

Minimizing total tardiness in a two-machine flowshop with uncertain and bounded processing times

Muberra Allahverdi

Department of Mathematics

Faculty of Engineering and Natural Sciences,

Ankara Yıldırım Beyazıt University, Ankara, Turkey

e-mail: mu.allahverdi@gmail.com

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. 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.

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

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 (2015). Allahverdi and Aydilek (2015) state that the total tardiness cost may also include the loss of goodwill, penalty cost in contracts, and damaged reputation. Allahverdi et al. (2016) point out that meeting customer due dates is a main concern for some manufacturing systems, while Ding et al. (2015) 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. (2005) or petroleum refineries, Arabameri and Salmasi (2013).

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 (2003) investigated the case of $F2 \mid Lt_{i,k} \leq t_{i,k} \leq Ut_{i,k} \mid C_{max}$ where they present some dominance relations for the problem. Allahverdi et al. (2021) provide heuristics for the problem of $F2 \mid Lt_{i,k} \leq t_{i,k} \leq Ut_{i,k}, no - wait \mid L_{max}$, while Aydilek et al. (2022) provide more effective heuristics for the same problem. Allahverdi and Allahverdi (2020) propose twelve algorithms based on Johnson's algorithm for the case of four machines with the objective of minimizing the makespan, $F4 \mid Lt_{i,k} \leq t_{i,k} \leq Ut_{i,k} \mid C_{max}$. Allahverdi and Allahverdi (2021), on the other hand, investigate the case of four machines with the objective of minimizing

the total completion time, $F4 \mid LBt_{i,m} \leq t_{i,m} \leq UBt_{i,m} \mid \sum C_i$. Scheduling with uncertain processing times, within lower and upper bounds, have also been considered for other shop scheduling environments, e.g., Sotskov and Egorova (2019), Sotskov et al. (2019, 2020), Arik (2020), Abtahi and Sahraeian (2021), Horng and Lin (2022), and Xu et al. (2022). Allahverdi (2022) 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 $F2 \mid LBt_{i,n} \leq t_{i,n} \leq UBt_{i,n} \mid \sum T_i$. 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 below gives the total tardiness for each instance

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

Table 1: Numerical Example

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 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.

2. Bibliographic Review

Papers addressing the problem of total tardiness include Framinan and Listen (2008) who propose a greedy algorithm and Vallada and Ruiz (2010) who propose a genetic algorithm for the flowshop scheduling problem with the objective of minimizing total tardiness. Moreover, Vallada et al. (2008) 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, Bozejko et al. (2020) proposes new elimination properties for accelerating the local search algorithm. Furthermore, Saber and Ranjbar (2022) 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. (2023) 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. (2022) 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. (2022) 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 (2013) and Seidgar et al (2014). Though this may be the case in some manufacturing settings, it is not so in others, Wang and Choi (2012) and Gonzalez-Neira et al (2017). 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 (1992). 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 (Allahverdi, 2001) 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

$$\delta_{[\alpha]}(\pi_2) = (t_{[1,1]} + t_{[2,1]} + \dots + t_{[\alpha,1]}) - (t_{[1,2]} + t_{[2,2]} + \dots + t_{[\alpha-1,2]})$$

$$= (t_{[1,1]} + t_{[2,1]} + \cdots + t_{j,1}) - (t_{[1,2]} + t_{[2,2]} + \cdots + t_{[\alpha-1,2]})$$

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

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

and

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

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

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

and

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

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

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

and

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

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

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, 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

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

Finally, we can combine all these cases to obtain,

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

$$= (w + 1)(t_{i,2} - t_{j,2}) + 2(d_j - d_i) \geq 0 \text{ by assumption c. } \square$$

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 $w>1$.

Algorithms

The problem of minimizing total tardiness in a single machine with deterministic job descriptors is known to be NP-hard, Du et al. (1990). 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.

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	$d_i - \frac{LBt_{i,1} + UBt_{i,1}}{2} - \frac{LBt_{i,2} + UBt_{i,2}}{2}$

Table 2: Description of each algorithm.

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 $[0,50]$).
 - 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, Aydilek et al (2017).

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 (1993) and Allahverdi and Aydilek (2010). 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

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

where $TT(A)$ is the total tardiness of the algorithm, $Min(A)$ is the total tardiness of the best performing algorithm, and $Max(A)$ is that of the worst performing algorithm. This value for the error was previously used by Allahverdi and Aydilek (2010). 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 Table 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 to 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 (Figures 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.

$$H_0: \mu(\text{alg 13}) = \mu(\text{alg 20})$$

$$H_1: \mu(\text{alg 13}) < \mu(\text{alg 20})$$

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:

$$H_0: \mu(\text{alg 4}) = \mu(\text{alg 20})$$

$$H_1: \mu(\text{alg 4}) < \mu(\text{alg 20})$$

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:

$$H_0: \mu(\text{alg 4}) = \mu(\text{alg 13})$$

$$H_1: \mu(\text{alg 4}) < \mu(\text{alg 13})$$

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.

$$H_0: \mu(\text{alg 4}) = \mu(\text{alg 9})$$

$$H_1: \mu(\text{alg 4}) < \mu(\text{alg 9})$$

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 ($F2 \mid LBt_{i,n} \leq t_{i,n} \leq UBt_{i,n} \mid \sum T_i$).

A dominance relation is established and nineteen algorithms (alg 1 to 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.

Alg	Number of Jobs										Avg
	100	200	300	400	500	600	700	800	900	1000	
alg 1	40.3 9	31.15	30.60	24.01	22.14	19.06	17.22	15.20	14.76	12.39	22.69
alg 2	30.8 0	26.88	27.51	21.33	19.68	16.81	15.30	13.60	13.20	11.27	19.64
alg 3	29.9 0	26.92	27.41	21.49	19.82	16.89	15.27	13.52	13.25	11.27	19.57
alg 4	0.16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02
alg 5	29.8 3	26.90	27.02	21.08	19.59	16.75	15.10	13.46	13.13	11.23	19.41
alg 6	30.3 6	26.91	27.48	21.35	19.72	16.91	15.29	13.57	13.32	11.25	19.62
alg 7	30.2 9	26.87	27.15	21.05	19.47	16.72	15.09	13.47	13.13	11.17	19.44
alg 8	30.2 5	26.80	27.23	21.39	19.57	16.73	15.25	13.55	13.20	11.26	19.52
alg 9	29.8 1	26.80	27.26	21.21	19.63	16.88	15.19	13.46	13.22	11.24	19.47
alg 10	30.4 6	26.78	27.15	21.29	19.56	16.86	15.20	13.49	13.16	11.18	19.51
alg 11	51.5 6	37.16	33.72	26.78	24.78	21.03	19.20	16.75	16.22	13.74	26.09
alg 12	51.4 2	37.42	33.59	26.81	24.81	21.14	19.21	16.83	16.12	13.62	26.10
alg 13	52.3 6	37.90	34.07	27.05	24.88	21.20	19.30	16.86	16.26	13.77	26.37
alg 14	52.1 2	37.50	34.17	27.10	24.92	21.25	19.20	16.90	16.28	13.73	26.32
alg 15	51.6 8	37.12	33.70	26.89	24.78	21.12	19.20	16.84	16.20	13.70	26.12
alg 16	52.6 9	37.72	34.13	27.03	24.90	21.22	19.24	16.92	16.29	13.77	26.39
alg 17	52.0 2	37.48	33.82	27.00	24.77	21.12	19.22	16.80	16.32	13.75	26.23
alg 18	52.1 5	37.45	34.03	26.91	24.95	21.21	19.24	16.87	16.26	13.66	26.27
alg 19	52.0 9	37.54	33.72	27.05	24.88	21.10	19.22	16.85	16.24	13.72	26.24
alg 20	99.1 9	100.0 0	100.0 0	100.0 0	100.0 0	100.0 0	100.0 0	100.0 0	100.0 0	100.0 0	99.92

Table 3: Errors of the algorithms for uniform distribution (T = 0.5 R = 0.3)

Alg	Number of Jobs										Avg
	100	200	300	400	500	600	700	800	900	1000	
alg 1	20.13	16.77	12.31	10.26	9.16	7.09	6.62	5.70	5.29	4.88	9.82
alg 2	15.46	14.58	10.80	9.39	8.40	6.26	5.99	5.14	4.74	4.35	8.51
alg 3	15.36	14.74	10.73	9.33	8.30	6.30	5.98	5.16	4.73	4.31	8.49
alg 4	0.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03
alg 5	15.14	14.53	10.68	9.33	8.31	6.22	5.96	5.18	4.69	4.32	8.44
alg 6	15.34	14.65	10.77	9.40	8.42	6.29	5.99	5.15	4.73	4.33	8.51
alg 7	14.96	14.67	10.69	9.33	8.26	6.22	5.94	5.13	4.69	4.32	8.42
alg 8	15.59	14.60	10.70	9.41	8.31	6.25	5.98	5.14	4.72	4.32	8.50
alg 9	15.23	14.63	10.73	9.33	8.27	6.30	5.94	5.15	4.74	4.33	8.47
alg 10	15.03	14.65	10.70	9.36	8.29	6.28	5.93	5.14	4.70	4.33	8.44
alg 11	24.95	18.83	13.64	11.56	10.20	7.98	7.39	6.45	5.86	5.33	11.22
alg 12	24.88	19.07	13.66	11.54	10.22	7.95	7.38	6.46	5.84	5.29	11.23
alg 13	24.91	19.09	13.98	11.55	10.28	7.99	7.40	6.48	5.90	5.39	11.30
alg 14	24.86	19.12	13.85	11.56	10.19	8.00	7.41	6.46	5.88	5.34	11.27
alg 15	24.85	18.93	13.75	11.57	10.17	7.92	7.38	6.47	5.89	5.33	11.23
alg 16	24.69	19.04	14.01	11.56	10.29	8.00	7.42	6.44	5.90	5.34	11.27
alg 17	24.81	19.05	13.89	11.57	10.24	7.92	7.43	6.48	5.91	5.35	11.27
alg 18	24.80	19.14	13.75	11.60	10.30	7.97	7.41	6.47	5.86	5.33	11.26
alg 19	24.75	18.97	13.85	11.57	10.22	7.96	7.39	6.45	5.88	5.32	11.24
alg 20	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	0	0	0	0	0	0	0	0	0	0	0

Table 4: Errors of the algorithms for uniform distribution (T = 0.5 R = 0.5)

Alg	Number of Jobs										Avg
	100	200	300	400	500	600	700	800	900	1000	
alg 1	69.8 1	56.5 1	50.1 9	41.33	38.73	32.80	32.19	27.95	26.65	22.58	39.87
alg 2	57.8 6	49.7 2	44.7 5	36.68	34.65	29.42	29.04	25.35	23.94	20.46	35.19
alg 3	57.4 9	49.8 3	45.0 9	36.69	34.80	29.35	29.19	25.21	24.05	20.35	35.21
alg 4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
alg 5	57.5 0	49.3 6	44.5 6	36.45	34.45	29.15	28.88	25.05	23.65	20.32	34.94
alg 6	57.7 7	49.5 3	44.8 1	36.72	34.69	29.41	28.99	25.19	23.90	20.35	35.14
alg 7	57.5 5	49.2 5	44.5 0	36.50	34.27	29.15	28.89	25.02	23.60	20.32	34.91
alg 8	57.8 8	48.9 8	44.5 4	36.69	34.52	29.30	28.82	25.22	23.81	20.32	35.01
alg 9	57.4 6	49.6 1	44.9 6	36.54	34.41	29.32	29.01	25.10	23.79	20.36	35.06
alg 10	57.8 0	49.5 2	44.6 3	36.53	34.61	29.25	28.97	25.17	23.78	20.33	35.06
alg 11	80.6 9	64.5 3	55.9 0	46.32	43.14	36.44	35.33	30.65	29.56	24.85	44.74
alg 12	80.1 6	64.4 7	55.9 9	46.00	43.19	36.19	35.55	30.74	29.49	24.87	44.67
alg 13	80.5 2	64.8 2	55.8 8	46.37	43.38	36.46	35.63	30.95	29.67	25.00	44.87
alg 14	80.2 5	64.9 7	56.1 8	46.31	43.25	36.45	35.74	30.91	29.77	24.94	44.88
alg 15	80.2 3	64.5 5	56.0 4	46.03	43.17	36.21	35.33	30.69	29.56	24.96	44.68
alg 16	80.4 2	64.6 9	56.0 5	46.34	43.39	36.32	35.69	30.91	29.78	24.93	44.85
alg 17	80.7 7	64.7 6	55.8 8	46.33	43.34	36.43	35.50	30.83	29.71	24.85	44.84
alg 18	79.8 9	64.4 4	56.2 8	46.04	43.21	36.20	35.71	30.83	29.59	24.89	44.71
alg 19	80.0 8	64.7 7	56.0 9	46.13	43.30	36.31	35.61	30.89	29.64	24.99	44.78
alg 20	95.2 4	98.6 4	99.6 7	100.0 0	100.0 0	100.0 0	100.0 0	100.0 0	100.0 0	100.0 0	99.36

Table 5: Errors of the algorithms for uniform distribution (T = 0.7 R = 0.3)

Alg	Number of Jobs										Avg
	100	200	300	400	500	600	700	800	900	1000	
alg 1	56.2 7	39.68	29.96	24.30	21.29	17.25	15.77	14.54	13.60	11.85	24.45
alg 2	48.3 0	35.03	26.17	21.33	19.41	15.41	14.14	13.01	12.34	10.66	21.58
alg 3	48.4 6	35.30	26.36	21.34	19.50	15.44	14.19	12.97	12.28	10.66	21.65
alg 4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
alg 5	48.5 2	34.89	26.23	21.20	19.23	15.32	14.07	12.90	12.14	10.65	21.52
alg 6	48.3 1	35.42	26.40	21.33	19.41	15.40	14.13	12.99	12.23	10.67	21.63
alg 7	48.4 0	35.11	26.13	21.22	19.30	15.36	14.08	12.95	12.14	10.62	21.53
alg 8	48.6 8	35.20	26.32	21.28	19.46	15.34	14.22	12.98	12.25	10.65	21.64
alg 9	48.4 5	35.36	26.34	21.32	19.43	15.40	14.10	12.96	12.24	10.63	21.62
alg 10	48.1 9	35.14	26.44	21.28	19.43	15.40	14.13	12.97	12.16	10.65	21.58
alg 11	64.3 0	45.01	33.16	26.60	23.49	18.88	17.51	16.03	14.92	13.20	27.31
alg 12	63.8 8	44.65	33.28	26.59	23.58	18.87	17.47	15.98	14.85	13.24	27.24
alg 13	64.3 3	44.80	33.15	26.71	23.53	18.95	17.63	16.08	15.04	13.28	27.35
alg 14	63.8 2	44.48	33.30	26.89	23.64	19.01	17.62	16.14	15.03	13.32	27.33
alg 15	64.1 5	44.85	33.19	26.57	23.51	18.85	17.56	16.05	14.92	13.25	27.29
alg 16	63.9 2	44.66	33.39	26.91	23.59	18.99	17.66	16.16	15.00	13.32	27.36
alg 17	64.1 9	44.79	33.09	26.68	23.51	18.95	17.55	15.99	15.00	13.28	27.30
alg 18	63.7 8	44.52	33.24	26.78	23.52	18.91	17.58	16.07	14.93	13.28	27.26
alg 19	63.9 8	44.68	33.19	26.68	23.67	18.88	17.59	16.05	14.95	13.29	27.30
alg 20	97.3 7	100.0 0	100.0 0	100.0 0	100.0 0	100.0 0	100.0 0	100.0 0	100.0 0	100.0 0	99.74

Table 6: Errors of the algorithms for uniform distribution (T = 0.7 R = 0.5)

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

Table 7: Errors of the algorithms for all the distributions over n, T, and R

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)

Table 8: 95% and 99% Confidence Intervals of the Errors of the Algorithms

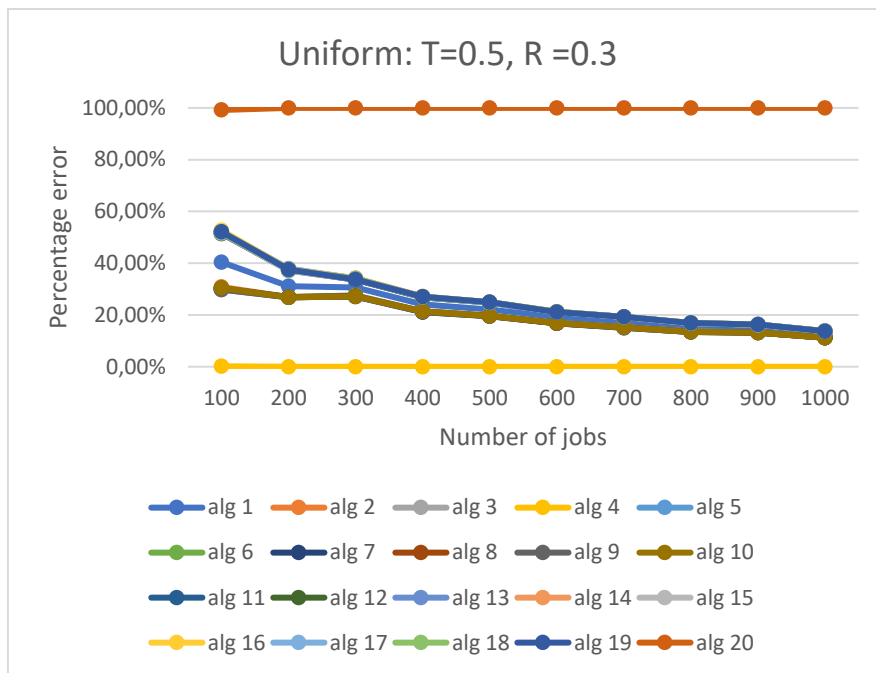


Figure 1: Errors of the algorithms for T=0.5 and R=0.3 - Uniform Distribution

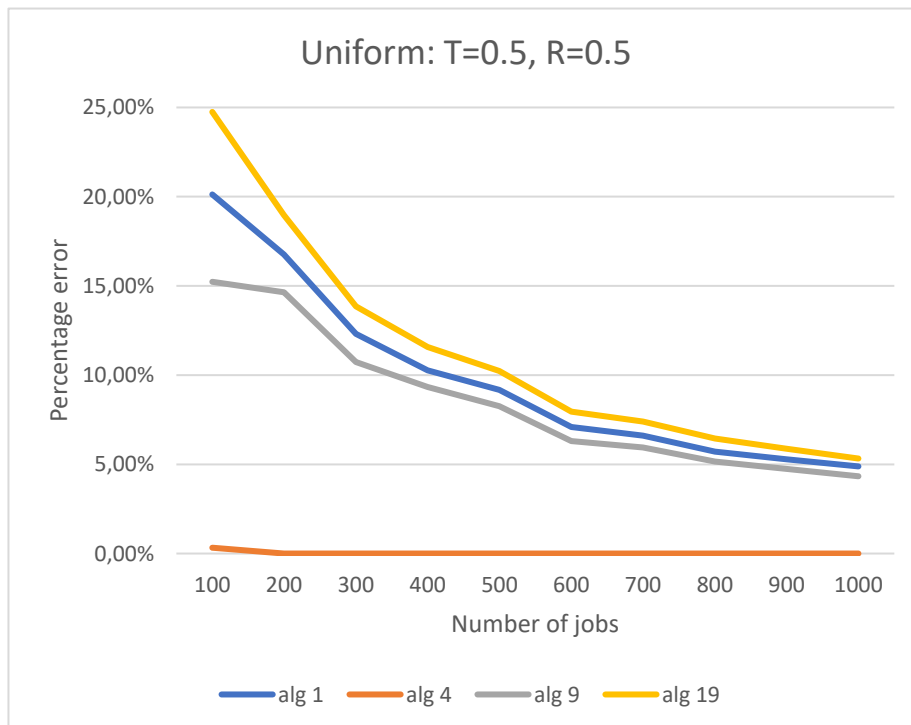


Figure 2: Errors of the four algorithms for T=0.5 and R=0.5 - Uniform Distribution

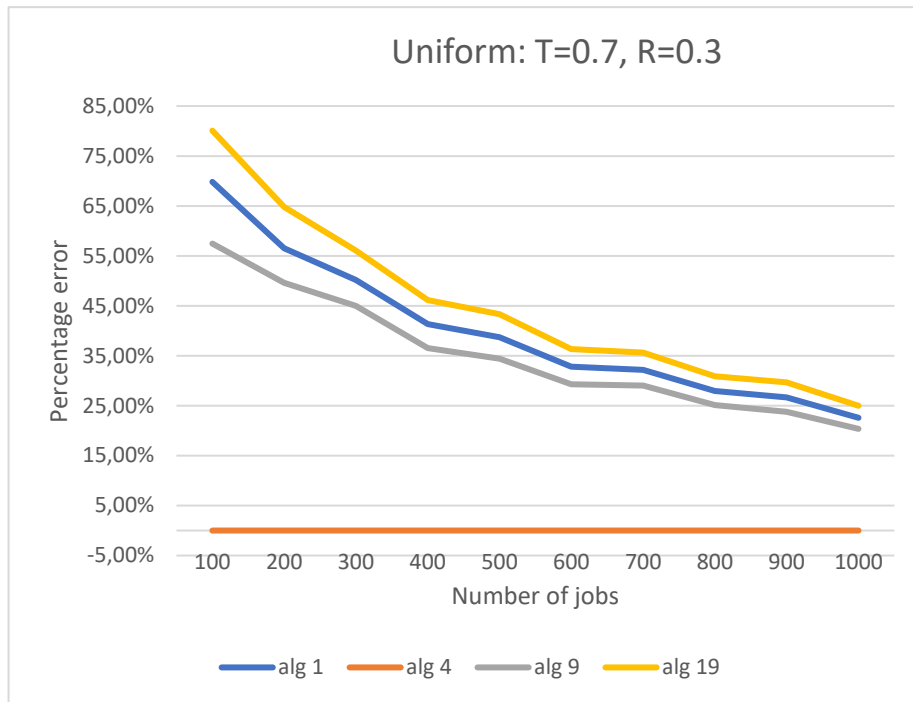


Figure 3: Errors of the four algorithms for T=0.7 and R=0.3 - Uniform Distribution

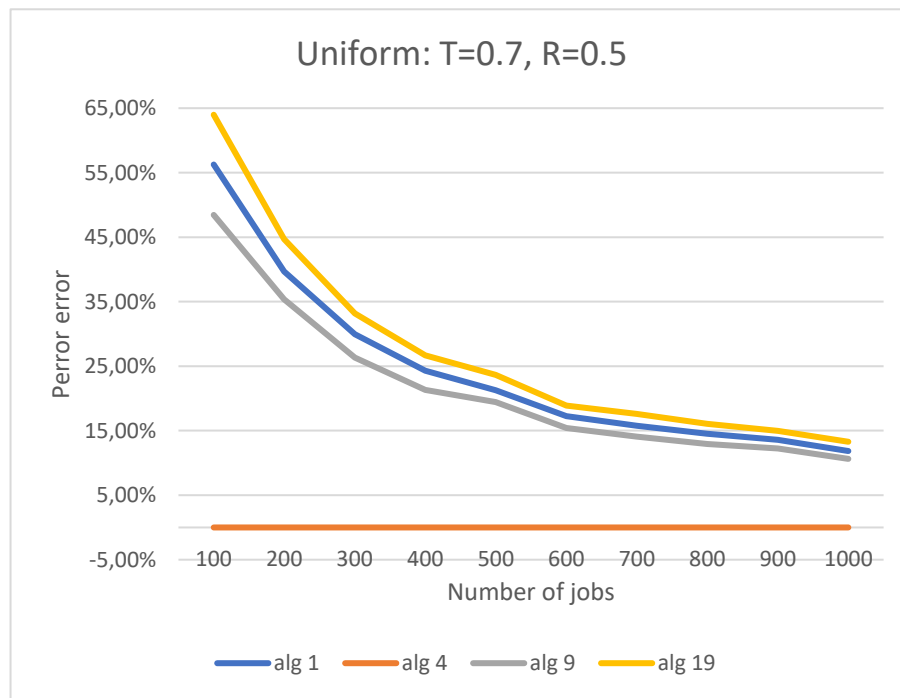


Figure 4: Errors of the four algorithms for T=0.7 and R=0.5 - Uniform Distribution

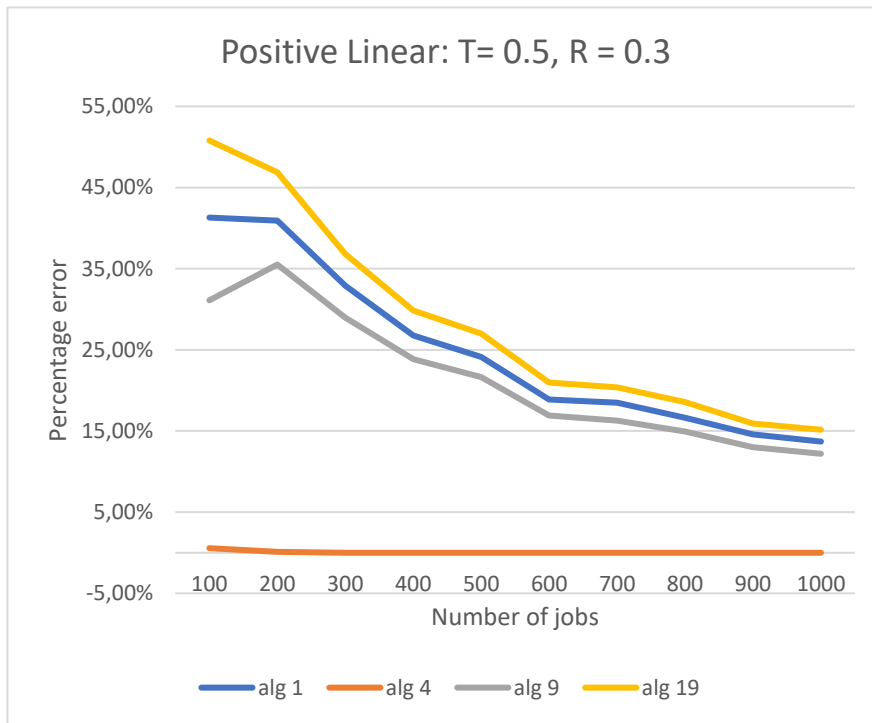


Figure 5: Errors of the four algorithms for T=0.5 and R=0.3 – Positive Linear Distribution

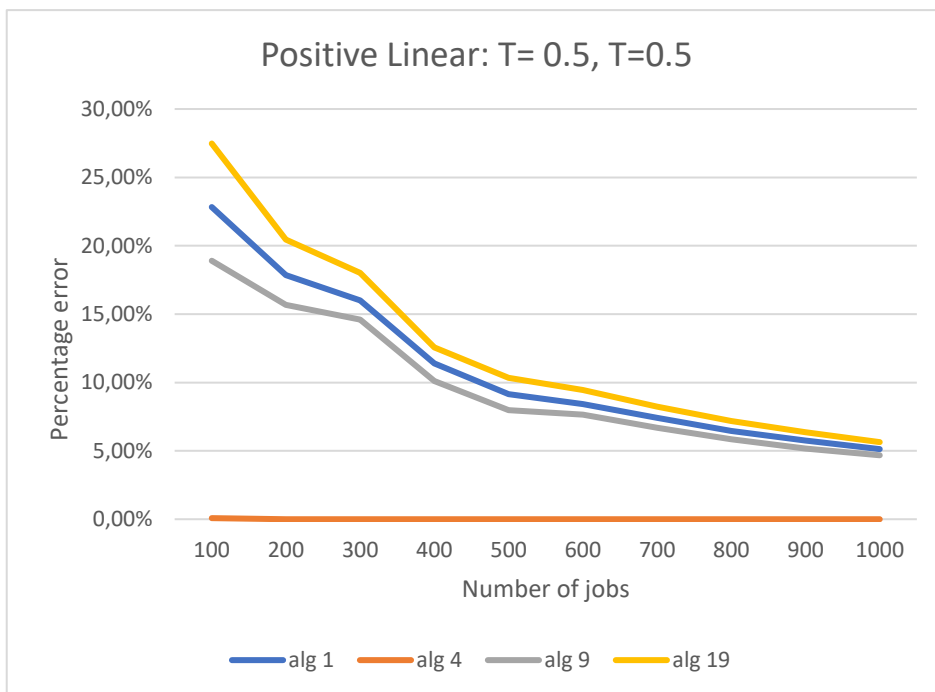


Figure 6: Errors of the four algorithms for T=0.5 and R=0.5 – Positive Linear Distribution

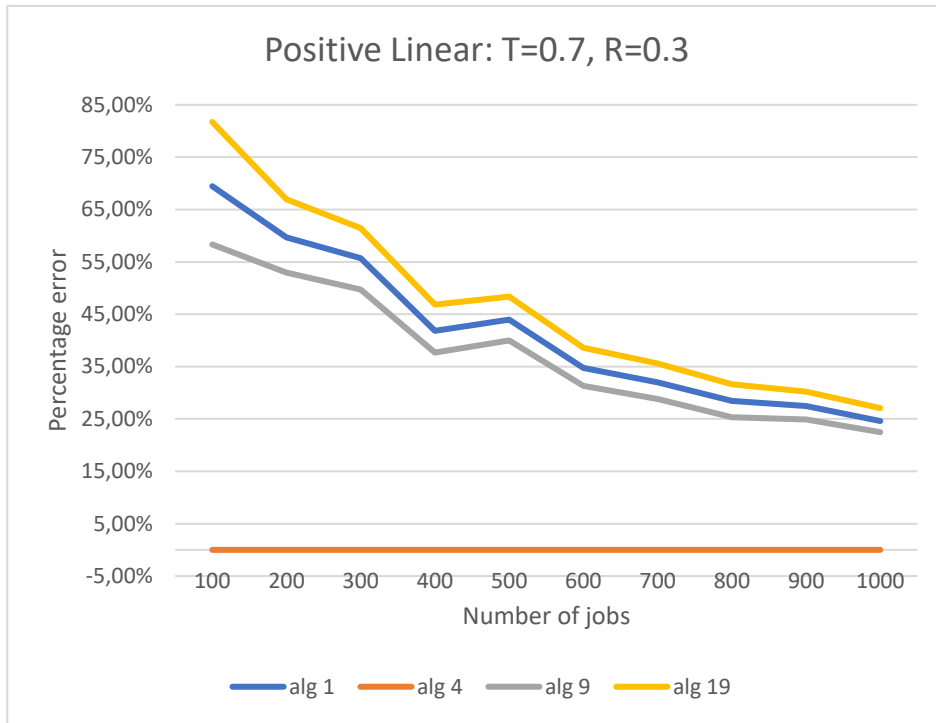


Figure 7: Errors of the four algorithms for T=0.7 and R=0.3 – Positive Linear Distribution

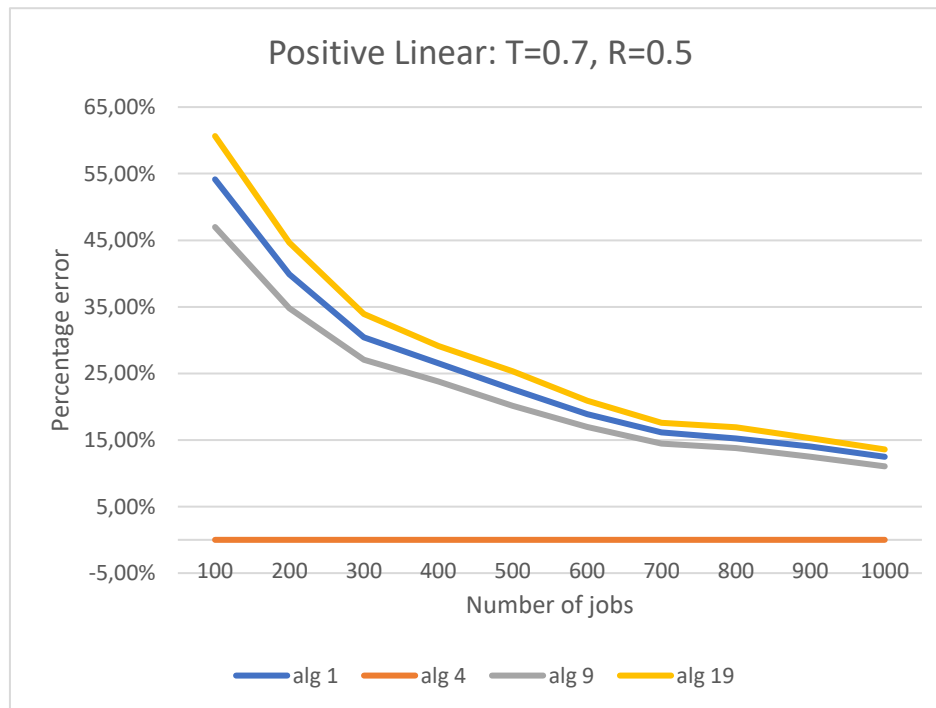


Figure 8: Errors of the four algorithms for T=0.7 and R=0.5 – Positive Linear Distribution

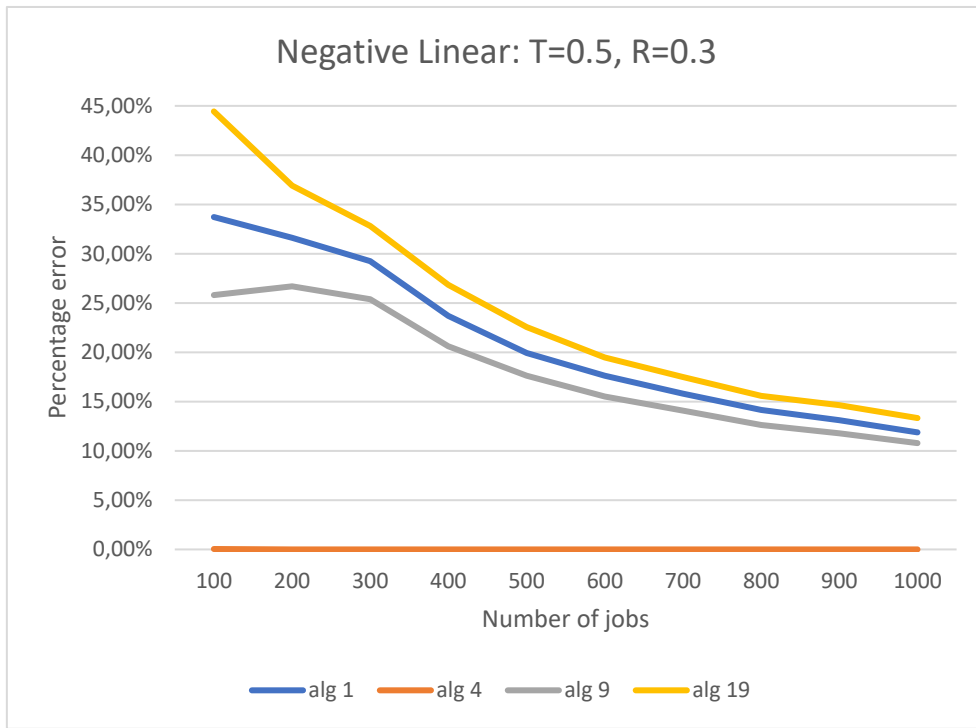


Figure 9: Errors of the four algorithms for T=0.5 and R=0.3 – Negative Linear Distribution

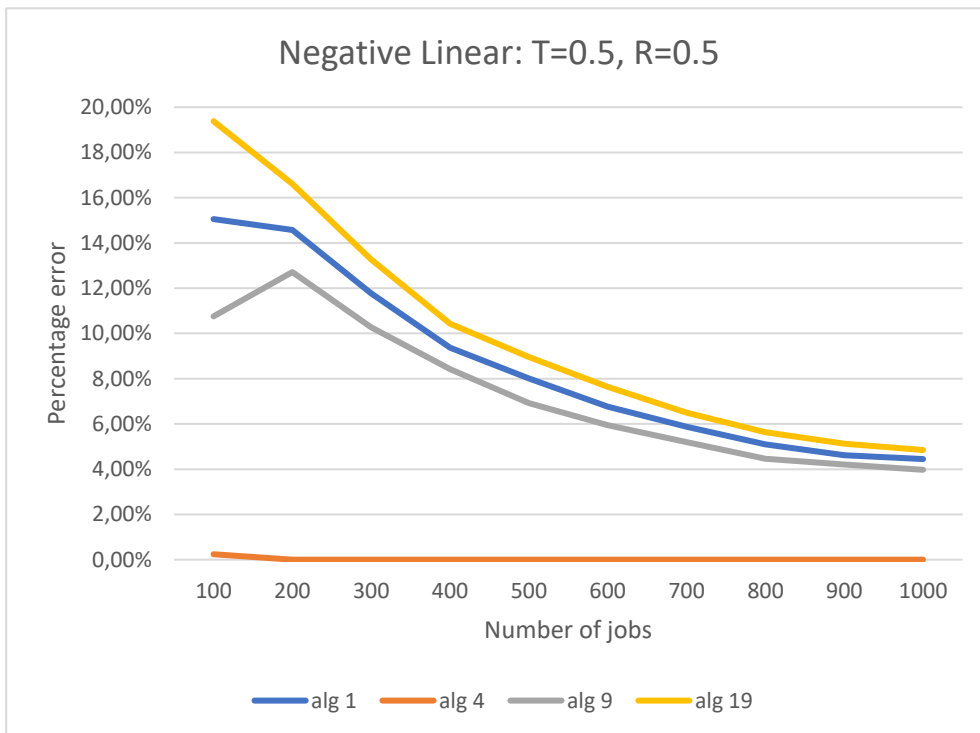


Figure 10: Errors of the four algorithms for T=0.5 and R=0.5 – Negative Linear Distribution

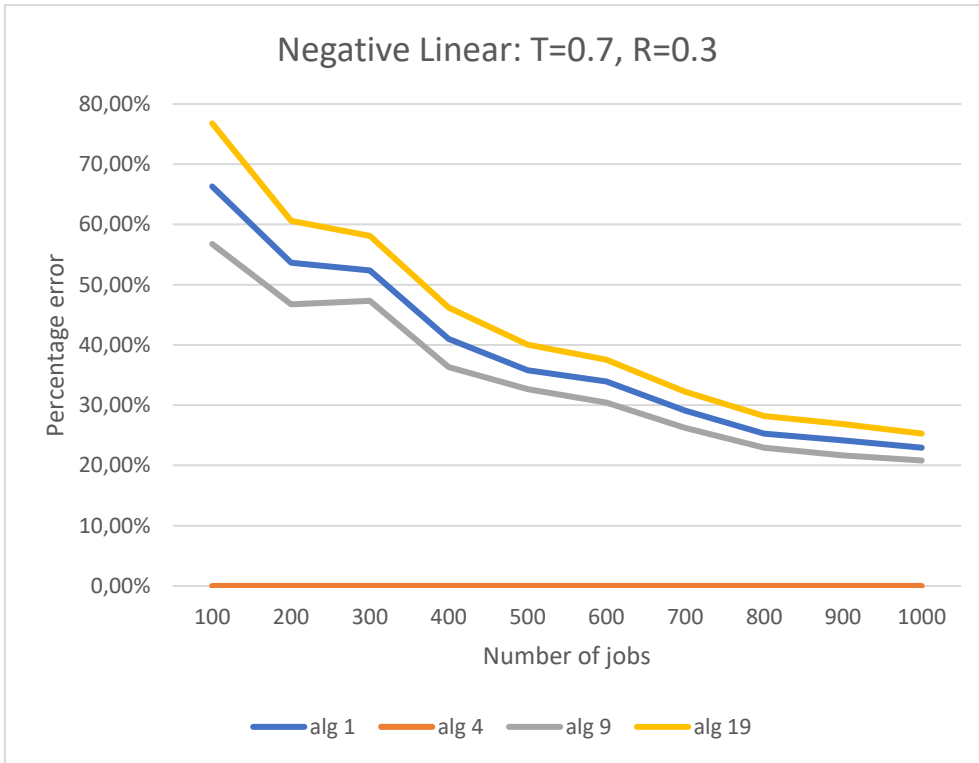


Figure 11: Errors of the four algorithms for T=0.7 and R=0.3 – Negative Linear Distribution

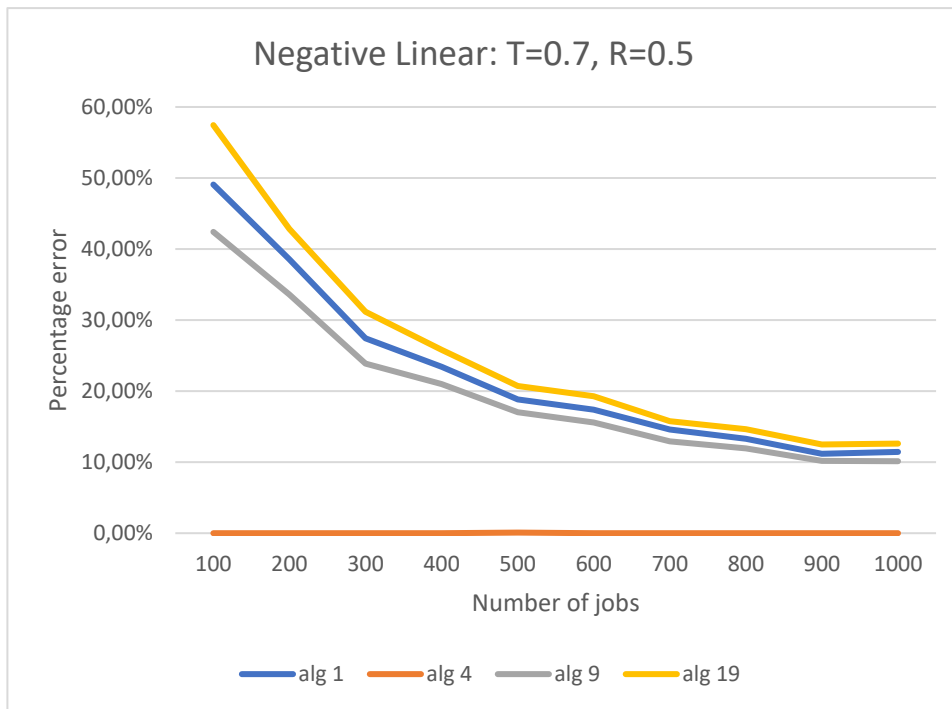


Figure 12: Errors of the four algorithms for T=0.7 and R=0.5 – Negative Linear Distribution

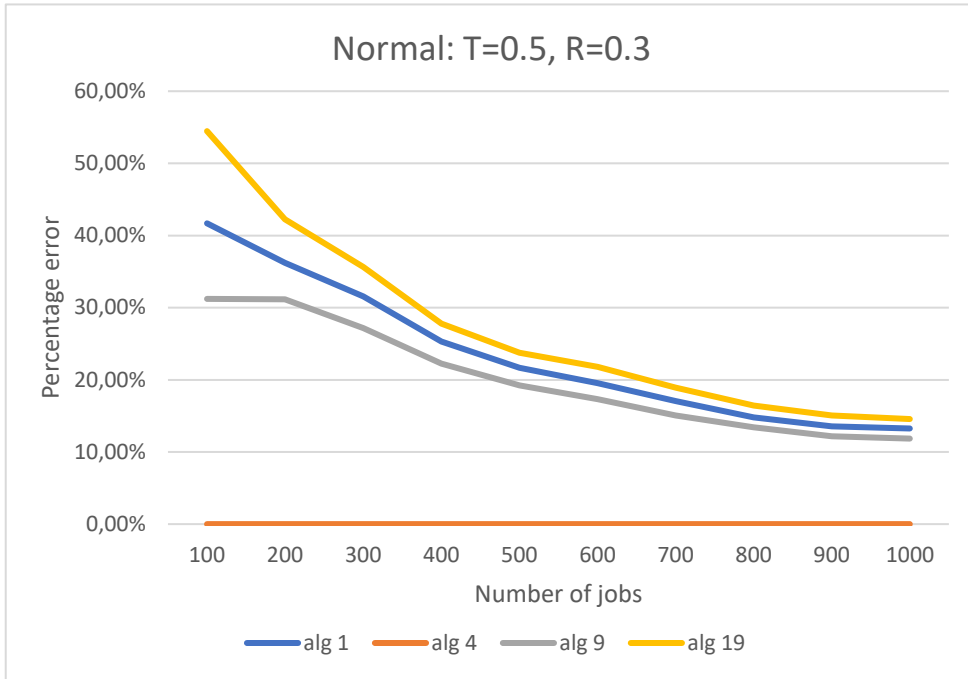


Figure 13: Errors of the four algorithms for T=0.5 and R=0.3 – Normal Distribution

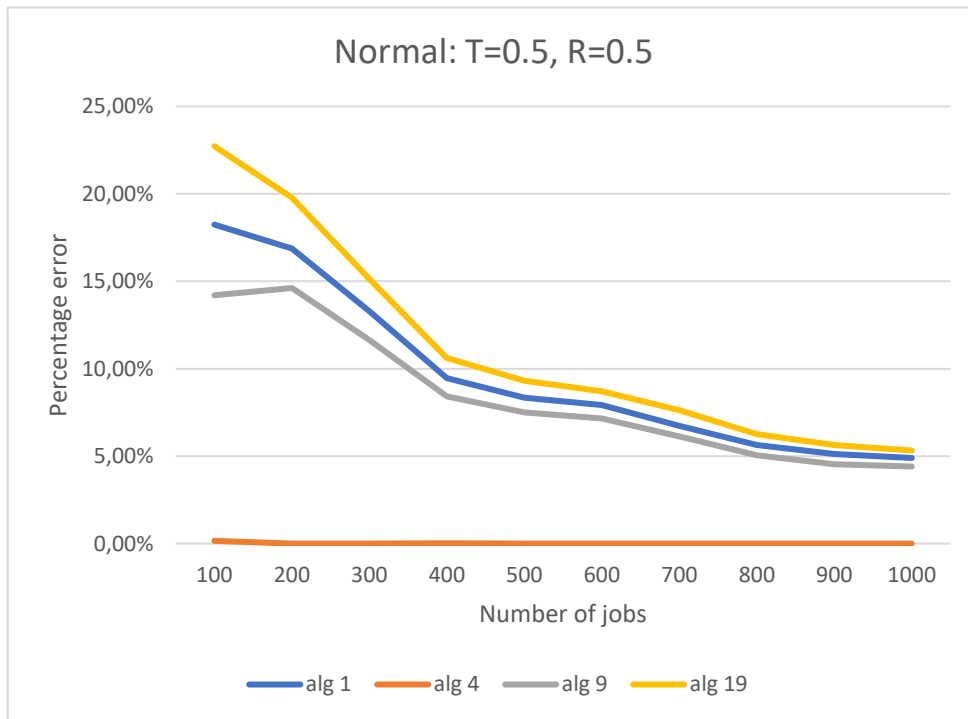


Figure 14: Errors of the four algorithms for T=0.5 and R=0.5 – Normal Distribution

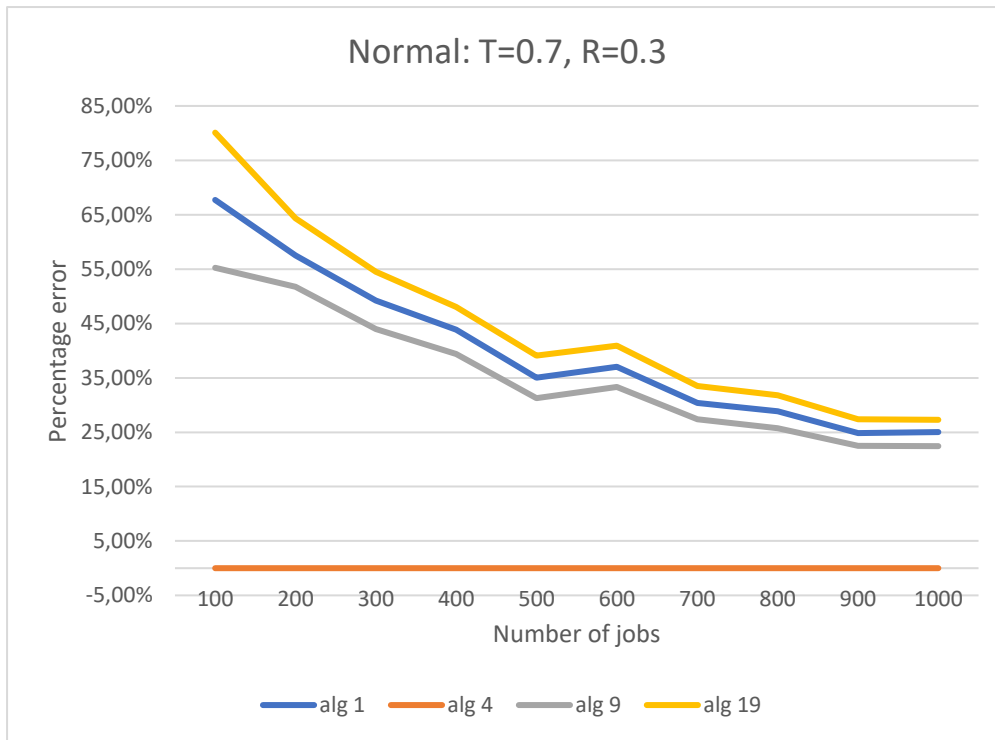


Figure 15: Errors of the four algorithms for T=0.7 and R=0.3 – Normal Distribution

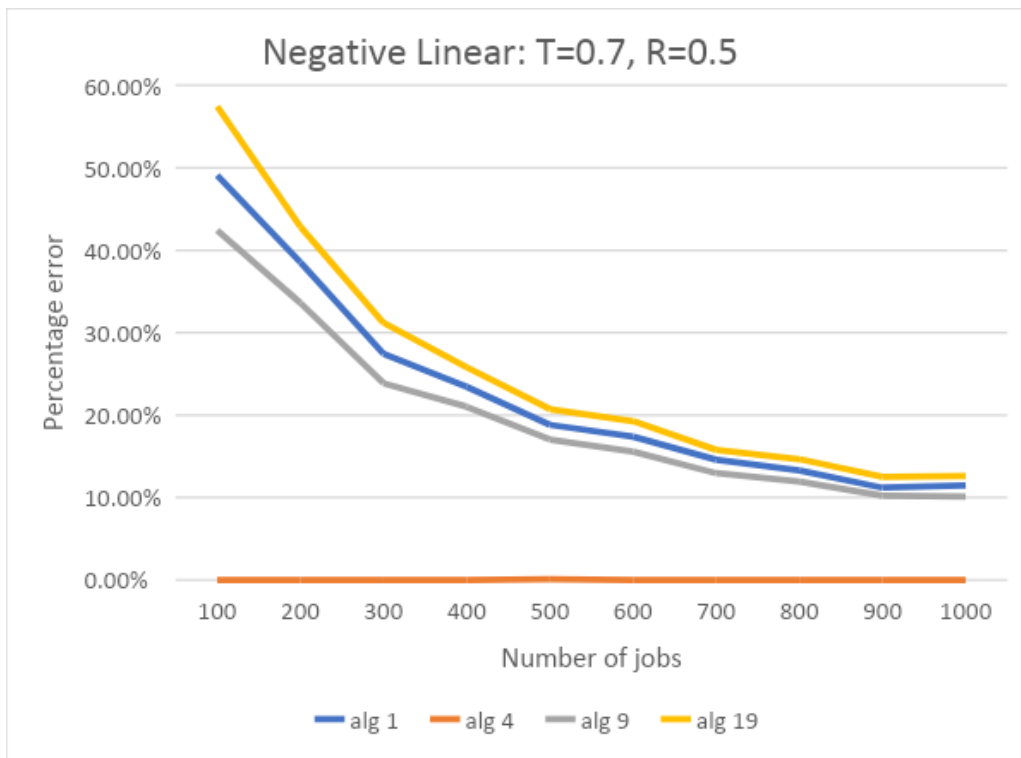


Figure 12: Errors of the four algorithms for T=0.7 and R=0.5 – Negative Linear Distribution

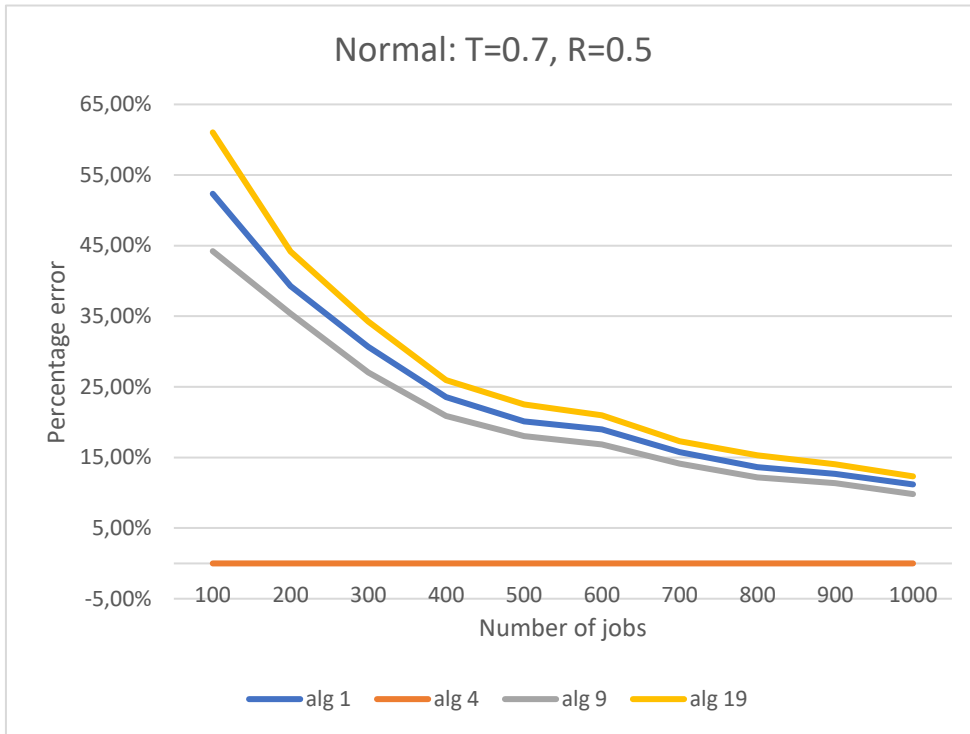


Figure 16: Errors of the four algorithms for T=0.7 and R=0.5 – Normal Distribution

References

- Abtahi, Z., Sahraeian, R. (2021). Robust and Stable Flow Shop Scheduling Problem under Uncertain Processing Times and Machines' Disruption. *International Journal of Engineering*, 34(4), 935-947.
- Allahverdi, A., 2001. The tricriteria two-machine flowshop scheduling problem. *International Transactions in Operational Research* 8, 403-425.
- Allahverdi, A. (2022). A survey of scheduling problems with uncertain interval/bounded processing/setup times. *Journal of Project Management*, 7(4), 255-264.
- Allahverdi, M., & Allahverdi, A. (2020). Algorithms for four-machine flowshop scheduling problem with uncertain processing times to minimize makespan. *RAIRO-Operations Research*, 54(2), 529-553.
- Allahverdi, M., & Allahverdi, A. (2021). Minimizing total completion time for flowshop scheduling problem with uncertain processing times. *RAIRO-Operations Research*, 55, S929-S946.
- Allahverdi, A., & Aydilek, H. (2010). Heuristics for the two-machine flowshop scheduling problem to minimize maximum lateness with bounded processing times. *Computers & Mathematics with Applications*, 60(5), 1374-1384.
- Allahverdi, A., Aydilek, H. (2015). The two stage assembly flowshop scheduling problem to minimize total tardiness. *Journal of Intelligent Manufacturing*, 26(2), 225-237.
- Allahverdi, A., Aydilek, H., Aydilek, A. (2016). Two-stage assembly scheduling problem for minimizing total tardiness with setup times. *Applied Mathematical Modelling*, 40(17-18), 7796-7815.
- Allahverdi, M., Aydilek, H., Aydilek, A., & Allahverdi, A. (2021). A better dominance relation and heuristics for two-machine no-wait flowshops with maximum lateness performance measure. *Journal of Industrial & Management Optimization*, 17(4), 1973.
- Allahverdi, A., & Sotskov, Y. (2003). Two-machine flowshop minimum-length scheduling problem with random and bounded processing times. *International Transactions in Operational Research*, 10(1), 65-76.
- Arabameri, S., Salmasi, N. (2013). Minimization of weighted earliness and tardiness for no-wait sequence-dependent setup times flowshop scheduling problem. *Computers & Industrial Engineering*, 64(4), 902-916.
- Arık, O. A. (2020). Single machine earliness/tardiness scheduling problem with grey processing times and the grey common due date. *Grey Systems: Theory and Application* (to appear).

- Aydilek, A., Aydilek, H., & Allahverdi, A. (2017). Algorithms for minimizing the number of tardy jobs for reducing production cost with uncertain processing times. *Applied mathematical modelling*, 45, 982-996.
- Aydilek, H., Aydilek, A., Allahverdi, M., & Allahverdi, A. (2022). More effective heuristics for a two-machine no-wait flowshop to minimize maximum lateness. *International Journal of Industrial Engineering Computations*, 13(4), 543-556.
- Bożejko, W., Uchroński, M., & Wodecki, M. (2020). Blocks for two-machines total weighted tardiness flow shop scheduling problem. *Bulletin of the Polish Academy of Sciences: Technical Sciences*, 31-41.
- de Athayde Prata, B., Fernandez-Viagas, V., Framinan, J. M., Rodrigues, C. D. (2022). Matheuristics for the flowshop scheduling problem with controllable processing times and limited resource consumption to minimize total tardiness. *Computers & Operations Research*, 105880.
- Ding, J., Song, S., Zhang, R., Gupta, J. N., Wu, C. (2015). Accelerated methods for total tardiness minimisation in no-wait flowshops. *International Journal of Production Research*, 53(4), 1002-1018.
- Du, J., Leung, J. Y. T. (1990). Minimizing total tardiness on one machine is NP-hard. *Mathematics of operations research*, 15(3), 483-495.
- Framinan, J. M., Leisten, R. (2008). Total tardiness minimization in permutation flow shops: a simple approach based on a variable greedy algorithm. *International Journal of Production Research*, 46(22), 6479-6498.
- Gonzalez-Neira, E. M., Ferone, D., Hatami, S., & Juan, A. A. (2017). A biased-randomized simheuristic for the distributed assembly permutation flowshop problem with stochastic processing times. *Simulation Modelling Practice and Theory*, 79, 23-36.
- Hong, S. C., Lin, S. S. (2022). Apply ordinal optimization to optimize the job-shop scheduling under uncertain processing times. *Arabian Journal for Science and Engineering*, 47(8), 9659-9671.
- Keshavarz, T., Salmasi, N. (2013). Makespan minimisation in flexible flowshop sequence-dependent group scheduling problem. *International Journal of Production Research*, 51(20), 6182-6193.
- Kim, Y. D. (1993). A new branch and bound algorithm for minimizing mean tardiness in two-machine flowshops. *Computers & Operations Research*, 20(4), 391-401.
- Rakrouki, M.A, Aljohani, A., Ladhari, T. (2023). Minimizing total tardiness in a two-machine flowshop scheduling problem with availability constraints. *Intelligent Automation and Soft Computing* 35 (1) , 1119-1134.

- Saber, R. G., Ranjbar, M. (2022). Minimizing the total tardiness and the total carbon emissions in the permutation flow shop scheduling problem. *Computers & Operations Research*, 138, 105604.
- Seidgar, H., Kiani, M., Abedi, M., & Fazlollahtabar, H. (2014). An efficient imperialist competitive algorithm for scheduling in the two-stage assembly flow shop problem. *International Journal of Production Research*, 52(4), 1240-1256.
- Seo, D. K., Klein, C. M., Jang, W. (2005). Single machine stochastic scheduling to minimize the expected number of tardy jobs using mathematical programming models. *Computers & Industrial Engineering*, 48(2), 153-161.
- Sotskov, Y. N., Egorova, N. G. (2019). The optimality region for a single-machine scheduling problem with bounded durations of the jobs and the total completion time objective. *Mathematics*, 7(5), 382.
- Sotskov, Y. N., Matsveichuk, N. M., Hatsura, V. D. (2020). Schedule execution for two-machine job-shop to minimize makespan with uncertain processing times. *Mathematics*, 8(8), 1314.
- Sotskov, Y. N., Matsveichuk, N. M., Hatsura, V. D. (2019). Two-machine job-shop scheduling problem to minimize the makespan with uncertain job durations. *Algorithms*, 13(1), 4.
- Tayanithi, P., Manivannan, S., & Banks, J. (1992). A knowledge-based simulation architecture to analyze interruptions in a flexible manufacturing system. *Journal of manufacturing systems*, 11(3), 195-214.
- Vallada, E., Ruiz, R. (2010). Genetic algorithms with path relinking for the minimum tardiness permutation flowshop problem. *Omega*, 38(1-2), 57-67.
- Vallada, E., Ruiz, R., Minella, G. (2008). Minimising total tardiness in the m-machine flowshop problem: A review and evaluation of heuristics and metaheuristics. *Computers & Operations Research*, 35(4), 1350-1373.
- Wang, K., Choi, S. H. (2012). A decomposition-based approach to flexible flow shop scheduling under machine breakdown. *International Journal of Production Research*, 50(1), 215-234.
- Wang, Z. Y., Pan, Q. K., Gao, L., Wang, Y. L. (2022). An effective two-stage iterated greedy algorithm to minimize total tardiness for the distributed flowshop group scheduling problem. *Swarm and Evolutionary Computation*, 74, 101143.
- Xu, W., Wu, W., Wang, Y., He, Y., Lei, Z. (2022). Flexible job-shop scheduling method based on interval grey processing time. *Applied Intelligence*, 1-16.