Journal of Industrial and Management Optimization

JIMO, 22 (4): 1905–1927.
DOI: 10.3934/jimo.2026070
Received: 08 December 2025
Revised: 03 March 2026
Accepted: 05 March 2026
Published: 19 March 2026

https://www.aimspress.com/journal/jimo

Research article

Improved algorithms for the two-machine flow shop scheduling problem with uncertain setup times to minimize total tardiness

Muberra Allahverdi[1,*] and Ali Allahverdi[2]

[1] Department of Mathematics, Faculty of Engineering and Natural Sciences, Ankara Yıldırım Beyazıt University, Ankara, Turkey
[2] Department of Industrial Engineering, Faculty of Engineering, Gazi University, Ankara, Turkey

* **Correspondence:** Email: mu.allahverdi@proton.me; Tel: +905523875520.

Abstract: We address the two-machine flow shop scheduling problem with the objective of minimizing total tardiness, where setup times are modeled as uncertain. Accordingly, setup times are represented using lower and upper bounds without assuming any probability distribution. We first develop a mathematical dominance relation for the problem and then present two novel heuristic algorithms that incorporate this dominance relation. Through extensive computational experiments using multiple problem sizes, due-date settings, and both symmetric and extreme uncertainty distributions, supported by statistical hypothesis testing, we show that both proposed algorithms significantly outperform the best existing algorithm in the literature. In particular, the computational results indicate that the error of the leading algorithm in the literature is approximately 38 to 115 times larger than that of our proposed algorithms.

Keywords: scheduling; flow shop; total tardiness; heuristic algorithms; uncertain setup times
Mathematics Subject Classification: 90B35, 90C27

1. Introduction

The flowshop production setting has numerous practical applications in industries such as electronics, semiconductor manufacturing, satellite production, chemical processing, and textile manufacturing (Fuchigami and Rangel [1]). A significant proportion of real-world production systems involve setup operations, and these setup times often play a critical role in overall system performance.

The objective considered in this study is the minimization of total tardiness. In many production environments, tardiness directly contributes to scheduling costs, contractual penalties, and customer dissatisfaction. Therefore, effective scheduling decisions that control total tardiness are of substantial practical importance.

Recent research has increasingly recognized that deterministic scheduling models may be insufficient when production parameters are subject to variability. In particular, uncertainty in setup times can significantly influence scheduling decisions and performance measures. Rahmativala and Ghahremani [2] develop a robust optimization framework for a flowshop scheduling problem with uncertain setup times and demonstrate that explicitly accounting for setup time variability affects both makespan and tardiness outcomes. At the same time, due-date related criteria such as total tardiness continue to play a central role in flowshop scheduling research because they directly reflect delivery performance and customer service considerations. A recent systematic review by de Athayde Prata et al. [3] confirms the sustained and growing interest in permutation flowshop problems with tardiness-based objectives. These findings further motivate the investigation of flow shop scheduling models that jointly consider uncertainty in setup durations and total tardiness minimization.

In this paper, we consider a two-machine flowshop scheduling problem in which setup times are uncertain. In many manufacturing environments, setup durations are inherently difficult to estimate precisely due to variability in operator skill levels, machine conditions, tooling wear, and the nature of product changeovers. In such cases, production planners are typically unable to specify exact setup times in advance and instead rely on reasonable lower and upper bounds derived from historical data or engineering judgment.

Assuming deterministic setup times in such settings may lead to schedules that are overly optimistic and insufficiently robust. On the other hand, fully stochastic models may be impractical when reliable probability distributions are unavailable. Therefore, we adopt an interval-based representation in which setup times are modeled using lower and upper bounds without assuming any probability distribution. This modeling framework reflects the type of information commonly available at the planning stage and provides a practical way to incorporate uncertainty while maintaining computational tractability.

Specifically, we address the two-machine flowshop scheduling problem to minimize total tardiness, where setup times are uncertain within given bounds and processing times are deterministic. We first establish a mathematical dominance relation for the problem. We then propose two novel heuristic algorithms that incorporate the developed dominance relation. Extensive computational experiments are conducted to compare the proposed algorithms with the best-performing existing algorithm in the literature.

The remainder of the paper is organized as follows. Section 2 reviews the related research. Section 3 presents the developed mathematical dominance relation. Section 4 provides a numerical example highlighting the impact of uncertainty on the scheduling decisions. Section 5 describes the proposed heuristic algorithms, PA1 and PA2. Section 6 reports the computational experiments. Section 7 presents the results of the statistical hypothesis testing. Finally, Section 8 concludes the paper and outlines directions for future research.

2. Related research

The two-machine flowshop scheduling problem has been extensively studied under various performance measures and modeling assumptions, and when uncertainty is present, different modeling frameworks have been proposed in the literature. Uncertainty in scheduling problems is typically modeled using a limited number of fundamental approaches. One common framework is stochastic programming, where uncertain parameters are represented by random variables with known probability distributions and decisions are evaluated in expectation or via risk measures (e.g., Birge and Louveaux [4]). An alternative framework is robust optimization, which does not require precise distributional information but instead assumes that uncertain parameters belong to a predefined uncertainty set and seeks solutions that perform well under worst-case realizations (e.g., Ben-Tal et al. [5]; Bertsimas et al. [6]). A commonly used and practically appealing specification of uncertainty sets is the interval (bounded) model, where parameters are assumed to lie between known lower and upper bounds. This interval-based representation is particularly suitable in production environments where reliable distributional information is unavailable but historical data or engineering judgment provide meaningful bounds. In scheduling, interval uncertainty has been adopted in several studies on processing and setup times, forming the basis for dominance relations and worst-case performance analysis.

The flow shop scheduling problem has been studied extensively from both theoretical and algorithmic perspectives. Comprehensive treatments of flow shop models, structural properties, and solution approaches can be found in Emmons and Vairaktarakis [7], which provide an in-depth overview of dominance relations and algorithmic frameworks for various flow shop environments. In addition, sequence-dependent and sequence-independent setup times have received considerable attention in the flow shop literature; see, for example, Gharbi et al. [8] for lower bounding strategies in two-machine flow shops with setup times. These studies form part of the broader theoretical foundation upon which the present work builds.

Beyond classical deterministic models, increasing attention has been devoted to time-dependent, position-dependent, and resource-dependent scheduling problems. Recent monographs and surveys (e.g., Gawiejnowicz [9]; Agnetis et al. [10]; Gawiejnowicz [11]) document the rapid development of models in which processing times vary over time or depend on system states. Furthermore, uncertainty-aware scheduling has become an important research direction, as evidenced by recent review papers such as Ksciuk et al. [12]. These contributions highlight the growing interest in integrating uncertainty and variability into scheduling theory, within which interval-based models for setup times constitute a natural and practically relevant subclass. Against this broader theoretical background, we next review contributions specifically related to setup times in two-machine flow shop environments.

Allahverdi [13] addressed the two-machine flowshop scheduling problem with the objective of minimizing total tardiness while disregarding setup times. Although omitting setup times may be appropriate in some production environments, it is not realistic in many practical settings (Pessan et al. [14]; Allahverdi et al. [15]). Treating setup times separately from processing times has been shown to improve modeling accuracy and scheduling performance (Allahverdi [16]; Aydilek et al. [17]; Allahverdi and Soroush [18]).

Most studies that model setup times separately assume deterministic setup durations (e.g., Feng and Kong [19]; Aqil [20]; Zhao [21]; Cheng et al. [22]; Jansen et al. [23]). However, setup times may be subject to considerable uncertainty in practice (Ertem et al. [24]; Joo and Xirouchakis [25]; Bruni et al. [26]). Kim and Bobrowski [27] emphasized that factors such as crew skills and unexpected tool breakdowns may cause setup durations to vary significantly. While early work on uncertain setup times was presented by Kim and Bobrowski [27], more recent studies continue to explore scheduling models that explicitly account for uncertainty in setup times. For example, Ertem et al. [24] study a single-machine scheduling problem with uncertain setup times. In addition, several robust optimization approaches have been proposed that explicitly incorporate uncertainty in setup and processing times (e.g., Yanıkoğlu & Yavuz [28]). More recently, Stanković et al. [29] investigate parallel machine scheduling environments where both processing and setup times are uncertain and develop metaheuristic solution methods for such problems. In the context of flow shop scheduling, Allahverdi and Allahverdi [30] address a two-machine no-wait flow shop problem with uncertain setup times and develop dominance relations along with algorithmic solution procedures. Similarly, Aydilek et al. [17] consider a two-machine flow shop problem with uncertain setup times and propose constructive algorithms with strong computational performance. Earlier work by Allahverdi and Allahverdi [31] investigated a two-machine no-wait flow shop with uncertain setup times by establishing dominance relations and algorithmic procedures.

Aydilek et al. [32] addressed the problem $F2/\ Ls_{p,k} \leqslant s_{p,k} \leqslant Us_{p,k}\ /\ C_{max}$ where they provided an algorithm with nine versions. They indicated by computational analysis that one version performed significantly better than the rest. Aydilek et al. [33] provided twenty-five implementations of an algorithm for the problem of $F2/\ Ls_{p,k} \leqslant s_{p,k} \leqslant Us_{p,k},\ Lt_{p,k} \leqslant t_{p,k} \leqslant Ut_{p,k}\ /\ C_{max}$ where it was shown that one of the implementations significantly outperforms the rest.

More recently, Allahverdi [13] addressed the problem of $F2/Lt_{p,k} \leqslant t_{p,k} \leqslant Ut_{p,k}\ /TT$ where TT denotes the total tardiness performance measure with deterministic setup times. She established a dominance relation and presented nineteen algorithms to address the problem. One of the algorithms was shown to perform considerably better than the rest.

We address the two-machine flowshop scheduling problem where setup times are uncertain within some lower and upper bounds, whereas processing times are known in advance, i.e., we address the problem $F2/Ls_{p,k} \leqslant s_{p,k} \leqslant Us_{p,k}\ /TT$. For this problem, Allahverdi and Allahverdi [34] presented three algorithms for solving the problem and showed that one algorithm outperforms the others. For the same problem, in this paper, we first provide a mathematical dominance relation. Next, we propose two new algorithms incorporating the mathematical dominance relation. We conduct extensive computational experiments to compare the proposed algorithms with the best-known existing algorithm in the literature (Allahverdi and Allahverdi [34]).

3. A dominance relation

In this section, we derive a mathematical dominance relation for the two-machine flowshop scheduling problem with uncertain setup times. This dominance relation is then incorporated into the algorithms presented in the following section.

We consider a two-machine flowshop scheduling problem in which a finite set of n independent jobs, denoted by $J = \{1, 2, \ldots, n\}$, must be processed on two machines M_1 and M_2, in this order. Each job requires a processing time on each machine, and sequence-independent setup times are incurred before processing a job on each machine. A permutation schedule is assumed; that is, the same job sequence is processed on both machines.

Associated with each job, there is a deterministic due date d_j. The objective is to determine a single common job sequence that minimizes the total tardiness of all jobs. While processing times and due dates are assumed to be known and deterministic, the setup times on both machines are uncertain. Specifically, for each job and machine, only lower and upper bounds on the setup times are available at the decision stage. No probabilistic information or exact realizations of setup times are assumed. Therefore, scheduling decisions, including dominance relations and heuristic algorithms, must be derived solely based on this interval information, ensuring validity for all feasible realizations of setup times within their bounds.

Let $t_{p,k}$ and $s_{p,k}$ represent the processing time and setup time of job p on machine k (k=1, 2), respectively. Also let $t_{[p,k]}$ and $s_{[p,k]}$ and be processing time and setup time of the job in position p on machine k for a given sequence, respectively. Similarly, let d_p and $d_{[p]}$ denote the due date of job p and the due date of the job in position p, respectively. Moreover, let $T_{[p]}$ represent the tardiness of the job in position p. The setup times are uncertain and satisfy the inequality

$$Ls_{p,k} \leq s_{p,k} \leq Us_{p,k},$$

where $Ls_{p,k}$ and $Us_{p,k}$ denote lower and upper bounds on the setup time $s_{p,k}$, respectively.

Let

$$SST_{[p,k]} = \sum_{r=1}^{p} s_{[r,k]} + t_{[r,k]},$$

and

$$\delta_{[p]} = SST_{[p,1]} - (SST_{[p-1,2]} + s_{[p,2]}), \tag{1}$$

and

$$\Delta_{[p]} = \max\{0, \delta_{[1]}, \delta_{[2]}, \dots, \delta_{[p]}\}. \tag{2}$$

Thus, the tardiness of the job in position p can be determined as

$$T_{[p]} = \max\{0, SST_{[p,2]} + \Delta_{[p]} - d_{[p]}\}. \tag{3}$$

Therefore, the total tardiness (TT) can be expressed as

$$TT = \sum_{p=0}^{n} TT_{[p]}. \tag{4}$$

Let ϕ_1 and ϕ_2 be two job sequences where the sequence ϕ_1 has jobs i and j in arbitrary positions α and β ($\alpha<\beta$), respectively. The sequence ϕ_2 is obtained from the sequence ϕ_1 by swapping the two jobs in positions α and β while keeping all other positions unchanged. If π_1 denotes the subsequence containing the jobs in positions 1, …, α-1, π_2 denotes the subsequence containing the jobs in positions α+1, …, β-1, and π_3 denotes the subsequence containing the jobs in positions β+1, …, n, then, the sequences ϕ_1 and ϕ_2 can be written as $\phi_1 = \{\pi_1, i, \pi_2, j, \pi_3\}$ and $\phi_2 = \{\pi_1, j, \pi_2, i, \pi_3\}$.

Theorem 1: Let $i, j \in J$, $i \neq j$, be two arbitrary jobs.

If

$$s_{j,1} + t_{j,1} + s_{i,2} \leq s_{i,1} + t_{i,1} + s_{j,2} \text{ and } t_{j,2} \geq t_{i,2}$$

then

$$\delta_{[g]}(\phi_2) \leq \delta_{[g]}(\phi_1) \text{ for all } g=1, ..., n. \quad (5)$$

Proof: Both sequences ϕ_1 and ϕ_2 have the same jobs in positions $=1, ..., \alpha\text{-}1$, therefore,

$$\delta_{[r]}(\phi_2) = \delta_{[r]}(\phi_1) \text{for } r=1, ..., \alpha\text{-}1. \quad (6)$$

For $g=\alpha$, it follows from equation (1) that

$$\delta_{[\alpha]}(\phi_1) = SST_{[\alpha-1,1]}+s_{i,1} + t_{i,1} - SST_{[\alpha-1,2]} - s_{i,2}, \quad (7)$$

and

$$\delta_{[\alpha]}(\phi_2) = SST_{[\alpha-1,1]}+s_{j,1} + t_{j,1} - SST_{[\alpha-1,2]} - s_{j,2}. \quad (8)$$

Since $s_{j,1} + t_{j,1} + s_{i,2} \leq s_{i,1} + t_{i,1} + s_{j,2}$, from Equations (7) and (8),

$$\delta_{[\alpha]}(\phi_2) \leq \delta_{[\alpha]}(\phi_1) \ . \quad (9)$$

For $g=\alpha+1, ..., \beta-1$, from Equation (1), it follows that

$$\delta_{[g]}(\phi_1) = SST_{[\alpha-1,1]}+s_{i,1} + t_{i,1} + \sum_{h=\alpha+1}^{g}(s_{[h,1]} + t_{[h,1]}) - SST_{[\alpha-1,2]} - s_{i,2} - t_{i,2} - \sum_{h=\alpha+1}^{g-1}(s_{[h,2]} + t_{[h,2]}) - s_{[g]}, \quad (10)$$

and

$$\delta_{[g]}(\phi_2) = SST_{[\alpha-1,1]}+s_{j,1} + t_{j,1} + \sum_{h=\alpha+1}^{g}(s_{[h,1]} + t_{[h,1]}) - SST_{[\alpha-1,2]} - s_{j,2} - t_{j,2} - \sum_{h=\alpha+1}^{g-1}(s_{[h,2]} + t_{[h,2]}) - s_{[g,2]}, \quad (11)$$

where $\sum_{h=\alpha+1}^{\alpha}(s_{[h,2]} + t_{[h,2]}) = 0$.

It is known that $s_{j,1} + t_{j,1} + s_{i,2} \leq s_{i,1} + t_{i,1} + s_{j,2}$ and $t_{j,2} \geq t_{i,2}$. Therefore, Equations (10) and (11) imply that

$$\delta_{[r]}(\phi_2) \leq \delta_{[r]}(\phi_1) \ \text{ for } g=\alpha+1, ..., \beta-1. \quad (12)$$

For $g=\beta$, it follows from Equation (1) that

$$\delta_{[\beta]}(\phi_1) = SST_{[\alpha-1,1]}+s_{i,1} + t_{i,1} + \sum_{h=\alpha+1}^{\beta-1}(s_{[h,1]} + t_{[h,1]}) + s_{j,1} + t_{j,1} - SST_{[\alpha-1,2]} - s_{i,2} - t_{i,2} - \sum_{h=\alpha+1}^{g-1}(s_{[h,2]} + t_{[h,2]}) - s_{j,2}, \quad (13)$$

and

$$\delta_{[\beta]}(\phi_2) = SST_{[\alpha-1,1]}+s_{j,1} + t_{j,1} + \sum_{h=\alpha+1}^{\beta-1}(s_{[h,1]} + t_{[h,1]}) + s_{i,1} + t_{i,1} - SST_{[\alpha-1,2]} -$$

$$s_{j,2} - t_{j,2} - \sum_{h=\alpha+1}^{g-1}(s_{[h,2]} + t_{[h,2]}) - s_{i,2}. \quad (14)$$

As $t_{j,2} \geq t_{i,2}$, Equations (13) and (14) give

$$\delta_{[\beta]}(\phi_2) \leq \delta_{[\beta]}(\phi_1). \quad (15)$$

For $g = \beta+1, \ldots, n$, it follows from Equation (1) that

$$\delta_{[g]}(\phi_1) = SST_{[\alpha-1,1]} + s_{i,1} + t_{i,1} + \sum_{h=\alpha+1}^{\beta-1}(s_{[h,1]} + t_{[h,1]}) + s_{j,1} + t_{j,1} + \sum_{h=\beta+1}^{g}(s_{[h,1]} + t_{[h,1]}) - SST_{[\propto-1,2]} - s_{i,2} - t_{i,2} - \sum_{h=\alpha+1}^{\beta-1}(s_{[h,2]} + t_{[h,2]}) - s_{j,2} - t_{j,2} - \sum_{h=\beta+1}^{g-1}(s_{[h,2]} + t_{[h,2]}) + s_{[g,2]}, \quad (16)$$

and

$$\delta_{[g]}(\phi_2) = SST_{[\alpha-1,1]} + s_{j,1} + t_{j,1} + \sum_{h=\alpha+1}^{\beta-1}(s_{[h,1]} + t_{[h,1]}) + s_{i,1} + t_{i,1} + \sum_{h=\beta+1}^{g}(s_{[h,1]} + t_{[h,1]}) - SST_{[\propto-1,2]} - s_{j,2} - t_{j,2} - \sum_{h=\alpha+1}^{\beta-1}(s_{[h,2]} + t_{[h,2]}) - s_{i,2} - t_{i,2} - \sum_{h=\beta+1}^{g-1}(s_{[h,2]} + t_{[h,2]}) + s_{[g,2]}. \quad (17)$$

Finally, from Equations (16) and (17) we have

$$\delta_{[g]}(\phi_2) = \delta_{[g]}(\phi_1) \text{ for } g=\beta+1, \ldots, n. \quad (18)$$

Hence, by Equations (6), (9), (12), (15), and (18), Equation (5) follows.

Theorem 2: Let $i, j \in J$, $i \neq j$, be two arbitrary jobs.

If $Us_{j,2} + t_{j,2} \leq Ls_{i,2} + t_{i,2}$,

1. $d_i \leq d_j$,
2. $Us_{j,1} + t_{j,1} + Us_{i,2} \leq Ls_{i,1} + t_{i,1} + Ls_{j,2}$, and
3. $t_{i,2} \leq t_{j,2}$

for sequences ϕ_1 and ϕ_2.
Then

$$TT(\phi_2) \leq TT(\phi_1). \quad (19)$$

Proof:

Case 1: $p = 1, \ldots, \alpha - 1$.

For $p = 1, \ldots, \alpha - 1$, we have $T_{[p]}(\phi_2) = T_{[p]}(\phi_1)$.

Case 2: $p = \alpha$.

$$T_{[\alpha]}(\phi_2) \;=\; SST_{[\alpha,\,2]} + \;\Delta_{[\alpha]} - d_{[p]}$$

$$=\; SST_{[\alpha-1,2]} + s_{[\alpha,2]} + t_{[\alpha,2]} + \Delta_{[\alpha]} - d_{[p]}$$

$$=\; SST_{[\alpha-1,2]} + s_{j,2} + t_{j,2} + \Delta_j - d_j,$$

and

$$T_{[\alpha]}(\phi_1) = SST_{[\alpha,2]} + \Delta_{[\alpha]} - d_{[\alpha]}$$

$$= SST_{[\alpha-1,2]} + s_{[\alpha,2]} + t_{[\alpha,2]} + \;\Delta_{[\alpha]} - d_{[\alpha]}$$

$$=\; SST_{[\alpha-1,2]} + s_{i,2} + t_{i,2} + \Delta_i - d_i.$$

Therefore,

$$T_{[\alpha]}(\phi_1) - T_{[\alpha]}(\phi_2) \;=\; SST_{[\alpha-1,2]} + s_{i,2} + t_{i,2} + \Delta_i - d_i - \left(SST_{[\alpha-1,2]} + s_{j,2} + t_{j,2} + \Delta_j - d_j\right)$$

$$= s_{i,2} + t_{i,2} - d_i - s_{j,2} - t_{j,2} + d_j \;\geq 0 \text{ by conditions 1 and 2.}$$

Case 3: $p = \alpha + 1, \dots, \beta - 1$.

For each p, we have

$$T_{[p]}(\phi_2) \;=\; SST_{[p,2]} + \Delta_{[p]} - d_{[p]} = SST_{[\alpha-1,2]} + s_{j,2} + t_{j,2} + s_{[\alpha+1,2]} + t_{[\alpha+1,2]} + \cdots + s_{[p,2]} + t_{[p,2]} + \Delta_{[p]} - d_{[p]},$$

and

$$T_{[p]}(\phi_1) = SST_{[p,2]} + \Delta_{[p]} - d_{[p]}$$

$$= SST_{[\alpha-1,2]} + s_{i,2} + t_{i,2} + s_{[\alpha+1,2]} + t_{[\alpha+1,2]} + \cdots + s_{[p,2]} + t_{[p,2]} + \Delta_{[p]} - d_{[p]}.$$

Hence,

$T_{[p]}(\phi_1) \;-\; T_{[p]}(\phi_2) = s_{i,2} + t_{i,2} - s_{j,2} - t_{j,2} \;\geq 0$ by condition 1.

Case 4: $p = \beta$.

We have

$$T_{[\beta]}(\phi_2) \;=\; SST_{[\beta,2]} + \Delta_{[\beta]} - d_{[\beta]}$$

$$= SST_{[\alpha-1,2]} + s_{[\alpha-1,2]} + t_{[\alpha-1,2]} + s_{j,2} + t_{j,2} + s_{[\alpha+1,2]} + t_{[\alpha+1,2]} + \cdots + s_{[\beta-1,2]} + t_{[\beta-1,2]} + s_{i,1} + t_{i,2} + \Delta_{[\beta]} - d_{[\beta]},$$

and

$$T_{[\beta]}(\phi_1) \;=\; SST_{[\beta,2]} + \Delta_{[\beta]} - d_{[\beta]}$$

$$= SST_{[\alpha-1,2]} + s_{[\alpha-1,2]} + t_{[\alpha-1,2]} + s_{i,2} + t_{i,2} + s_{[\alpha+1,2]} + t_{[\alpha+1,2]} + \cdots + s_{[\beta-1,2]} + t_{[\beta-1,2]} + s_{j,1} + t_{j,2} + \Delta_{[\beta]} - d_{[\beta]}.$$

So that

$$T_{[\beta]}(\phi_1) - T_{[\beta]}(\phi_2) = 0.$$

Case 5: $p = \beta + 1, \dots, n$.

For each p, we have

$$T_{[p]}(\phi_2) = SST_{[p,2]} + \Delta_{[p]} - d_{[p]}$$

$$= SST_{[\alpha-1,2]} + s_{j,2} + t_{j,2} + s_{[\alpha+1,2]} + t_{[\alpha+1,2]} + \cdots + s_{[\beta-1,2]} + t_{[\beta-1,2]} + s_{i,2} + t_{i,2} + \cdots + s_{[p,2]} + t_{[p,2]} + \Delta_{[p]} - d_{[p]},$$

and

$$T_{[p]}(\phi_1) = SST_{[p,2]} + \Delta_{[p]} - d_{[p]}$$

$$= SST_{[\alpha-1,2]} + s_{i,2} + t_{i,2} + s_{[\alpha+1,2]} + t_{[\alpha+1,2]} + \cdots + s_{[\beta-1,2]} + t_{[\beta-1,2]} + s_{j,2} + t_{j,2} + \cdots + s_{[p,2]} + t_{[p,2]} + \Delta_{[p]} - d_{[p]}.$$

Therefore,

$$T_{[p]}(\phi_1) - T_{[p]}(\phi_2) = 0.$$

Combining Cases 1, 2, 3, 4, and 5, Equation (19) follows.

4. A numerical example highlighting the impact of uncertainty

To illustrate the structure of the uncertain scheduling problem studied in this paper and to provide additional insight into its implications, we present a small numerical example. The purpose of this example is not only to clarify the model components, but also to demonstrate the fundamental differences between the deterministic and uncertain versions of the problem.

Consider a two-machine flow shop environment with three jobs. For each job, the processing times on both machines are known and deterministic, whereas the setup times on each machine are uncertain and lie within specified lower and upper bounds. In addition, each job has an associated due date. The processing times, setup time bounds, and due dates for all three jobs are summarized in Table 1.

Table 1. Data for the numerical example.

Jobs	$t_{p,1}$	$t_{p,2}$	$Ls_{p,1}$	$Us_{p,1}$	$Ls_{p,2}$	$Us_{p,2}$	d_p
1	19	20	10	20	10	14	30
2	12	22	2	5	3	6	60
3	20	10	5	10	4	12	50

With three jobs, there are six possible job sequences, namely (1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), and (3, 2, 1). However, based on Theorem 2 established earlier in this paper, job 2 must precede job 1 in any optimal solution. This dominance property allows us to eliminate all sequences in which job 1 appears before job 2.

Consequently, the sequences (1, 2, 3), (1, 3, 2), and (3, 1, 2) are excluded from further consideration. The set of candidate optimal sequences is therefore reduced to the remaining three sequences: (2, 1, 3), (2, 3, 1), and (3, 2, 1).

Next, in order to examine the effect of setup time uncertainty, we construct two specific instances that satisfy the lower and upper bounds given in Table 1. These instances represent two possible realizations of the uncertain setup times and are reported in Table 2.

Table 2. Setup times for the two instances.

Job	Instance 1		Instance 2	
	$s_{p,1}$	$s_{p,2}$	$s_{p,1}$	$s_{p,2}$
1	10	10	20	14
2	2	3	5	6
3	5	4	10	12

For each instance, we evaluate all remaining feasible sequences and compute the corresponding total tardiness values. The results are reported in Table 3.

Table 3. Total tardiness for both instances.

Sequence	TT for Instance 1	TT for Instance 2
(2, 1, 3)	66	94
(2, 3, 1)	58	87
(3, 2, 1)	62	85

As shown in Table 3, the optimal sequence depends critically on the realized setup times. For Instance 1, the sequence (2, 3, 1) yields the minimum total tardiness and is therefore optimal. In contrast, for Instance 2, the optimal sequence is (3, 2, 1), which outperforms the other alternatives under this realization.

This example clearly highlights a key challenge of the uncertain scheduling problem. Unlike the deterministic case, where a single optimal solution can be identified based on fixed parameter values, the uncertain case admits infinitely many possible instances corresponding to different realizations of setup times within their bounds. Since the optimal sequence varies across these instances, it is impossible to identify a single sequence that is guaranteed to be optimal for all realizations. Therefore, the notion of optimality must be reconsidered in the presence of uncertainty, motivating the robust approaches developed in this paper.

5. Algorithms PA1 and PA2

As the minimization of total tardiness is NP(Nondeterministic Polynomial time)-hard even with a single machine, different algorithms were presented in the scheduling literature for our problem. In this section, we present two new algorithms for the problem. The algorithms are obtained by using the information on di, Lsi,1, Lsi,2, Usi,1, Usi,2, ti,1, and ti,2. The description of the algorithms is given as:

Algorithm PAi (i = 1 or 2)

Input:

n // number of jobs

$\{d_i\}$ // due dates, i = 1,...,n

$\{t_{i,1}, t_{i,2}\}$ // processing times

T, R // Tardiness factor and due date range

Output:

π_i // final job sequence obtained by PAi

Begin

// Generate bounds of setup times (see Section 6)

1. For i = 1 to n do

2. Generate $Ls_{i,1}$, $Ls_{i,2}$, $Us_{i,1}$, $Us_{i,2}$

End For

// Generate setup times within bounds (see Section 6)

3. For i = 1 to n do

4. Generate $s_{i,1}$ and $s_{i,2}$ such that:

$Ls_{i,k} \le s_{i,k} \le Us_{i,k}$, $k = 1,2$

End For

// Generate processing times (see Section 6)

5. For i = 1 to n do

6. Generate $t_{i,1}$ and $t_{i,2}$

End For

// Select parameter combination (see Section 6)

7. Select values of T and R

// Compute priority index according to algorithm (*PA1* or *PA2*)

8. For i = 1 to n do

9. If algorithm=*PA1* then

10. Priority (i) = $d_i + 0.5 * Us_{i,2} + 0.2 * Ls_{i,2} + t_{i,1} + 0.5 * t_{i,2}$

11. Else if *PA2* then

12. Priority (i) = $d_i + [Us_{i,1} + 0.5 * (Ls_{i,2} + Us_{i,2})] + t_{i,1} + t_{i,2}$

13. End If

End For

14. Sort jobs in nondecreasing order of Priority (i)

15. Let π_i be the resulting sequence

// Pairwise improvement phase based on Theorem 2

16. For $g = 1$ to $n-1$ do

17. For $r = g + 1$ to n do

18. If jobs in positions g and r of π_i satisfy the dominance condition in Theorem 2 then

19. Swap jobs in positions g and r

20. End If

21. End For

22. End For

23. Return π_i

End

It is important to emphasize that the proposed algorithms do not rely on sampling or generating specific realizations of the uncertain setup times. Instead, only the lower and upper bounds of the setup times are utilized throughout the algorithmic framework. These bounds are directly incorporated into the priority indices and dominance relations, allowing the algorithms to account for uncertainty without assuming any probability distribution or scenario realization. As a result, the proposed approach follows an interval-based uncertainty representation, in which scheduling decisions are guided by conservative and optimistic bounds rather than by repeated deterministic evaluations under sampled data.

The weighted priority index used in Step 5 of PA1 reflects the asymmetric roles of due dates, processing times, and interval-based setup times in the considered flowshop with uncertain setups. Since only lower and upper bounds of setup times are available, these bounds provide different types of information: the upper bound represents a conservative estimate that guards against worst-case delays, while the lower bound reflects a more optimistic scenario. Assigning a larger weight to the upper bound of the setup time on the second machine emphasizes robustness against downstream congestion, which has a stronger impact on total tardiness, whereas a smaller weight on the lower bound helps balance excessive conservatism.

Similarly, deterministic processing times on both machines are included explicitly, with a higher weight on the first-machine processing time to account for its role in determining job release times to the second machine. The due date term remains unweighted, as tardiness is directly driven by due-date proximity. The selected weights therefore reflect a trade-off between robustness to setup-time uncertainty and sensitivity to deterministic processing requirements.

The final parameter values were obtained through systematic calibration over a discrete and limited set of candidate weights, rather than continuous fine-tuning, and were then fixed for all subsequent experiments. This procedure aims to balance performance and generality while avoiding instance-specific overfitting.

After preliminary computational testing, it was observed that *PA1* consistently outperformed *PA2*, which motivated further calibration of its priority index. Initially, Step 5 of *PA1* was defined as $d_i + \left(UBs_{i,2} + UBs_{i,2}\right) + t_{i,1} + t_{i,2}$ for each i. To systematically examine the impact of different components, Step 5 was generalized to $(a)d_i + (b)UBs_{i,2} + (c)UBs_{i,2} + (d)t_{i,1} + (e)t_{i,2}$ where $a, b, c, d, e \in \{0.2, 0.5, 1\}$. All combinations of these values were evaluated on instances with $n = 100$ jobs. The combination a = 1, b = 0.5, c = 0.2, d = 1, e = 0.5 consistently yielded the best overall performance and was therefore fixed and used in all subsequent experiments.

6. Computational experiments

We compare the proposed heuristics (PA1 and PA2) with PH2 (Allahverdi and Allahverdi [34]), which is the only established state-of-the-art heuristic specifically developed for the same problem setting with uncertain setup times and the total tardiness objective; comparing against other heuristics designed for deterministic or structurally different problems would not provide a meaningful or fair evaluation.

We conducted our computational experiments by using the Python computer language. All numerical experiments were performed on a MacBook Air (Sequoia 15.6) equipped with an Apple M1 processor (8-core CPU) and 16 GB of RAM, running macOS. The proposed algorithms were implemented in Python 3.13.5 using standard scientific computing libraries. All simulations were executed under identical conditions in a single-threaded environment. The average computational time per execution was approximately 3 seconds.

The upper bounds Usi,1 and Usi,2 for the setup times were generated uniformly from the interval [1,100]. The lower bounds were then generated uniformly from the interval [Usi,j – 50, Usi,j] for j = 1, 2. If the generated number was a negative lower bound, the lower bound is set to 1. Generating the Usi,1 and Usi,2 in this way is common in the scheduling literature, e.g., Aydilek et al. [33].

To generate due dates, two variables R and T are used as tightness and range factors. Taking 0.25, 0.5 and 0.75 for R and T, there are a total of nine different combinations of R and T. For each combination, due dates, di, are generated from the uniform distribution

$$U\left(\left(Ls_{i,2}+t_{i,2}\right)\left(1-T-\frac{R}{2}\right),\left(Ls_{i,2}+t_{i,2}\right)\left(1-T+\frac{R}{2}\right)\right).$$

This method is also commonly used in scheduling literature to generate due dates.

Finally, processing times ti,1, and ti,2 were generated uniformly from the range [1, 100], which is also common in the scheduling literature, e.g., Aydilek et al. [33].

Once the results are obtained, the error of algorithm Ai is computed using the formula,

$$\text{Error }(A_i) = \frac{TT(Ai)-\min(\text{TT(PH2), TT(PA1), TT(PA2)})}{\max(\text{TT(PH2), TT(PA1), TT(PA2)}-\min(\text{TT(PH2), TT(PA1), TT(PA2)}}\times 100,$$

where TT(Ai) denotes the tardiness of Ai. Hence, the error for the maximum total tardiness is 100 and that of the minimum is 0.

Since the setup times are unknown, it would not be wise to assume a certain distribution. Therefore, we generate the setup times from a wide range of different distributions: uniform, normal, positive linear, and negative linear to include both symmetric and skewed distributions. For each of these four distributions, five different n values were considered, namely n = 100, 200, 300, 400, and 500. For each n-value, 50 replications were carried out. Finally, for each replication, 9 cases (depending on the nine different combinations of R and T) were carried out. Therefore, a total of 4 x 5 x 50 x 9 = 9,000 cases were considered.

Tables 4–7 report 4 samples out of a total of 36 (9 x 4) tables. Specifically, Tables 4–7 report the results (average error and std) for uniform distribution (for combination of T = 0.25 and R = 0.5), normal distribution (for combination of T = 0.5 and R = 0.5), negative linear distribution (for combination of T = 0.75 and R = 0.75), and positive linear distribution (for combination of T = 0.5 and R = 0.75), respectively. The data given in the tables are the averages of the 50 replications. The remaining tables are omitted to save space since the other tables were similar. On the other hand, Table 8 and 9 report the overall average errors and standard deviations of the errors for uniform and normal distributions, and positive linear and negative linear distributions, respectively, over all 9 combinations of T and R values.

Table 4. Overall avg. errors for uniform distribution when T = 0.25 and R = 0.5.

n	Algorithm	Error	Std
100	PH2	100	0
	PA1	1.27088	2.22372
	PA2	3.85854	5.46198
200	PH2	100	0
	PA1	1.42947	2.6733
	PA2	2.5513	3.96801
300	PH2	100	0
	PA1	0.78096	1.69468
	PA2	2.39668	2.33842
400	PH2	100	0
	PA1	0.54091	1.69634
	PA2	3.23424	3.1817
500	PH2	100	0
	PA1	0.34248	0.77971
	PA2	2.33306	2.52749
Avg.	PH2	100	0
	PA1	0.87294	1.81355
	PA2	2.87476	3.49552

Table 5. Overall avg. errors for normal distribution when T = 0. 5 and R = 0.25.

N	Algorithm	Error	Std
100	PH2	100	0
	PA1	1.50268	2.93228
	PA2	3.00311	4.05835
200	PH2	100	0
	PA1	0.92497	2.015
	PA2	2.57226	3.34509
300	PH2	100	0
	PA1	0.84598	1.72615
	PA2	2.03967	2.37637
400	PH2	100	0
	PA1	0.47773	1.09401
	PA2	2.4045	2.39412
500	PH2	100	0
	PA1	0.33049	0.88337
	PA2	2.30591	2.2549
Avg.	PH2	100	0
	PA1	0.81637	1.73016
	PA2	2.46509	2.88577

Table 6. Overall avg. errors for negative linear distribution when T = 0.75 and R = 0.75.

n	Algorithm	Error	Std
100	PH2	100	0
	PA1	1.50268	2.93228
	PA2	3.00311	4.05835
200	PH2	100	0
	PA1	0.92497	2.015
	PA2	2.57226	3.34509
300	PH2	100	0
	PA1	0.84598	1.72615
	PA2	2.03967	2.37637
400	PH2	100	0
	PA1	0.47773	1.09401
	PA2	2.4045	2.39412
500	PH2	100	0
	PA1	0.33049	0.88337
	PA2	2.30591	2.2549
Avg.	PH2	100	0
	PA1	0.81637	1.73016
	PA2	2.46509	2.88577

Table 7. Overall avg. errors for positive linear distribution when T = 0.5 and R = 0.75.

n	Algorithm	Error	Std
100	PH2	100	0
	PA1	1.67914	3.27967
	PA2	3.51764	4.63216
200	PH2	100	0
	PA1	0.74144	1.58411
	PA2	2.89552	3.24725
300	PH2	100	0
	PA1	0.73766	1.39247
	PA2	2.06158	2.49616
400	PH2	100	0
	PA1	0.47763	1.06075
	PA2	2.50196	2.77436
500	PH2	100	0
	PA1	0.47685	1.18729
	PA2	1.94381	2.02262
Avg.	PH2	100	0
	PA1	0.82254	1.70086
	PA2	2.5841	3.03451

Table 8. Overall avg. errors for uniform and normal distributions for all combinations T and R.

Distribution	n	Algorithm	Error	Std
Uniform	100	PH2	100	0
		PA1	1.77589	3.68318
		PA2	3.34029	4.44321
	200	PH2	100	0
		PA1	1.08776	2.13205
		PA2	2.3364	2.96133
	300	PH2	100	0
		PA1	0.93544	1.92805
		PA2	2.20001	2.68311
	400	PH2	100	0
		PA1	0.37757	0.9958
		PA2	2.46045	2.35946
	500	PH2	100	0
		PA1	0.32301	0.85464
		PA2	2.2582	2.30728
	Avg.	PH2	100	0
		PA1	0.899934	1.918744
		PA2	2.51907	2.950878
Normal	100	PH2	100	0
		PA1	2.1281	4.18225
		PA2	2.8601	4.17326
	200	PH2	100	0
		PA1	0.8423	1.82563
		PA2	2.91136	3.45225
	300	PH2	100	0
		PA1	0.55504	1.35312
		PA2	2.65654	2.50806
	400	PH2	100	0
		PA1	0.43801	1.06165
		PA2	2.20085	2.19276
	500	PH2	100	0
		PA1	0.30761	0.83732
		PA2	2.2501	2.08817
	Avg.	PH2	100	0
		PA1	0.854212	1.851994
		PA2	2.57579	2.8829

Table 9. Overall avg. errors for positive and negative linear distributions for all combinations T and R.

Distribution	n	Algorithm	Error	Std
Positive	100	PH2	100	0
Linear		PA1	1.78194	3.13841
		PA2	3.77827	4.82904
	200	PH2	100	0
		PA1	1.05978	2.28577
		PA2	2.48506	3.26917
	300	PH2	100	0
		PA1	0.56774	1.34251
		PA2	2.71764	2.85816
	400	PH2	100	0
		PA1	0.44038	1.06889
		PA2	2.32959	2.30702
	500	PH2	100	0
		PA1	0.41126	1.03508
		PA2	2.10662	2.114
	Avg.	PH2	100	0
		PA1	0.819114	1.774132
		PA2	2.69104	3.075478
Negative	100	PH2	100	0
Linear		PA1	2.1281	3.60026
		PA2	2.8601	4.93643
	200	PH2	100	0
		PA1	1.10529	2.21628
		PA2	2.63827	3.17331
	300	PH2	100	0
		PA1	0.64729	1.41541
		PA2	2.28052	2.78657
	400	PH2	100	0
		PA1	0.60725	1.26366
		PA2	1.98086	2.49616
	500	PH2	100	0
		PA1	0.46666	1.00688
		PA2	2.01931	2.21696
	Avg.	PH2	100	0
		PA1	0.921686	1.900498
		PA2	2.539446	3.121886

The overall average errors of (over the considered 9,000 cases) the proposed algorithms PA1 and PA2, and the best existing algorithm PH2 are 0.873737, 2.5813365, and 100, respectively. The average errors of the algorithms are shown in Figure 1 against the number of jobs, n. Figure 2 shows the errors of the proposed algorithms PA1 and PA2 against n excluding PH2 since the error of PH2 is always 100 and it is hard to compare the performance PA1 and PA2. Note that the standard deviation (std) of PH2 is zero since PH2 always give the worst outcome. Similarly, Figure 3 shows the standard deviations of the errors of the proposed algorithms PA1 and PA2 against n.

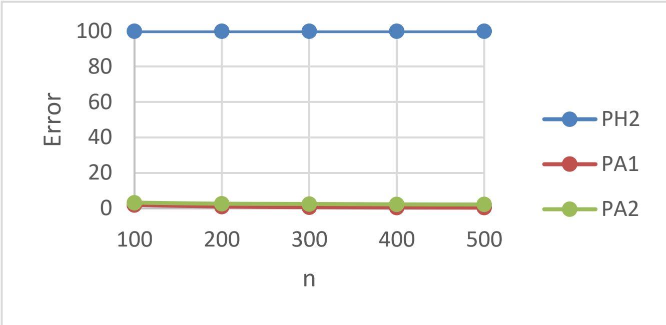

Figure 1. Overall average errors of the algorithms versus n.

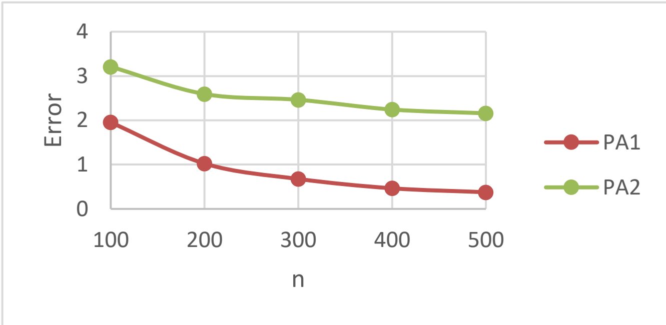

Figure 2. Overall average errors of the proposed algorithms versus n.

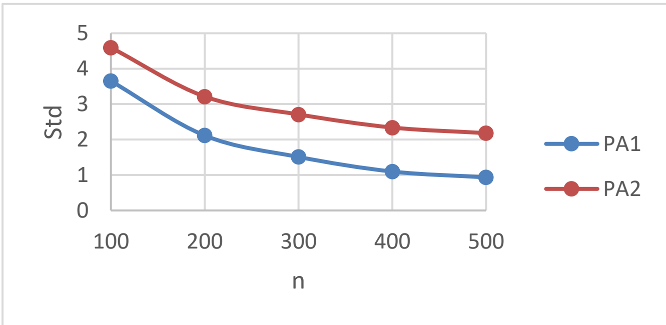

Figure 3. Overall std of the proposed algorithms versus n.

The results indicates that the overall average error of the best existing algorithm PH2 is about 40 times larger than that of the proposed algorithm PA2, and it is more than 110 times larger than that of the proposed algorithm PA1. Therefore, the proposed algorithms PA1 and PA2 performs much better than the best existing algorithm PH2 in the literature. Another advantage of the proposed algorithms PA1 and PA2 is that their performances get better as the number of jobs (n) increases. Furthermore, the error of PA2 is about 3 times larger than that of PA1. Hence, the proposed algorithm PA1 is the best algorithm for the problem.

Next, we statistically show that the proposed algorithm PA2 is better than the best existing algorithm PH2, and the proposed algorithm PA1 is better than the proposed algorithm PA2.

7. Hypothesis testing

To check if the proposed algorithm PA2 is statistically significantly better than the PH2 algorithm (the existing best algorithm), we perform a paired t-test. We have the null hypothesis,

$$H_0: \mu_{PA2} = \mu_{PH2},$$

and the alternative hypothesis,

$$H_1: \mu_{PA2} < \mu_{PH2},$$

where μ_{PA2} and μ_{PH2} are the sample means of the total tardiness of PA2 and PH2, respectively.

The null hypothesis was rejected at a significance level of 0.001 for all combinations of T and R, and for all the four distributions, i.e., for each of the 36 cases. Therefore, it follows that the proposed algorithm PA2 is statistically significantly better than that of the best-known algorithm PH2.

We have also conducted a test of hypotheses to compare the performances of the two proposed algorithms of PA1 and PA2. To check if the proposed algorithm PA1 is statistically significantly better than the proposed algorithm PA2, we perform a paired t-test. We have the null hypothesis,

$$H_0: \mu_{PA1} = \mu_{PA2},$$

and the alternative hypothesis,

$$H_1: \mu_{PA1} < \mu_{PA2},$$

where μ_{PA1} and μ_{PA2} are the sample means of the total tardiness of PA1 and PA2, respectively.

The null hypothesis was rejected at a significance level of 0.001 for all combinations of T and R, and for all the four distributions. Therefore, it follows that the proposed algorithm PA1 is statistically significantly better than that of proposed algorithm PA2. Therefore, the proposed algorithm PA1 is the best algorithm for the problem.

8. Conclusion

In this study, we investigated the two-machine flowshop scheduling problem with uncertain setup times under the objective of minimizing total tardiness. We first developed a mathematical dominance relation tailored to the interval-based uncertainty setting. We then proposed two new heuristic algorithms, PA1 and PA2, incorporating the developed dominance relation. Extensive computational experiments were conducted using four uncertainty distributions, five problem sizes, and nine different due-date parameter combinations, resulting in 9,000 test instances. The results demonstrated that both proposed algorithms significantly outperformed the best existing algorithm (PH2), and statistical hypothesis testing confirmed the superiority of PA1 and PA2. Among the proposed approaches, PA1 emerged as the most effective overall. Furthermore, the performance improvement became more pronounced as the problem size increased.

We have explicitly incorporated the developed dominance relation into the proposed algorithms. The computational analysis showed that incorporating the dominance relation reduced the solution error to below 5%. Although the numerical improvement was modest because PA1 already captured most implied job-ordering preferences, the dominance relation provides a formal theoretical justification for these preferences and enables pruning in enumeration-based procedures. Dominance relations are typically employed within implicit enumeration techniques such as branch-and-bound algorithms or dynamic programming.

Future research may focus on developing implicit enumeration algorithms such as [35-40] that exploit the proposed dominance relation, as such techniques may further improve solution quality for small- and medium-sized instances under uncertainty. In addition, extending the current framework to environments where both setup times and processing times are uncertain, as well as to three- and four-machine flowshop settings, represents promising research directions. Moreover, conducting application-driven case studies incorporating additional real-world constraints (such as sequence-dependent setups, machine eligibility restrictions, or industry-specific due-date policies) would further demonstrate the practical impact of the proposed methods.

Author contributions

Muberra Allahverdi and Ali Allahverdi contributed equally to the conceptualization, methodology, data analysis, and writing the manuscript. Both authors approved the final version of the manuscript

Use of Generative-AI tools declaration

AI-assisted tools (ChatGPT by OpenAI) were used solely to improve the grammar and clarity of the language and did not contribute to the scientific content of the manuscript.

Conflict of interest

The authors have no financial or non-financial conflicts of interest to report.

Acknowledgment

The authors would like to thank the three anonymous referees and the Associate Editor for their valuable comments and suggestions on the two revisions of the paper. Their feedback significantly improved the current version of the manuscript.

References

1. H. Y. Fuchigami, S. Rangel, A survey of case studies in production scheduling: analysis and perspectives, *J. Comput. Sci.*, **25** (2018), 425–436. https://doi.org/10.1016/j.jocs.2017.06.004
2. S. Rahmativala, J. Ghahremani, A robust optimization method for hybrid flow shop scheduling with uncertain setup times, *Decis. Anal. J.,* 2025, 100609. https://doi.org/10.1016/j.dajour.2025.100609
3. B. D. A. Prata, L. R. D. Abreu, V. Fernandez-Viagas, A systematic review of permutation flow shop scheduling with due-date-related objectives, *Comput. Oper. Res.,* **177** (2025), 106989. https://doi.org/10.1016/j.cor.2025.106989

4. J. R. Birge, F. Louveaux, *Introduction to stochastic programming,* Springer, 2011. https://doi.org/10.1007/978-1-4614-0237-4
5. A. Ben-Tal, L. E. Ghaoui, A. Nemirovski, *Robust optimization,* University Press, 2018. https://doi.org/10.1515/9781400831050
6. D. Bertsimas, D. B. Brown, C. Caramanis, Theory and applications of robust optimization, *SIAM Rev.*, **53** (2011), 464–501. https://doi.org/10.1137/080734510
7. H. Emmons, G. Vairaktarakis, *Flow shop scheduling: Theoretical results, algorithms, and applications*, Springer, **18** (2012), 334. https://doi.org/10.1007/978-1-4614-5152-5
8. A. Gharbi, T. Ladhari, M. K. Msakni, M. Serairi, The two-machine flowshop scheduling problem with sequence-independent setup times: New lower bounding strategies, *Eur. J. Oper. Res.,* **231** (2013), 69–78. https://doi.org/10.1016/j.ejor.2013.05.031
9. S. Gawiejnowicz, *Models and algorithms of time-dependent scheduling*, Springer Berlin Heidelberg, 2020. https://doi.org/10.1007/978-3-662-59362-2
10. A. Agnetis, J. C. Billaut, M. Pinedo, D. Shabtay, Fifty years of research in scheduling–Theory and applications, *Eur. J. Oper. Res.*, **327** (2025), 367–393. https://doi.org/10.1016/j.ejor.2025.01.034
11. S. Gawiejnowicz, Theory and methodology of time-dependent scheduling: Past, present and future, *Eur. J. Oper. Res.*, **329** (2025), 719–746. https://doi.org/10.1016/j.ejor.2025.05.024
12. J. Ksciuk, S. Kuhlemann, K. Tierney, A. Koberstein, Uncertainty in maritime ship routing and scheduling: A Literature review, *Eur. J. Oper. Res.*, **308** (2023), 499–524. https://doi.org/10.1016/j.ejor.2022.08.006
13. M. Allahverdi, Minimizing total tardiness in a two-machine flowshop with uncertain and bounded processing times, *RAIRO-Oper. Res.*, **57** (2023), 1353–1375. https://doi.org/10.1051/ro/2023023
14. C. Pessan, J. L. Bouquard, E. Neron, An unrelated parallel machines model for an industrial production resetting problem, *Eur. J. Ind. Eng.*, **2** (2008), 153–171. https://doi.org/10.1504/EJIE.2008.017349
15. A. Allahverdi, C.T. Ng, T. E. Cheng, M.Y. Kovalyov, A survey of scheduling problems with setup times or costs, *Eur. J. Oper. Res.*, **187** (2008), 985–1032. https://doi.org/10.1016/j.ejor.2006.06.060
16. A. Allahverdi, The third comprehensive survey on scheduling problems with setup times/costs, *Eur. J. Oper. Res.*, **246** (2015), 345–378. https://doi.org/10.1016/j.ejor.2015.04.004
17. H. Aydilek, A. Aydilek, A. Allahverdi, Algorithms to minimize total completion time in a two-machine flowshop problem with uncertain set-up times, *Eng. Optim.*, **53** (2021), 1417–1430. https://doi.org/10.1080/0305215X.2020.1796999
18. A. Allahverdi, H. M. Soroush, The significance of reducing setup times/setup costs, *Eur. J. Oper. Res.*, **187** (2008), 978–984. https://doi.org/10.1016/j.ejor.2006.09.010
19. Y. J. Feng, J. L. Kong, Multi-objective hybrid flow-shop scheduling in parallel sequential mode while considering handling time and setup time, *Appl. Sci.*, **13** (2023), 3563. https://doi.org/10.3390/app13063563
20. S. Aqil, Effective population-based meta-heuristics with neh and grasp heuristics minimizing total weighted flow time in no-wait flow shop scheduling problem under sequence-dependent setup time constraint, *Arab. J. Sci. Eng.,* **49** (2024), 12235–12258. https://doi.org/10.1007/s13369-023-08642-7
21. S. Zhao, Scheduling jobs with general truncated learning effects including proportional setup times, *Comp. Appl. Math.*, **41** (2022), 146. https://doi.org/10.1007/s40314-022-01851-0

22. C. Y. Cheng, P. Pourhejazy, K. C. Ying, Y. H. Liao, New benchmark algorithms for No-wait flowshop group scheduling problem with sequence-dependent setup times, *Appl. Soft Comput.*, **111** (2021), 107705. https://doi.org/10.1016/j.asoc.2021.107705
23. K. Jansen, K. M. Klein, M. Maack, M. Rau, Empowering the configuration-IP: new PTAS results for scheduling with setup times, *Math. Program.*, **195** (2022), 367–401. https://doi.org/10.1007/s10107-021-01719-3
24. M. Ertem, F. Ozcelik, T. Saraç, Single machine scheduling problem with stochastic sequence-dependent setup times, *Int. J. Prod. Res.,* **57** (2019), 3273–3289. https://doi.org/10.1080/00207543.2019.1581383
25. B. J. Joo, P. Xirouchakis, A production scheduling problem with uncertain sequence-dependent set-up times and random yield, *Int. J. Prod. Res.*, **53** (2015), 2820–2835. https://doi.org/10.1080/00207543.2014.998790
26. M. E. Bruni, S. Khodaparasti, E. Demeulemeester, The distributionally robust machine scheduling problem with job selection and sequence-dependent setup times, *Comput. Oper. Res.*, **123** (2020), 105017. https://doi.org/10.1016/j.cor.2020.105017
27. S. C. Kim, P. M. Bobrowski, Scheduling jobs with uncertain setup times and sequence dependency, *Omega*, **25** (1997), 437–447. https://doi.org/10.1016/S0305-0483(97)00013-3
28. İ. Yanıkoğlu, T. Yavuz, Branch-and-price approach for robust parallel machine scheduling with sequence-dependent setup times, *Eur. J. Oper. Res.*, **301** (2022), 875–895. https://doi.org/10.1016/j.ejor.2021.11.023
29. A. Stanković, N. Simić, M. Stanković, Metaheuristic optimization for stochastic job scheduling in parallel machine systems with uncertain processing and setup times, *J. Eng. Manag. Syst. Eng.*, **4** (2025), 206–217. https://doi.org/10.56578/jemse040304
30. M. Allahverdi, A. Allahverdi, Minimizing total completion time in a two-machine no-wait flowshop with uncertain and bounded setup times, *J. Ind. Manag. Optim.*, **16** (2020), 2439–2457. https://doi.org/10.3934/jimo.2019062
31. A. Allahverdi, M. Allahverdi, Two-machine no-wait flowshop scheduling problem with uncertain setup times to minimize maximum lateness, *Comp. Appl. Math.*, **37** (2018), 6774–6794. https://doi.org/10.1007/s40314-018-0694-3
32. A. Aydilek, H. Aydilek, A. Allahverdi, Increasing the profitability and competitiveness in a production environment with random and bounded setup times, *Int. J. Prod. Res.*, **51** (2013), 106–117. https://doi.org/10.1080/00207543.2011.652263
33. A. Aydilek, H. Aydilek, A. Allahverdi, Production in a two-machine flowshop scheduling environment with uncertain processing and setup times to minimize makespan, *Int. J. Prod. Res.*, **53** (2015), 2803–2819. https://doi.org/10.1080/00207543.2014.997403
34. A. Allahverdi, M. Allahverdi, Minimizing total tardiness in flowshops with uncertain setup times, *GITMA*, 2025, 82–87.
35. M. Jafari-Nikpay, Z. Arabzade-Nosratabad, F. Momayezi, Optimizing smart home energy management with branch and bound method: A novel model for daily and weekly scheduling of smart appliances, *Spectr. Oper. Res.*, **3** (2026), 252–274. https://doi.org/10.31181/sor31202643
36. J. Ahn, H. J. Kim, A branch and bound algorithm for scheduling of flexible manufacturing systems, *IEEE Trans. Autom. Sci. Eng.*, **21** (2023), 4382–4396. https://doi.org/10.1109/TASE.2023.3296087

37. W. Bożejko, J. Pempera, M. Uchroński, M. Wodecki, Quantum annealing-driven branch and bound for the single machine total weighted number of tardy jobs scheduling problem, *Future Gener. Comput. Syst.*, **155** (2024), 245–255. https://doi.org/10.1016/j.future.2024.02.016
38. X. Li, Z. W. He, N. M. Wang, A branch-and-bound algorithm for the proactive resource-constrained project scheduling problem with a robustness maximization objective, *Comput. Oper. Res.*, **166** (2024), 106623. https://doi.org/10.1016/j.cor.2024.106623
39. L. Liu, M. Urgo, A branch-and-bound approach to minimise the value-at-risk of the makespan in a stochastic two-machine flow shop, *Int. J. Prod. Res.*, **62** (2024), 2107–2123. https://doi.org/10.1080/00207543.2023.2217279
40. Z. Wang, Q. Zeng, A branch-and-bound approach for AGV dispatching and routing problems in automated container terminals, *Comput. Ind. Eng.*, **166** (2022), 107968. https://doi.org/10.1016/j.cie.2022.107968