Contents lists available at GrowingScience

Journal of Project Management

homepage: www.GrowingScience.com

An improved algorithm to minimize the total completion time in a two-machine no-wait flowshop with uncertain setup times

Muberra Allahverdi[a*]

[a]*Department of Mathematical Sciences, School of Natural Sciences, Kean University, 1000 Morris Ave, Union, NJ 07083, United States*

C H R O N I C L E

Article history:
Received: June 17, 2021
Received in revised format: July 30, 2021
Accepted: September 18, 2021
Available online:
September 18, 2021

Keywords:
Flowshop scheduling
No-wait
Setup time
Uncertainty
Total completion time

A B S T R A C T

Since scheduling literature has a wide range of uncertainties, it is crucial to take these into account when solving performance measure problems. Otherwise, the performance may severely be affected in a negative way. In this paper, an algorithm is proposed to minimize the total completion time (TCT) of a two-machine no-wait flowshop with uncertain setup times within lower and upper bounds. The results are compared to the best existing algorithm in scheduling literature: the programming language Python is used to generate random samples with respect to various distributions, and the TCT of the proposed algorithm is compared to that of the best existing one. Results reveal that the proposed one significantly outperforms the best one given in literature for all considered distributions. Specifically, the average percentage improvement of the proposed algorithm over the best existing one is over 90%. A test of hypothesis is conducted to further confirm the results.

1. Introduction

A two-machine flowshop is a manufacturing model with two machines and a set of jobs, each of which has two operations, where the first operation is performed on the first machine and the second on the second machine. Certain manufacturing settings require that these operations move from the first machine to the next with no idle time in between. This might be necessary, for instance, in cases where heat is involved and waiting would cause certain materials to cool down, thereby negatively affecting the performance, Baker and Trietsch (2009). Such a flowshop where no idle time is permitted is called a no-wait flowshop and is used extensively in many industries, including chemical, plastic, and pharmaceutical. Certain scheduling problems such as aircraft landing, patient scheduling, and bakery production require no-wait flowshops as well, Allahverdi (2016), Hall and Sriskandarajah (1996). Research regarding no-wait flowshops is growing, addressing many problems with different performance measures. Such papers include Ying and Lin (2018) and Li et al. (2018), addressing the makespan and total flow time, respectively. Since uncertainty is common in scheduling problems (Soroush (1999), Soroush (2007)), it is crucial to take these into account while optimizing a certain performance measure. In some manufacturing settings, for instance, certain job descriptions (e.g. processing times, setup times, due dates) are unpredictable. Many scheduling papers address such problems: Seo et al. (2005) addresses the case of minimizing the expected number of tardy jobs given normally distributed processing times, Cunningham and Dutta (1973) and Ku and Niu (1986) consider the case where jobs have processing times that are exponentially distributed, and Kalczynski and Kamburowski (2006) consider the problem where job processing times follow a Weibull distribution.

* Corresponding author.
E-mail address: mu.allahverdi@gmail.com (M. Allahverdi)

doi: 10.5267/j.jpm.2021.9.001

The time required to prepare a certain machine for a particular operation is known as the setup time of that operation. In particular, $s_{i,j}$ denotes the setup time for the operation of job i on machine j. To minimize the total completion time (TCT) of a scheduling problem, it is crucial to consider setup times as well along with processing times. This is especially true for settings where setup times are long enough to make a difference in the total completion time, as neglecting them in such cases considerably impacts the performance. The paper Kopanos et al. (2009) discusses such cases at length. In fact, setup times should be considered separately from processing times in order to eliminate waste, increase productivity, improve resource utilization, and meet deadlines, Allahverdi (2015). Nonetheless, only 10% of the scheduling literature address setup times despite its common presence in manufacturing settings, (Allahverdi (2015) and Kopanos et al. (2009)).
In some manufacturing environments, setup times need not be deterministic but are rather unpredictable and prone to change. Hence, it is not only important to consider setup times in a solution but to consider the unpredictability as well, which may be due to a wide range of reasons such as the breakdown of equipment, inadequate crew skills, and a shortage of necessary tools, Kim and Bobrowski (1997). Papers considering uncertain setup times include Allahverdi (2005), Allahverdi (2006a), Allahverdi (2006b), Allahverdi et al. (2003), addressing the problems $(F2|Ls_{i,k} \leq s_{i,k} \leq Us_{i,k}|C_{max}, \sum C_j)$, $(F2|Ls_{i,k} \leq s_{i,k} \leq Us_{i,k}|C_{max})$, $(F2|Ls_{i,k} \leq s_{i,k} \leq Us_{i,k}|\sum C_j)$, and $(F2|Ls_{i,k} \leq s_{i,k} \leq Us_{i,k}|L_{max})$, respectively. The first establishes a dominance relation for a two-machine flowhsop with respect to makespan and total completion time. The rest consider the same problem with respect to C_{max} (makespan), $\sum C_j$ (total completion time), and L_{max} (maximum lateness), respectively.

Allahverdi and Allahverdi (2020) address the scheduling problem of minimizing total completion time with uncertain setup times where only the lower and upper bounds are known. It establishes an algorithm to minimize the total completion time of such a problem. In this paper, we propose a new algorithm which significantly outperforms the one given in paper Allahverdi and Allahverdi (2020). The two algorithms are compared for four different distributions: uniform, positive linear, negative linear, and normal. Furthermore, a test of hypothesis is conducted to confirm the effectiveness of the new algorithm. The remainder of the paper is as follows: Section 1 explains the proposed algorithm and how it is applied. Section 2 describes the test that was conducted using the programming language python to compare the proposed algorithm to the best existing one in literature. Section 3 discusses and analyzes the results obtained from the test described in section 2. Section 4 conducts a test of hypothesis to determine the effectiveness of the proposed algorithm over the existing algorithm in literature. Section 5 constructs a 95% confidence interval for the percentage improvements of the proposed algorithm over the existing one. Section 6 summarizes and concludes the results obtained in the paper.

Notation

The following notation will be used throughout this paper.

$s_{j,k}$: Setup time of job j on machine k
$t_{j,k}$: Processing time of job j on machine k
$Us_{j,k}$: Upper bound on the setup time of job j on machine k
$Ls_{j,k}$: Lower bound on the setup time of job j on machine k
n : number of jobs
Δ : The range below the upper bound of a setup time where the lower bound is selected from. In particular, if $Us_{j,k}$ is randomly generated from within the range (1,100), then the lower bound $Ls_{j,k}$ would be generated from the range ($Us_{j,k} - \Delta$, $Us_{j,k}$), provided that $Us_{j,k} - \Delta$ is greater than or equal to 1. Otherwise, $Ls_{j,k}$ is generated from the range (1, $Us_{j,k}$).

2. An improved algorithm

Minimizing TCT for a two machine no-wait flowshop with uncertain setup times is known to be NP-Hard. Since there is an optimal solution for the case of a single machine, we will transform a two-machine problem into a single machine one. Given processing times and lower and upper bounds of setup times $t_{j,k}$, $Ls_{j,k}$, and $Us_{j,k}$ for k = 1,2, we define the processing times of a single machine problem,

$$t1 = [t_{j,1} + t_{j,2} + 0.5(Ls_{j,1} + Us_{j,1}) + 0.25(Ls_{j,2} + Us_{j,2}) \text{ for } i = 1, \cdots, n].$$

We then apply the SPT to order the jobs based on the induced processing times $t1$.

3. Testing the improved algorithm

The algorithm is compared with the best algorithm in Allahverdi and Allahverdi (2020) for the n values 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000 and Δ values 20,25,30,35,40,45, where n denotes the number of jobs and Δ determines how the lower bound of setup times is generated based on the upper bound, as explained in the notation section. For each n and Δ combination, r = 100 replications are conducted and the average and standard deviations of those replications are taken. In particular, the following steps are taken to compare the two algorithms.

1. Select the number of jobs n, the values for Δ, and the number of replications r.

2. For each n and Δ combination, do the following:

(a) For R = 1, do the following:

i. Randomly generate values for processing times $t_{j,k}$ for $i = 1, \cdots, n$ and $k = 1,2$.

ii. Randomly generate values for upper and lower bounds of setup times $Us_{j,k}$ and $Ls_{j,k}$ for $i = 1, \cdots, n$ and $k = 1,2$.

iii. Transform the 2-machine problem into a single machine problem by giving different weights to the processing times and lower and upper bounds of setup times. Denote the processing times of the new single machine by $t1$.

iv. Apply the SPT on $t1$ and denote the sequence obtained by $st1$. We compare the TCT of $st1$ to $st2$, the sequence obtained from the algorithm given in paper Allahverdi and Allahverdi (2020).

v. Generate setup times within the lower and upper bounds (a number of different distributions are considered while generating the setup times).

vi. Compute the TCT of $st1$ and $st2$ given setup and processing times.

vii. Compute the error for the TCT's using the formula

$$\text{Error} = \frac{TCT(sti) - \min(TCT(st1), TCT(st2))}{\min(TCT(st1), TCT(st2))}$$

for $i = 1,2$.

(b) Let $R = R + 1$. If $R < r$, repeat step (i). Otherwise, continue.

(c) Take the average and standard deviations of TCT's obtained from all the replications.

4. Computational results

For each combination of n and Δ, 100 replications are performed for any given distribution. So a total of $10 \times 6 \times 4 \times 100 = 24{,}000$ different cases are considered. The results are listed in Tables 1-4 for the uniform, positive linear, negative linear, and normal distributions, respectively. The first two columns in the table are the considered n and Δ values. The third column is the average of the errors obtained in the replications using the proposed algorithm. The fourth column is the average of the errors using the algorithm from Allahverdi and Allahverdi (2020). The fifth is the standard deviation of the errors using the proposed algorithm. Similarly, the sixth column is the standard deviation of the errors using the algorithm from Allahverdi and Allahverdi (2020). Finally, the last column is the percentage improvement of the proposed algorithm over the existing one using the formula (x-y)/x where x is the value obtained from column 4 and y is the value obtained from column 3.

Fig. 1 compares the percentage improvement of the four different distributions with respect to the n values. As the n values increase, the percentage improvement seems to increase as well, which further indicates the effectiveness of the proposed algorithm over the existing one.

Table 1
Comparing the proposed algorithm to the existing one – Uniform Distribution

n	delta	new_average	old_average	new_std	old_std	per imp
100	20	0.23	1.35	0.56	1.27	0.83
100	25	0.34	1.07	0.75	1.12	0.68
100	30	0.34	0.97	0.64	1.1	0.65
100	35	0.26	1.21	0.65	1.16	0.79
100	40	0.14	1.38	0.34	1.14	0.90
100	45	0.29	1.09	0.56	1.13	0.73
200	20	0.14	0.93	0.43	0.86	0.85
200	25	0.11	1.08	0.33	0.94	0.90
200	30	0.09	1.09	0.26	0.92	0.92
200	35	0.18	0.98	0.4	1	0.82
200	40	0.12	1.03	0.33	0.98	0.88
200	45	0.16	0.91	0.37	0.89	0.82
300	20	0.12	0.98	0.29	0.85	0.88
300	25	0.1	0.93	0.28	0.78	0.89
300	30	0.08	1.06	0.22	0.87	0.92
300	35	0.14	0.84	0.31	0.74	0.83
300	40	0.11	0.98	0.29	0.77	0.89
300	45	0.13	0.95	0.39	0.81	0.86

Table 1
Comparing the proposed algorithm to the existing one – Uniform Distribution (Continued)

n	delta	new_average	old_average	new_std	old_std	per imp
400	20	0.04	0.98	0.16	0.73	0.96
400	25	0.08	0.94	0.2	0.83	0.91
400	30	0.06	0.85	0.21	0.72	0.93
400	35	0.05	0.85	0.18	0.64	0.94
400	40	0.07	0.79	0.22	0.68	0.91
400	45	0.09	0.76	0.23	0.75	0.88
500	20	0.04	0.81	0.17	0.65	0.95
500	25	0.04	1	0.15	0.71	0.96
500	30	0.07	0.93	0.22	0.74	0.92
500	35	0.04	0.94	0.14	0.67	0.96
500	40	0.06	0.92	0.17	0.73	0.93
500	45	0.05	0.8	0.15	0.61	0.94
600	20	0.02	0.84	0.1	0.66	0.98
600	25	0.02	1.04	0.09	0.59	0.98
600	30	0.02	0.88	0.09	0.62	0.98
600	35	0.04	0.9	0.15	0.64	0.96
600	40	0.06	0.78	0.22	0.63	0.92
600	45	0.02	0.86	0.08	0.52	0.98
700	20	0.02	0.94	0.12	0.66	0.98
700	25	0.03	0.84	0.11	0.58	0.96
700	30	0.02	0.8	0.07	0.54	0.98
700	35	0.03	0.83	0.1	0.59	0.96
700	40	0.03	0.81	0.16	0.53	0.96
700	45	0.04	0.77	0.14	0.6	0.95
800	20	0.01	0.94	0.06	0.52	0.99
800	25	0.01	0.91	0.06	0.5	0.99
800	30	0.02	0.84	0.09	0.54	0.98
800	35	0.01	0.79	0.06	0.5	0.99
800	40	0.02	0.89	0.11	0.59	0.98
800	45	0.02	0.87	0.08	0.6	0.98
900	20	0.01	0.87	0.08	0.54	0.99
900	25	0.02	0.78	0.1	0.51	0.97
900	30	0.03	0.81	0.12	0.55	0.96
900	35	0.01	0.76	0.07	0.46	0.99
900	40	0.01	0.79	0.09	0.5	0.99
900	45	0.02	0.83	0.1	0.5	0.98
1000	20	0.01	0.85	0.05	0.55	0.99
1000	25	0.01	0.78	0.08	0.49	0.99
1000	30	0.02	0.84	0.08	0.5	0.98
1000	35	0.01	0.78	0.03	0.47	0.99
1000	40	0.02	0.72	0.1	0.49	0.97
1000	45	0.02	0.82	0.08	0.48	0.98

Table 2
Comparing the proposed algorithm to the existing one – Positive Linear Distribution

n	**delta**	**new av**	**old av**	**new std**	**old std**	**per imp**
100	20	0.3	1.3	0.72	1.2	0.77
100	25	0.29	1.09	0.59	1.2	0.73
100	30	0.28	0.92	0.6	0.96	0.70
100	35	0.24	1.33	0.58	1.27	0.82
100	40	0.28	1.24	0.59	1.26	0.77
100	45	0.22	1.26	0.55	1.17	0.83
200	20	0.22	0.95	0.45	0.97	0.77
200	25	0.15	0.92	0.35	0.89	0.84
200	30	0.09	1.02	0.25	0.89	0.91
200	35	0.14	1.07	0.33	0.95	0.87
200	40	0.13	0.85	0.31	0.89	0.85
200	45	0.15	0.92	0.39	0.85	0.84
300	20	0.1	0.89	0.29	0.75	0.89
300	25	0.12	0.89	0.36	0.72	0.87
300	30	0.15	0.82	0.34	0.71	0.82
300	35	0.06	1	0.16	0.78	0.94
300	40	0.08	1.04	0.21	0.87	0.92
300	45	0.13	0.85	0.34	0.78	0.85
400	20	0.04	0.98	0.19	0.63	0.96
400	25	0.09	0.74	0.21	0.71	0.88
400	30	0.08	0.83	0.26	0.66	0.90
400	35	0.06	0.88	0.19	0.7	0.93

Table 2
Comparing the proposed algorithm to the existing one – Positive Linear Distribution

n	delta	new av	old av	new std	old std	per imp
400	40	0.07	0.93	0.27	0.6	0.92
400	45	0.06	0.9	0.22	0.67	0.93
500	20	0.05	0.91	0.18	0.64	0.95
500	25	0.05	0.83	0.15	0.6	0.94
500	30	0.06	0.74	0.18	0.62	0.92
500	35	0.08	0.76	0.24	0.62	0.89
500	40	0.05	0.73	0.15	0.58	0.93
500	45	0.02	0.78	0.11	0.61	0.97
600	20	0.03	0.78	0.1	0.58	0.96
600	25	0.01	0.9	0.06	0.57	0.99
600	30	0.05	0.82	0.18	0.57	0.94
600	35	0.04	0.78	0.15	0.53	0.95
600	40	0.02	0.85	0.08	0.56	0.98
600	45	0.02	0.79	0.09	0.6	0.97
700	20	0.02	0.84	0.11	0.53	0.98
700	25	0.01	0.8	0.06	0.56	0.99
700	30	0.03	0.72	0.13	0.51	0.96
700	35	0.03	0.81	0.1	0.52	0.96
700	40	0.02	0.86	0.09	0.62	0.98
700	45	0.03	0.72	0.12	0.51	0.96
800	20	0.01	0.84	0.07	0.49	0.99
800	25	0.02	0.75	0.09	0.51	0.97
800	30	0.02	0.89	0.07	0.56	0.98
800	35	0.03	0.77	0.1	0.55	0.96
800	40	0.02	0.83	0.1	0.49	0.98
800	45	0.03	0.75	0.12	0.52	0.96
900	20	0.01	0.84	0.05	0.52	0.99
900	25	0.01	0.86	0.06	0.53	0.99
900	30	0.03	0.81	0.1	0.53	0.96
900	35	0.02	0.74	0.07	0.51	0.97
900	40	0.03	0.77	0.12	0.54	0.96
900	45	0.02	0.8	0.1	0.52	0.98
1000	20	0.01	0.72	0.06	0.48	0.99
1000	25	0.02	0.75	0.06	0.49	0.97
1000	30	0.01	0.78	0.05	0.56	0.99
1000	35	0.02	0.81	0.1	0.5	0.98
1000	40	0.01	0.73	0.06	0.47	0.99
1000	45	0.01	0.81	0.04	0.48	0.99

Table 3
Comparing the proposed algorithm to the existing one– Negative Linear Distribution

n	delta	new av	old av	new std	old std	per imp
100	20	0.28	1.15	0.57	1.2	0.76
100	25	0.36	1.1	0.7	1.13	0.67
100	30	0.25	1.03	0.5	1.05	0.76
100	35	0.21	1.24	0.47	1.21	0.83
100	40	0.22	1.05	0.5	1.04	0.79
100	45	0.38	1.04	0.73	1.15	0.63
200	20	0.12	1.1	0.41	0.91	0.89
200	25	0.11	0.99	0.29	0.92	0.89
200	30	0.11	1.04	0.3	0.94	0.89
200	35	0.13	1.13	0.38	0.97	0.88
200	40	0.16	0.99	0.35	0.94	0.84
200	45	0.16	0.93	0.34	0.91	0.83
300	20	0.04	0.9	0.13	0.69	0.96
300	25	0.07	0.99	0.24	0.78	0.93
300	30	0.05	0.98	0.2	0.69	0.95
300	35	0.05	0.96	0.16	0.7	0.95
300	40	0.09	1.02	0.25	0.81	0.91
300	45	0.07	0.94	0.2	0.77	0.93
400	20	0.05	0.88	0.17	0.73	0.94
400	25	0.04	0.93	0.15	0.71	0.96
400	30	0.04	0.99	0.15	0.71	0.96
400	35	0.07	0.92	0.18	0.73	0.92
400	40	0.08	0.84	0.21	0.78	0.90
400	45	0.07	0.92	0.18	0.8	0.92

Table 3

Comparing the proposed algorithm to the existing one– Negative Linear Distribution (Continued)

n	delta	new av	old av	new std	old std	per imp
500	20	0.02	1	0.11	0.68	0.98
500	25	0.04	0.96	0.14	0.73	0.96
500	30	0.02	0.85	0.13	0.63	0.98
500	35	0.05	0.79	0.18	0.57	0.94
500	40	0.08	0.79	0.24	0.66	0.90
500	45	0.04	0.93	0.14	0.74	0.96
600	20	0.03	0.82	0.11	0.57	0.96
600	25	0.04	0.84	0.17	0.63	0.95
600	30	0.03	0.85	0.12	0.64	0.96
600	35	0.06	0.86	0.16	0.64	0.93
600	40	0.03	0.83	0.12	0.6	0.96
600	45	0.04	0.84	0.17	0.57	0.95
700	20	0.01	0.9	0.06	0.63	0.99
700	25	0.02	0.91	0.07	0.58	0.98
700	30	0.01	0.94	0.04	0.65	0.99
700	35	0.04	0.77	0.13	0.54	0.95
700	40	0.02	0.89	0.07	0.55	0.98
700	45	0.04	0.83	0.12	0.56	0.95
800	20	0.03	0.89	0.11	0.54	0.97
800	25	0.02	0.87	0.08	0.58	0.98
800	30	0.01	0.85	0.06	0.51	0.99
800	35	0.04	0.75	0.12	0.54	0.95
800	40	0.05	0.79	0.17	0.51	0.94
800	45	0.01	0.92	0.06	0.53	0.99
900	20	0.01	0.89	0.06	0.55	0.99
900	25	0	0.83	0.03	0.48	1.00
900	30	0.02	0.87	0.07	0.5	0.98
900	35	0.04	0.8	0.19	0.52	0.95
900	40	0.01	0.77	0.05	0.48	0.99
900	45	0.01	0.81	0.06	0.5	0.99
1000	20	0.01	0.84	0.04	0.49	0.99
1000	25	0.02	0.78	0.09	0.47	0.97
1000	30	0.01	0.84	0.06	0.49	0.99
1000	35	0.01	0.79	0.05	0.53	0.99
1000	40	0.01	0.83	0.04	0.46	0.99
1000	45	0.02	0.75	0.07	0.47	0.97

Table 4

Comparing the proposed algorithm to the existing one – Normal Distribution

n	delta	new av	old av	new std	old std	per imp
100	20	0.24	1.37	0.53	1.28	0.82
100	25	0.38	1.04	0.79	1.07	0.63
100	30	0.32	1.1	0.63	1.15	0.71
100	35	0.28	1.12	0.58	1.37	0.75
100	40	0.33	0.89	0.62	1.05	0.63
100	45	0.29	1.41	0.77	1.24	0.79
200	20	0.13	1.05	0.37	0.91	0.88
200	25	0.1	0.99	0.28	0.91	0.90
200	30	0.13	1.05	0.35	0.91	0.88
200	35	0.19	1.11	0.5	1.06	0.83
200	40	0.12	0.98	0.29	0.84	0.88
200	45	0.1	1.09	0.25	0.97	0.91
300	20	0.11	0.92	0.28	0.83	0.88
300	25	0.09	0.79	0.26	0.75	0.89
300	30	0.1	0.99	0.28	0.86	0.90
300	35	0.07	0.9	0.18	0.82	0.92
300	40	0.07	0.96	0.25	0.71	0.93
300	45	0.1	0.91	0.27	0.79	0.89
400	20	0.04	0.86	0.12	0.66	0.95
400	25	0.05	1	0.17	0.73	0.95
400	30	0.04	0.93	0.16	0.69	0.96
400	35	0.06	0.84	0.2	0.67	0.93
400	40	0.08	0.86	0.23	0.73	0.91
400	45	0.06	0.81	0.17	0.72	0.93
500	20	0.01	0.89	0.04	0.63	0.99
500	25	0.06	0.86	0.19	0.61	0.93
500	30	0.05	0.84	0.16	0.65	0.94
500	35	0.05	0.88	0.15	0.67	0.94

Table 4
Comparing the proposed algorithm to the existing one – Normal Distribution (Continued)

n	delta	new av	old av	new std	old std	per imp
500	40	0.06	0.87	0.18	0.65	0.93
500	45	0.06	0.85	0.18	0.7	0.93
600	20	0.03	0.87	0.13	0.58	0.97
600	25	0.03	0.81	0.1	0.6	0.96
600	30	0.03	0.88	0.1	0.66	0.97
600	35	0.03	0.84	0.11	0.54	0.96
600	40	0.04	0.8	0.12	0.62	0.95
600	45	0.06	0.83	0.2	0.59	0.93
700	20	0	0.87	0	0.54	1.00
700	25	0.01	0.9	0.06	0.57	0.99
700	30	0.01	0.88	0.07	0.51	0.99
700	35	0.03	0.83	0.13	0.55	0.96
700	40	0.03	0.77	0.12	0.55	0.96
700	45	0.04	0.72	0.15	0.57	0.94
800	20	0.01	0.84	0.06	0.52	0.99
800	25	0.01	0.84	0.07	0.5	0.99
800	30	0.03	0.82	0.09	0.54	0.96
800	35	0.01	0.85	0.06	0.49	0.99
800	40	0.03	0.71	0.11	0.47	0.96
800	45	0.01	0.85	0.06	0.52	0.99
900	20	0.01	0.81	0.07	0.49	0.99
900	25	0.01	0.9	0.03	0.49	0.99
900	30	0.02	0.78	0.09	0.51	0.97
900	35	0.01	0.8	0.05	0.51	0.99
900	40	0.03	0.79	0.1	0.54	0.96
900	45	0.01	0.75	0.04	0.48	0.99
1000	20	0.01	0.84	0.03	0.49	0.99
1000	25	0.01	0.82	0.04	0.44	0.99
1000	30	0.03	0.8	0.13	0.45	0.96
1000	35	0.02	0.84	0.09	0.49	0.98
1000	40	0.01	0.7	0.06	0.53	0.99
1000	45	0.02	0.88	0.07	0.51	0.98

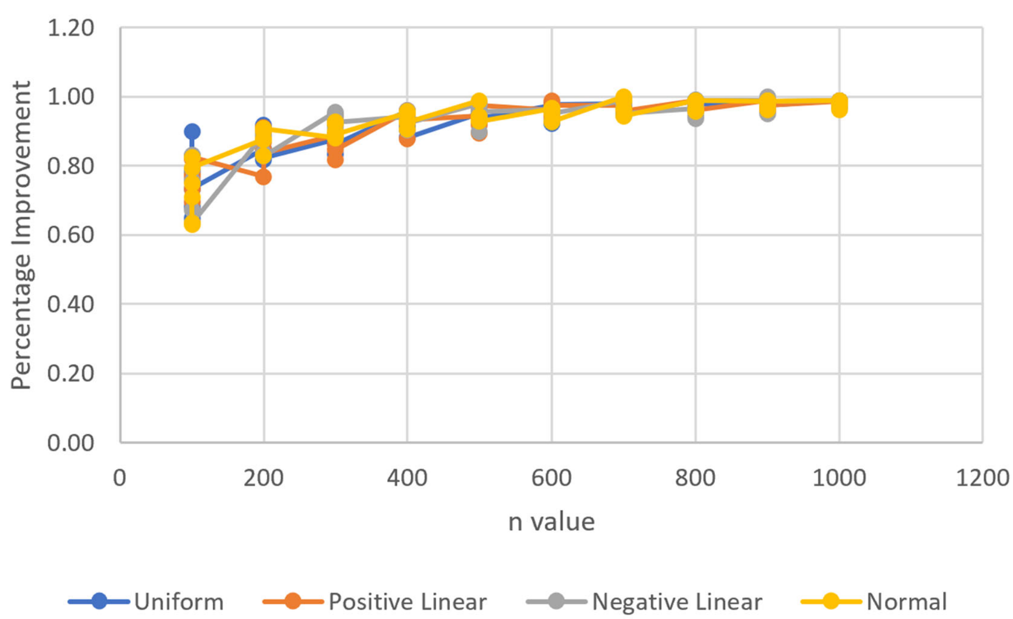

Fig. 1. Percentage Improvement vs n

As seen in the tables, the average and median of the percentage improvement is generally the same across different distributions with just a difference of 1%, as can be seen in Table 5. This confirms the effectiveness of the proposed algorithm over the existing one and indicates that it is not dependent on a particular distribution but rather works on all four of them.

Table 5
Averages and medians for percentage improvement

Distribution	Average per imp	Median of per imp
Uniform	0.92	0.96
Positive Linear	0.92	0.95
Negative Linear	0.93	0.95
Normal	0.93	0.95

Furthermore, the percentage improvement seems to improve with greater n, which is advantageous, since it implies that the algorithm will likely work for even larger values of n, perhaps with a greater effectiveness.

5. Hypothesis testing

A hypothesis test is conducted for a difference of means to determine the degree of improvement obtained by the new algorithm. We want to check whether the average total completion time for the proposed algorithm is indeed lower than that of the existing one. From now on, the best existing algorithm in literature is denoted as old-algorithm and the proposed algorithm is denoted as new-algorithm.
Let μ_0 be the population mean for the TCT of old-algorithm and μ_1 be that of new-algorithm. We define the null and alternative hypotheses as follows:

$H_0 : \mu_0 - \mu_1 = 0$
$H_1 : \mu_0 - \mu_1 > 0$

Hence, if μ_1 is considerably less than μ_0, we reject the null hypothesis that new-algorithm gives us similar total completion times as old-algorithm. Otherwise, we fail to reject the null hypothesis. The level of significance is taken to be $\alpha = 0.01$.
Given a certain distribution, 100 replications are performed for every combination of n and Δ, so the sample size is large enough to use the Z-test. Since the significance level α was taken to be 0.01 and since $P(Z \leq 2.33) = 0.99$, we reject the null hypothesis if the Z-score is greater than 2.33 and fail to reject it otherwise. The calculated Z-scores are listed in Table 6 for each n and Δ combination for all considered distributions. As seen in the table, all the Z-scores for every combination of n and Δ are much greater than 2.33, clearly rejecting the null hypothesis. Furthermore, the Z-scores seem to increase as n increases, which seems to indicate that this result is true for greater values of n as well.

Table 6
Z-scores

n	Δ	Uniform	Positive Linear	Negative Linear	Normal
100	20	8.07	7.15	6.55	8.16
100	25	5.42	5.98	5.57	4.96
100	30	4.95	5.65	6.71	5.95
100	35	7.14	7.81	7.93	5.65
100	40	10.42	6.9	7.19	4.59
100	45	6.34	8.04	4.85	7.67
200	20	8.22	6.83	9.82	9.37
200	25	9.74	8.05	9.12	9.35
200	30	10.46	10.06	9.43	9.44
200	35	7.43	9.25	9.6	7.85
200	40	8.8	7.64	8.27	9.68
200	45	7.78	8.23	7.93	9.88
300	20	9.58	9.82	12.25	9.25
300	25	10.02	9.57	11.27	8.82
300	30	10.92	8.51	12.95	9.84
300	35	8.72	11.81	12.67	9.89
300	40	10.57	10.73	10.97	11.82
300	45	9.12	8.46	10.94	9.7
400	20	12.58	14.29	11.07	12.22
400	25	10.07	8.78	12.26	12.67
400	30	10.53	10.57	13.09	12.57
400	35	12.03	11.31	11.31	11.16
400	40	10.07	13.07	9.41	10.19
400	45	8.54	11.91	10.37	10.14
500	20	11.46	12.94	14.23	13.94
500	25	13.23	12.61	12.38	12.52
500	30	11.14	10.53	12.9	11.8
500	35	13.15	10.23	12.38	12.09
500	40	11.47	11.35	10.11	12.01
500	45	11.94	12.26	11.82	10.93
600	20	12.28	12.74	13.61	14.13
600	25	17.09	15.53	12.26	12.82

Table 6
Z-scores (Continued)

n	Δ	Uniform	Positive Linear	Negative Linear	Normal
600	30	13.73	12.88	12.59	12.73
600	35	13.08	13.43	12.13	14.7
600	40	10.79	14.67	13.07	12.03
600	45	15.97	12.69	13.45	12.36
700	20	13.71	15.15	14.06	16.11
700	25	13.72	14.03	15.23	15.53
700	30	14.32	13.11	14.28	16.9
700	35	13.37	14.73	13.14	14.16
700	40	14.09	13.41	15.69	13.15
700	45	11.85	13.17	13.79	11.54
800	20	17.77	16.77	15.61	15.86
800	25	17.87	14.1	14.52	16.44
800	30	14.98	15.42	16.36	14.43
800	35	15.49	13.24	12.84	17.02
800	40	14.5	16.2	13.77	14.09
800	45	14.04	13.49	17.06	16.05
900	20	15.75	15.89	15.91	16.16
900	25	14.62	15.94	17.26	18.13
900	30	13.86	14.46	16.84	14.68
900	35	16.12	13.99	13.73	15.42
900	40	15.35	13.38	15.75	13.84
900	45	15.89	14.73	15.89	15.36
1000	20	15.21	14.68	16.88	16.91
1000	25	15.51	14.79	15.88	18.33
1000	30	16.19	13.7	16.81	16.44
1000	35	16.35	15.49	14.65	16.46
1000	40	14	15.2	17.76	12.94
1000	45	16.44	16.61	15.36	16.71

6. Confidence interval

The following table lists the 95% confidence intervals with respect to the new algorithm for the four distributions: uniform, positive linear, negative linear, and normal. It is evident that the confidence intervals are narrow, which is advantageous as they indicate the accuracy in the calculated means.

Table 7
95% Confidence Intervals for new-algorithm

Δ	n	Uniform	Positive Lin.	Negative Lin.	Normal
20	100	(0.12, 0.34)	(0.18, 0.42)	(0.19, 0.37)	(0.14, 0.34)
25	100	(0.19, 0.49)	(0.19, 0.39)	(0.24, 0.48)	(0.23, 0.53)
30	100	(0.21, 0.47)	(0.18, 0.38)	(0.17, 0.33)	(0.2, 0.44)
35	100	(0.13, 0.39)	(0.14, 0.34)	(0.13, 0.29)	(0.17, 0.39)
40	100	(0.07, 0.21)	(0.18, 0.38)	(0.14, 0.3)	(0.21, 0.45)
45	100	(0.18, 0.4)	(0.13, 0.31)	(0.26, 0.5)	(0.14, 0.44)
20	200	(0.06, 0.22)	(0.15, 0.29)	(0.05, 0.19)	(0.06, 0.2)
25	200	(0.05, 0.17)	(0.09, 0.21)	(0.06, 0.16)	(0.05, 0.15)
30	200	(0.04, 0.14)	(0.05, 0.13)	(0.06, 0.16)	(0.06, 0.2)
35	200	(0.1, 0.26)	(0.09, 0.19)	(0.07, 0.19)	(0.09, 0.29)
40	200	(0.06, 0.18)	(0.08, 0.18)	(0.1, 0.22)	(0.06, 0.18)
45	200	(0.09, 0.23)	(0.09, 0.21)	(0.1, 0.22)	(0.05, 0.15)
20	300	(0.06, 0.18)	(0.05, 0.15)	(0.02, 0.06)	(0.06, 0.16)
25	300	(0.05, 0.15)	(0.06, 0.18)	(0.03, 0.11)	(0.04, 0.14)
30	300	(0.04, 0.12)	(0.09, 0.21)	(0.02, 0.08)	(0.05, 0.15)
35	300	(0.08, 0.2)	(0.03, 0.09)	(0.02, 0.08)	(0.03, 0.11)
40	300	(0.05, 0.17)	(0.05, 0.11)	(0.05, 0.13)	(0.02, 0.12)
45	300	(0.05, 0.21)	(0.07, 0.19)	(0.04, 0.1)	(0.05, 0.15)
20	400	(0.01, 0.07)	(0.01, 0.07)	(0.02, 0.08)	(0.02, 0.06)
25	400	(0.04, 0.12)	(0.06, 0.12)	(0.02, 0.06)	(0.02, 0.08)
30	400	(0.02, 0.1)	(0.04, 0.12)	(0.02, 0.06)	(0.01, 0.07)
35	400	(0.01, 0.09)	(0.03, 0.09)	(0.04, 0.1)	(0.02, 0.1)
40	400	(0.03, 0.11)	(0.03, 0.11)	(0.05, 0.11)	(0.03, 0.13)
45	400	(0.04, 0.14)	(0.02, 0.1)	(0.04, 0.1)	(0.03, 0.09)
20	500	(0.01, 0.07)	(0.02, 0.08)	(0, 0.04)	(0, 0.02)
25	500	(0.01, 0.07)	(0.03, 0.07)	(0.02, 0.06)	(0.02, 0.1)
30	500	(0.03, 0.11)	(0.03, 0.09)	(0, 0.04)	(0.02, 0.08)
35	500	(0.01, 0.07)	(0.04, 0.12)	(0.02, 0.08)	(0.02, 0.08)
40	500	(0.03, 0.09)	(0.03, 0.07)	(0.04, 0.12)	(0.02, 0.1)

Table 7
95% Confidence Intervals for new-algorithm (Continued)

Δ	n	Uniform	Positive Lin.	Negative Lin.	Normal
45	500	(0.02, 0.08)	(0, 0.04)	(0.02, 0.06)	(0.02, 0.1)
20	600	(0, 0.04)	(0.01, 0.05)	(0.01, 0.05)	(0, 0.06)
25	600	(0, 0.04)	(0, 0.02)	(0.01, 0.07)	(0.01, 0.05)
30	600	(0, 0.04)	(0.02, 0.08)	(0.01, 0.05)	(0.01, 0.05)
35	600	(0.01, 0.07)	(0.02, 0.06)	(0.03, 0.09)	(0.01, 0.05)
40	600	(0.02, 0.1)	(0.01, 0.03)	(0.01, 0.05)	(0.02, 0.06)
45	600	(0, 0.04)	(0.01, 0.03)	(0.01, 0.07)	(0.02, 0.1)
20	700	(0, 0.04)	(0, 0.04)	(0, 0.02)	(0, 0)
25	700	(0.01, 0.05)	(0, 0.02)	(0.01, 0.03)	(0, 0.02)
30	700	(0.01, 0.03)	(0.01, 0.05)	(0, 0.02)	(0, 0.02)
35	700	(0.01, 0.05)	(0.01, 0.05)	(0.02, 0.06)	(0, 0.06)
40	700	(0, 0.06)	(0.01, 0.03)	(0.01, 0.03)	(0.01, 0.05)
45	700	(0.01, 0.07)	(0.01, 0.05)	(0.02, 0.06)	(0.01, 0.07)
20	800	(0, 0.02)	(0, 0.02)	(0.01, 0.05)	(0, 0.02)
25	800	(0, 0.02)	(0.01, 0.03)	(0.01, 0.03)	(0, 0.02)
30	800	(0, 0.04)	(0.01, 0.03)	(0, 0.02)	(0.01, 0.05)
35	800	(0, 0.02)	(0.01, 0.05)	(0.02, 0.06)	(0, 0.02)
40	800	(0, 0.04)	(0, 0.04)	(0.02, 0.08)	(0.01, 0.05)
45	800	(0, 0.04)	(0.01, 0.05)	(0, 0.02)	(0, 0.02)
20	900	(-0.01, 0.03)	(0, 0.02)	(0, 0.02)	(0, 0.02)
25	900	(0, 0.04)	(0, 0.02)	(0, 0)	(0, 0.02)
30	900	(0.01, 0.05)	(0.01, 0.05)	(0.01, 0.03)	(0, 0.04)
35	900	(0, 0.02)	(0.01, 0.03)	(0.01, 0.07)	(0, 0.02)
40	900	(-0.01, 0.03)	(0.01, 0.05)	(0, 0.02)	(0.01, 0.05)
45	900	(0, 0.04)	(0, 0.04)	(0, 0.02)	(0, 0.02)
20	1000	(0, 0.02)	(0, 0.02)	(0, 0.02)	(0, 0.02)
25	1000	(-0.01, 0.03)	(0.01, 0.03)	(0.01, 0.03)	(0, 0.02)
30	1000	(0, 0.04)	(0, 0.02)	(0, 0.02)	(0, 0.06)
35	1000	(0, 0.02)	(0, 0.04)	(0, 0.02)	(0, 0.04)
40	1000	(0, 0.04)	(0, 0.02)	(0, 0.02)	(0, 0.02)
45	1000	(0, 0.04)	(0, 0.02)	(0.01, 0.03)	(0.01, 0.03)

7. Conclusion

Minimizing the total completion time (TCT) for a two-machine no-wait flowshop where setup times are unknown and bounded is an NP-hard problem. To find an effective algorithm which minimizes the TCT, the variability and uncertainty of setup times must be taken into account. Otherwise, the uncertainty of setup times may negatively impact the algorithm and result in poor performance. Such an algorithm which considers uncertainty in setup times is established in Allahverdi and Allahverdi (2020). In this paper, a significantly more effective algorithm is proposed and compared to the existing one in literature. The comparison is done through a test of hypothesis where random processing times and setup times (within lower and upper bounds) are generated, and the TCT is computed after applying both the existing algorithm and the proposed one. A total of 100 replications are considered for each case. Moreover, four different distributions are considered to obtain a better picture on this algorithm's performance in real life. A Z-test is conducted, and the results strongly confirm that the proposed algorithm performs substantially better than the existing one. In fact, the average percentage improvement of the proposed algorithm over the existing one is $92 - 93\%$ with a median of $95 - 96\%$. The fact that the average and medians are similar imply the accurate representation of the average for the samples taken.

References

Allahverdi, A. (2005). Two-machine flowshop scheduling problem to minimize makespan with bounded setup and processing times. *Int. Journal of Agile Manufacturing*, *8,*145–153.

Allahverdi, A. (2006a). Two-machine flowshop scheduling problem to minimize maximum lateness with bounded setup and processing times. *Kuwait Journal of Science and Engineering*, *33*(2), 233–251.

Allahverdi, A. (2006b). Two-machine flowshop scheduling problem to minimize total completion time with bounded setup and processing times. *International Journal of Production Economics*, *103*, 386–400.

Allahverdi, A. (2015). The third comprehensive survey on scheduling problems with setup times/costs. *European Journal of Operational Research*, *246*(2), 345–378.

Allahverdi, A. (2016). A survey of scheduling problems with no-wait in process. *European Journal of Operational Research*, *255*(3), 665–686.

Allahverdi, A., Aldowaisan, T., & Sotskov, Y. (2003). Two-machine flowshop scheduling problem to minimize makespan or total completion time with random and bounded setup times. *International Journal of Mathematics and Mathematical Sciences*, *39*, 2475–2486.

Allahverdi, M., & Allahverdi, A. (2020). Minimizing total completion time in a two-machine no-wait flowshop with uncertain and bounded setup times. *Journal of Industrial and Management Optimization*, *16*(5), 2439–2457.

Baker, K. R., & Trietsch, D. (2009). *Principles of Sequencing and Scheduling*.
Cunningham, A. A., & Dutta, S. K. (1973). Scheduling jobs, with exponentially distributed processing times, on two machines of a flow shop. *Naval Research Logistics Quarterly, 20*(1), 69-81.
Hall, N. G., & Sriskandarajah, C. (1996). A survey of machine scheduling problems with blocking and no-wait in process. *Operations Research, 44*(3), 510-525.
Kalczynski, P. J., & Kamburowski, J. (2006). A heuristic for minimizing the expected makespan in two-machine flow shops with consistent coefficients of variation. *European Journal of Operational Research, 169*(3), 742-750.
Kim, S. C., & Bobrowski, P. M. (1997). Scheduling jobs with uncertain setup times and sequence dependency. *Omega, 25*(4), 437-447.
Kopanos, G. M., Laínez, J. M., & Puigjaner, L. (2009). An efficient mixed-integer linear programming scheduling framework for addressing sequence-dependent setup issues in batch plants. *Industrial & Engineering Chemistry Research, 48*(13), 6346-6357.
Ku, P. S., & Niu, S. C. (1986). On Johnson's two-machine flow shop with random processing times. *Operations Research*, *34*(1), 130-136.
Li, X., Yang, Z., Ruiz, R., Chen, T., & Sui, S. (2018). An iterated greedy heuristic for no-wait flow shops with sequence dependent setup times, learning and forgetting effects. *Information Sciences*, *453*, 408–425.
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.
Soroush, H. M. (1999). Sequencing and due-date determination in the stochastic single machine problem with earliness and tardiness costs. *European Journal of Operational Research*, *113*(2), 450-468.
Soroush, H. M. (2007). Minimizing the weighted number of early and tardy jobs in a stochastic single machine scheduling problem. *European Journal of Operational Research*, *181*(1), 266-287.
Ying, K. C., & Lin, S. W. (2018). Minimizing makespan for no-wait flowshop scheduling problems with setup times. *Computers & Industrial Engineering*, *121*, 73-81.