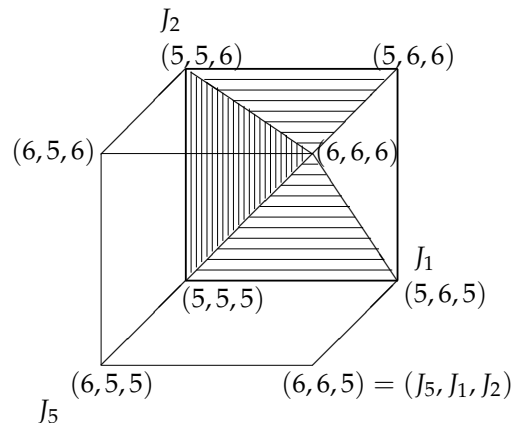


Figure 4. The pyramid $\mathcal{PCO}\{J_{k_1}[l_{k_1}^1, u_{k_1}^1]\hat{J}_1^1\} = \mathcal{PCO}\{J_5[5,6](J_5, J_1, J_2)\}$ of the optimality subset for the job J_5 , which is based on the semi-interval $[5,6]$ for the permutation $\hat{J}_1^1 = (J_5, J_1, J_2)$ that is a part of the permutation π_3 .

Similarly, we can calculate the following values: $OS(J_{k_2}, \pi_k) = OS(J_1, \pi_3) = \frac{8}{15}, OS(J_{k_3}, \pi_k) = OS(J_2, \pi_3) = \frac{13}{15}, OS(J_{k_4}, \pi_k) = OS(J_3, \pi_3) = \frac{13}{15}, OS(J_{k_5}, \pi_k) = OS(J_4, \pi_3) = 1\frac{37}{60}, OS(J_{k_6}, \pi_k) = OS(J_6, \pi_3) = 1\frac{1}{12}, OS(J_{k_7}, \pi_k) = OS(J_8, \pi_3) = \frac{3}{4}, OS(J_{k_8}, \pi_k) = OS(J_7, \pi_3) = 5\frac{3}{4}$. Using the equality (13), we calculate the quasi-perimeter of the optimality region for the permutation π_3 as follows: $Per(\pi_3, T) = \sum_{r=1}^8 OS(J_{3_r}, \pi_3) = 3\frac{8}{15} + \frac{8}{15} + \frac{13}{15} + \frac{13}{15} + 1\frac{37}{60} + 1\frac{1}{12} + \frac{3}{4} + 5\frac{3}{4} = 15$.

It should be noted that the above quasi-perimeter $Per(\pi_3, T)$ is equal to the length $15 = 17 - 2$ of the scope $[\hat{p}_{3_1}^L, \hat{p}_{3_8}^U] = [2, 17]$ of the single section $s_1^{\pi_3}$ of the permutation $\pi_3 = s_1^{\pi_3}$. The following theorem shows that such an equality is not accidental.

Theorem 8. If the condition $\mathcal{OR}(\pi_k, T) \neq \emptyset$ holds, one can calculate the quasi-perimeter $Per(\pi_k, T)$ of the optimality region for the permutation $\pi_k \in S$ as follows:

$$Per(\pi_k, T) = \sum_{s_j^{\pi_k} \in S(\pi_k)} (\hat{p}_{k_{m_j}}^U - \hat{p}_{k_j}^L), \tag{18}$$

where $(\hat{p}_{k_{m_j}}^U - \hat{p}_{k_j}^L)$ is a length of the scope of the section $s_j^{\pi_k} \in S(\pi_k)$.

Proof. Due to Corollary 2, the above condition $\mathcal{OR}(\pi_k, T) \neq \emptyset$ implies the equality $\pi_k = (s_j^{\pi_k}, s_{m_j+1}^{\pi_k}, \dots, s_w^{\pi_k})$. Therefore, the following equality also holds:

$$Per(\pi_k, T) = \sum_{s_j^{\pi_k} \in S(\pi_k)} Per(s_j^{\pi_k}, T), \tag{19}$$

where $Per(s_j^{\pi_k}, T)$ denotes a quasi-perimeter of the optimality region for the section $s_j^{\pi_k}$. Let $OS(\mathcal{J}_i^j, \pi_k)$ denote a quasi-perimeter of the optimality region for the permutation $\hat{J}_i^j = (J_{k_i}, J_{k_{i+1}}, \dots, J_{k_{|\mathcal{J}_i^j|}})$ of all jobs from the set $\mathcal{J}_i^j = \{J_{k_i}, J_{k_{i+1}}, \dots, J_{k_{|\mathcal{J}_i^j|}}\}$.

For calculating the quasi-perimeter $Per(\pi_k, T)$ of the optimality region $\mathcal{OR}(\pi_k, T)$, we will use a summation of the values $OS(J_{k_r}, [l_i^j(s_j^{\pi_k}), u_i^j(s_j^{\pi_k})])$ through all jobs J_{k_r} in the permutation \hat{J}_r^j . Thus, instead of the formulas (13)–(15), we use the equality (16) and the equalities (20) and (21):

$$Per(s_j^{\pi_k}, T) = \sum_{i=j}^{m_j} OS(\mathcal{J}_i^j, \pi_k), \tag{20}$$

$$OS(\mathcal{J}_i^j, \pi_k) = \sum_{r=i}^{|\mathcal{J}_i^j|} OS(J_{k_r}, [l_i^j(s_j^{\pi_k}), u_i^j(s_j^{\pi_k})]). \tag{21}$$

The equality (20) follows from the above definition of $Per(s_j^{\pi_k}, T)$. The equality (21) follows from the definition of $OS(\mathcal{J}_i^j, \pi_k)$. Using the equalities (16) and (21) we obtain

$$OS(\mathcal{J}_i^j, \pi_k) = \sum_{r=i}^{|\mathcal{J}_i^j|} OS(J_{k_r}, [l_i^j(s_j^{\pi_k}), u_i^j(s_j^{\pi_k})]) = |\mathcal{J}_i^j| \frac{u_i^j(s_j^{\pi_k}) - l_i^j(s_j^{\pi_k})}{|\mathcal{J}_i^j|} = u_i^j(s_j^{\pi_k}) - l_i^j(s_j^{\pi_k}). \tag{22}$$

Using the equalities (20) and (22) we obtain

$$Per(s_j^{\pi_k}, T) = \sum_{i=j}^{m_j} u_i^j(s_j^{\pi_k}) - l_i^j(s_j^{\pi_k}) = \hat{p}_{k_{m_j}}^U - \hat{p}_{k_j}^L. \tag{23}$$

Using the equalities (19) and (23) we obtain the desired equality (18) as follows:

$$Per(\pi_k, T) = \sum_{s_j^{\pi_k} \in S(\pi_k)} Per(s_j^{\pi_k}, T) = \sum_{s_j^{\pi_k} \in S(\pi_k)} (\hat{p}_{k_{m_j}}^U - \hat{p}_{k_j}^L).$$

Theorem 8 is proven. \square

5. The Largest Quasi-Perimeter of the Optimality Region $\mathcal{OR}(\pi_k, T)$ for the Problem

$$1|\hat{p}_i^L \leq p_i \leq \hat{p}_i^U| \Sigma C_i$$

We call the permutation $\pi_k \in S$ an effective permutation, if this permutation has the largest quasi-perimeter $Per(\pi_k, T)$ of the optimality region $\mathcal{OR}(\pi_k, T)$ among all permutations in the set S . The following claim follows directly from Theorem 8.

Corollary 3. *If the following equality holds:*

$$\sum_{s_j^{\pi_k} \in S(\pi_k)} (\hat{p}_{k_{m_j}}^U - \hat{p}_{k_j}^L) = \hat{p}_{k_n}^U - \hat{p}_{k_1}^L \tag{24}$$

for the permutation $\pi_k \in S$, then this permutation is effective.

Since the equality (24) holds for the permutation $\pi_3 = (J_5, J_1, J_2, J_3, J_4, J_6, J_8, J_7)$, one can conclude that this permutation is effective due to Corollary 3.

We next show how to find an effective permutation $\pi_k \in S$ in the general case of the problem $1|\hat{p}_i^L \leq p_i \leq \hat{p}_i^U| \Sigma C_i$. Similarly to the proof of Theorem 4 given in [23], we can prove the following claim.

Theorem 9. *Let all jobs from the set \mathcal{J} be fixed in their blocks from the set B . Then the effective permutation $\pi_k \in S$ may be constructed in $O(n \log n)$ time.*

Thus, due to Theorem 9, the main problem, which must be solved for the construction of the effective permutation $\pi_k \in S$, is the optimal distribution of all non-fixed jobs between the effective sub-permutations of the jobs fixed in the block B_1 , those fixed in the block B_2 , and so on, those fixed in the block B_m . Let \mathcal{J}^{non} denote a set of all non-fixed jobs of the set \mathcal{J} . The following lemma shows

that we also need to find optimal positions for some fixed jobs of the set \mathcal{J} in the desired effective permutation $\pi_k \in S$.

Theorem 10. *An effective sub-permutation of all jobs, which are fixed in the block $B_r = \{J_{r_1}, J_{r_2}, \dots, J_{r_{|B_r|}}\} \subseteq B$, exists if and only if there is no job $J_{r_d} \in B_r$ such that the following conditions hold simultaneously:*

$$p_{r_d}^L = \min\{p_{r_i}^L : J_{r_i} \in B_r\} < \min\{p_{r_i}^L : J_{r_i} \in B_r \setminus \{J_{r_d}\}\}; \tag{25}$$

$$p_{r_d}^U = \max\{p_{r_i}^U : J_{r_i} \in B_r\} > \max\{p_{r_i}^U : J_{r_i} \in B_r \setminus \{J_{r_d}\}\}. \tag{26}$$

Proof. Sufficiency. Let there be no job $J_{r_d} \in B_r$ such that conditions (25) and (26) hold.

Hence, there exist at least two different jobs $J_{r_i} \in B_r$ and $J_{r_j} \in B_r$ such that the inequalities $p_{r_i}^L = \min\{p_{r_i}^L : J_{r_i} \in B_r\}$ and $p_{r_j}^U = \max\{p_{r_j}^U : J_{r_j} \in B_r\}$ hold. The effective sub-permutation of all jobs, which are fixed in the block $B_r = \{J_{r_1}, J_{r_2}, \dots, J_{r_{|B_r|}}\} \subseteq B$, looks as follows $(J_{r_i}, \dots, J_{r_j})$, where all jobs from the set $B_r \setminus \{J_{r_i}, J_{r_j}\}$ are located between jobs J_{r_i} and J_{r_j} and their order may be arbitrary.

Necessity. Let there exist a job $J_{r_d} \in B_r$ such that both conditions (25) and (26) hold.

Hence, an optimal position of the job J_{r_d} is either the first position or the last position in the effective sub-permutation of all jobs, which are fixed in the block $B_r = \{J_{r_1}, J_{r_2}, \dots, J_{r_{|B_r|}}\}$. This choice for the job J_{r_d} depends from the positions of other such jobs in the blocks $B_l \in B \setminus \{B_r\}$ in the effective permutation π_k and from the positions of jobs from the set \mathcal{J}^{non} in the effective permutation π_k . \square

The following Algorithm 2 is based on Theorem 10.

Algorithm 2: Construction of the effective permutation of the jobs fixed in the block B_r

Input: The segments $[p_{r_i}^L, p_{r_i}^U]$ for all jobs from the set \mathcal{J} , which are fixed in the block $B_r = \{J_{r_1}, J_{r_2}, \dots, J_{r_{|B_r|}}\}$, $i \in \{1, 2, \dots, |B_r|\}$.

Output: The effective sub-permutation $\pi_{B_r} = (J_{r_1}, J_{r_1}, \dots, J_{r_{|S_r|}})$ of the subset S_r of the set B_r , $S_r \subseteq B_r$, which are fixed in the block B_r .

- Step 1: Find a job J_{r_i} such that $p_{r_i}^L = \min\{p_{r_j}^L : j \in \{1, 2, \dots, |B_r|\}\}$;
- Step 2: Find a job J_{r_k} such that $p_{r_k}^U = \max\{p_{r_j}^U : j \in \{1, 2, \dots, |B_r|\} \setminus \{i\}\}$;
- Step 3: **IF** $p_{r_k}^U \geq p_{r_i}^U$ **THEN** $\pi_{B_r} = (J_{r_1}, \dots, J_{r_k})$, where jobs from the set $B_r \setminus \{J_{r_i}, J_{r_k}\}$ are ordered arbitrarily, set $S_{r_1} = \emptyset, S_{r_2} = \emptyset, S_r = B_r$ **GOTO** step 11;
- Step 4: Find a job J_{r_l} such that $p_{r_l}^L = \min\{p_{r_j}^L : j \in \{1, 2, \dots, |B_r|\} \setminus \{i\}\}$;
- Step 5: **IF** $p_{r_l}^L = p_{r_i}^L$ **THEN** $\pi_{B_r} = (J_{r_1}, \dots, J_{r_l})$, where jobs from the set $B_r \setminus \{J_{r_l}, J_{r_i}\}$ are ordered arbitrarily, set $S_{r_1} = \emptyset, S_{r_2} = \emptyset, S_r = B_r$ **GOTO** step 11;
- Step 6: Find a job J_{r_o} such that $p_{r_o}^U = \max\{p_{r_j}^U : j \in \{1, 2, \dots, |B_r|\} \setminus \{i, l\}\}$;
- Step 7: **IF** $p_{r_o}^U \geq p_{r_i}^U$ **THEN** $\pi_{B_r} = (J_{r_1}, \dots, J_{r_o})$, where jobs $B_r \setminus \{J_{r_l}, J_{r_o}, J_{r_i}\}$ are ordered arbitrarily, set $S_{r_1} = \{J_{r_i}\}, S_{r_2} = \emptyset, S_r = B_r \setminus \{J_{r_i}\}$ **GOTO** step 11;
- Step 8: Find a job J_{r_q} such that $p_{r_q}^L = \min\{p_{r_j}^L : j \in \{1, 2, \dots, |B_r|\} \setminus \{i, l\}\}$;
- Step 9: **IF** $p_{r_q}^L = p_{r_l}^L$ **THEN** $\pi_{B_r} = (J_{r_q}, \dots, J_{r_l})$, where jobs from the set $B_r \setminus \{J_{r_q}, J_{r_l}, J_{r_i}\}$ are ordered arbitrarily, set $S_{r_1} = \{J_{r_i}\}, S_{r_2} = \emptyset, S_r = B_r \setminus \{J_{r_i}\}$ **GOTO** step 11;
- Step 10: $\pi_{B_r} = (J_{r_1}, \dots, J_{r_{|S_r|}})$, where jobs from the set $S_r = B_r \setminus \{J_{r_i}, J_{r_l}\}$ are ordered arbitrarily, set $S_{r_1} = \{J_{r_i}\}, S_{r_2} = \{J_{r_l}\}$;
- Step 11: The obtained sub-permutation π_{B_r} is effective **STOP**.

The asymptotic complexity of Algorithm 2 is equal to $O(n)$. The set S_{r_1} is either empty or contains a single job $J_{r_d} \in B_r$ for which both conditions (25) and (26) hold. The set S_{r_2} is either empty or contains a single job $J_{r_z} \in B_r$ for which both conditions $p_{r_z}^L = \min\{p_{r_i}^L : J_{r_i} \in B_r \setminus \{J_{r_d}\}\} < \min\{p_{r_i}^L : J_{r_i} \in B_r \setminus \{J_{r_d}, J_{r_z}\}\}$ and $p_{r_z}^U = \max\{p_{r_i}^U : J_{r_i} \in B_r \setminus \{J_{r_d}\}\} > \max\{p_{r_i}^U : J_{r_i} \in B_r \setminus \{J_{r_d}, J_{r_z}\}\}$ hold.

Let \mathcal{J}^{fix} denote a set of all jobs $J_{r_d} \in B_r$, which are fixed in their blocks $B_r \in B$ and both conditions (25) and (26) hold. The following Algorithm 3 constructs an effective permutation $\pi_k \in S$ for the general case of the problem $1|p_i^L \leq p_i \leq p_i^U|\Sigma C_i$.

Algorithm 3: Construction of the effective permutation $\pi_k \in S$

Input: The segments $[p_i^L, p_i^U]$ for all jobs $J_i \in \mathcal{J}$;
 the effective sub-permutations $\pi_{B_r} = (J_{r_1}, J_{r_2}, \dots, J_{r_{|S_r|}})$, sets S_{r_1} and sets S_{r_2}
 for all blocks $B_r \in B$.

Output: The effective job permutation $\pi_k \in S$.

Step 1: Construct a sub-permutation $\pi_p = (\dots, \pi_{B_1}, \dots, \pi_{B_2}, \dots, \pi_{B_m}, \dots)$;

FOR $r = 1$ **to** $|B|$ **DO**
 IF $S_{r_2} = \emptyset$ **THEN** $\hat{l}_r = p_{r_1}^L, \hat{u}_r = p_{r_{|S_r|}}^U$ **ELSE**
 $\hat{l}_r = p_{r_d}^L, \hat{u}_r = p_{r_d}^U, J_{r_d} \in S_{r_2}$;
 END FOR

Step 2: Construct sets of jobs $\mathcal{J}^{fix} = \cup_{r=1}^m S_{r_1}, \tilde{\mathcal{J}} = \mathcal{J}^{fix} \cup \mathcal{J}^{non}$, sort jobs of the set $\tilde{\mathcal{J}} = \{J_{f_1}, J_{f_2}, \dots, J_{f_{|\tilde{\mathcal{J}}|}}\}$ by increasing of the mid-points of the segments $[p_{f_i}^L, p_{f_i}^U]$, obtain set $\tilde{\mathcal{J}} = (J_{f_1}, J_{f_2}, \dots, J_{f_{|\tilde{\mathcal{J}}|}})$;

Step 3: Construct set $D = \cup_{i=1}^n (p_i^L \cup p_i^U)$, sort set D by increasing $D = (b_1, b_2, \dots, b_q)$, $1 \leq q \leq 2n$, construct intervals $I_i = [b_i, b_{i+1}], i \in \{1, 2, \dots, q - 1\}$;

Step 4: $S_{res} = \emptyset, N_{res} = \emptyset, sNumRes = 1, k = 1,$
 $\hat{l}_{m+1} = \max\{p_i^U : i \in \{1, 2, \dots, n\}\}, \hat{u}_{m+1} = \max\{p_i^L : i \in \{1, 2, \dots, n\}\};$

Step 5: **IF** $k \leq |\tilde{\mathcal{J}}|$ **THEN** $iNum = 1, bNum = 1, sNum = 1, S_k = \emptyset, N_k = \emptyset$
ELSE GOTO step 10;

Step 6: **FOR** $j = 1$ **to** $q - 1$ **DO**
 IF $[b_j, b_{j+1}] \cap (p_{f_k}^L, p_{f_k}^U) \neq \emptyset$ **AND** $\hat{l}_{bNum} > b_j$ **THEN** $S_{k_sNum} := S_{k_sNum} \cup I_j$;
 IF $\hat{u}_{bNum} = b_{j+1}$ **THEN** $bNum := bNum + 1$;
 IF $\hat{u}_{bNum-1} = b_{j+1}$ **AND** $S_{k_sNum} \neq \emptyset$ **THEN** $N_{k_sNum} := bNum - 1,$
 $sNum := sNum + 1$;
END FOR

Step 7: $s = 1, S_{tmp} = \emptyset$;
FOR $i = 1$ **to** $sNum - 1$ **DO**
 FOR $j = 1$ **to** $sNumRes$ **DO**
 $S_{tmp_s} = S_{res_j} \cup S_{k_i}, N_{tmp_s} = (N_{res_j}, N_{k_i}), s := s + 1$;
 END FOR
END FOR

Step 8: $j = 1, S_d = \emptyset, N_d = \emptyset$;
WHILE $j < s$ **DO** $i = 1$;
 WHILE $i < s$ **DO**
 IF $S_{tmp_i} \subseteq S_{tmp_j}$ **AND** $i \neq j$ **AND** $S_{tmp_j} \notin S_d$ **THEN**
 $S_d := S_d \cup S_{tmp_i}, N_d := N_d \cup N_{tmp_i}; i := i + 1$;
 END j := j + 1;
 END $S_{nd} := S_{tmp} \setminus S_d, N_{nd} := N_{tmp} \setminus N_d$;

Step 9: $S_{res} = S_{nd}, N_{res} = N_{nd}, sNumRes = |S_{res}|, k := k + 1$ **GOTO** step 5;

Step 10: $maxP = 0$;
FOR $i = 1$ **to** $sNumRes$ **DO** $P = 0$;


```

FOR EACH  $I_j$  IN  $S_{res_i}$  DO  $P := P + b_{j+1} - b_j$  END FOR;
IF  $P > \max P$  THEN  $\max P = P$ ,  $\max Num = i$  END FOR
Step 11: FOR  $x = 1$  to  $m + 1$  DO  $S_x = \emptyset$  END FOR
FOR  $k = 1$  to  $|\tilde{\mathcal{J}}|$  DO  $x = N_{\max Num_k}$ ,  $S_x := S_x \cup J_k$ ; END FOR
Step 12:  $\hat{u}_0 = \min\{p_i^L : i \in \{1, 2, \dots, n\}\}$ ;
FOR  $x = 1$  to  $m + 1$  DO sort jobs of the set  $S_x$  by increasing of the
mid-points of the segments  $[\max\{p_i^L, \hat{u}_{x-1}\}, \min\{p_i^U, \hat{l}_x\}]$  and obtain
sub-permutation  $\pi_{r_x}$ ;
END FOR
Step 13: FOR  $x = 1$  to  $m$  DO
IF  $S_{x_2} = \emptyset$  THEN  $\pi_k = (\pi_k, \pi_{r_x}, \pi_{B_x})$  ELSE
IF  $S_{x_1} \subseteq S_x$  THEN  $\pi_k := (\pi_k, \pi_{r_x}, \pi_{B_x}, S_{x_2})$  ELSE
 $\pi_k := (\pi_k, \pi_{r_x}, S_{x_2}, \pi_{B_x})$ ;
END FOR
 $\pi_k := (\pi_k, S_{m+1})$  STOP.

```

6. Computational Results

In the computational experiments, we tested six classes of hard instances $1|p_i^L \leq p_i \leq p_i^U|\sum C_i$. Algorithms 1–3 were coded in C# and tested on a PC with Intel Core (TM) 2 Quad, 2.5 GHz, 4.00 GB RAM. For all tested instances, inequalities $p_i^L < p_i^U$ hold for all jobs $J_i \in \mathcal{J}$. Table 2 presents computational results for randomly generated instances of the problem $1|p_i^L \leq p_i \leq p_i^U|\sum C_i$ with $n \in \{50, 100, 500, 1000, 5000, 10,000\}$.

The segments of possible durations have been randomly generated as follows. An integer center C of the segment $[p_i^L, p_i^U]$ was generated using a uniform distribution in the range $[1, 100]$. The lower bound p_i^L of the possible duration p_i was determined using the equality $p_i^L = C \cdot (1 - \frac{\delta}{100})$, where δ denotes the maximal relative error of the durations p_i due to the given segments $[p_i^L, p_i^U]$. The upper bound p_i^U was determined using the equality $p_i^U = C \cdot (1 + \frac{\delta}{100})$. For each job $J_i \in \mathcal{J}$, the point \underline{p}_i was generated using a uniform distribution in the range $[p_i^L, p_i^U]$. In order to generate instances, where all jobs \mathcal{J} belonged to a single block, the segments $[p_i^L, p_i^U]$ of the possible durations were modified as follows: $[\tilde{p}_i^L, \tilde{p}_i^U] = [p_i^L + \underline{p} - \underline{p}_i, p_i^U + \underline{p} - \underline{p}_i]$, where $\underline{p} = \max_{i=1}^n \underline{p}_i$.

Since the inclusion $\underline{p} \in [\tilde{p}_i^L, \tilde{p}_i^U]$ holds, each constructed instance contained a single block, $|B| = 1$. The maximum absolute error of the uncertain durations p_i , $J_i \in \mathcal{J}$, is equal to $\max_{i=1}^n (p_i^U - p_i^L)$, and the maximum relative error of the uncertain durations p_i , $J_i \in \mathcal{J}$, is not greater than $2\delta\%$. We say that these instances belong to class 1.

Three distribution laws were used in our computational experiments to determine the factual durations of the jobs. If inequality $p_i^L < p_i^U$ holds, then the factual duration of the job J_i becomes known only after completing the job J_i .

We call the uniform distribution as the distribution law with number 1, the gamma distribution with the parameters $\alpha = 9$ and $\beta = 2$ as the distribution law with number 2, and the gamma distribution with the parameters $\alpha = 4$, and $\beta = 2$ as the distribution law with number 3. In each instance of class 1, for generating the factual durations for different jobs of the set \mathcal{J} , the number of the distribution law was randomly chosen from the set $\{1, 2, 3\}$. We solved 15 series of the randomly generated instances from class 1. Each series contained 10 instances with the same combination of n and δ .

In the computational experiments, we answered the question of how large the obtained relative error $\Delta = \frac{\gamma_{p^*}^k - \gamma_{p^*}^t}{\gamma_{p^*}^t} \cdot 100\%$ of the value $\gamma_{p^*}^k$ of the objective function $\gamma = \sum_{i=1}^n C_i$ was for the effective permutation π_k with respect to the actually optimal objective function value $\gamma_{p^*}^t$ calculated for the factual durations $p^* = (p_1^*, p_2^*, \dots, p_n^*) \in T$, which were known after completing all the jobs.

The number n of jobs in the instance is given in column 1 in Table 2. The half of the maximum possible errors δ of the random durations (in percentage) is given in column 2. Column 3 gives the average error Δ for the effective permutation π_k . Column 4 presents the average CPU-time in seconds. The smallest errors, average errors, largest errors for the tested series of the instances are presented in the last rows of Table 2.

Table 2. Computational results for randomly generated instances with a single block (class 1).

n	δ (%)	Δ	CPU-Time (s)
1	2	3	4
50	1	0.088066	0.028202
50	5	0.29217	0.028702
50	10	0.451719	0.027502
100	1	0.083836	0.040702
100	5	0.25303	0.040202
100	10	0.442234	0.038802
500	1	0.090923	0.162809
500	5	0.268353	0.160009
500	10	0.446225	0.162509
1000	1	0.09579	0.309918
1000	5	0.266479	0.310618
1000	10	0.443648	0.312518
5000	1	0.097144	1.196568
5000	5	0.264383	1.531488
5000	10	0.455035	1.556389
10,000	1	0.094943	3.103378
10,000	5	0.265045	3.073576
10,000	10	0.452539	2.993571
Minimum		0.083836	0.027502
Average		0.269454	0.837637
Maximum		0.455035	3.103378

In the second part of our computational experiments, Algorithms 1–3 were applied to randomly generated instances from other classes 2–6 of the problem $1|p_i^L \leq p_i \leq p_i^U|\Sigma C_i$. We randomly generated non-fixed jobs J_1, J_2, \dots, J_s , which belong to blocks B_1, B_2, \dots, B_m of the randomly generated $n - s$ fixed jobs. The lower bound p_i^L and the upper bound p_i^U on the feasible values of $p_i \in R_+^1$ of the durations of the fixed jobs, $p_i \in [p_i^L, p_i^U]$, were generated as follows.

We determined a bound of blocks $[\tilde{b}_i^L, \tilde{b}_i^U]$ for generating the cores of the blocks $[b_i^L, b_i^U] \subseteq [\tilde{b}_i^L, \tilde{b}_i^U]$ and for generating the segments $[p_i^L, p_i^U]$ for the durations of $|B_i|$ jobs from all blocks $B_i, i \in \{1, 2, \dots, m\}, [b_i^L, b_i^U] \subseteq [p_i^L, p_i^U] \subseteq [\tilde{b}_i^L, \tilde{b}_i^U]$.

Each instance in class 2 or in class 3 had a single non-fixed job J_v , whose bounds were determined as follows: $p_{J_v}^L \leq \tilde{b}_1^L \leq \tilde{b}_1^U < \tilde{b}_2^L \leq \tilde{b}_2^U < \tilde{b}_3^L \leq \tilde{b}_3^U \leq p_{J_v}^U$. Classes 2 and 3 of the solved instances differed one from another by the numbers of non-fixed jobs and the distribution laws used for choosing the factual durations of the jobs \mathcal{J} .

Each instance from classes 4 and 5 had two non-fixed jobs. In each instance from classes 2, 4, 5 and 6, for generating the factual durations for the jobs \mathcal{J} , the numbers of the distribution laws were randomly chosen from the set $\{1, 2, 3\}$, and they were indicated in column 4 in Table 3. In the instances of class 6, the cores of the blocks were determined in order to generate different numbers of non-fixed jobs in different instances. The numbers of non-fixed jobs were randomly chosen from the set $\{2, 3, \dots, 8\}$.

The numbers n of the jobs are presented in column 1 in Table 3. Column 2 represents the number $|B|$ of blocks in the solved instance and column 3 the number of non-fixed jobs. The distribution laws used for determining the factual durations of the jobs are indicated in column 4 in Table 3. Column 6 presents average numbers λ of the maximal number of the variants of the non-dominated distributions

of the jobs $\mathcal{J}^{fix} \cup \mathcal{J}^{non}$ in the effective permutation π_k while it was constructed on the iterations of Algorithm 3. Each solved series contained 10 instances with the same combination of n and the other parameters. The obtained smallest, average and largest values of Δ for each series of the tested instances are presented in column 5 in Table 3 at the end of series.

Table 3. Computational results for randomly generated instances from classes 2–6.

n	$ B $	$ \mathcal{J}^{non} $	Laws	Δ	Average λ	CPU-Time (s)
1	2	3	4	5	6	7
Class 2						
50	3	1	1,2,3	0.54205	4	0.16921
100	3	1	1,2,3	0.281253	4	0.305017
500	3	1	1,2,3	0.177597	4	0.952555
1000	3	1	1,2,3	0.121447	4	1.561289
5000	3	1	1,2,3	0.111056	4	9.481842
10,000	3	1	1,2,3	0.105322	4	18.933383
	Minimum			0.105322	4	0.16921
	Average			0.223121	4	5.233883
	Maximum			0.54205	4	18.933383
Class 3						
50	3	1	1	0.575038	4	0.098006
100	3	1	1	0.284279	4	0.334319
500	3	1	1	0.132735	4	0.647537
1000	3	1	1	0.114245	4	1.389479
5000	3	1	1	0.160372	4	9.290531
10,000	3	1	1	0.149278	4	12.189497
	Minimum			0.114245	4	0.098006
	Average			0.235991	4	3.991562
	Maximum			0.575038	4	12.189497
Class 4						
50	3	2	1,2,3	0.670408	6.5	0.408923
100	3	2	1,2,3	0.402251	6.2	0.791245
500	3	2	1,2,3	0.084687	6.2	3.866421
1000	3	2	1,2,3	0.084137	6.4	8.715098
5000	3	2	1,2,3	0.066305	6.2	35.98006
10,000	3	2	1,2,3	0.061258	6.2	78.877412
	Minimum			0.061258	6.2	0.408923
	Average			0.228174	6.3	21.43986
	Maximum			0.670408	6.5	78.877412
Class 5						
50	5	2	1,2,3	0.498197	16.2	0.969455
100	5	2	1,2,3	0.157349	15	2.035616
500	5	2	1,2,3	0.069192	15	6.832491
1000	5	2	1,2,3	0.059916	15	13.53107
5000	5	2	1,2,3	0.045986	15	86.01118
10,000	5	2	1,2,3	0.047765	15	129.1963
	Minimum			0.045986	15	0.969455
	Average			0.146401	15.2	39.762685
	Maximum			0.498197	16.2	129.1963

Table 3. Cont.

n	$ B $	$ \mathcal{J}^{non} $	Laws	Δ	Average λ	CPU-Time (s)
1	2	3	4	5	6	7
Class 6						
50	2	2–4	1,2,3	1.086983	3.9	0.496728
100	2	2–4	1,2,3	0.839207	4	0.945754
500	2	2–6	1,2,3	0.843448	3.8	4.976885
1000	2	2–8	1,2,3	0.874078	3.6	13.604478
5000	2	2–8	1,2,3	0.790634	3.7	71.244575
10,000	2	2–8	1,2,3	0.768925	3.7	153.430476
		Minimum		0.768925	3.6	0.496728
		Average		0.867212	3.8	40.783149
		Maximum		1.086983	4	153.430476

7. Concluding Remarks

The uncertain scheduling problem $1|p_i^L \leq p_i \leq p_i^U| \sum C_i$ attracts the attention of the researchers since this problem is applicable in real-life scheduling and is commonly used in many multiple-resource scheduling systems, where one of the available machines is the bottleneck and uncertain machine. The optimal scheduling decisions allow the plant to reduce the costs of productions due to a better utilization of the available machines.

In Sections 2–5, we used a notion of the optimality region of a job permutation π_k and proved useful properties of the optimality region $\mathcal{OR}(\pi_k, T)$. We investigated the permutation π_k with the largest quasi-perimeter of the optimality region. Using these properties, we derived algorithms for constructing a job permutation π_k with the largest quasi-perimeter of the optimality region $\mathcal{OR}(\pi_k, T)$.

From the computational experiments, it follows that the effective permutation π_k is close to the optimal permutation, which can be determined after completing all jobs when their durations became known. We tested classes 1–6 of the problems $1|p_i^L \leq p_i \leq p_i^U| \sum C_i$. The minimal, average and maximal errors Δ of the objective function values were 0.045986, 0.313658 and 1.086983, respectively, for the effective permutations.

An attractive direction for a further research is a generalization of the obtained results to the problem $1|p_i^L \leq p_i \leq p_i^U| \sum w_i C_i$, where the given jobs may have different weights. It is also useful to find precedence constraints on the set of jobs such that the effective job permutation may be constructed similarly to Section 5.

Author Contributions: Y.N. proved theoretical results; Y.N. and N.E. jointly conceived and designed the algorithms; N.E. performed the experiments; Y.N. and N.E. analyzed the data; Y.N. wrote the paper.

Funding: This research received no external funding.

Acknowledgments: The authors are grateful to anonymous referees for their useful remarks and suggestions.

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

References

- Davis, W.J.; Jones, A.T. A real-time production scheduler for a stochastic manufacturing environment. *Int. J. Prod. Res.* **1988**, *1*, 101–112. [\[CrossRef\]](#)
- Pinedo, M. *Scheduling: Theory, Algorithms, and Systems*; Prentice-Hall: Englewood Cliffs, NJ, USA, 2002.
- Daniels, R.L.; Kouvelis, P. Robust scheduling to hedge against processing time uncertainty in single stage production. *Manag. Sci.* **1995**, *41*, 363–376. [\[CrossRef\]](#)
- Sabuncuoglu, I.; Goren, S. Hedging production schedules against uncertainty in manufacturing environment with a review of robustness and stability research. *Int. J. Comput. Integr. Manuf.* **2009**, *22*, 138–157. [\[CrossRef\]](#)
- Sotskov, Y.N.; Werner, F. *Sequencing and Scheduling with Inaccurate Data*; Nova Science Publishers: Hauppauge, NY, USA, 2014.

6. Pereira, J. The robust (minmax regret) single machine scheduling with interval processing times and total weighted completion time objective. *Comput. Oper. Res.* **2016**, *66*, 141–152. [[CrossRef](#)]
7. Grabot, B.; Geneste, L. Dispatching rules in scheduling: A fuzzy approach. *Int. J. Prod. Res.* **1994**, *32*, 903–915. [[CrossRef](#)]
8. Kasperski, A.; Zielinski, P. Possibilistic minmax regret sequencing problems with fuzzy parameters. *IEEE Trans. Fuzzy Syst.* **2011**, *19*, 1072–1082. [[CrossRef](#)]
9. Özelkan, E.C.; Duckstein, L. Optimal fuzzy counterparts of scheduling rules. *Eur. J. Oper. Res.* **1999**, *113*, 593–609. [[CrossRef](#)]
10. Sotskov, Y.N.; Egorova, N.M.; Lai, T.-C. Minimizing total weighted flow time of a set of jobs with interval processing times. *Math. Comput. Model.* **2009**, *50*, 556–573. [[CrossRef](#)]
11. Tanaev, V.S.; Sotskov, Y.N.; Strusevich, V.A. *Scheduling Theory: Multi-Stage Systems*; Kluwer Academic Publishers: Dordrecht, The Netherlands, 1994.
12. Graham, R.L.; Lawler, E.L.; Lenstra, J.K.; Rinnooy Kan, A.H.G. Optimization and approximation in deterministic sequencing and scheduling. *Ann. Discr. Appl. Math.* **1979**, *5*, 287–326.
13. Smith, W.E. Various optimizers for single-stage production. *Naval Res. Logist. Q.* **1956**, *3*, 59–66. [[CrossRef](#)]
14. Burdett, R.L.; Kozan, E. Techniques to effectively buffer schedules in the face of uncertainties. *Comput. Ind. Eng.* **2015**, *87*, 16–29. [[CrossRef](#)]
15. Goren, S.; Sabuncuoglu, I. Robustness and stability measures for scheduling: Single-machine environment. *IIE Trans.* **2008**, *40*, 66–83. [[CrossRef](#)]
16. Kasperski, A.; Zielinski, P. A 2-approximation algorithm for interval data minmax regret sequencing problems with total flow time criterion. *Oper. Res. Lett.* **2008**, *36*, 343–344. [[CrossRef](#)]
17. Kouvelis, P.; Yu, G. *Robust Discrete Optimization and Its Application*; Kluwer Academic Publishers: Boston, MA, USA, 1997.
18. Lu, C.-C.; Lin, S.-W.; Ying, K.-C. Robust scheduling on a single machine total flow time. *Comput. Oper. Res.* **2012**, *39*, 1682–1691. [[CrossRef](#)]
19. Yang, J.; Yu, G. On the robust single machine scheduling problem. *J. Combin. Optim.* **2002**, *6*, 17–33. [[CrossRef](#)]
20. Lebedev, V.; Averbakh, I. Complexity of minimizing the total flow time with interval data and minmax regret criterion. *Discr. Appl. Math.* **2006**, *154*, 2167–2177. [[CrossRef](#)]
21. Harikrishnan, K.K.; Ishii, H. Single machine batch scheduling problem with resource dependent setup and processing time in the presence of fuzzy due date. *Fuzzy Optim. Decis. Mak.* **2005**, *4*, 141–147. [[CrossRef](#)]
22. Allahverdi, A.; Aydilek, H.; Aydilek, A. Single machine scheduling problem with interval processing times to minimize mean weighted completion times. *Comput. Oper. Res.* **2014**, *51*, 200–207. [[CrossRef](#)]
23. Lai, T.-C.; Sotskov, Y.N.; Egorova, N.G.; Werner, F. The optimality box in uncertain data for minimising the sum of the weighted job completion times. *Int. J. Prod. Res.* **2018**, *56*, 6336–6362. [[CrossRef](#)]
24. Sotskov, Y.N.; Egorova, N.M. Single machine scheduling problem with interval processing times and total completion time objective. *Algorithms* **2018**, *75*, 66. [[CrossRef](#)]
25. Sotskov, Y.N.; Lai, T.-C. Minimizing total weighted flow time under uncertainty using dominance and a stability box. *Comput. Oper. Res.* **2012**, *39*, 1271–1289. [[CrossRef](#)]



© 2019 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 (<http://creativecommons.org/licenses/by/4.0/>).