

MINIMIZING TOTAL TARDINESS IN A TWO-MACHINE FLOWSHOP WITH UNCERTAIN AND BOUNDED PROCESSING TIMES

MUBERRA ALLAHVERDI*

Abstract. The two-machine flowshop scheduling problem with the performance measure of total tardiness is addressed. This performance measure is essential since meeting deadlines is a crucial part of scheduling and a major concern for some manufacturing systems. The processing times on both machines are uncertain variables and within some lower and upper bounds. This is due to uncertainty being an integral part of some manufacturing settings, making it impossible to predict processing times in advance. To the best of the author's knowledge, this problem is addressed for the first time in this paper. A dominance relation is established and nineteen algorithms are proposed. These algorithms are extensively evaluated through randomly generated data for different numbers of jobs and four different distributions, representing both symmetric and non-symmetric distributions. Computational experiments show that the presented algorithms perform extremely well when compared with a random solution. In particular, the best of the considered 19 algorithms reduces the error of the random solution by 99.99% and the error of the worst algorithm (among the 19 algorithms) by 99.96%. The results are confirmed by a test of hypothesis and this algorithm is recommended.

Mathematics Subject Classification. 90B36.

Received October 18, 2022. Accepted February 16, 2023.

1. INTRODUCTION

Performance measures in the scheduling literature may be categorized into two groups: those based on job completion times such as makespan and mean completion time, and those based on job due dates such as maximum lateness and total tardiness. The former is used more widely, while the latter is used when a penalty exists for completing jobs past their due dates. The performance measure of total tardiness is used when the objective is to minimize the sum of the gaps between job due dates and their completion times, given that job completion times are later than their due dates. The scheduling cost gets higher as the sum of the gaps increases, Allahverdi and Aydilek [7]. Allahverdi and Aydilek [7] state that the total tardiness cost may also include the loss of goodwill, penalty cost in contracts, and damaged reputation. Allahverdi *et al.* [9] point out that meeting customer due dates is a main concern for some manufacturing systems, while Ding *et al.* [17] point out that customer satisfaction is attained by fulfilling job due dates. Furthermore, performance measures such as total tardiness are used to evaluate a manager's performance in automobile manufacturing and semiconductor industries, Seo *et al.* [27] or petroleum refineries, Arabameri and Salmasi [11].

Keywords. Flowshop scheduling, total tardiness, uncertain processing times, algorithm.

Department of Mathematics, Faculty of Engineering and Natural Sciences, Ankara Yıldırım Beyazıt University, Ankara, Turkey.

*Corresponding author: m.allahverdi.abdulhafiz@aybu.edu.tr

Thus, considering the performance measure of total tardiness has many advantages. It should be noted that such problems cannot be answered by other performance measures, such as maximum lateness which is only concerned with the *maximum* gap between the due dates and completion times rather than the *total* of nonnegative gaps.

We now introduce some notation. Let $t_{i,k}$ denote the processing time of job i on machine k and let $LBt_{i,k}$ and $UBt_{i,k}$ denote the lower and upper bounds of $t_{i,k}$, respectively. Let n denote the number of jobs. Finally, let d_i denote the due date of job i . Various scheduling environments have been studied for the case when processing times are uncertain within some given upper and lower bounds. Allahverdi and Sotskov [8] investigated the case of $F2|Lt_{i,k} \leq t_{i,k} \leq Ut_{i,k}|C_{\max}$ where they present some dominance relations for the problem. Allahverdi *et al.* [10] provide heuristics for the problem of $F2|Lt_{i,k} \leq t_{i,k} \leq Ut_{i,k}, \text{ no-wait}|L_{\max}$, while Aydilek *et al.* [14] provide more effective heuristics for the same problem. Allahverdi and Allahverdi [4] propose twelve algorithms based on Johnson's algorithm for the case of four machines with the objective of minimizing the makespan, $F4|Lt_{i,k} \leq t_{i,k} \leq Ut_{i,k}|C_{\max}$. Allahverdi and Allahverdi [5], on the other hand, investigate the case of four machines with the objective of minimizing the total completion time, $F4|LBt_{i,m} \leq t_{i,m} \leq UBt_{i,m}|\sum C_i$. Scheduling with uncertain processing times, within lower and upper bounds, have also been considered for other shop scheduling environments, *e.g.*, Sotskov and Egorova [28], Sotskov *et al.* [29, 30], Arık [12], Abtahi and Sahraeian [1], Horng and Lin [21], and Xu *et al.* [36]. Allahverdi [3] surveys the scheduling problems with uncertain job descriptors within lower and upper bounds.

1.1. Problem description

In this paper, we study the stochastic problem of the two-machine flowshop with uncertain processing times (within upper and lower bounds) with the performance measure of total tardiness. That is, we consider the problem $F2|LBt_{i,n} \leq t_{i,n} \leq UBt_{i,n}|\sum T_i$. A dominance relation is provided and a total of 19 algorithms are proposed and compared with each other and with a random sequence of jobs. The dominance relation is established in the Section 3 and the algorithms are presented in Section 4. Computational experiments are given in Section 5 while concluding remarks are made in Section 6. Before we get into the addressed problem, we discuss some related papers in Section 2.

Before we proceed, we present a numerical example of the problem at hand. Suppose we have the following upper and lower bounds for the processing times for three jobs, $15 \leq t_{1,1} \leq 100$, $5 \leq t_{2,1} \leq 70$, $15 \leq t_{3,1} \leq 150$, $50 \leq t_{1,2} \leq 80$, $4 \leq t_{2,2} \leq 130$, and $30 \leq t_{3,2} \leq 200$ with due dates $d_1 = 100, d_2 = 50, d_3 = 150$. There are a total of six ways to arrange the jobs: (1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1). Given these upper and lower bounds, we take a look at three different instances.

Instance 1: $t_{1,1} = 40, t_{2,1} = 60, t_{3,1} = 70, t_{1,2} = 65, t_{2,2} = 89, t_{3,2} = 199$

Instance 2: $t_{1,1} = 50, t_{2,1} = 30, t_{3,1} = 80, t_{1,2} = 70, t_{2,2} = 77, t_{3,2} = 94$

Instance 3: $t_{1,1} = 43, t_{2,1} = 36, t_{3,1} = 22, t_{1,2} = 67, t_{2,2} = 50, t_{3,2} = 43$.

Table 1 gives the total tardiness for each instance.

As seen from Table 1, the best sequence for the first instance is (1, 2, 3), for the second instance is (2, 1, 3), and for the third one is (2, 3, 1). Hence, even for such a simple problem with only 3 jobs there is no optimal solution that solves this scheduling problem. Furthermore, these are just three instances that we look at in this example; there are an infinite number of instances and there is no way to know the best sequence for each instance. This illustrates the difficulty of this scheduling problem, especially in real world scenarios where the number of jobs is much larger than 3. Even for such a simple case, there is no known solution.

In this paper, we attempt to solve this problem through algorithms and theoretical means. In particular, along with a dominance relation we provide 19 algorithms to help solve this problem and compare these algorithms through statistical analysis to determine the most effective one.

TABLE 1. Numerical example.

Sequence	Total tardiness		
	Instance 1	Instance 2	Instance 3
(1, 2, 3)	392	308	173
(1, 3, 2)	512	345	166
(2, 1, 3)	476	255	135
(2, 3, 1)	610	285	132
(3, 1, 2)	726	439	164
(3, 2, 1)	750	446	147

2. BIBLIOGRAPHIC REVIEW

Papers addressing the problem of total tardiness include Framinan and Listen [19] who propose a greedy algorithm and Vallada and Ruiz [32] who propose a genetic algorithm for the flowshop scheduling problem with the objective of minimizing total tardiness. Moreover, Vallada *et al.* [33] present an evaluation and review of heuristics that exist in the literature for flowshop scheduling problems with the same objective function. For the case of the two-machine, Bożejko *et al.* [15] proposes new elimination properties for accelerating the local search algorithm. Furthermore, Saber and Ranjbar [25] propose a mixed-integer programming model for the permutation flowshop to minimize both total tardiness and total carbon emissions. They also present a multi-objective decomposition-based heuristic algorithm for the problem. Rakrouki *et al.* [24] propose five different algorithms, including the genetic algorithm, to minimize total tardiness for the two-machine flowshop problem with the unavailability periods of the machines. de Athayde Prata *et al.* [16] address the flowshop scheduling problem where the machines can operate at different speeds with the objective of minimizing total tardiness. They propose two heuristic algorithms which are based on mixed-integer linear programming formulation, while Wang *et al.* [35] present a heuristic for the distributed flowshop scheduling problem with sequence-dependent setup times to minimize total tardiness.

The papers mentioned above, and in fact the vast majority of scheduling literature, consider processing times to be deterministic, Keshavarz and Salmasi [22] and Seidgar *et al.* [26]. Though this may be the case in some manufacturing settings, it is not so in others, Wang and Choi [34] and Gonzalez-Neira *et al.* [20]. Uncertainty is an integral part of manufacturing and production, as a multitude of factors may cause processing times to change. These factors include the condition of available tools, machine operator fatigue, and disruption in the manufacturing environment. Other factors include the absence of past data used to predict processing times, the lack of experience, or untested processing tools, Tayanithi *et al.* [31]. Therefore, it is essential to consider cases when processing times are uncertain where only the upper and lower bounds are known in advance.

3. DOMINANCE RELATIONS

In this section, a dominance relation is established for the two-machine flowshop scheduling problem.

Let $t_{p,k}$ represent the processing time of job p on machine k ($k = 1, 2$). Let $t_{[p,k]}$ be the processing time of the job in position p on machine k for a given sequence. Likewise, let d_p and $d_{[p]}$ represent the due date of job p and the due date of the job in position p , respectively. Furthermore, let $T_{[p]}$ symbolize the tardiness of the job in position p . Processing times are uncertain and satisfy the inequality $LBt_{p,k} \leq t_{p,k} \leq UBt_{p,k}$, where $LBt_{p,k}$ and $UBt_{p,k}$ denote lower and upper bounds on the processing time $t_{p,k}$, respectively. Let

$$ST_{[p,k]} = \sum_{r=1}^p t_{[r,k]},$$

$$\delta_{[p]} = ST_{[p,1]} - ST_{[p-1,2]}$$

and

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

Thus, $ST_{[p,k]}$ is the sum of all processing times on machine k starting from position 1 up to position p , while $\delta_{[p]}$ is the difference between the sum of processing times on machines 1 and 2 starting from position 1 up to position p for machine 1 and $p - 1$ for machine 2. The maximum of $\delta_{[1]}, \dots, \delta_{[p]}$ is denoted by $\Delta_{[p]}$. Then, the tardiness of the job in position p [2] can be computed as

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

Hence, total tardiness (TT) is computed as

$$TT = \sum_{p=0}^n T_{[p]}.$$

Consider two job sequences π_1 and π_2 where the sequence π_1 has job i in an arbitrary position α and job j in an arbitrary position β , where $\alpha < \beta$. The sequence π_2 is derived from π_1 by interchanging the jobs in positions α and β (*i.e.*, job j is in position α and job i in position β). The sequences π_1 and π_2 are written as $\pi_1 = \{\sigma_1, i, \sigma_2, j, \sigma_3\}$ and $\pi_2 = \{\sigma_1, j, \sigma_2, i, \sigma_3\}$, where σ_1, σ_2 , and σ_3 are subsequences denoting the jobs in positions $1, \dots, \alpha - 1$, those in $\alpha + 1, \dots, \beta - 1$, and those in $\beta + 1, \dots, n$, respectively.

Theorem 1. $\delta_{[r]}(\pi_2) \leq \delta_{[r]}(\pi_1)$ for $r = 1, \dots, n$ if

- (a) $UBt_{j,1} \leq Lbt_{i,1}$ and
- (b) $Lbt_{j,2} \geq UBt_{i,2}$.

Proof. We look at different cases separately. We want to show that $\delta_{[r]}(\pi_1) - \delta_{[r]}(\pi_2) \geq 0 \forall r = 1, 2, \dots, n$.

Case 1. Let $r = 1, \dots, \alpha - 1$.

By definition, we have $\delta_{[r]}(\pi_2) = \delta_{[r]}(\pi_1)$.

Case 2. Let $r = \alpha$.

We have

$$\begin{aligned} \delta_{[\alpha]}(\pi_1) &= (t_{[1,1]} + t_{[2,1]} + \dots + t_{[\alpha,1]}) - (t_{[1,2]} + t_{[2,2]} + \dots + t_{[\alpha-1,2]}) \\ &= (t_{[1,1]} + t_{[2,1]} + \dots + t_{i,1}) - (t_{[1,2]} + t_{[2,2]} + \dots + t_{[\alpha-1,2]}) \end{aligned}$$

and

$$\begin{aligned} \delta_{[\alpha]}(\pi_2) &= (t_{[1,1]} + t_{[2,1]} + \dots + t_{[\alpha,1]}) - (t_{[1,2]} + t_{[2,2]} + \dots + t_{[\alpha-1,2]}) \\ &= (t_{[1,1]} + t_{[2,1]} + \dots + t_{j,1}) - (t_{[1,2]} + t_{[2,2]} + \dots + t_{[\alpha-1,2]}). \end{aligned}$$

So that $\delta_{[\alpha]}(\pi_1) - \delta_{[\alpha]}(\pi_2) = t_{i,1} - t_{j,1} \geq 0$ by the assumption $UBt_{j,1} \leq Lbt_{i,1}$.

Case 3. Let $r = \alpha + 1, \dots, \beta - 1$.

Here

$$\begin{aligned} \delta_{[r]}(\pi_1) &= (t_{[1,1]} + t_{[2,1]} + \dots + t_{[\alpha,1]} + \dots + t_{[r,1]}) - (t_{[1,2]} + t_{[2,2]} + \dots + t_{[r-1,2]}) \\ &= (t_{[1,1]} + t_{[2,1]} + \dots + t_{i,1} + \dots + t_{[r,1]}) - (t_{[1,2]} + t_{[2,2]} + \dots + t_{[\alpha-1,2]} + t_{i,2} + \dots + t_{[r-1,2]}) \end{aligned}$$

and

$$\begin{aligned} \delta_{[r]}(\pi_2) &= (t_{[1,1]} + t_{[2,1]} + \dots + t_{[\alpha,1]} + \dots + t_{[r,1]}) - (t_{[1,2]} + t_{[2,2]} + \dots + t_{[r-1,2]}) \\ &= (t_{[1,1]} + t_{[2,1]} + \dots + t_{j,1} + \dots + t_{[r,1]}) - (t_{[1,2]} + t_{[2,2]} + \dots + t_{[\alpha-1,2]} + t_{j,2} + \dots + t_{[r-1,2]}). \end{aligned}$$

Therefore, $\delta_{[\alpha]}(\pi_1) - \delta_{[\alpha]}(\pi_2) = t_{i,1} - t_{i,2} - t_{j,1} + t_{j,2} = (t_{i,1} - t_{j,1}) - (t_{i,2} - t_{j,2})$. Here, $(t_{i,1} - t_{j,1}) \geq 0$ by the assumption $UBt_{j,1} \leq Lbt_{i,1}$ and $(t_{i,2} - t_{j,2}) \leq 0$ by the assumption $Lbt_{j,2} \geq UBt_{i,2}$, so that $(t_{i,1} - t_{j,1}) - (t_{i,2} - t_{j,2}) \geq 0$.

Case 4. Let $r = \beta$.

Here

$$\begin{aligned}\delta_{[\alpha]}(\pi_1) &= (t_{[1,1]} + t_{[2,1]} + \dots + t_{[\alpha,1]} + \dots + t_{[\beta,1]}) - (t_{[1,2]} + t_{[2,2]} + \dots + t_{[\alpha,2]} + \dots + t_{[\beta-1,2]}) \\ &= (t_{[1,1]} + t_{[2,1]} + \dots + t_{i,1} + \dots + t_{j,1}) - (t_{[1,2]} + t_{[2,2]} + \dots + t_{[\alpha-1,2]} + t_{i,2} + \dots + t_{[\beta-1,2]})\end{aligned}$$

and

$$\begin{aligned}\delta_{[\alpha]}(\pi_2) &= (t_{[1,1]} + t_{[2,1]} + \dots + t_{[\alpha,1]} + \dots + t_{[\beta,1]}) - (t_{[1,2]} + t_{[2,2]} + \dots + t_{[\alpha,2]} + \dots + t_{[\beta-1,2]}) \\ &= (t_{[1,1]} + t_{[2,1]} + \dots + t_{j,1} + \dots + t_{i,1}) - (t_{[1,2]} + t_{[2,2]} + \dots + t_{[\alpha-1,2]} + t_{j,2} + \dots + t_{[\beta-1,2]}).\end{aligned}$$

Hence, $\delta_{[\alpha]}(\pi_1) - \delta_{[\alpha]}(\pi_2) = t_{i,1} + t_{j,1} - t_{i,2} - t_{j,2} = t_{j,2} - t_{i,2} \geq 0$ by the assumption $LBt_{j,2} \geq UBt_{i,2}$.

Case 5. Let $r = \beta + 1, \dots, n$ (where n is the length of π_1).

$$\begin{aligned}\delta_{[\alpha]}(\pi_1) &= (t_{[1,1]} + t_{[2,1]} + \dots + t_{[\alpha,1]} + \dots + t_{[\beta,1]} + \dots + t_{[r,1]}) \\ &\quad - (t_{[1,1]} + t_{[2,2]} + \dots + t_{[\alpha,1]} + \dots + t_{[\beta,1]} + \dots + t_{[r-1,2]}) \\ &= (t_{[1,1]} + t_{[2,1]} + \dots + t_{i,1} + \dots + t_{j,1} + \dots + t_{[r,1]}) \\ &\quad - (t_{[1,2]} + t_{[2,2]} + \dots + t_{i,2} + \dots + t_{j,2} + \dots + t_{[r-1,2]})\end{aligned}$$

and

$$\begin{aligned}\delta_{[\alpha]}(\pi_2) &= (t_{[1,1]} + t_{[2,1]} + \dots + t_{[\alpha,1]} + \dots + t_{[\beta,1]} + \dots + t_{[r,1]}) \\ &\quad - (t_{[1,1]} + t_{[2,2]} + \dots + t_{[\alpha,1]} + \dots + t_{[\beta,1]} + \dots + t_{[r-1,2]}) \\ &= (t_{[1,1]} + t_{[2,1]} + \dots + t_{j,1} + \dots + t_{i,1} + \dots + t_{[r,1]}) \\ &\quad - (t_{[1,2]} + t_{[2,2]} + \dots + t_{j,2} + \dots + t_{i,2} + \dots + t_{[r-1,2]}).\end{aligned}$$

Clearly, we have $\delta_{[\alpha]}(\pi_1) - \delta_{[\alpha]}(\pi_2) = 0$, and we are done. □

Theorem 2. *If the following three conditions are true,*

- (a) $UBt_{j,1} \leq LBt_{i,1}$.
- (b) $UBt_{i,2} \leq LBt_{j,2}$.
- (c) $UBt_{j,2} + 2d_i \leq LBt_{i,2} + 2d_j$.

Then $TT(\pi_2) \leq TT(\pi_1)$. Therefore, job j should precede job i in order to minimize total tardiness.

Proof. Note that $TT = \sum_{p=0}^n T_{[p]}$. As in the Theorem 1, we look at different cases separately. We will later combine them to get the result.

Case 1. $p = 1, \dots, \alpha - 1$.

Obviously, we have $T_{[p]}(\pi_2) = T_{[p]}(\pi_1)$ for $p = 1, \dots, \alpha - 1$.

Case 2. $p = \alpha$.

We have $T_{[\alpha]}(\pi_1) = \max\{0, ST_{[\alpha,2]} + \Delta_{[\alpha]} - d_{[\alpha]}\}$ and $T_{[\alpha]}(\pi_2) = \max\{0, ST_{[\alpha,2]} + \Delta_{[\alpha]} - d_{[\alpha]}\}$. It suffices to consider the case when $ST_{[\alpha,2]} + \Delta_{[\alpha]} - d_{[\alpha]}$ is positive for both $T_{[\alpha]}(\pi_1)$ and $T_{[\alpha]}(\pi_2)$. By the lemma, $\delta_{[r]}(\pi_2) \leq \delta_{[r]}(\pi_1)$ for $r = 1, \dots, n$, so that $\Delta_{[r]}(\pi_1) = \Delta_{[r]}(\pi_2)$ for $r = 1, \dots, n$. Therefore,

$$\begin{aligned}T_{[\alpha]}(\pi_1) - T_{[\alpha]}(\pi_2) &= [(t_{[1,2]} + t_{[2,2]} + \dots + t_{i,2}) + \Delta_{[\alpha]} - d_i] - [(t_{[1,2]} + t_{[2,2]} + \dots + t_{j,2}) + \Delta_{[\alpha]} - d_j] \\ &= (t_{i,2} - t_{j,2}) + (d_j - d_i).\end{aligned}$$

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

For each p , we have

$$\begin{aligned} T_{[p]}(\pi_1) - T_{[p]}(\pi_2) &= [(t_{[1,2]} + t_{[2,2]} + \dots + t_{i,2} + \dots + t_{[p,2]}) + \Delta_{[p]} - d_{[p]}] \\ &\quad - [(t_{[1,2]} + t_{[2,2]} + \dots + t_{j,2} + \dots + t_{[p,2]}) + \Delta_{[p]} - d_{[p]}] \\ &= (t_{i,2} - t_{j,2}). \end{aligned}$$

Case 4. $p = \beta$.

We have the following,

$$\begin{aligned} T_{[\beta]}(\pi_1) - T_{[\beta]}(\pi_2) &= [(t_{[1,2]} + t_{[2,2]} + \dots + t_{i,2} + \dots + t_{j,2}) + \Delta_{[\beta]} - d_i] \\ &\quad - [(t_{[1,2]} + t_{[2,2]} + \dots + t_{j,2} + \dots + t_{i,2}) + \Delta_{[\beta]} - d_j] \\ &= d_j - d_i. \end{aligned}$$

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

For each p , we have

$$\begin{aligned} T_{[p]}(\pi_1) - T_{[p]}(\pi_2) &= [(t_{[1,2]} + t_{[2,2]} + \dots + t_{i,2} + \dots + t_{j,2} + \dots + t_{[p,2]}) + \Delta_{[p]} - d_{[p]}] \\ &\quad - [(t_{[1,2]} + t_{[2,2]} + \dots + t_{j,2} + \dots + t_{i,2} + \dots + t_{[p,2]}) + \Delta_{[p]} - d_{[p]}] = 0. \end{aligned}$$

Finally, we can combine all these cases to obtain,

$$\begin{aligned} T(\pi_1) - T(\pi_2) &= (t_{i,2} - t_{j,2}) + (d_j - d_i) + w(t_{i,2} - t_{j,2}) + (d_j - d_i) \\ &= (w + 1)(t_{i,2} - t_{j,2}) + 2(d_j - d_i) \geq 0 \end{aligned}$$

by assumption c.

□

Notice that w denotes the number of jobs in σ_2 . Even if $w = 0$, *i.e.*, jobs i and j are adjacent, this inequality holds by condition c in the statement of Theorem 2. Of course, the inequality holds if $w > 1$.

4. ALGORITHMS

The problem of minimizing total tardiness in a single machine with deterministic job descriptors is known to be NP-hard, Du *et al.* [18]. It follows that this problem is certainly NP-hard for the case of 2 machines with stochastic job descriptors. Thus we aim to develop algorithms to minimize total tardiness for the problem in question.

To minimize the total tardiness, we compare the sequences obtained from 19 different algorithms, along with a random sequence. The algorithms are described in the following table. The sequence of jobs associated to each algorithm is obtained by ordering the associated list in Table 1 in nondecreasing order.

Steps for the algorithm

These are the steps used in the algorithm. Let n be the number of jobs and m the number of replications. For each n , do the following:

- (1) Create an empty list called `alg_list` with 20 empty subsets (one for each algorithm).
- (2) Create an empty list called `er_alg_list` with 20 empty subsets (one for each algorithm).
- (3) Perform these steps m times:
 - (a) Generate upper and lower bounds for processing times. For each job i , the upper bound $UBt_{i,k}$ was generated from values between 1 and 100 and the lower bound $LBt_{i,k}$ was generated from $[UBt_{i,k} - \delta, UBt_{i,k}]$ (where δ was uniformly generated from the interval $[0, 50]$).

TABLE 2. Description of each algorithm.

Algorithm	List
1	d_i
2	$d_i + LBt_{i,1} + LBt_{i,2}$
3	$d_i + LBt_{i,1} + UBt_{i,2}$
4	$d_i + 10(UBt_{i,1} + LBt_{i,2})$
5	$d_i + UBt_{i,1} + UBt_{i,2}$
6	$d_i + LBt_{i,1} + \frac{LBt_{i,2} + UBt_{i,2}}{2}$
7	$d_i + UBt_{i,1} + \frac{LBt_{i,2} + UBt_{i,2}}{2}$
8	$d_i + \frac{LBt_{i,1} + UBt_{i,1}}{2} + LBt_{i,2}$
9	$d_i + \frac{LBt_{i,1} + UBt_{i,1}}{2} + UBt_{i,2}$
10	$d_i + \frac{LBt_{i,1} + UBt_{i,1}}{2} + \frac{LBt_{i,2} + UBt_{i,2}}{2}$
11	$d_i - LBt_{i,1} - LBt_{i,2}$
12	$d_i - LBt_{i,1} - UBt_{i,2}$
13	$d_i - UBt_{i,1} - LBt_{i,2}$
14	$d_i - UBt_{i,1} - UBt_{i,2}$
15	$d_i - LBt_{i,1} - \frac{LBt_{i,2} + UBt_{i,2}}{2}$
16	$d_i - UBt_{i,1} - \frac{LBt_{i,2} + UBt_{i,2}}{2}$
17	$d_i - \frac{LBt_{i,1} + UBt_{i,1}}{2} - LBt_{i,2}$
18	$d_i - \frac{LBt_{i,1} + UBt_{i,1}}{2} - UBt_{i,2}$
19	$d_i - \frac{LBt_{i,1} + UBt_{i,1}}{2} - \frac{LBt_{i,2} + UBt_{i,2}}{2}$

- (b) Randomly pick processing times for machines 1 and 2 lying between the lower and upper bounds of the processing times.
- (c) Based on the upper and lower bounds of the processing times, generate due dates (a list of four due dates each corresponding to a combination of T and R , explained below).
- (d) For each due date combination (there are a total of four of them), do the following:
 - (1) Generate a sequence based on each algorithm.
 - (2) Create a list `seq_list` which contains the sequence obtained from EDD as its first element, followed by the sequences induced by the algorithms, followed by a random sequence.
 - (3) Create another list `tt_list` which stores the total tardiness for each sequence in `seq_list`.
 - (4) Finally, append the results to the algorithm list we created above, `alg_list` (so the total tardiness we got from the 1st algorithm is stored in the first sublist of `alg_list`, the second total tardiness associated with the 2nd algorithm is stored in the 2nd sublist, and so on).

- (5) Compute the errors of the algorithms as follows: $(TT_H - TT_min)/(TT_max - TT_min)$ where TT_H is the total tardiness of the algorithm, TT_min is the minimum total tardiness (that of the best algorithm), and TT_max is the greatest total tardiness (that of the worst algorithm).
- (6) Similarly, append the errors obtained to the list `er_alg_list`.
- (4) Now take the average of the errors obtained from all replications for each algorithm.

5. COMPUTATIONAL EXPERIMENTS

Computational experiments were carried out using the algorithms described above. To test the effectiveness of the algorithm for a wide range of cases, the number of jobs was varied from 100 to 1000 with an increment of 100. That is, the test was performed for $n = 100, 200, 300, \dots, 1000$.

Given a value of n , upper and lower bounds for job processing times were generated for each replication. The upper bounds $UBt_{i,k}$ were uniformly generated from the range $[1, 100]$. Then the lower bounds $LBt_{i,k}$ were generated from $[UBt_{i,k} - \delta, UBt_{i,k}]$, where δ was uniformly generated from the interval $[0, 50]$. Negative lower bounds were replaced by 1.

Once the upper $UBt_{i,k}$ and lower bounds $LBt_{i,k}$ of processing times were generated, an instance $t_{i,k}$ was generated between $LBt_{i,k}$ and $UBt_{i,k}$. Even though processing times are assumed to be uncertain, instances between $UBt_{i,k}$ and $LBt_{i,k}$ had to be generated to carry out computational experiments. Nonetheless, it would not be wise to generate the instances using only the uniform distribution, as it is possible that instances of processing times follow a different distribution. Accordingly, four different distributions were considered: uniform, positive linear, negative linear, and normal. Since the uniform and normal distributions are symmetric, while the positive and negative linear distributions are skewed, the considered distributions represent a wide range of distributions, Aydılek *et al.* [13].

Finally, due dates were generated based on the upper and lower processing times in the following way: Johnson's algorithm was applied to $LBt_{i,1}$ and $LBt_{i,2}$ to obtain a sequence, which was used to calculate the makespan C_{max} . Denote the result by $C_{max}(LB)$. Similarly, another makespan was obtained by applying Johnson's algorithm to $UBt_{i,1}$ and $UBt_{i,2}$, denoted by $C_{max}(UB)$. Then $C_{max}(AVG)$ was computed as the average of $C_{max}(LB)$ and $C_{max}(UB)$. Finally due dates were generated uniformly from the range $[C_{max}(AVG) \times (1 - T - \frac{R}{2}), C_{max}(AVG) \times (1 - T + \frac{R}{2})]$, where T is the tardiness factor and R is the relative range of the due dates, see Kim [23] and Allahverdi and Aydılek [6]. Such a method for generating due dates is standard in scheduling literature. The values for T and R are usually taken from the range 0 to 1. For the computations in this paper, various T and R values were tested to see which produced reasonable due dates. In particular, tests were performed to select two values a and b for T and two values c and d for R . The values of a, b, c, d were selected from the list $(0.1, 0.3, 0.5, 0.7, 0.9)$. Every combination of a, b, c, d (*i.e.*, $(T = a, R = c)$, $(T = a, R = d)$, $(T = b, R = c)$, $(T = b, R = d)$) values were checked, so that a total of $4 \times 3 \times 4 \times 3 = 144$ cases were analyzed. Finally, $T = 0.5, 0.7$ and $R = 0.3, 0.5$ were found to generate reasonable due dates.

Furthermore, during this process, it was discovered that algorithm 4 (alg 4) performed the best for different values of T and R . Therefore, once the T and R values were determined, attempts were made to improve algorithm 4 even further. Initially, the formula used for algorithm 4 was $d_i + (UBt_{i,1} + LBt_{i,2})$. The computations were carried out for the formulas $d_i + w(UBt_{i,1} + LBt_{i,2})$, where $w = 2, 3, 5, 7, 10$, and $w = 10$ was found to be the most effective. Accordingly, the formula associated with algorithm 4 was modified as $d_i + 10(UBt_{i,1} + LBt_{i,2})$.

The error of a given algorithm was calculated as

$$100 * \frac{TT(A) - \text{Min}(A)}{\text{Max}(A) - \text{Min}(A)}$$

where $TT(A)$ is the total tardiness of the algorithm, $\text{Min}(A)$ is the total tardiness of the best performing algorithm, and $\text{Max}(A)$ is that of the worst performing algorithm. This value for the error was previously used by Allahverdi and Aydılek [6]. Notice that by definition the worst algorithm gets an error of 100 while the best algorithm gets an error of zero.

For each value of n (10 different values are considered), a total of 50 replications were carried out so that the output is the average error of those 50 replications. Finally, the four different distributions were considered to randomly generate an instance based on the upper and lower bounds for the processing times. Therefore, a total of $10 \times 50 \times 4 = 2000$ cases were considered to test the algorithm.

5.1. Algorithm comparisons

The errors of the algorithms for the four combinations of T and R are given in Tables 3–6 for the case of the uniform distribution. The tables for the other three distributions (normal, positive linear, negative linear) are omitted to save space where only the overall average errors, over n , T , and R values, are given in Table 7. However, summaries of these tables are given in Figures 5–16.

Figures 1–4 summarize the errors given in Tables 3–6, respectively. As seen in Figure 1, the difference between the errors of the algorithms (alg 1–alg 19) and that of the random sequence (alg 20) is very large. In fact, the gap keeps increasing with increasing n values. Therefore, the random sequence was omitted in the rest of the figures (Figs. 2–16) to present a clearer comparison between the given algorithms. Furthermore, it can clearly be seen in Figure 1 that alg 2, alg 3, alg 5, alg 6, alg 7, alg 8, alg 9, alg 10 produce a very similar outcome. The same is true for alg 11–alg 19. This is true for all the figures, so that representatives (alg 9 and alg 19) were chosen from these two groups of algorithms along with alg 1, which is EDD, and alg 4 (the best performing algorithm) and the rest were omitted to simplify the figures for the remaining ones.

As mentioned previously and as seen in the figures, algorithm 4 performs the best among all the considered algorithms. Furthermore, it appears that alg 2 to alg 10 perform better than alg 1, which is based on EDD (earliest due date), while alg 11 through alg 19 perform worse.

The figures indicate that the 19 algorithms perform better for the T and R combinations $T = 0.5$, $R = 0.3$ and $T = 0.5$, $R = 0.5$ compared to $T = 0.7$, $R = 0.3$ and $T = 0.7$, $R = 0.5$. Nonetheless, the performance of the algorithms increases with increasing n values for all of the considered combinations.

The overall average percentage errors across the four distributions and the four combinations of T and R are 24.19, 21.29, 21.30, 0.01, 21.13, 21.29, 21.11, 21.20, 21.20, 21.20, 27.27, 27.26, 27.42, 27.41, 27.26, 27.42, 27.34, 27.33, 27.34, 99.78 for alg 1, alg 2, alg 3, alg 4, alg 5, alg 6, alg 7, alg 8, alg 9, alg 10, alg 11, alg 12, alg 13, alg 14, alg 15, alg 16, alg 18, alg 19, and alg 20, respectively. This shows that the error of the worst considered algorithm (alg 13) is 27.42 which reduces the error of the random sequence by more 72%, *i.e.*, $(99.78 - 27.42)/99.78$.

5.2. Statistical analysis

Statistical analysis of the algorithm comparisons have been conducted through confidence intervals and a test of hypothesis. The 95% and 99% confidence intervals of the algorithm errors are given in Table 8. Notice that the errors of alg 13 and alg 20 do not overlap, indicating that alg 13 performs much better than alg 20. Similarly, the errors of alg 4 and alg 20 do not overlap, and neither those of alg 4 and alg 13 or those of alg 4 and alg 9. Hence, alg 4 is statistically the best performing algorithm.

Test of hypotheses have been conducted through a two-sample t test. The following null (H_0) and alternative (H_1) hypotheses testing were performed.

$$\begin{aligned} H_0: \mu(\text{alg } 13) &= \mu(\text{alg } 20) \\ H_1: \mu(\text{alg } 13) &< \mu(\text{alg } 20) \end{aligned}$$

where $\mu(\text{alg } k)$ denotes the overall average error of the algorithm alg k . The null hypothesis was rejected at a significance level of $\alpha = 0.01$.

On the other hand, the best algorithm (alg 4) reduces the error of the worst algorithm (alg 20) by 99.99%, $(99.78 - 0.01)/99.78$. This result was statistically confirmed by conducting the following two hypotheses:

$$\begin{aligned} H_0: \mu(\text{alg } 4) &= \mu(\text{alg } 20) \\ H_1: \mu(\text{alg } 4) &< \mu(\text{alg } 20). \end{aligned}$$

TABLE 5. Errors of the algorithms for uniform distribution ($T = 0.7$, $R = 0.3$).

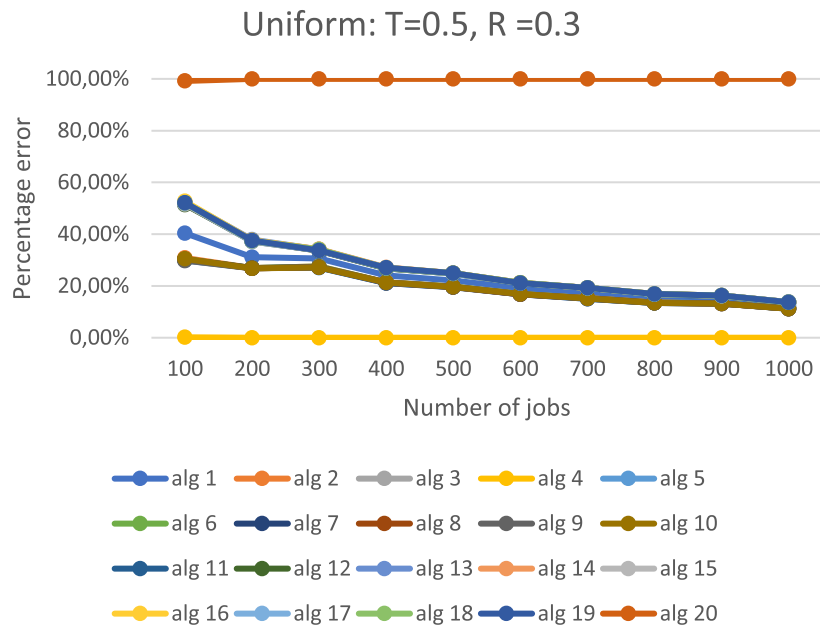
[illegible]

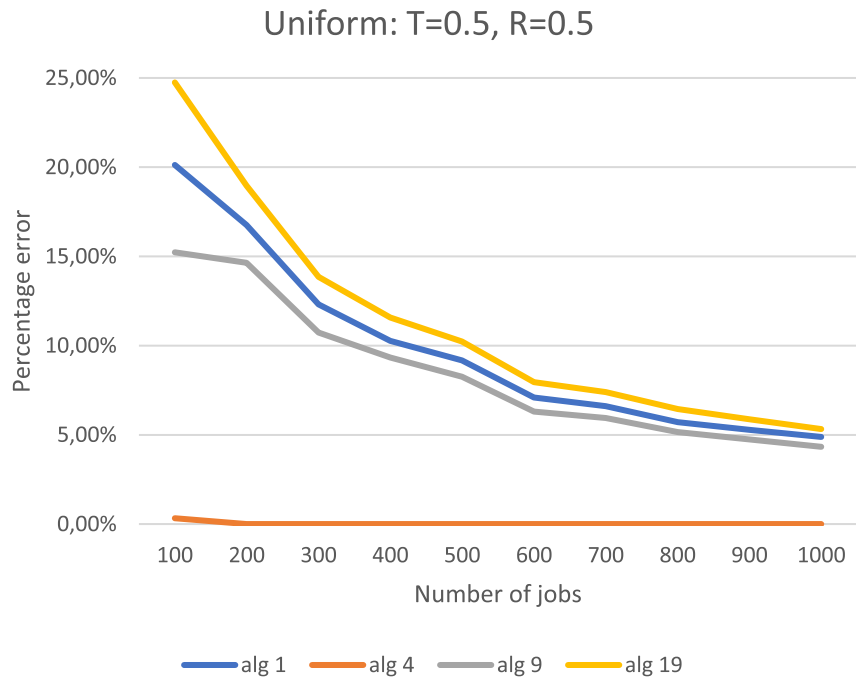
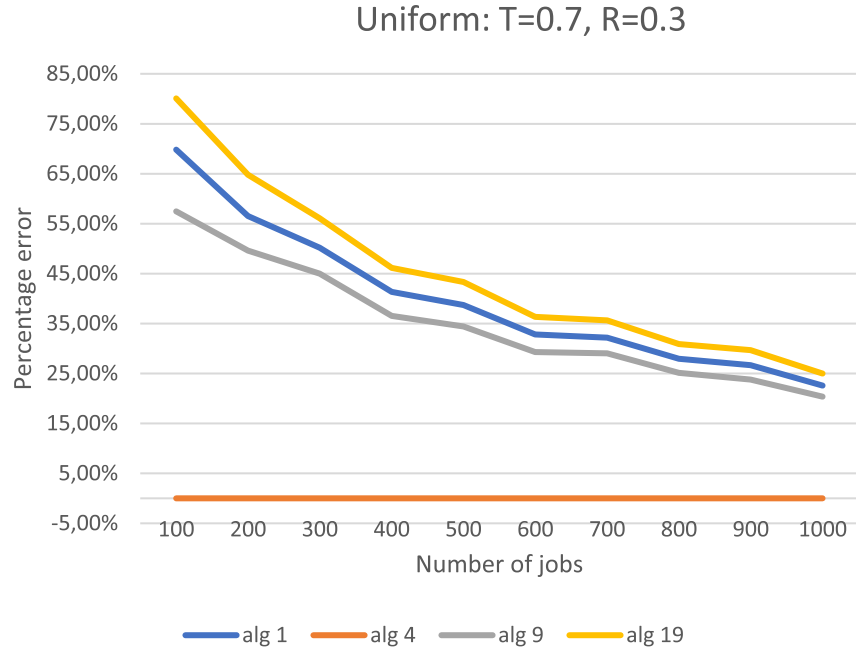
TABLE 6. Errors of the algorithms for uniform distribution ($T = 0.7$, $R = 0.5$).

[illegible]

TABLE 7. Errors of the algorithms for all the distributions over n , T , and R .

Algorithms	Uniform	Positive Linear	Negative Linear	Normal	Average
alg 1	24.21	25.67	22.65	24.22	24.19
alg 2	21.23	22.74	19.90	21.28	21.29
alg 3	21.23	22.74	19.92	21.29	21.30
alg 4	0.01	0.02	0.01	0.00	0.01
alg 5	21.07	22.50	19.83	21.11	21.13
alg 6	21.22	22.76	19.89	21.29	21.29
alg 7	21.07	22.51	19.77	21.08	21.11
alg 8	21.17	22.62	19.83	21.19	21.20
alg 9	21.15	22.61	19.86	21.18	21.20
alg 10	21.15	22.62	19.84	21.19	21.20
alg 11	27.34	28.79	25.63	27.32	27.27
alg 12	27.31	28.77	25.62	27.34	27.26
alg 13	27.47	29.00	25.73	27.47	27.42
alg 14	27.45	28.98	25.74	27.48	27.41
alg 15	27.33	28.76	25.61	27.33	27.26
alg 16	27.47	28.99	25.73	27.48	27.42
alg 17	27.41	28.91	25.68	27.38	27.34
alg 18	27.38	28.86	25.66	27.43	27.33
alg 19	27.39	28.86	25.68	27.41	27.34
alg 20	99.75	99.85	99.82	99.68	99.78

FIGURE 1. Errors of the algorithms for $T = 0.5$ and $R = 0.3$ – Uniform Distribution.

FIGURE 2. Errors of the four algorithms for $T = 0.5$ and $R = 0.5$ – Uniform Distribution.FIGURE 3. Errors of the four algorithms for $T = 0.7$ and $R = 0.3$ – Uniform Distribution.

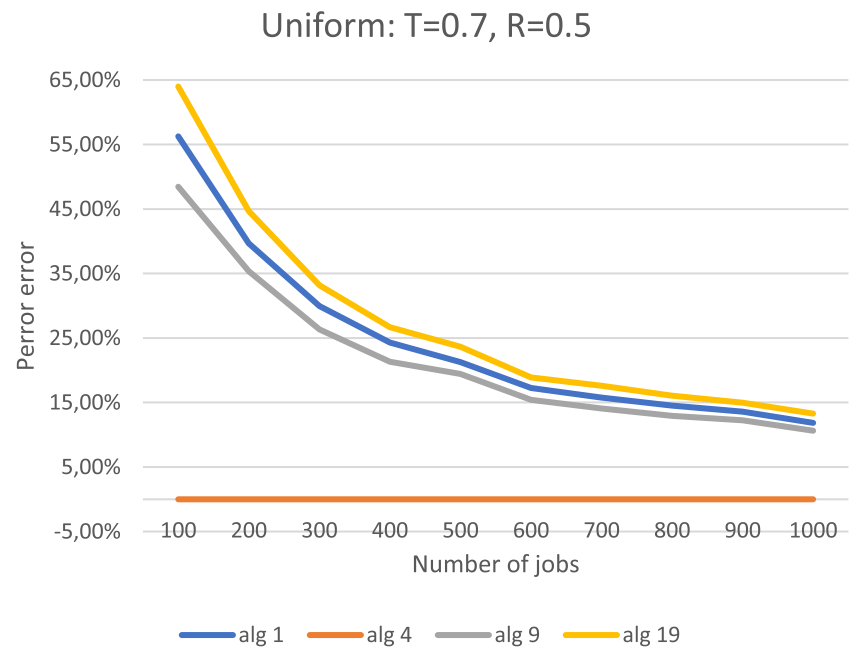


FIGURE 4. Errors of the four algorithms for $T = 0.7$ and $R = 0.5$ – Uniform Distribution.

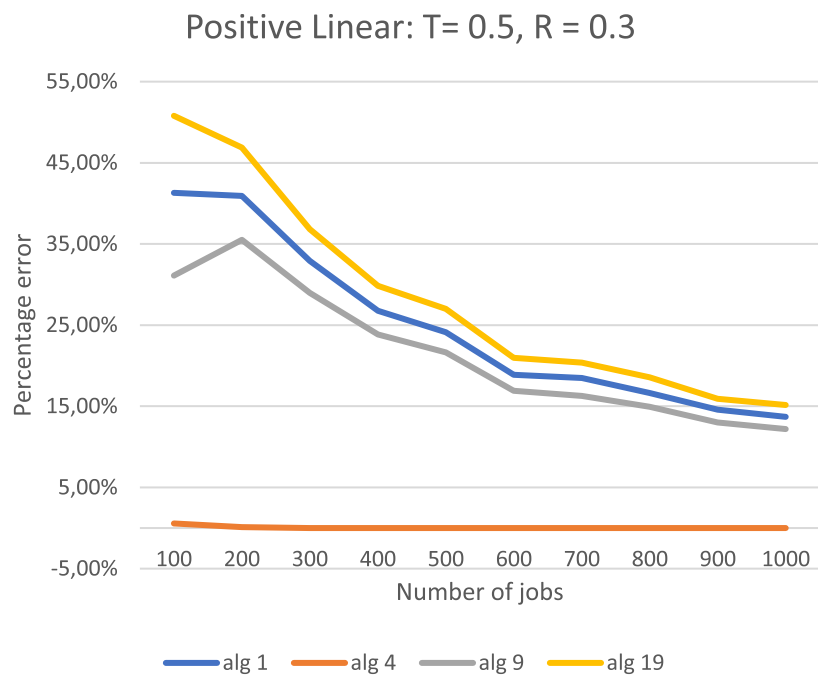
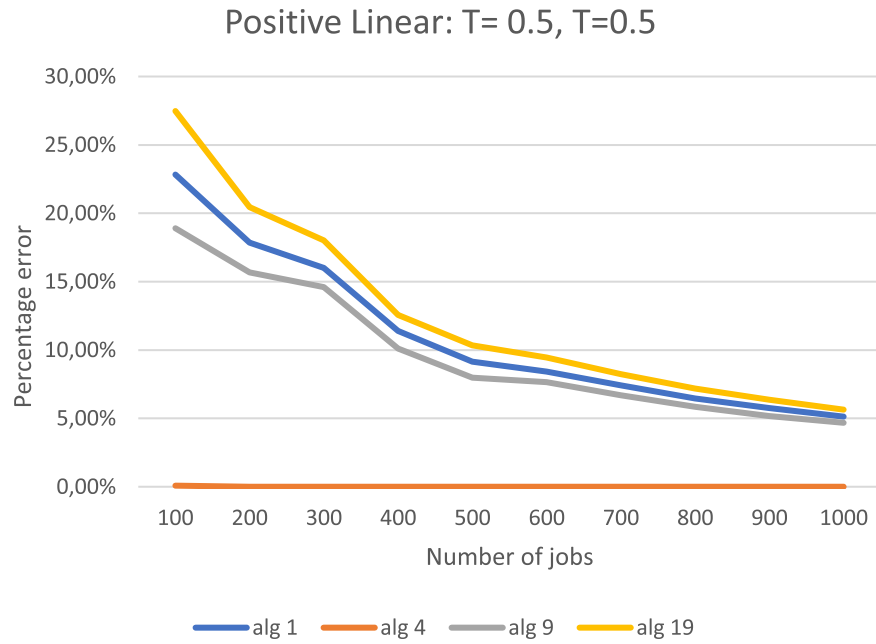
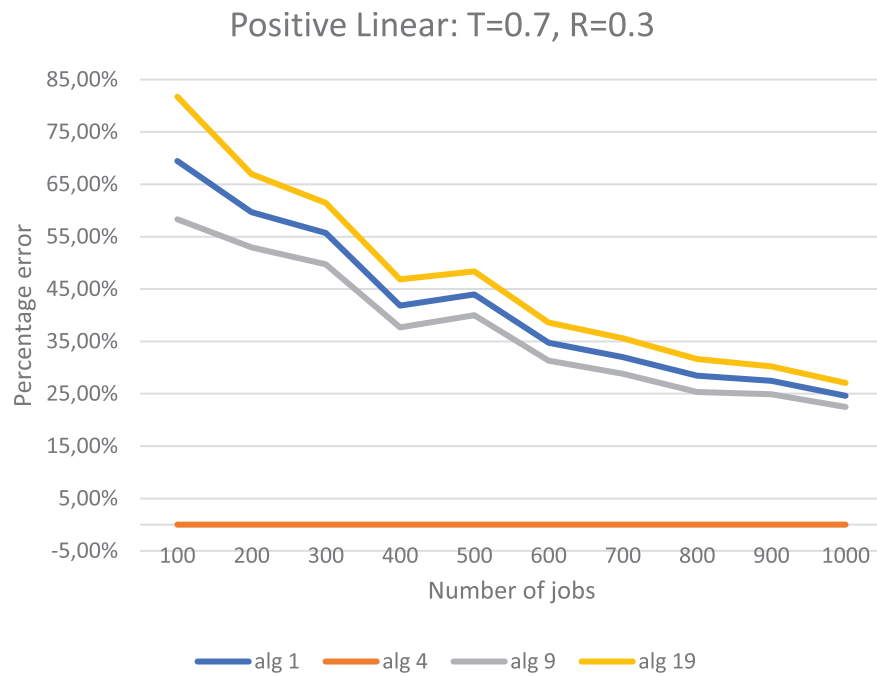


FIGURE 5. Errors of the four algorithms for $T = 0.5$ and $R = 0.3$ – Positive Linear Distribution.

FIGURE 6. Errors of the four algorithms for $T = 0.5$ and $R = 0.5$ – Positive Linear Distribution.FIGURE 7. Errors of the four algorithms for $T = 0.7$ and $R = 0.3$ – Positive Linear Distribution.

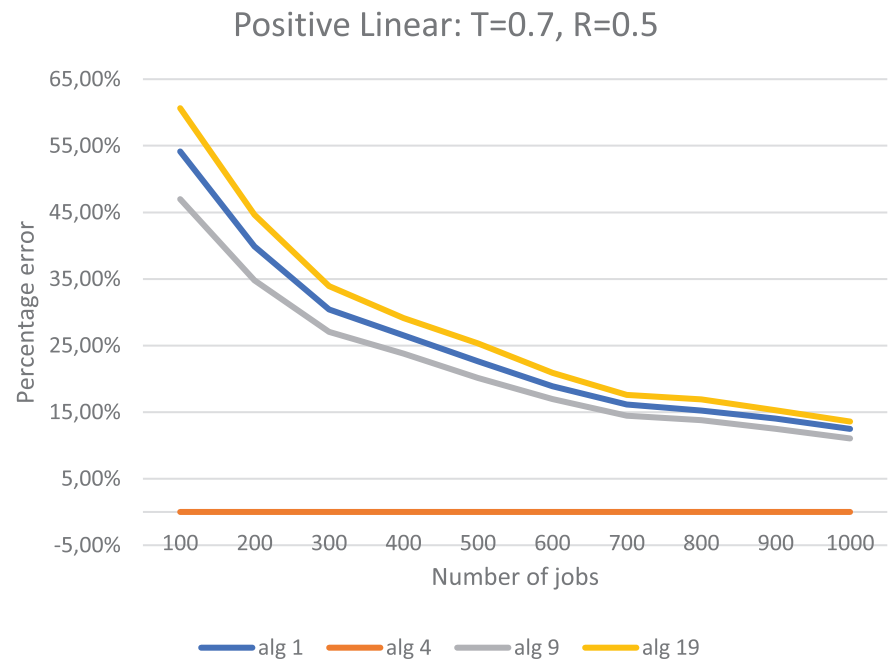


FIGURE 8. Errors of the four algorithms for $T = 0.7$ and $R = 0.5$ – Positive Linear Distribution.

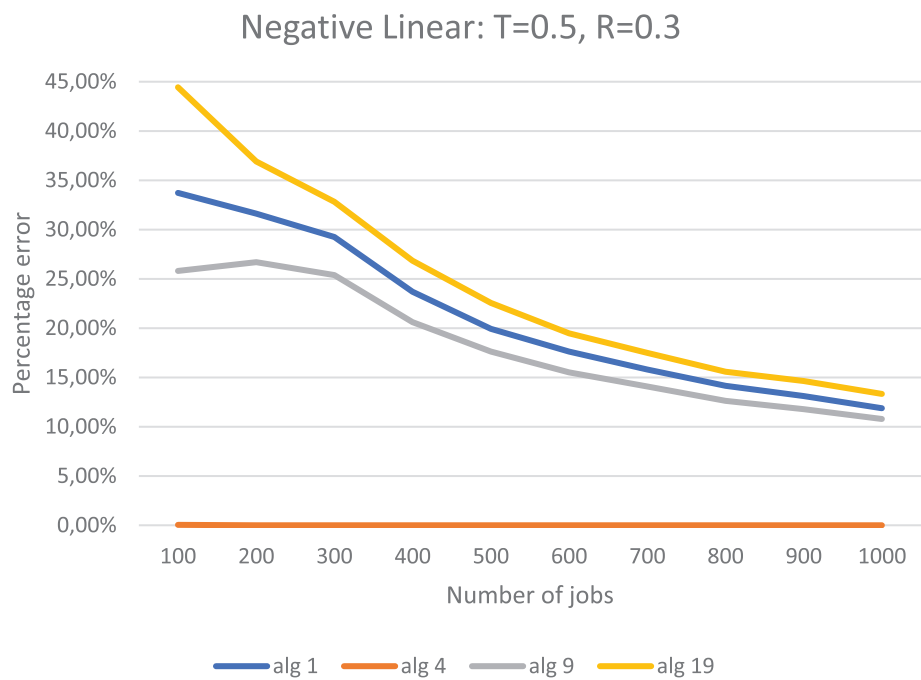
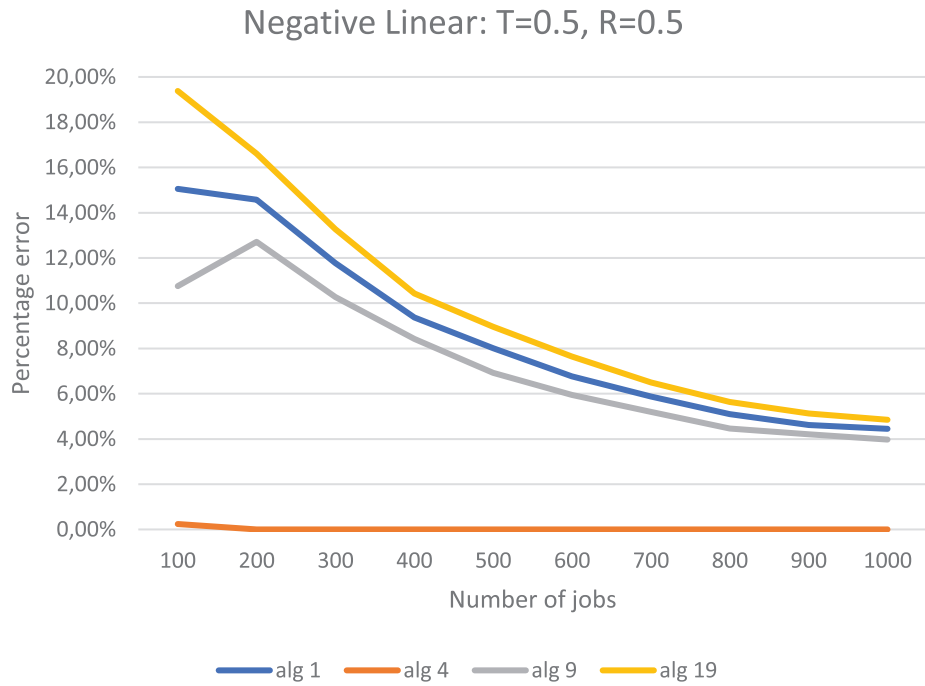
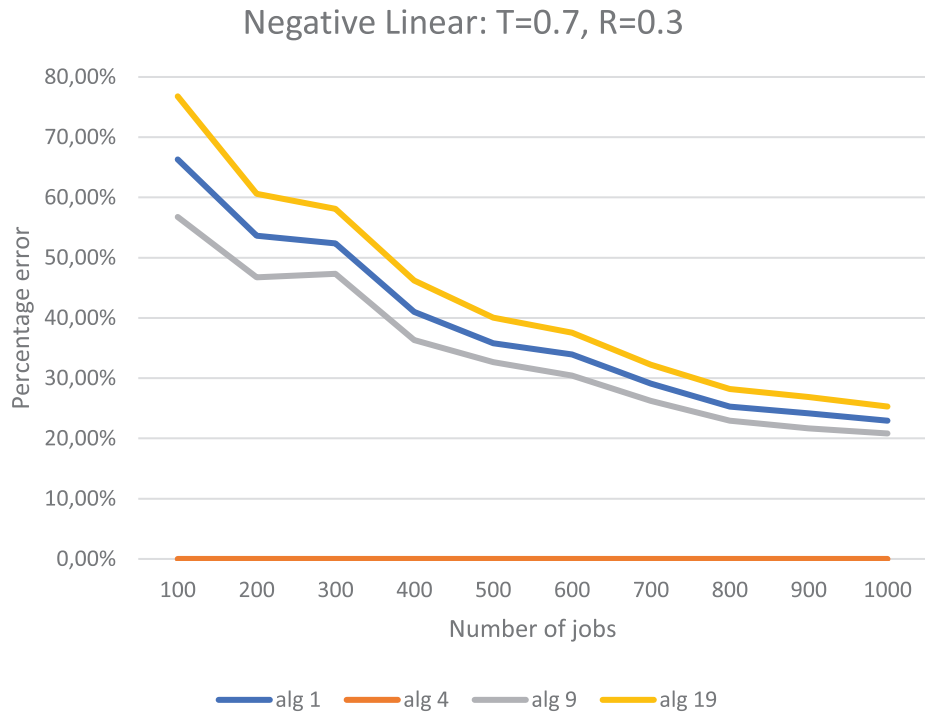


FIGURE 9. Errors of the four algorithms for $T = 0.5$ and $R = 0.3$ – Negative Linear Distribution.

FIGURE 10. Errors of the four algorithms for $T = 0.5$ and $R = 0.5$ – Negative Linear Distribution.FIGURE 11. Errors of the four algorithms for $T = 0.7$ and $R = 0.3$ – Negative Linear Distribution.

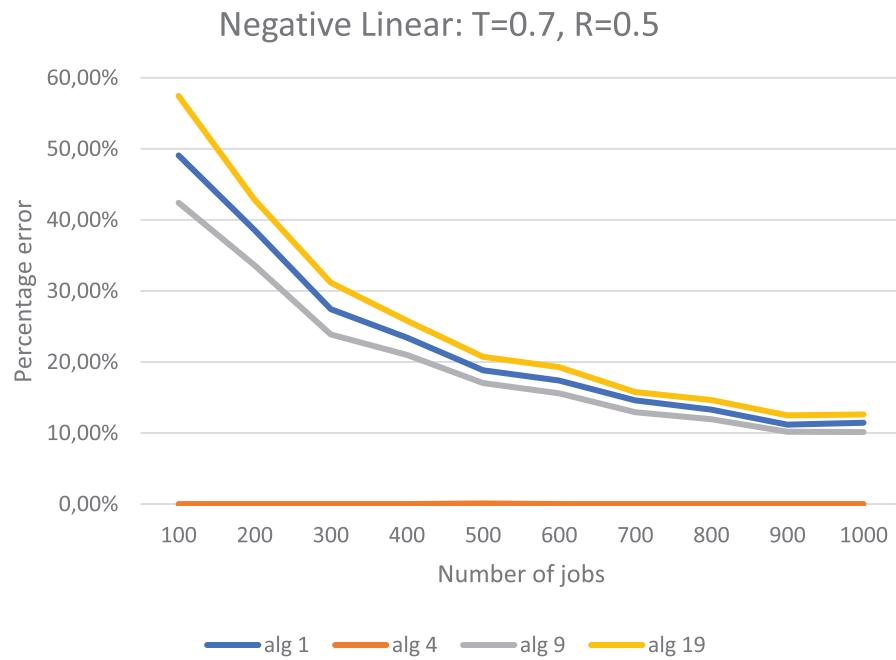


FIGURE 12. Errors of the four algorithms for $T = 0.7$ and $R = 0.5$ – Negative Linear Distribution.

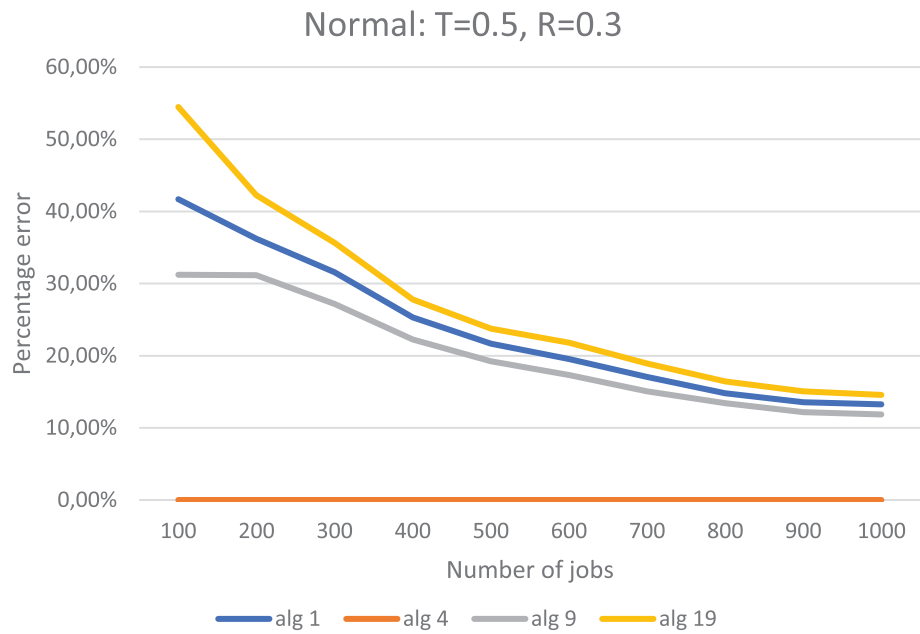
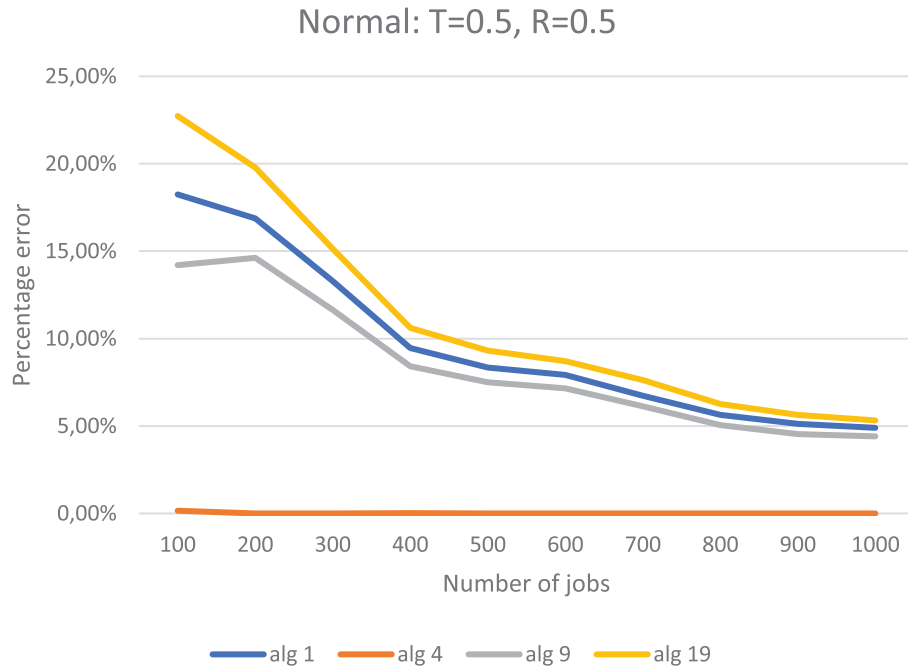
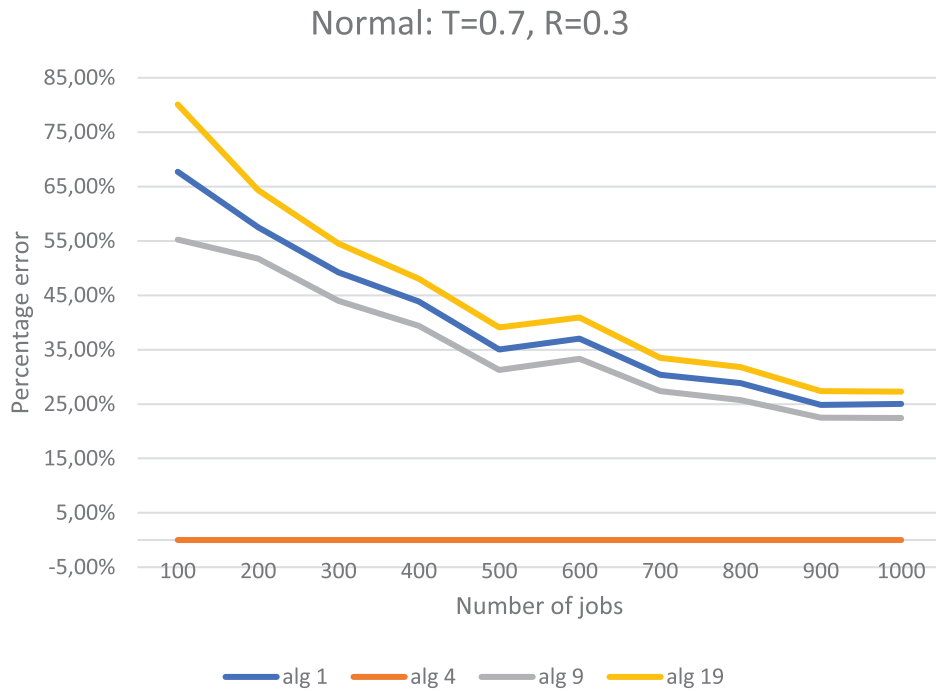


FIGURE 13. Errors of the four algorithms for $T = 0.5$ and $R = 0.3$ – Normal Distribution.

FIGURE 14. Errors of the four algorithms for $T = 0.5$ and $R = 0.5$ – Normal Distribution.FIGURE 15. Errors of the four algorithms for $T = 0.7$ and $R = 0.3$ – Normal Distribution.

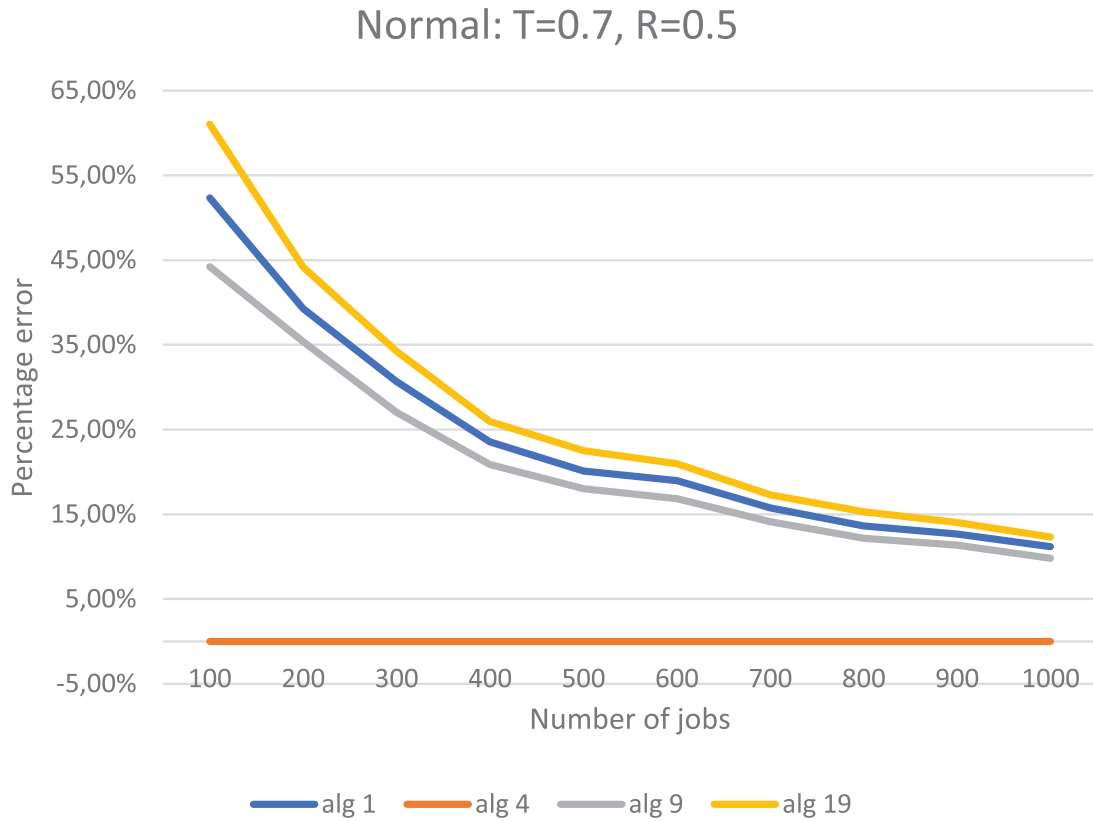


FIGURE 16. Errors of the four algorithms for $T = 0.7$ and $R = 0.5$ – Normal Distribution.

The null hypothesis was rejected at a significance level of $\alpha = 0.01$.

Finally, alg 4 reduces the error of alg 13 (the worst algorithm among alg 1 – alg 19) by 99.96%, $(27.42 - 0.01)/27.42$. This result is statistically confirmed by conducting the following two hypotheses:

$$\begin{aligned} H_0: \mu(\text{alg 4}) &= \mu(\text{alg 13}) \\ H_1: \mu(\text{alg 4}) &< \mu(\text{alg 13}). \end{aligned}$$

Thus, the null hypothesis is rejected at a significance level of $\alpha = 0.01$.

Finally, the performance of alg 4 and alg 9 is statistically compared by using a two-sample t test, since alg 9 is compared to alg 4 in all the figures. As seen in the figures, the performance of alg 9 is the closest to alg 4 among the considered algorithms. The following H_0 and H_1 hypotheses testing are performed.

$$\begin{aligned} H_0: \mu(\text{alg 4}) &= \mu(\text{alg 9}) \\ H_1: \mu(\text{alg 4}) &< \mu(\text{alg 9}). \end{aligned}$$

The null hypothesis is rejected at a significance level of $\alpha = 0.01$ and alg 4 is recommended.

6. CONCLUSION

This paper studies the case of a two-machine flowshop where the processing times are uncertain, within some upper and lower bounds. The objective is to find the sequence that minimizes the total tardiness $(F2|LBt_{i,n} \leq t_{i,n} \leq UBt_{i,n}|\sum T_i)$.

TABLE 8. 95% and 99% confidence intervals of the errors of the algorithms.

Algorithms	95% CI	(LB, UB)	99% CI	(LB, UB)
alg 1	(24.01773, 24.30916)		(24.03016, 24.34710)	
alg 2	(21.12659, 21.40045)		(21.13827, 21.43610)	
alg 3	(21.13588, 21.40909)		(21.14754, 21.44466)	
alg 4	(0.010370, 0.011810)		(0.010430, 0.011990)	
alg 5	(20.97435, 21.23245)		(20.98536, 21.26605)	
alg 6	(21.12835, 21.40593)		(21.14019, 21.44207)	
alg 7	(20.95210, 21.21710)		(20.96340, 21.25160)	
alg 8	(21.04310, 21.31196)		(21.05456, 21.34696)	
alg 9	(21.04745, 21.31355)		(21.05881, 21.34819)	
alg 10	(21.04105, 21.30950)		(21.05250, 21.34445)	
alg 11	(27.09158, 27.39641)		(27.10458, 27.43609)	
alg 12	(27.08200, 27.38643)		(27.09498, 27.42606)	
alg 13	(27.23286, 27.54846)		(27.24632, 27.58954)	
alg 14	(27.22809, 27.54195)		(27.24148, 27.58280)	
alg 15	(27.07908, 27.38327)		(27.09206, 27.42287)	
alg 16	(27.23390, 27.54828)		(27.24731, 27.58921)	
alg 17	(27.16226, 27.47412)		(27.17556, 27.51472)	
alg 18	(27.15113, 27.46072)		(27.16434, 27.50102)	
alg 19	(27.15611, 27.46422)		(27.16925, 27.50433)	
alg 20	(99.76598, 99.78379)		(99.76674, 99.78611)	

A dominance relation is established and nineteen algorithms (alg 1–alg 19) are proposed. The algorithms are compared with each other and with a random solution, denoted as alg 20. The proposed algorithms were tested by using four different distributions for generating processing times: uniform, positive linear, negative linear, and normal distributions, representing both symmetric and skewed distributions.

Computational experiments indicate that the worst algorithm, among the proposed nineteen algorithms, reduces the error of a random solution by more than 72%. They also indicated that the best algorithm (alg 4) reduces the error of the worst algorithm (among the proposed nineteen algorithms) by 99.96%. The results are statistically confirmed. Therefore, alg 4 is recommended for the considered problem.

In this paper, setup times were assumed to be included within processing times. However, this assumption may not be valid for certain scheduling environments. Therefore, it would be interesting to consider the problem of minimizing total tardiness for a two machine flowshop with uncertain setup times for future research. Furthermore, it would be interesting to consider the problem of a three machine flowshop with uncertain processing times with the objective to minimize total tardiness.

REFERENCES

- [1] Z. Abtahi and R. Sahraeian, Robust and stable flow shop scheduling problem under uncertain processing times and machines' disruption. *Int. J. Eng.* **34** (2021) 935–947.
- [2] A. Allahverdi, The tricriteria two-machine flowshop scheduling problem. *Int. Trans. Oper. Res.* **8** (2001) 403–425.
- [3] A. Allahverdi, A survey of scheduling problems with uncertain interval/bounded processing/setup times. *J. Project Manage.* **7** (2022) 255–264.
- [4] M. Allahverdi and A. Allahverdi, Algorithms for four-machine flowshop scheduling problem with uncertain processing times to minimize makespan. *RAIRO: Oper. Res.* **54** (2020) 529–553.
- [5] M. Allahverdi and A. Allahverdi, Minimizing total completion time for flowshop scheduling problem with uncertain processing times. *RAIRO: Oper. Res.* **55** (2021) S929–S946.
- [6] A. Allahverdi and H. Aydilek, Heuristics for the two-machine flowshop scheduling problem to minimize maximum lateness with bounded processing times. *Comput. Math. App.* **60** (2010) 1374–1384.
- [7] A. Allahverdi and H. Aydilek, The two stage assembly flowshop scheduling problem to minimize total tardiness. *J. Intell. Manuf.* **26** (2015) 225–237.

- [8] A. Allahverdi and Y. Sotskov, Two-machine flowshop minimum-length scheduling problem with random and bounded processing times. *Int. Trans. Oper. Res.* **10** (2003) 65–76.
- [9] A. Allahverdi, H. Aydilek and A. Aydilek, Two-stage assembly scheduling problem for minimizing total tardiness with setup times. *Appl. Math. Modell.* **40** (2016) 7796–7815.
- [10] M. Allahverdi, H. Aydilek, A. Aydilek and A. Allahverdi, A better dominance relation and heuristics for two-machine no-wait flowshops with maximum lateness performance measure. *J. Ind. Manage. Optim.* **17** (2021) 1973.
- [11] S. Arabameri and N. Salmasi, Minimization of weighted earliness and tardiness for no-wait sequence-dependent setup times flowshop scheduling problem. *Comput. Ind. Eng.* **64** (2013) 902–916.
- [12] O.A. Arik, Single machine earliness/tardiness scheduling problem with grey processing times and the grey common due date. *Grey Syst.: Theory App.* **11** (2021) 95–109.
- [13] A. Aydilek, H. Aydilek and A. Allahverdi, Algorithms for minimizing the number of tardy jobs for reducing production cost with uncertain processing times. *Appl. Math. Modell.* **45** (2017) 982–996.
- [14] H. Aydilek, A. Aydilek, M. Allahverdi and A. Allahverdi, More effective heuristics for a two-machine no-wait flowshop to minimize maximum lateness. *Int. J. Ind. Eng. Comput.* **13** (2022) 543–556.
- [15] W. Bożejko, M. Uchroński and M. Wodecki, Blocks for two-machines total weighted tardiness flow shop scheduling problem. *Bull. Polish Acad. Sci. Tech. Sci.* **68** (2020) 31–41.
- [16] B. de Athayde Prata, V. Fernandez-Viagas, J.M. Framinan and C.D. Rodrigues, Matheuristics for the flowshop scheduling problem with controllable processing times and limited resource consumption to minimize total tardiness. *Comput. Oper. Res.* **145** (2022) 105880.
- [17] J. Ding, S. Song, R. Zhang, J.N. Gupta and C. Wu, Accelerated methods for total tardiness minimisation in no-wait flowshops. *Int. J. Prod. Res.* **53** (2015) 1002–1018.
- [18] J. Du and J.Y.T. Leung, Minimizing total tardiness on one machine is NP-hard. *Math. Oper. Res.* **15** (1990) 483–495.
- [19] J.M. Framinan and R. Leisten, Total tardiness minimization in permutation flow shops: a simple approach based on a variable greedy algorithm. *Int. J. Prod. Res.* **46** (2008) 6479–6498.
- [20] E.M. Gonzalez-Neira, D. Ferone, S. Hatami and A.A. Juan, A biased-randomized simheuristic for the distributed assembly permutation flowshop problem with stochastic processing times. *Simul. Modell. Pract. Theory* **79** (2017) 23–36.
- [21] S.C. Horng and S.S. Lin, Apply ordinal optimization to optimize the job-shop scheduling under uncertain processing times. *Arabian J. Sci. Eng.* **47** (2022) 9659–9671.
- [22] T. Keshavarz and N. Salmasi, Makespan minimisation in flexible flowshop sequence-dependent group scheduling problem. *Int. J. Prod. Res.* **51** (2013) 6182–6193.
- [23] Y.D. Kim, A new branch and bound algorithm for minimizing mean tardiness in two-machine flowshops. *Comput. Oper. Res.* **20** (1993) 391–401.
- [24] M.A. Rakrouki, A. Aljohani and T. Ladhari, Minimizing total tardiness in a two-machine flowshop scheduling problem with availability constraints. *Intell. Autom. Soft Comput.* **35** (2023) 1119–1134.
- [25] R.G. Saber and M. Ranjbar, Minimizing the total tardiness and the total carbon emissions in the permutation flow shop scheduling problem. *Comput. Oper. Res.* **138** (2022) 105604.
- [26] H. Seidgar, M. Kiani, M. Abedi and H. Fazlollahtabar, An efficient imperialist competitive algorithm for scheduling in the two-stage assembly flow shop problem. *Int. J. Prod. Res.* **52** (2014) 1240–1256.
- [27] D.K. Seo, C.M. Klein and W. Jang, Single machine stochastic scheduling to minimize the expected number of tardy jobs using mathematical programming models. *Comput. Ind. Eng.* **48** (2005) 153–161.
- [28] Y.N. Sotskov and N.G. Egorova, The optimality region for a single-machine scheduling problem with bounded durations of the jobs and the total completion time objective. *Mathematics* **7** (2019) 382.
- [29] Y.N. Sotskov, N.M. Matsveichuk and V.D. Hatsura, Two-machine job-shop scheduling problem to minimize the makespan with uncertain job durations. *Algorithms* **13** (2019) 4.
- [30] Y.N. Sotskov, N.M. Matsveichuk, V.D. Hatsura, Schedule execution for two-machine job-shop to minimize makespan with uncertain processing times. *Mathematics* **8** (2020) 1314.
- [31] P. Tayanithi, S. Manivannan and J. Banks, A knowledge-based simulation architecture to analyze interruptions in a flexible manufacturing system. *J. Manuf. Syst.* **11** (1992) 195–214.
- [32] E. Vallada and R. Ruiz, Genetic algorithms with path relinking for the minimum tardiness permutation flowshop problem. *Omega* **38** (2010) 57–67.
- [33] E. Vallada, R. Ruiz and G. Minella, Minimising total tardiness in the m-machine flowshop problem: a review and evaluation of heuristics and metaheuristics. *Comput. Oper. Res.* **35** (2008) 1350–1373.
- [34] K. Wang and S.H. Choi, A decomposition-based approach to flexible flow shop scheduling under machine breakdown. *Int. J. Prod. Res.* **50** (2012) 215–234.
- [35] Z.Y. Wang, Q.K. Pan, L. Gao and Y.L. Wang, An effective two-stage iterated greedy algorithm to minimize total tardiness for the distributed flowshop group scheduling problem. *Swarm Evol. Comput.* **74** (2022) 101143.
- [36] W. Xu, W. Wu, Y. Wang, Y. He and Z. Lei, Flexible job-shop scheduling method based on interval grey processing time. *Appl. Intell.* (2022) 1–16. DOI: [10.1007/s10489-022-04213-9](https://doi.org/10.1007/s10489-022-04213-9).

**Please help to maintain this journal in open access!**

This journal is currently published in open access under the Subscribe to Open model (S2O). We are thankful to our subscribers and supporters for making it possible to publish this journal in open access in the current year, free of charge for authors and readers.

Check with your library that it subscribes to the journal, or consider making a personal donation to the S2O programme by contacting subscribers@edpsciences.org.

More information, including a list of supporters and financial transparency reports, is available at <https://edpsciences.org/en/subscribe-to-open-s2o>.