Communications in Combinatorics and Optimization
Vol. xx, No. x (xxxx), pp. 1-15
https://doi.org/10.22049/cco.2025.30140.2332

CCO
Commun. Comb. Optim.

Research Article

The minimum cost problem of downgrading minimum lateness scheduling under uncertainty

Hiep Thanh Tran[1,2,3], Duc Minh Phung[2,4], Kien Trung Nguyen[5,*]

[1]Department of Applied Mathematics, Faculty of Applied Sciences, Ho Chi Minh University of Technology (HCMUT), 268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam
ttthiep.sdh241@hcmut.edu.vn, hieptt@fe.edu.vn

[2]Vietnam National University Ho Chi Minh City, Linh Trung Ward, Thu Duc City, Ho Chi Minh City, Vietnam

[3]Department of Mathematics, FPT University (HCM City Campus), Vietnam

[4]Department of Mathematics and Physics, University of Information Technology, Ho Chi Minh City, Vietnam
ducpm@uit.edu.vn

[5]Faculty of Mathematics and Informatics, Teacher College, Can Tho University, Can Tho, Vietnam
*trungkien@ctu.edu.vn

Received: 12 November 2024; Accepted: 29 June 2025
Published Online: 10 July 2025

Abstract: The minimum lateness scheduling problem seeks to create a schedule that minimizes the largest lateness of the system. This paper deals with the challenge of increasing the processing time of jobs in a minimum cost such that the minimum lateness attains a given bound. It is called the minimum cost problem of downgrading minimum lateness scheduling. Additionally, the modifying costs are represented as intervals, and we apply the minmax regret criterion to address this uncertainty. Our contribution is an $O(n^2)$ algorithm for solving the corresponding robust problem.

Keywords: robust optimization, downgrading problem, scheduling, lateness.

AMS Subject classification: 90B35, 90C27

1. Introduction

The scheduling problem is an important topic in optimization theory, with numerous applications in production, management, computer systems, etc. Generally, scheduling theory focuses on the challenge of sequencing a finite set of jobs utilizing a system

* *Corresponding Author*

with constrained resources. For concepts and algorithms, readers can refer to Brucker [8] and Pinedo [21], or the surveys by Lawler [17], Lenstra [18], and Xiong et al. [25] among other relevant literature.
Recently, there has been growing interest in modifying parameters to improve network behavior with respect to a given objective. This has led to the study of a class of problems known as up- and downgrading optimization. Specifically, the upgrading (downgrading) optimization problem involves adjusting parameters within a budget to minimize (maximize) the optimal objective. The seminal work on downgrading optimization in networks is by Fulkerson and Harding [13], who aimed to maximize the shortest path through edge length modifications. Since then, other up- and downgrading combinatorial optimization problems have been extensively investigated; for example, see [11, 12, 24] for studies on degrading various types of spanning tree problems by modifying edge lengths on graphs, or refer to [9, 10] for research on upgrading combinatorial optimization with a bottleneck function in a general setting, where network optimization is a special case.. The up- and downgrading location problem has been also focused. Gassner [14, 15] introduced the up- and downgrading 1-median and 1-center on the plane and on networks. She also developed combinatorial algorithms that solve the corresponding problems in polynomial time. Then, Sepasian [23] studied the upgrading 1-center problem on weighted trees and proposed an $O(n^2)$ algorithm. Plastria [22] developed a polynomial-time algorithm to solve the up- and downgrading 1-median problem on the plane endowed by the Euclidean norm. Afrashteh et al. [1] investigated the upgrading selective obnoxious p-median problem on trees and solved the problem in polynomial time for both cases $p = 1$ and $p > 1$. The upgrading vertex cover problem was considered by Baldomero-Naranjo et al. [6]. They devised a polynomial time algorithm for the problem. The literature on the up/downgrading optimization problem primarily focuses on its deterministic version, where all data are precisely known. A natural open question is how to extend these problems to a more general, nondeterministic setting.
In practice, input data often contain uncertainties due to a lack of information or changes in the environment. Decision-makers aim to protect against the worst-case scenarios in such situations. Solutions are typically evaluated against the optimal solution that could have been achieved if the actual values of the uncertain parameters were known. To address these challenges, the robust optimization technique has been introduced in [7, 16]. Robust optimization seeks solutions for scenarios where the parameters are imprecise, uncertain, and generally incompletely known. A popular robust optimization model aims to find a solution that minimizes the largest deviation from the optimum to consider the worst-case across all scenarios in the uncertainty set (regret criterion). For minmax regret combinatorial optimization with models and algorithms, we can see [2, 4, 5, 19, 20] and references therein. Anh et al. [3] recently studied the problem of downgrading the makespan objective with interval cost coefficients and devised polynomial-time algorithms to solve it. The relevant models considered the problems under finitely many scenarios or interval data, and one technique used was the analysis of worst-case scenarios, which leads to maximum regret.

In cybersecurity and system resilience, where delaying certain tasks can mitigate risks, counteract adversarial actions, or allow critical preparations. For instance, in defensive scheduling against cyber-attacks, an attacker might aim to optimize task completion to minimize lateness, while a defender may increase processing times strategically to ensure a minimum delay, buying time for system defense mechanisms. Another relevant application arises in manufacturing and supply chain management, where adjusting job processing times can help manage workload distribution and enforce production constraints. In situations where delays beyond a threshold are necessary (e.g., controlled release of products, inventory balancing, or maintenance scheduling), companies may modify processing times at minimal cost to meet operational requirements. These applications can be modeled as a minimum cost problem for downgrading minimum lateness scheduling. Furthermore, the cost of modifying the processing time for a job depends on many factors and is not deterministic, but is estimated within an interval. This leads to uncertainty in costs. Hence, we apply the min-max regret criterion to obtain a reasonable solution.
This paper is organized as follows. The introduction section is dedicated to the literature review. Section 2 introduces the basic concepts of the minimum lateness problem and presents the deterministic downgrading minimum lateness problem. We propose a linear-time algorithm to solve this problem, where the due dates are already sorted. In Section 3, we address the robust downgrading minimum lateness problem, formulating it as n linear programming problems, where n is the number of jobs. Since each subproblem can be solved in linear time, this results in an $O(n^2)$ algorithm that solves the robust problem. Finally, we summarize the results of the paper and provide an outlook in Section 4.

2. The minimum cost problem of downgrading minimum lateness scheduling

2.1. Classical minimum lateness scheduling problem

Let us introduce the (single machine) minimum lateness scheduling problem (MLSP). Given n jobs $J_1, J_2, \ldots, J_n$ with corresponding processing times $p_1, p_2, \ldots, p_n$ and due dates $d_1, d_2, \ldots, d_n$. A schedule π is a permutation of $1, 2, \ldots, n$. Specifically, it is the sequence of jobs processed by the machine in such an order $J_{\pi(1)}, J_{\pi(2)}, \ldots, J_{\pi(n)}$. The completion time of job $J_{\pi(i)}$ is

$$C_{\pi(i)} = \sum_{j=1}^{i} p_{\pi(j)}$$

for $i = 1, \ldots, n$. If the completion time of job $J_{\pi(i)}$ exceeds the deadline, its lateness is measured by the gap between its completion time and the corresponding due date, say

$$L_{\pi(i)} = \max\{0, C_{\pi(i)} - d_{\pi(i)}\},$$

and the lateness of schedule π is the maximum lateness across all jobs,

$$L_\pi = \max_{i=1}^{n} L_{\pi(i)}.$$

The goal of the MLSP is to identify a schedule π that minimizes the maximum lateness of all jobs. Let us denote Π by the set of all schedules, the MLSP is stated as $\min_{\pi\in\Pi} L_\pi$. The optimal schedule of the MLSP is characterized as in the following result.

Theorem 1 (Jackson's rule [8]). *The optimal schedule π^* of the MLSP satisfies*

$$d_{\pi^*(1)} \leq d_{\pi^*(2)} \leq \ldots \leq d_{\pi^*(n)}.$$

Due to Jackson's rule, we process the jobs in the sequence of non-decreasing due dates and the optimal solution π^* does not depend on the processing times. From here onwards, we assume without loss of generality that

$$d_1 \leq d_2 \leq \ldots \leq d_n.$$

This means that the identity permutation, $\pi(i) = i \quad \forall i = 1, \ldots, n$, presents the optimal schedule.
Let π^* be an optimal schedule of the MLSP, the minimum lateness is $L_{\pi^*} = \max_{i=1}^{n} L_{\pi^*(i)}$.

Example 1. Let us consider the input of the MLSP as in Table 1, where the due dates are already sorted according to the indices.

i	1	2	3	4	5	6
p_i	4	2	4	3	2	1
d_i	6	8	10	11	12	14

Table 1. **An instance of the MLSP in Example 1**

By Jackson's rule, the identity schedule is the optimal one. We can compute the completion time and the lateness of each job as in Table 2.

i	1	2	3	4	5	6
C_i	4	6	10	13	15	16
L_i	0	0	0	2	3	2

Table 2. **Completion time and lateness of each job**

According to Table 2, the minimum lateness is 3.

2.2. Deterministic minimum cost problem of downgrading minimum lateness scheduling

Let an instance of the MLSP be given with due dates that are already sorted. Based on the motivation outlined in the previous section, the scheduling system should be degraded to a minimum lateness time Δ for maintenance or cybersecurity purposes. This means the lateness of all jobs should be downgraded to a value Δ, which is larger than $\min_{\pi\in\Pi} L_\pi$, by augmenting the processing times of jobs. Precisely, the processing time of each job J_i is increased by x_i, with a cost of w_i per unit of augmentation for $i = 1, \ldots, n$. The modified processing time of job J_i is denoted as $\tilde{p}_i = p_i + x_i$. Subsequently, the completion time with respect to the new processing times is given as $\tilde{C}_i = \sum_{j=1}^{i} \tilde{p}_j$, and the corresponding lateness is $\tilde{L}_i = \max\{0, \tilde{C}_i - d_i\}, \quad i = 1, \ldots, n$. Then, we state the minimum cost problem of downgrading minimum lateness scheduling (MCDMLS) as:

- The modified minimum lateness is at least Δ, i.e., $\max\limits_{i=1}^{n} \tilde{L}_i \geq \Delta$.
- The cost $\sum_{i=1}^{n} w_i x_i$ is minimized with $x_i \geq 0$ for $i = 1, \ldots, n$.

According to the previous statement and Jackson's rule, the minimum lateness can be attained by scheduling jobs in ordering $1, 2 \ldots, n$. Moreover, the inequality $\max\limits_{i=1}^{n} \tilde{L}_i \geq \Delta$ holds if there exists an index $i \in \{1, 2, \ldots, n\}$ such that $\tilde{L}_i \geq \Delta$. The MCDMLS can be formulated as

$$\begin{aligned} \text{(MC)} \quad \min \quad & \sum_{i=1}^{n} w_i x_i \\ \text{s.t.} \quad & x \in \bigcup_{i=1}^{n} \mathcal{X}_i. \end{aligned}$$

Here, x is a vector in $\mathbb{R}^n$ and the set $\mathcal{X}_i$ is identified as

$$\begin{aligned} \mathcal{X}_i = \Big\{ x = (x_j) \in \mathbb{R}^n : \; & \tilde{p}_j = p_j + x_j \quad \forall j = 1, \ldots, n, \\ & \tilde{C}_i = \sum_{j=1}^{i} \tilde{p}_j \quad \forall i = 1, \ldots, n, \\ & \tilde{L}_i \geq \tilde{C}_i - d_i, \;\; \tilde{L}_i \geq \Delta, \;\; \tilde{L}_i \geq 0, \\ & x_j \geq 0 \quad \forall i = 1, \ldots, n. \Big\} \end{aligned}$$

for $i = 1, \ldots, n$.
A special property concerning the optimal solution of (MC) is presented below.

Proposition 1. *An optimal solution to (MC) exists where the processing time of exactly one job is augmented.*

Proof. Let $k = \arg\max_{i=1}^{n} \tilde{L}_i$ be the index corresponding to the maximum lateness of all jobs. We know that the minimum lateness equals the threshold, i.e., $\tilde{L}_k = \Delta$. As

$$\tilde{L}_k = \tilde{C}_k - d_k = C_k - d_k + \sum_{j=1}^{k} x_j,$$

the optimal cost equals the optimal value of the problem

$$\min\left\{\sum_{j=1}^{k} w_j x_j : C_k - d_k + \sum_{j=1}^{k} x_j = \Delta, x_j \geq 0 \quad \forall j = 1, \ldots, k\right\}.$$

By setting $\Delta^k = \Delta + d_k - C_k$, we simplify the problem as

$$\min\left\{\sum_{j=1}^{k} w_j x_j : \sum_{j=1}^{k} x_j = \Delta^k, x_j \geq 0 \quad \forall j = 1, \ldots, k\right\}.$$

Let $i_0 = \arg\min\{w_j : j = 1, \ldots, k\}$, then the optimal solution is $w_{i_0}\Delta^k$. Hence, the optimal solution of (MC) is x^* with $x^*_{i_0} = \Delta^k$ and $x^*_j = 0$ for $j \neq i_0$. This proves the proposition. □

By Proposition 1, we focus on the modification $x_j := \Delta^j$ for $j = 1, \ldots, n$ for each candidate optimal solution. One can compute all candidate optimal objective $\min_{i=1,\ldots,j} w_i \Delta^j$ for $j = 1, \ldots, n$ and take the smallest one that is the optimal objective. The detailed computation is presented in Algorithm 1.

Algorithm 1 Solves the MCDMLS

Input: Jobs J_i corresponding to the due dates d_i with $d_1 \leq d_2 \leq \ldots \leq d_n$ and processing times p_i for $i = 1, \ldots, n$. Downgrading level Δ and costs $w_1, w_2, \ldots, w_n$.
Compute $C_j = \sum_{l=1}^{j} p_l$ and $\Delta^j := \Delta + d_j - C_j$ for $j = 1, \ldots, n$.
Set $\lambda := w_1$, $Val := \lambda\Delta^1$.
for i=2,...,n **do**
  Set $\lambda := \min\{\lambda, w_i\}$ and $Val := \min\{Val, \lambda\Delta^i\}$.
end for
Output: The optimal objective value $Val = w_{j_0}\Delta^{i_0}$ and the corresponding optimal solution x^* with $x^*_{j_0} = \Delta^{i_0}$ and $x^*_j = 0$ for $j \neq j_0$.

Theorem 2. *Algorithm 1 solves the MCDMLS in linear time, assuming that the due dates are sorted in advance.*

Proof. For any index i in $\{1, \ldots, n\}$, we find $\min_{j=1}^{i}\{w_j\Delta^i\} = \lambda\Delta^i$. The smallest value among them is the optimal value of the MCDMLS. Hence, the algorithm is correct. Next, let us examine the complexity of the algorithm. We can compute C_j and Δ^j for $j = 1, \ldots, n$ in linear time by reduction. Then, we update λ and Val in constant time in each iteration. Therefore, Algorithm 1 runs in linear time. □

We illustrate Algorithm 1 by an example as follows.

Example 2. We consider an instance of the MCDMLS with processing times and due dates given in Example 1. Moreover, augmentation costs are in Table 3.

i	1	2	3	4	5	6
w_i	4	2	3	3	4	4

Table 3. **Augmentation costs of jobs in Example 2**

Assume that downgrading level Δ equals 15. Then, we compute Δ^i for $i = 1, \dots, 6$ as Table 4.

i	1	2	3	4	5	6
Δ^i	17	17	15	13	12	13

Table 4. **Computation of Δ^i for $i = 1, \dots, 6$**

Applying Algorithm 1, we can solve the corresponding MCDMLS as in the following iterations.

Iter.	1	2	3	4	5	6
λ	4	2	2	2	2	2
Val	68	34	30	26	24	24

Table 5. **Computation of λ and Val in each iteration**

Hence, the optimal objective is 24 with optimal solution $(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*) = (0, 12, 0, 0, 0, 0)$.

3. The robust minimum cost problem of downgrading minimum lateness scheduling

In real-life situations, the costs relative to augmenting processing times are not exactly known with no distribution function. Then, the concept of robust optimization is employed to yield reasonable solutions. The so-called minmax regret criterion is to minimize the maximum loss with respect to the optimal solution across all scenarios. In this paper, we consider the robust (minmax regret) MCDMLS (RobMCDMLS) with interval costs, i.e., $w_i \in [\underline{w}_i, \overline{w}_i]$. Any assignment of costs to a specified vector in $\prod_{i=1}^{n} [\underline{w}_i, \overline{w}_i]$ is called a scenario. This means a scenario $s \in S$ corresponds to $w_i^s \in [\underline{w}_i, \overline{w}_i]$ for $i = 1, \dots, n$. Let S be the set of all scenarios and let the set of all feasible solutions be $\mathcal{F} = \bigcup_{i=1}^n \mathcal{F}_i$, where

$$\mathcal{F}_i = \left\{ x \in \mathbb{R}^n : C_i - d_i + \sum_{j=1}^{i} x_j \geq \Delta, x_j \geq 0 \ \forall j = 1, \ldots, n \right\}.$$

For a solution $x \in \mathcal{F}$ and a scenario $s \in S$, we associate the regret function

$$R^s(x) = \sum_{i=1}^{n} w_i^s x_i - Val^s.$$

Here, $\sum_{i=1}^n w_i^s x_i$ is the objective value at x and Val^s is the optimal value with respect to scenario s in S. Let us assume that $Val^s = w_{i_0}^s \Delta^k$ with $i_0 \leq k$ and $i_0, k \in \{1, \ldots, n\}$. Then, we can write

$$\begin{aligned} R^s(x) &= \textstyle\sum_{i=1}^n w_i^s x_i - w_{i_0}^s \Delta^k \\ &= \textstyle\sum_{i \neq i_0} w_i^s x_i + w_{i_0}^s (x_{i_0} - \Delta^k). \end{aligned}$$

We analyze the two cases:

Case 1: If $x_{i_0} \geq \Delta^k$, we set $w_i^s = \overline{w}_i$ for $i = 1, \ldots, n$ to attain

$$\max_{s \in S} R^s(x) = \sum_{i=1}^{n} \overline{w}_i x_i - \overline{w}_{i_0} \Delta^k.$$

Case 2: If $x_{i_0} < \Delta^k$, we set $w_i^s = \overline{w}_i$ for $i = 1, \ldots, n, i \neq i_0$ and $w_{i_0}^s = \underline{w}_{i_0}$ to attain

$$\begin{aligned} \max_{s \in S} R^s(x) &= \textstyle\sum_{i \neq i_0} \overline{w}_i x_i - \underline{w}_{i_0} \Delta^k + \underline{w}_{i_0} x_{i_0} \\ &= \textstyle\sum_{i=1}^n \overline{w}_i x_i - \left((\overline{w}_{i_0} - \underline{w}_{i_0}) x_{i_0} + \underline{w}_{i_0} \Delta^k \right). \end{aligned}$$

For any index $i \in \{1, \ldots, n\}$, we denote by $k_i := \arg\min_{j=i,\ldots,n} \Delta^j$. We can find k_i for $i = 1, \ldots, n$ in linear time. Then, the maximum regret $R(x) = \max\limits_{s \in S} R^s(x)$ can be expressed as in the formulation

$$\begin{aligned} R(x) &= \max_{i=1}^{n} \left\{ \textstyle\sum_{j=1}^n \overline{w}_j x_j - \overline{w}_i \Delta^{k_i}, \sum_{j=1}^n \overline{w}_j x_j - \left(\delta_i x_i + \underline{w}_i \Delta^{k_i} \right) \right\} \\ &= \textstyle\sum_{j=1}^n \overline{w}_j x_j - \min\limits_{i=1}^{n} \left\{ \overline{w}_i \Delta^{k_i}, \delta_i x_i + \underline{w}_i \Delta^{k_i} \right\}, \end{aligned}$$

where $\delta_i = \overline{w}_i - \underline{w}_i$ for $i = 1, \ldots, n$. Hence, the RobMCDMLS is state as $\min\limits_{x \in \mathcal{F}} R(x)$ and it is formulated as the program

$$\begin{aligned} \text{(RobMC)} \quad \min \quad & \sum_{j=1}^{n} \overline{w}_j x_j - \xi \\ \text{s.t.} \quad & \xi \leq \overline{w}_j \Delta^{k_j} \quad \forall j = 1, \ldots, n, \\ & \xi \leq \delta_j x_j + \underline{w}_j \Delta^{k_j} \quad \forall j = 1, \ldots, n, \\ & x \in \mathcal{F}. \end{aligned}$$

As $\xi \leq \overline{w}_j \Delta^{k_j}$ for all $j = 1, \ldots, n$, we can set $\overline{\xi} = \min_{j=1}^{n} \overline{w}_j \Delta^{k_j}$ and attain $\xi \leq \overline{\xi}$. Let $\alpha_j = \underline{w}_j \Delta^{k_j}$, we can reformulate (RobMC) as

$$
\begin{aligned}
\text{(RobMC')} \quad \min \quad & \sum_{j=1}^{n} \overline{w}_j x_j - \xi \\
s.t. \quad & x_j \geq \frac{\xi - \alpha_j}{\delta_j} \quad \forall j = 1, \ldots, n, \\
& x \in \mathcal{F}, \\
& 0 \leq \xi \leq \overline{\xi}.
\end{aligned}
$$

Let us fix a value $\xi \in \left[0, \overline{\xi}\right]$ and denote by

$$
I^{<}(\xi) = \{j \in \{1, \ldots, n\} : \alpha_j < \xi\}
$$

and

$$
I_i^{<}(\xi) = \{j \in \{1, \ldots, i\} : \alpha_j < \xi\}.
$$

Then, we first set $x_j^{\xi} := \dfrac{\xi - \alpha_j}{\delta_j}$ for $j \in I^{<}(\xi)$ and consider the objective

$$
\sum_{j \in I^{<}(\xi)} \overline{w}_j x_j^{\xi} + \sum_{j=1}^{n} \overline{w}_j y_j - \xi,
$$

where $y = (y_i) \in \mathcal{F}^{\xi}$ and $\mathcal{F}^{\xi} = \bigcup_{i=1}^{n} \mathcal{F}_i^{\xi}$ with

$$
\mathcal{F}_i^{\xi} = \left\{ y : C_i - d_i + \sum_{j \in I_i^{<}(\xi)} x_j^{\xi} + \sum_{j=1}^{i} y_j \geq \Delta, y_j \geq 0 \quad \forall j = 1, \ldots, n \right\}.
$$

We specify an index i and get the subproblem of (RobMC') as follows:

$$
\begin{aligned}
(P_i) \quad \min \quad & \sum_{j=1}^{n} \overline{w}_j x_j - \xi \\
s.t. \quad & x_j \geq \frac{\xi - \alpha_j}{\delta_j} \quad \forall j = 1, \ldots, n, \\
& C_i - d_i + \sum_{j=1}^{i} x_j \geq \Delta, \\
& x_j \geq 0 \quad \forall j = 1, \ldots, n, \\
& 0 \leq \xi \leq \overline{\xi}.
\end{aligned}
$$

Let us denote

$$\Delta^i(\xi) = \Delta - C_i + d_i - \sum_{j\in I_i^<(\xi)} x_j^\xi = \Delta^i - \sum_{j\in I_i^<(\xi)} x_j^\xi.$$

We also find $\overline{\xi}_i$ such that $\Delta^i(\overline{\xi}_i) = 0$. Then, as $\Delta^i(\overline{\xi}) \geq 0$, we get $\xi \leq \overline{\xi}_i$ for all $i = 1, \dots, n$. This updates the value $\overline{\xi}$ by $\overline{\xi} = \min_{i=1,\dots,n}\{\overline{\xi}, \overline{\xi}_i\}$. We solve

$$\min\left\{\sum_{j=1}^{i} \overline{w}_j y_j : y \in \mathcal{F}_i^\xi\right\},$$

where

$$\mathcal{F}_i^\xi = \left\{y : \sum_{j=1}^{i} y_j \geq \Delta^i(\xi), y_j \geq 0 \quad \forall j = 1, \dots, n\right\}.$$

and get the optimal objective $\overline{w}_{i_\xi}\Delta^i(\xi)$. Then, the corresponding objective value of (P_i) can be presented as below.

$$\mathrm{Cost}_i(\xi) = \sum_{j\in I^<(\xi)} \overline{w}_j x_j^\xi + \overline{w}_{i_\xi}\Delta^i(\xi) - \xi.$$

Let us sort the elements in the set $\{0, \alpha_1, \alpha_2, \dots, \alpha_n, \overline{\xi}\} \cap [0, \overline{\xi}]$ non-decreasingly in order to get the set of breakpoints

$$\mathcal{B} = \{z_1, \dots, z_m\}$$

with $z_1 < z_2 < \dots < z_m$ and $m \leq n + 1$.
For $\xi \in [z_j, z_{j+1}]$, we characterize the function $\mathrm{Cost}_i(\xi)$. We can write

$$\mathrm{Cost}_i(\xi) = \sum_{j\in I^<(z_{j+1})} \overline{w}_j \frac{\xi - \alpha_j}{\delta_j} + \min_{q=1}^{i} \overline{w}_q \Delta^i(\xi) - \xi.$$

The function $\sum_{j\in I^<(z_{j+1})} \overline{w}_j \dfrac{\xi - \alpha_j}{\delta_j}$ is linear as the set $I^<(z_{j+1})$ is fixed. Similarly, we also get the linearity of $\Delta^i(\xi) = \Delta^i - \sum_{j\in I_i^<(z_{j+1})} \dfrac{\xi - \alpha_j}{\delta_j}$. Hence, the function $\mathrm{Cost}_i(\xi)$ is linear for ξ in $\xi \in [z_j, z_{j+1}]$.
We attain the following result concerning the optimal value of (RobMC').

Proposition 2. *The optimal objective of the (RobMC') is attained at z_j for $j \in \{1, \dots, m\}$.*

Proof. The problem (RobMC') is equal to $\min_{\xi \in [0,\bar{\xi}]} \text{Cost}_i(\xi)$. As the function $\text{Cost}_i(\xi)$ is linear for $\xi \in [z_j, z_{j+1}]$, the local minimizer of $\min_{\xi \in [z_j, z_{j+1}]} \text{Cost}_i(\xi)$ is attain at either z_j or z_{j+1}. Hence, the result of the proposition is proved. □

Due to Proposition 2, we can compute the objective value at each breakpoint $z_1, \dots,$ z_m. Then, we take the smallest value among them, that is also the optimal objective. For details, we refer to Algorithm 2.

Algorithm 2 Solves the Robust MCDMLS

Input: An instance of the Robust DMLP with sequence of due dates $d_1 \le d_2 \le \dots \le d_n$.
Compute α_j, δ_j, $\bar{\xi}$ for $j = 1, \dots, n$.
Find the set $\{z_1, z_2, \dots, z_m\}$.
Compute $A_i = \sum_{j \in I^<(z_i)} \overline{w}_j x_j^{z_i}$ and $\Delta^k(z_i)$ for $i = 1, \dots, m$ and $k = 1, \dots, n$.

Find the sets $I^<(z_i), I_j^<(z_i)$ for j=1,..., n.
for i=2,...,m **do**
  Compute $Cost(z_i) = A_i + \min_{j=1}^{n} \overline{w}_j \Delta^{k_j}(z_i) - z_i$.
end for
Output: The optimal value $Val = \arg\min_{i=1}^{m} Cost(z_i)$ and the corresponding optimal solution

$$x_j^* := \begin{cases} \dfrac{Val - \alpha_j}{\delta_j}, & \text{if } \alpha_j < Val, \\ 0, & \text{if } \alpha_j \ge Val, \end{cases}$$

and $x_{j_0}^* := x_{j_0}^* + \Delta^{k_{j_0}}(Val)$ with $j_0 = \arg\min_{j=1}^{n} \overline{w}_j \Delta^{k_j}(Val)$.

Theorem 3. *The Robust MCDMLS can be solved in $O(n^2)$ time.*

Proof. The correctness of the algorithm is due to comparing all candidate values and taking the smallest one. We computes all $I^<(z_i)$, $I_j^<(z_i)$ in $O(n^2)$ time. Then, we can update A_i and $\Delta^k(z_i)$ for $k = 1, \dots, n$ in $O(n^2)$ time. In the for loop of the algorithm, we compute $\min_{j=1}^{n} \overline{w}_j \Delta^{k_j}(z_i)$ in linear time in each iteration. Finally, the corresponding optimal solution can be computed in linear time. Therefore, the total complexity of Algorithm 2 is $O(n^2)$. □

The following example illustrates Algorithm 2.

Example 3. Given processing times and due dates of jobs as in Example 1. We consider the uncertain costs as in Table 6.

i	1	2	3	4	5	6
$\underline{w}_i$	3	2	2	3	4	4
$\overline{w}_i$	4	4	6	4	5	6

Table 6. **Interval costs concerning augmentations of processing times**

We next compute δ_i and α_i as in Table 7.

i	1	2	3	4	5	6
δ_i	1	2	4	1	1	2
α_i	36	24	24	36	48	52

Table 7. **Computation of δ_i and α_i**

Moreover, $\overline{\xi} = 408/11$ then $\mathcal{B} = \{0, 24, 36, \frac{408}{11}\}$. Note that $\Delta^5(\frac{408}{11}) = 0$. We also compute

$$I^<(z_1) = I_i^<(z_1) = \emptyset,\ I^<(z_2) = \emptyset,\ I^<(z_3) = \{2,3\},\ I^<(z_4) = \{1,2,3,4\}, \ldots$$

In detail, the sets $I_j(z_i)$ for $i = 1, \ldots, 4$ and $j = 1, \ldots, 6$ are given in Table 8

i	j					
	1	2	3	4	5	6
1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
2	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
3	$\emptyset$	$\{2\}$	$\{2,3\}$	$\{2,3\}$	$\{2,3\}$	$\{2,3\}$
4	$\{1\}$	$\{1,2\}$	$\{1,2,3\}$	$\{1,2,3,4\}$	$\{1,2,3,4\}$	$\{1,2,3,4\}$

Table 8. **The sets $I_j^<(z_i)$ for $i = 1, \ldots, 4$ and $j = 1, \ldots, 6$**

i	A_i	$\Delta^1(z_i)$	$\Delta^2(z_i)$	$\Delta^3(z_i)$	$\Delta^4(z_i)$	$\Delta^5(z_i)$	$\Delta^6(z_i)$
1	0	17	17	15	13	12	13
2	0	17	17	15	13	12	13
3	42	17	11	6	4	3	4
4	600/11	175/11	103/11	45/11	1	0	1

Table 9. **Computation in each iteration**

We can compute Cost(z_1) = 48, Cost(z_2)= 24, Cost(z_3) = 18, Cost(z_4) = 17.45. It shows that $\xi = 408/11$ is the optimal solution with optimal objective 17.45. This yields an optimal solution

$$(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*) = (12/11, 72/11, 36/11, 12/11, 0, 0).$$

4. Conclusions

This paper addresses the problem of augmenting the processing time of jobs such that the lateness of an optimal schedule reaches a level Δ. Moreover, the augmenting cost with respect to each job can take any value in an interval. We call the problem the robust minimum cost problem of downgrading minimum lateness scheduling. We first develop a linear time algorithm for the deterministic problem by leveraging the characteristic of an optimal solution, provided that due dates are already sorted. For the robust problem, we decompose it into sub-problems and solve all of them in $O(n^2)$ time. In this paper, we do not impose specific bounds on modifying variables. Adding such a constraint can lead to NP-hardness. Future research on the problem of up-/downgrading scheduling with other objectives, such as makespan, lateness, tardiness,... is a promising topic.

Acknowledgements. We sincerely thank anonymous reviewers for insightful comments and constructive feedback, which helped to improve the paper. This research is funded by Vietnam National University HoChiMinh City (VNU-HCM) under grant number C2025-26-02.

Conflict of Interest: The authors declare that they have no conflict of interest.

Data Availability: Data sharing is not applicable to this article as no data sets were generated or analyzed during the current study.

References

[1] E. Afrashteh, B. Alizadeh, and F. Baroughi, *Optimal approaches for upgrading selective obnoxious p-median location problems on tree networks*, Ann. Oper. Res. **289** (2020), no. 2, 153–172.
https://doi.org/10.1007/s10479-020-03561-4.

[2] H. Aissi, C. Bazgan, and D. Vanderpooten, *Min–max and min–max regret versions of combinatorial optimization problems: A survey*, European J. Oper. Res. **197** (2009), no. 2, 427–438.
https://doi.org/10.1016/j.ejor.2008.09.012.

[3] L.Q. Anh, H.M. Le, K.T. Nguyen, and L.X. Thanh, *An algorithmic approach to the robust downgrading makespan scheduling problem*, Appl. Set-Valued Anal. Optim **6** (2024), no. 3, 263–273.
https://doi.org/10.23952/asvao.6.2024.3.02.

[4] I. Averbakh, *Minmax regret linear resource allocation problems*, Oper. Res. Lett. **32** (2004), no. 2, 174–180.
https://doi.org/10.1016/S0167-6377(03)00091-9.

[5] O. Bachtler, S.O. Krumke, and H. Minh Le, *Robust single machine makespan scheduling with release date uncertainty*, Oper. Res. Lett. **48** (2020), no. 6, 816–

819.
https://doi.org/10.1016/j.orl.2020.10.003.

[6] M. Baldomero-Naranjo, J. Kalcsics, A. Marín, and A.M. Rodríguez-Chía, *Upgrading edges in the maximal covering location problem*, European J. Oper. Res. **303** (2022), no. 1, 14–36.
https://doi.org/10.1016/j.ejor.2022.02.001.

[7] A. Ben-Tal, L.E. Ghaoui, and A. Nemirovski, *Robust Optimization*, Princeton University Press, Princeton, New Jersey, 2009.

[8] P. Brucker, *Scheduling Algorithms (fifth edition)*, Springer Verlag, Heidelberg, 2007.

[9] R.E. Burkard, B. Klinz, and J. Zhang, *Bottleneck capacity expansion problems with general budget constraints*, RAIRO Oper. Res. **35** (2001), no. 1, 1–20.
https://doi.org/10.1051/ro:2001100.

[10] R.E. Burkard, Y. Lin, and J. Zhang, *Weight reduction problems with certain bottleneck objectives*, European J. Oper. Res. **153** (2004), no. 1, 191–199.
https://doi.org/10.1016/S0377-2217(02)00713-0.

[11] K.U. Drangmeister, S.O. Krurnke, M.V. Marathe, H. Noltemeier, and S.S. Ravi, *Modifying edges of a network to obtain short subgraphs*, Theor. Comput. Sci. **203** (1998), no. 1, 91–121.
https://doi.org/10.1016/S0304-3975(97)00290-9.

[12] G.N. Frederickson and R. Solis-Oba, *Increasing the weight of minimum spanning trees*, J. Algorithms. **33** (1999), no. 2, 244–266.
https://doi.org/10.1006/jagm.1999.1026.

[13] D.R. Fulkerson and G.C. Harding, *Maximizing the minimum source-sink path subject to a budget constraint*, Math. Program. **13** (1977), no. 1, 116–118.
https://doi.org/10.1007/BF01584329.

[14] E. Gassner, *A game-theoretic approach for downgrading the 1-median in the plane with Manhattan metric*, Ann. Oper. Res. **172** (2009), no. 1, 393–404.
https://doi.org/10.1007/s10479-009-0641-1.

[15] ______, *Up-and downgrading the 1-center in a network*, European J. Oper. Res. **198** (2009), no. 2, 370–377.
https://doi.org/10.1016/j.ejor.2008.09.013.

[16] P. Kouvelis and G. Yu, *Robust Discrete Optimization and Its Applications*, Springer Science & Business Media, 2013.

[17] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys, *Sequencing and scheduling: Algorithms and complexity*, Logistics of Production and Inventory, Handbooks in Operations Research and Management Science, vol. 4, Elsevier, 1993, pp. 445–522.
https://doi.org/10.1016/S0927-0507(05)80189-6.

[18] J.K. Lenstra, A.H.G. Rinnooy Kan, and P. Brucker, *Complexity of machine scheduling problems*, Ann. Discrete Math., vol. 1, Elsevier, 1977, pp. 343–362.
https://doi.org/10.1016/S0167-5060(08)70743-X.

[19] K.T. Nguyen and N.T. Hung, *The minmax regret inverse maximum weight problem*, Appl. Math. Comput. **407** (2021), Article ID: 126328.

https://doi.org/10.1016/j.amc.2021.126328.

[20] T.H.N. Nhan, K.T. Nguyen, and H. Nguyen-Thu, *The minmax regret reverse 1-median problem on trees with uncertain vertex weights*, Asia-Pac. J. Oper. Res. **40** (2023), no. 3, Article ID: 2250033. https://doi.org/10.1142/S0217595922500336.

[21] M.L. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, Springer New York, 2012.

[22] F. Plastria, *Up-and downgrading the euclidean 1-median problem and knapsack Voronoi diagrams*, Ann. Oper. Res. **246** (2016), no. 1, 227–251. https://doi.org/10.1007/s10479-014-1587-5.

[23] A.R. Sepasian, *Upgrading the 1-center problem with edge length variables on a tree*, Discrete Optim. **29** (2018), 1–17. https://doi.org/10.1016/j.disopt.2018.02.002.

[24] A.R. Sepasian and E. Monabbati, *Upgrading min–max spanning tree problem under various cost functions*, Theor. Comput. Sci. **704** (2017), 87–91. https://doi.org/10.1016/j.tcs.2017.08.006.

[25] H. Xiong, S. Shi, D. Ren, and J. Hu, *A survey of job shop scheduling problem: The types and models*, Comput. Oper. Res. **142** (2022), Article ID: 105731. https://doi.org/10.1016/j.cor.2022.105731.