

MINIMIZING TOTAL COMPLETION TIME FOR FLOWSHOP SCHEDULING PROBLEM WITH UNCERTAIN PROCESSING TIMES

MUBERRA ALLAHVERDI^{1,*} AND ALI ALLAHVERDI²

Abstract. The four-machine flowshop scheduling problem is investigated with the objective of minimizing total completion time. Job processing times are uncertain where only the lower and upper bounds are known. This problem is common in some manufacturing environments. Some mathematical (dominance) relations are established, and an algorithm (with ten scenarios) is proposed. The proposed algorithm converts the four-machine problem to a single machine problem for which an optimal solution is known for the deterministic problem. The difference among the scenarios is related to the weights assigned to the lower and upper bounds of processing times on the machines. The proposed algorithm is further improved by the established mathematical relations and are evaluated based on extensive computational experiments. The computational results indicate that three scenarios of the proposed algorithm perform much better than the others, and the errors of these three scenarios get better as the size of the problem increases. The results are statistically verified by constructing the confidence intervals.

Mathematics Subject Classification. 90-XX.

Received October 4, 2019. Accepted February 19, 2020.

1. INTRODUCTION

We consider the flowshop scheduling problem which is very common in the production scheduling. Practical research and case studies in production scheduling was recently surveyed [20] where they considered different industries including furniture, electronics, food, and pharmaceuticals. Fuchigami and Rangel [20] analyzed the considered practical problems and concluded that about 70% of considered real life case studies were in the flowshop scheduling area. Fuchigami and Rangel [20] also described the applications of four-machine flowshop scheduling problem, which is the focus of this paper. It should be noted that they described the case when processing times are certain. As we will describe below, there exist environments where processing times are uncertain. Therefore, in this paper, we address the environments where processing times are uncertain.

The performance measure of total completion time is an important performance measure in production scheduling as it affects the inventory levels and lead times [18]. Moreover, minimizing this performance measure not only enables a time-based competition but also results in reduced work in process [19]. Furthermore, the total completion time needs to be minimized for companies to remain competitive [17].

Keywords. Scheduling, total completion time, uncertain processing times.

¹ Department of Mathematical Sciences, School of Natural Sciences, Kean University, 1000 Morris Ave, Union, NJ 07083, USA.

² Department of Industrial and Management Systems Engineering, Kuwait University, P.O. Box 5969, Safat, Kuwait.

*Corresponding author: muallahv@kean.edu

Processing times are assumed to be known with fixed values in the vast majority of the scheduling literature as pointed out [22, 27]. This assumption is certainly valid for some manufacturing systems. On the other hand, the assumption is not valid for some other real-world manufacturing systems as a wide range of uncertainties exist in them [21, 30]. Lack of experience, machine operator fatigue, condition of auxiliary devices for holding jobs at appropriate positions on machines, untested processing technology, conditions of the tools are some factors causing uncertainty in job processing times, *e.g.*, Tayanithi *et al.* [29]. Moreover, past data are not available for new jobs where only some bounds can be predicted. Hence, processing times cannot be modeled as deterministic for such manufacturing systems.

Kouvelis and Yu [23] pointed out that modeling job processing times by certain probability distributions is not appropriate for all manufacturing systems as factors such as worker skill levels, supplier yield problems, machine conditions determine uncertainty in processing times. Although processing time distributions may not be known, it has been observed that upper and lower bounds on processing times are easy to obtain in many manufacturing systems, *e.g.*, Sotskov *et al.* [28].

We consider manufacturing systems with uncertain processing times where only lower and upper bounds are known. The processing times are random variables where the probability distributions are not known. Let $t_{j,m}$ denote the processing time of job j ($j \in I = \{1, 2, \dots, n\}$) on machine m ($m \in M = \{1, \dots, 4\}$). $t_{j,m}$ satisfies $LBt_{j,m} \leq t_{j,m} \leq UBt_{j,m}$ where $LBt_{j,m}$ denotes the lower bound and $UBt_{j,m}$ denotes the upper bound of $t_{j,m}$. The only known information about jobs is the lower and upper bounds.

The considered problem has been investigated in the literature for two or three machines. More specifically, dominance relations were given [12] for the problem of $F2/LBt_{j,m} \leq t_{j,m} \leq UBt_{j,m}/C_{\max}$. Allahverdi and Aydilek [6] also investigated the same problem where they proposed several polynomial time algorithms. They indicated by computational experiments that one of their algorithms yields close to the optimal solution. The two machine problem with uncertain processing times was addressed [7] with respect to maximum lateness, *i.e.*, $F2/LBt_{j,m} \leq t_{j,m} \leq UBt_{j,m}/L_{\max}$. They provided different heuristics and showed that one of their heuristics performs well.

Sotskov *et al.* [28] established some dominance relations for the $F2/LBt_{j,m} \leq t_{j,m} \leq UBt_{j,m}/\sum C_i$ problem where the objective was to minimize total completion time. Aydilek and Allahverdi [16] also addressed the same problem and proposed different heuristics, which use the lower and upper bounds on job processing times. They indicated by computational experiments that the performance of one of their heuristics was superior for the considered different distributions of processing times. The three-machine case was investigated [28]. They established some dominance relations for the $F3/LBt_{j,m} \leq t_{j,m} \leq UBt_{j,m}/\sum C_i$ problem. Allahverdi and Allahverdi [9] provided some dominance relations and proposed several algorithms for the $F4/LBt_{j,m} \leq t_{j,m} \leq UBt_{j,m}/C_{\max}$ problem.

Different scheduling environments have also been investigated in the scheduling literature when job processing times are uncertain and only the lower and upper bounds are known, *e.g.*, [13–15, 24, 25].

We investigate the four-machine flowshop scheduling problem with the objective of minimizing total completion time with uncertain processing times within some bounds, *i.e.*, $F4/LBt_{j,m} \leq t_{j,m} \leq UBt_{j,m}/\sum C_i$ in this paper. We establish some dominance relations and propose an algorithm with ten scenarios for the problem. Formulation of the problem along with some dominance relations are presented in the next section. The proposed algorithm is described in Section 3, while evaluation of the proposed algorithm with ten scenarios is conducted in Section 4. Lastly, Section 5 presents the conclusions.

2. NOTATION AND DOMINANCE RELATIONS

In this section, we first denote the notation used in the paper in Section 2.1 while the problem formulation and dominance relations are described in Section 2.2.

2.1. Notation

n : Number of jobs
 m : Number of machines
 TCT : Total completion time
 $t_{j,m}$: Processing time of job j on machine m
 $t_{[j,m]}$: Processing time of the job in position j on machine m
 $C_{[j]}$: Completion time of the job in position j
 $LBt_{r,k}$: Lower bound of the processing time of job r on machine k
 $UBt_{r,k}$: Upper bound of the processing time of job r on machine k
 $\Omega_{[j,m]}$: Total idle time on the k th machine up to the completion time of the job in position j for machine m

$$SP_{[j,m]} = t_{[1,k]} + \dots + t_{[j,m]} \quad \text{for } k = 1, \dots, 4.$$

2.2. Problem formulation and dominance relations

Let $\omega_{[j,2]} = SP_{[j,1]} - SP_{[j-1,2]}$ where $SP_{[0,2]} = 0$, then it is known that $\Omega_{[j,2]} = \max\{\omega_{[1,2]}, \omega_{[2,2]}, \dots, \omega_{[j,2]}\}$ [1]. Furthermore, let $\omega_{[j,3]} = \Omega_{[j,2]} + SP_{[j,2]} - SP_{[j-1,3]}$ where $SP_{[0,3]} = 0$, it is also known that $\Omega_{[j,3]} = \max\{\omega_{[1,3]}, \omega_{[2,3]}, \dots, \omega_{[j,3]}\}$ [2]. Moreover, let $\omega_{[j,4]} = \Omega_{[j,3]} + SP_{[j,3]} - SP_{[j-1,4]}$ where $SP_{[0,4]} = 0$, then, it can be shown that $\Omega_{[j,4]} = \max\{\omega_{[1,4]}, \omega_{[2,4]}, \dots, \omega_{[j,4]}\}$. Therefore, the completion time of the job in position j is computed as

$$C_{[j]} = SP_{[j,4]} + \Omega_{[j,4]} \quad \text{for } j = 1, 2, \dots, n. \quad (2.1)$$

Therefore, the total completion time is given by

$$TCT = \sum_{j=1}^n C_{[j]}. \quad (2.2)$$

Consider a job sequence ϕ_1 such that job h is an arbitrary position β and job g is in position $\beta + 1$. Let sequence ϕ_2 be derived from sequence ϕ_1 by interchanging only jobs h and g . Let δ_1 denote a subsequence containing the jobs in positions $1, 2, \dots, \beta - 1$, and δ_2 denote a subsequence containing the jobs in positions $\beta + 1, \dots, n$ where n denotes the number of jobs. Note that both subsequences δ_1 and δ_2 comprise of the same jobs in both sequences ϕ_1 and ϕ_2 . Therefore, the sequences ϕ_1 and ϕ_2 can be written as $\phi_1 = \{\delta_1, h, g, \delta_2\}$ and $\phi_2 = \{\delta_1, g, h, \delta_2\}$.

It should be noted that

$$LBt_{r,k} \leq t_{r,k} \leq UBt_{r,k} \quad \text{for } r = g, h \quad \text{and} \quad k = 1, \dots, 4, \quad (2.3)$$

where $LBt_{r,k}$ and $UBt_{r,k}$ denote lower and upper bounds of the processing time of job r on machine k ($t_{r,k}$).

For the two sequences ϕ_1 and ϕ_2 ,

$$\omega_{[\beta,2]}(\phi_1) = SP_{[\beta-1,1]}(\phi_1) + t_{h,1} - SP_{[\beta-1,2]}(\phi_1) \quad (2.4)$$

$$\omega_{[\beta,2]}(\phi_2) = SP_{[\beta-1,1]}(\phi_2) + t_{g,1} - SP_{[\beta-1,2]}(\phi_2) \quad (2.5)$$

$$\omega_{[\beta+1,2]}(\phi_1) = SP_{[\beta-1,1]}(\phi_1) + t_{h,1} + t_{g,1} - SP_{[\beta-1,2]}(\phi_1) - t_{h,2} \quad (2.6)$$

$$\omega_{[\beta+1,2]}(\phi_2) = SP_{[\beta-1,1]}(\phi_2) + t_{g,1} + t_{h,1} - SP_{[\beta-1,2]}(\phi_2) - t_{g,2}. \quad (2.7)$$

In the equations (2.4–2.7)

$$SP_{[\beta-1,1]}(\phi_1) = SP_{[\beta-1,1]}(\phi_2) \quad \text{and} \quad SP_{[\beta-1,2]}(\phi_1) = SP_{[\beta-1,2]}(\phi_2) \quad (2.8)$$

since both two sequences ϕ_1 and ϕ_2 have the same jobs in positions $1, \dots, \beta - 1$.

Moreover, it follows by definition of $\omega_{[j,3]}$ that

$$\omega_{[\beta,3]}(\phi_1) = \max \{ \Omega_{[\beta-1,2]}(\phi_1), \omega_{[\beta,2]}(\phi_1) \} + \text{SP}_{[\beta-1,2]}(\phi_1) + t_{h,2} - \text{SP}_{[\beta-1,3]}(\phi_1) \quad (2.9)$$

$$\omega_{[\beta,3]}(\phi_2) = \max \{ \Omega_{[\beta-1,2]}(\phi_2), \omega_{[\beta,2]}(\phi_2) \} + \text{SP}_{[\beta-1,2]}(\phi_2) + t_{g,2} - \text{SP}_{[\beta-1,3]}(\phi_2) \quad (2.10)$$

$$\begin{aligned} \omega_{[\beta+1,3]}(\phi_1) &= \max \{ \Omega_{[\beta-1,2]}(\phi_1), \omega_{[\beta,2]}(\phi_1), \omega_{[\beta+1,2]}(\phi_1) \} \\ &\quad + \text{SP}_{[\beta-1,2]}(\phi_1) + t_{h,2} + t_{g,2} - \text{SP}_{[\beta-1,3]}(\phi_1) - t_{h,3} \end{aligned} \quad (2.11)$$

$$\begin{aligned} \omega_{[\beta+1,3]}(\phi_2) &= \max \{ \Omega_{[\beta-1,2]}(\phi_2), \omega_{[\beta,2]}(\phi_2), \omega_{[\beta+1,2]}(\phi_2) \} \\ &\quad + \text{SP}_{[\beta-1,2]}(\phi_2) + t_{g,2} + t_{h,2} - \text{SP}_{[\beta-1,3]}(\phi_1) - t_{g,3}. \end{aligned} \quad (2.12)$$

Given that both sequences have the same jobs in positions $1, \dots, \beta-1$,

$$\text{SP}_{[\beta-1,3]}(\phi_1) = \text{SP}_{[\beta-1,3]}(\phi_2). \quad (2.13)$$

It also follows by definition of $\omega_{[j,4]}$ that

$$\omega_{[\beta,4]}(\phi_1) = \max \{ \Omega_{[\beta-1,3]}(\phi_1), \omega_{[\beta,3]}(\phi_1) \} + \text{SP}_{[\beta-1,3]}(\phi_1) + t_{h,3} - \text{SP}_{[\beta-1,4]}(\phi_1) \quad (2.14)$$

$$\omega_{[\beta,4]}(\phi_2) = \max \{ \Omega_{[\beta-1,3]}(\phi_2), \omega_{[\beta,3]}(\phi_2) \} + \text{SP}_{[\beta-1,3]}(\phi_2) + t_{g,3} - \text{SP}_{[\beta-1,4]}(\phi_2) \quad (2.15)$$

$$\begin{aligned} \omega_{[\beta+1,4]}(\phi_1) &= \max \{ \Omega_{[\beta-1,3]}(\phi_1), \omega_{[\beta,3]}(\phi_1), \omega_{[\beta+1,3]}(\phi_1) \} \\ &\quad + \text{SP}_{[\beta-1,3]}(\phi_1) + t_{h,3} + t_{g,3} - \text{SP}_{[\beta-1,4]}(\phi_1) - t_{h,4} \end{aligned} \quad (2.16)$$

$$\begin{aligned} \omega_{[\beta+1,4]}(\phi_2) &= \max \{ \Omega_{[\beta-1,3]}(\phi_2), \omega_{[\beta,3]}(\phi_2), \omega_{[\beta+1,3]}(\phi_2) \} \\ &\quad + \text{SP}_{[\beta-1,3]}(\phi_2) + t_{g,3} + t_{h,3} - \text{SP}_{[\beta-1,4]}(\phi_1) - t_{g,4}. \end{aligned} \quad (2.17)$$

Since the sequences ϕ_1 and ϕ_2 have the same jobs in positions $1, \dots, \beta-1$,

$$\text{SP}_{[\beta-1,4]}(\phi_1) = \text{SP}_{[\beta-1,4]}(\phi_2). \quad (2.18)$$

Theorem 2.1. *If $\text{UB}t_{g,r} \leq \text{LB}t_{h,r}$ for $r = 1, 2, 3$ hold, then*

$$\max \{ \Omega_{[\beta-1,4]}(\phi_2), \omega_{[\beta,4]}(\phi_2) \} \leq \max \{ \Omega_{[\beta-1,4]}(\phi_1), \omega_{[\beta,4]}(\phi_1) \}.$$

Proof of Theorem 2.1. It follows from equations (2.3–2.5) and (2.8) that

$$\omega_{[\beta,2]}(\phi_2) \leq \omega_{[\beta,2]}(\phi_1). \quad (2.19)$$

Since $\text{UB}t_{g,1} \leq \text{LB}t_{h,1}$. Moreover, it follows from equations (2.3), (2.9), (2.10), (2.13), and (2.19) that

$$\omega_{[\beta,3]}(\phi_2) \leq \omega_{[\beta,3]}(\phi_1). \quad (2.20)$$

Since $\text{UB}t_{g,2} \leq \text{LB}t_{h,2}$. Now it follows from equations (2.3), (2.13–2.15), (2.18), (2.20) that

$$\omega_{[\beta,4]}(\phi_2) \leq \omega_{[\beta,4]}(\phi_1). \quad (2.21)$$

Since $\text{UB}t_{g,3} \leq \text{LB}t_{h,3}$. Then, by equation (2.21)

$$\max \{ \Omega_{[\beta-1,4]}(\phi_2), \omega_{[\beta,4]}(\phi_2) \} \leq \max \{ \Omega_{[\beta-1,4]}(\phi_1), \omega_{[\beta,4]}(\phi_1) \}.$$

Since $\Omega_{[\beta-1,4]}(\phi_1) = \Omega_{[\beta-1,4]}(\phi_2)$ as a result of both sequences having the same jobs in positions $1, \dots, \beta-1$. \square

Theorem 2.2. *If $\text{UB}t_{g,r} \leq \text{LB}t_{h,r}$ for $r = 1, 2, 3$ and $\text{UB}t_{h,r} \leq \text{LB}t_{g,r+1}$ for $r = 1, 2, 3$ hold, then*

$$\max \{ \Omega_{[\beta-1,4]}(\phi_2), \omega_{[\beta,4]}(\phi_2), \omega_{[\beta+1,4]}(\phi_2) \} \leq \max \{ \Omega_{[\beta-1,4]}(\phi_1), \omega_{[\beta,4]}(\phi_1), \omega_{[\beta+1,4]}(\phi_1) \}.$$

Proof of Theorem 2.2. It follows from equations (2.5), (2.7) and (2.8) that

$$\omega_{[\beta+1,2]}(\phi_2) \leq \omega_{[\beta,2]}(\phi_2). \quad (2.22)$$

Since $UBt_{h,1} \leq LBt_{g,2}$. Furthermore, it follows from equations (2.10), (2.12), (2.13), (2.22) that

$$\omega_{[\beta+1,3]}(\phi_2) \leq \omega_{[\beta,3]}(\phi_2). \quad (2.23)$$

Since $UBt_{h,2} \leq LBt_{g,3}$. By equations (2.15), (2.17), (2.18) and (2.23),

$$\omega_{[\beta+1,4]}(\phi_2) \leq \omega_{[\beta,4]}(\phi_2). \quad (2.24)$$

Since $UBt_{h,3} \leq LBt_{g,4}$. Since $UBt_{g,r} \leq LBt_{h,r}$ for $r = 1, 2, 3$, equation (2.21) is satisfied.

It then follows from equations (2.21) to (2.24) that

$$\max \{ \Omega_{[\beta-1,4]}(\phi_2), \omega_{[\beta,4]}(\phi_2), \omega_{[\beta+1,4]}(\phi_2) \} \leq \max \{ \Omega_{[\beta-1,4]}(\phi_1), \omega_{[\beta,4]}(\phi_1), \omega_{[\beta+1,4]}(\phi_1) \}.$$

□

Theorem 2.3. *If $UBt_{g,r} \leq LBt_{h,r}$ for $r = 1, 2, 3$ and $UBt_{g,r} \leq LBt_{g,r+1}$ for $r = 1, 2, 3$ hold, then*

$$\max \{ \Omega_{[\beta-1,4]}(\phi_2), \omega_{[\beta,4]}(\phi_2), \omega_{[\beta+1,4]}(\phi_2) \} \leq \max \{ \Omega_{[\beta-1,4]}(\phi_1), \omega_{[\beta,4]}(\phi_1), \omega_{[\beta+1,4]}(\phi_1) \}.$$

Proof of Theorem 2.3. By equations (2.4), (2.5), (2.7), and (2.8),

$$\omega_{[\beta,2]}(\phi_2) \leq \omega_{[\beta,2]}(\phi_1) \quad \text{and} \quad (2.25)$$

$$\omega_{[\beta+1,2]}(\phi_2) \leq \omega_{[\beta,2]}(\phi_1). \quad (2.26)$$

Since $UBt_{g,1} \leq LBt_{h,1}$ and $UBt_{g,1} \leq LBt_{g,2}$. Furthermore, by equations (2.9), (2.10), (2.12), (2.13), (2.25), and (2.26),

$$\omega_{[\beta,3]}(\phi_2) \leq \omega_{[\beta,3]}(\phi_1) \quad \text{and} \quad (2.27)$$

$$\omega_{[\beta+1,3]}(\phi_2) \leq \omega_{[\beta,3]}(\phi_1). \quad (2.28)$$

Since $UBt_{g,2} \leq LBt_{h,2}$ and $UBt_{g,2} \leq LBt_{g,3}$.

Moreover, it follows from equations (2.14), (2.17), (2.18), (2.27), and (2.28) that

$$\omega_{[\beta+1,4]}(\phi_2) \leq \omega_{[\beta,4]}(\phi_1). \quad (2.29)$$

Since $UBt_{g,3} \leq LBt_{g,4}$. Since $UBt_{g,r} \leq LBt_{h,r}$ for $r = 1, 2, 3$, equation (2.21) is satisfied, then it follows from equations (2.21) and (2.25–2.29) that

$$\max \{ \Omega_{[\beta-1,4]}(\phi_2), \omega_{[\beta,4]}(\phi_2), \omega_{[\beta+1,4]}(\phi_2) \} \leq \max \{ \Omega_{[\beta-1,4]}(\phi_1), \omega_{[\beta,4]}(\phi_1), \omega_{[\beta+1,4]}(\phi_1) \}.$$

□

Theorem 2.4. *If the following condition (i) and either (iia) or (iib) hold*

- (i) $UBt_{g,4} \leq LBt_{h,4}$
- (iia) $UBt_{g,r} \leq LBt_{h,r}$ for $r = 1, 2, 3$ and $UBt_{h,r} \leq LBt_{g,r+1}$ for $r = 1, 2, 3$
- (iib) $UBt_{g,r} \leq LBt_{h,r}$ for $r = 1, 2, 3$ and $UBt_{g,r} \leq LBt_{g,r+1}$ for $r = 1, 2, 3$

then

$$[C_{[\beta]}(\phi_2) + C_{[\beta+1]}(\phi_2)] \leq [C_{[\beta]}(\phi_1) + C_{[\beta+1]}(\phi_1)].$$

Proof of Theorem 2.4. For jobs in positions β and $\beta + 1$ in the two sequences ϕ_1 and ϕ_2 ,

$$C_{[\beta]}(\phi_1) = \text{SP}_{[\beta-1,4]}(\phi_1) + t_{h,4} + \max\{\Omega_{[\beta-1,4]}(\phi_1), \omega_{[\beta,4]}(\phi_1)\} \quad (2.30)$$

$$C_{[\beta]}(\phi_2) = \text{SP}_{[\beta-1,4]}(\phi_2) + t_{g,4} + \max\{\Omega_{[\beta-1,4]}(\phi_2), \omega_{[\beta,4]}(\phi_2)\} \quad (2.31)$$

$$C_{[\beta+1]}(\phi_1) = \text{SP}_{[\beta-1,4]}(\phi_1) + t_{h,4} + t_{g,4} + \max\{\Omega_{[\beta-1,4]}(\phi_1), \omega_{[\beta,4]}(\phi_1), \omega_{[\beta+1,4]}(\phi_1)\} \quad (2.32)$$

$$C_{[\beta+1]}(\phi_2) = \text{SP}_{[\beta-1,4]}(\phi_2) + t_{g,4} + t_{h,4} + \max\{\Omega_{[\beta-1,4]}(\phi_2), \omega_{[\beta,4]}(\phi_2), \omega_{[\beta+1,4]}(\phi_2)\}. \quad (2.33)$$

We obtain from equation (2.18) that

$$\begin{aligned} [C_{[\beta]}(\phi_2) + C_{[\beta+1]}(\phi_2)] - [C_{[\beta]}(\phi_1) + C_{[\beta+1]}(\phi_1)] &= t_{g,4} - t_{h,4} \\ \max\{\Omega_{[\beta-1,4]}(\phi_2), \omega_{[\beta,4]}(\phi_2)\} + \max\{\Omega_{[\beta-1,4]}(\phi_2), \omega_{[\beta,4]}(\phi_2), \omega_{[\beta+1,4]}(\phi_2)\} \\ - \max\{\Omega_{[\beta-1,4]}(\phi_1), \omega_{[\beta,4]}(\phi_1)\} - \max\{\Omega_{[\beta-1,4]}(\phi_1), \omega_{[\beta,4]}(\phi_1), \omega_{[\beta+1,4]}(\phi_1)\}. \end{aligned} \quad (2.34)$$

By Theorem 2.1

$$\max\{\Omega_{[\beta-1,4]}(\phi_2), \omega_{[\beta,4]}(\phi_2)\} \leq \max\{\Omega_{[\beta-1,4]}(\phi_1), \omega_{[\beta,4]}(\phi_1)\}. \quad (2.35)$$

Since $\text{UB}t_{g,r} \leq \text{LB}t_{h,r}$ for $r = 1, 2, 3$ from conditions (iia) and (iib) of Theorem 2.4. Moreover, it follows from Theorems 2.2 and 2.3 that

$$\max\{\Omega_{[\beta-1,4]}(\phi_2), \omega_{[\beta,4]}(\phi_2), \omega_{[\beta+1,4]}(\phi_2)\} \leq \max\{\Omega_{[\beta-1,4]}(\phi_1), \omega_{[\beta,4]}(\phi_1), \omega_{[\beta+1,4]}(\phi_1)\} \quad (2.36)$$

if either conditions (iia) or (iib) of Theorem 2.4 holds. Therefore, it follows from equations (2.34) to (2.36) and the condition (i) of Theorem 2.4 that

$$[C_{[\beta]}(\phi_2) + C_{[\beta+1]}(\phi_2)] \leq [C_{[\beta]}(\phi_1) + C_{[\beta+1]}(\phi_1)].$$

□

Theorem 2.5. For $r = \beta + 2, \dots, n$, if either (iia) or (iib) of Theorem 2.4 holds,

$$\omega_{[r,4]}(\phi_2) \leq \omega_{[r,4]}(\phi_1).$$

Proof of Theorem 2.5. For $r = \beta + 2, \dots, n$

$$\begin{aligned} \omega_{[r,4]}(\phi_1) &= \max\{\Omega_{[\beta-1,3]}(\phi_1), \omega_{[\beta,3]}(\phi_1), \omega_{[\beta+1,3]}(\phi_1), \dots, \omega_{[r,3]}(\phi_1)\} \\ &\quad + \text{SP}_{[\beta-1,3]}(\phi_1) + t_{h,3} + t_{g,3} + \sum_{p=\beta+2}^r t_{[p,3]}(\phi_1) \\ &\quad - \text{SP}_{[\beta-1,4]}(\phi_1) - t_{h,4} - t_{g,4} - \sum_{p=\beta+2}^{r-1} t_{[p,4]}(\phi_1) \end{aligned} \quad (2.37)$$

$$\begin{aligned} \omega_{[r,4]}(\phi_2) &= \max\{\Omega_{[\beta-1,3]}(\phi_2), \omega_{[\beta,3]}(\phi_2), \omega_{[\beta+1,3]}(\phi_2), \dots, \omega_{[r,3]}(\phi_2)\} \\ &\quad + \text{SP}_{[\beta-1,3]}(\phi_2) + t_{g,3} + t_{h,3} + \sum_{p=\beta+2}^r t_{[p,3]}(\phi_2) \\ &\quad - \text{SP}_{[\beta-1,4]}(\phi_2) - t_{g,4} - t_{h,4} - \sum_{p=\beta+2}^{r-1} t_{[p,4]}(\phi_2) \end{aligned} \quad (2.38)$$

where $\sum_{p=\beta+2}^{\beta+1} t_{[p,4]}(\phi_1) = \sum_{p=\beta+2}^{\beta+1} t_{[p,4]}(\phi_2) = 0$. Since both sequences ϕ_1 and ϕ_2 have the same jobs in positions $\beta+2, \dots, n$

$$\sum_{p=\beta+2}^r t_{[p,3]}(\phi_1) = \sum_{p=\beta+2}^r t_{[p,3]}(\phi_2) \quad (2.39)$$

$$\sum_{p=\beta+2}^{r-1} t_{[p,4]}(\phi_1) = \sum_{p=\beta+2}^{r-1} t_{[p,4]}(\phi_2) \quad (2.40)$$

$$\omega_{[r,3]}(\phi_1) = \omega_{[r,3]}(\phi_2). \quad (2.41)$$

Then, it follows from equation (2.20) and either equation (2.23) or equation (2.28) that $\max\{\omega_{[\beta,3]}(\phi_2), \omega_{[\beta+1,3]}(\phi_2)\} \leq \max\{\omega_{[\beta,3]}(\phi_1), \omega_{[\beta+1,3]}(\phi_1)\}$.

Hence, equations (2.13), (2.18) and (2.37–2.41) imply

$$\omega_{[r,4]}(\phi_2) \leq \omega_{[r,4]}(\phi_1). \quad (2.42)$$

□

Theorem 2.6. For $r = \beta+2, \dots, n$

$$C_{[r,4]}(\phi_2) \leq C_{[r,4]}(\phi_1)$$

provided that the conditions in either Theorem 2.2 or Theorem 2.3 hold.

Proof of Theorem 2.6. For $r = \beta+2, \dots, n$

$$\begin{aligned} C_{[r]}(\phi_1) &= \text{SP}_{[\beta-1,4]}(\phi_1) + t_{h,4} + t_{g,4} + \sum_{p=\beta+2}^r t_{[p,4]}(\phi_1) \\ &\quad + \max\{\Omega_{[\beta-1,4]}(\phi_1), \omega_{[\beta,4]}(\phi_1), \omega_{[\beta+1,4]}(\phi_1), \dots, \omega_{[r,4]}(\phi_1)\} \end{aligned} \quad (2.43)$$

$$\begin{aligned} C_{[r]}(\phi_2) &= \text{SP}_{[\beta-1,4]}(\phi_2) + t_{h,4} + t_{g,4} + \sum_{p=\beta+2}^r t_{[p,4]}(\phi_2) \\ &\quad + \max\{\Omega_{[\beta-1,4]}(\phi_2), \omega_{[\beta,4]}(\phi_2), \omega_{[\beta+1,4]}(\phi_2), \dots, \omega_{[r,4]}(\phi_2)\}. \end{aligned} \quad (2.44)$$

Note that $\sum_{p=\beta+2}^r t_{[p,4]}(\phi_1) = \sum_{p=\beta+2}^r t_{[p,4]}(\phi_2)$ as both sequences have the same jobs in those positions, and hence, from equations (2.18), (2.43), and (2.44)

$$\begin{aligned} C_{[r]}(\phi_2) - C_{[r]}(\phi_1) &= \max\{\Omega_{[\beta-1,4]}(\phi_2), \omega_{[\beta,4]}(\phi_2), \omega_{[\beta+1,4]}(\phi_2), \dots, \omega_{[r,4]}(\phi_2)\} \\ &\quad - \max\{\Omega_{[\beta-1,4]}(\phi_1), \omega_{[\beta,4]}(\phi_1), \omega_{[\beta+1,4]}(\phi_1), \dots, \omega_{[r,4]}(\phi_1)\}. \end{aligned} \quad (2.45)$$

Then, it follows from equation (2.45) and either Theorem 2.2 or Theorem 2.3 that $C_{[r,4]}(\phi_2) \leq C_{[r,4]}(\phi_1)$. □

Lemma 2.7. For $r = 1, \dots, \beta-1$,

$$C_{[r,4]}(\phi_2) = C_{[r,4]}(\phi_1).$$

Proof of Lemma 2.7. The proof follows from that fact that both sequences have the same jobs in positions $r = 1, \dots, \beta-1$. □

Theorem 2.8. Assume jobs g and h are adjacent in a sequence and they satisfy condition (i) and either (iia) or (iib) given below,

- (i) $\text{UB}t_{g,4} \leq \text{LB}t_{h,4}$
- (iia) $\text{UB}t_{g,r} \leq \text{LB}t_{h,r}$ for $r = 1, 2, 3$ and $\text{UB}t_{h,r} \leq \text{LB}t_{g,r+1}$ for $r = 1, 2, 3$

(iib) $UBt_{g,r} \leq LBt_{h,r}$ for $r = 1, 2, 3$ and $UBt_{g,r} \leq LBt_{g,r+1}$ for $r = 1, 2, 3$

then, $TCT(\phi_2) \leq TCT(\phi_1)$. In other words, given that jobs g and h are adjacent, job g should precede job h in order to minimize total completion time.

Proof of Theorem 2.8. It follows from Lemma 2.7 that

$$C_{[r,4]}(\phi_2) = C_{[r,4]}(\phi_1) \quad \text{for } r = 1, \dots, \beta - 1. \quad (2.46)$$

Moreover, from Theorem 2.4 we have

$$[C_{[\beta]}(\phi_2) + C_{[\beta+1]}(\phi_2)] \leq [C_{[\beta]}(\phi_1) + C_{[\beta+1]}(\phi_1)]. \quad (2.47)$$

Also, by Theorem 2.6

$$C_{[r,4]}(\phi_2) \leq C_{[r,4]}(\phi_1) \quad \text{for } r = \beta + 2, \dots, n. \quad (2.48)$$

Finally, it follows from equations (2.2) and (2.46–2.48) that

$$TCT(\phi_2) \leq TCT(\phi_1).$$

□

3. AN ALGORITHM WITH TEN SCENARIOS (SCENARIOS: AL1–AL10)

The problem investigated in this paper is strongly NP-hard since a special case of this problem is strongly NP-hard. Specifically, the two-machine flowshop scheduling problem with deterministic processing time (*i.e.*, $k = 2$ and $U_{t,i,k} = L_{t,i,k}$ for all $i = 1, 2, \dots, n$) is strongly NP-hard, Pinedo [26]. Therefore, it is unlikely that the considered problem has a polynomial time solution. Hence, approximate solutions, algorithms or heuristics, need to be investigated for the problem. It is known from scheduling literature that advanced algorithms or meta-heuristics cannot be used for problems with uncertain processing times, *e.g.*, Allahverdi and Aydilek [6]. This is because advanced algorithms or meta-heuristics are based on the exact values of $t_{j,k}$ and small changes in processing times significantly affects the quality of the solution. However, the value of $t_{j,k}$ could be any value between $LBt_{i,k}$ and $UBt_{i,k}$ where even the distributions of $t_{j,k}$'s are not known. Therefore, a decision on constructing a schedule should be based only on the bounds of processing times ($LBt_{i,k}$ and $UBt_{i,k}$), *i.e.*, it cannot be based on $t_{j,k}$'s as they are not known until the processing times have been completed.

The sequence obtained by ordering jobs based on the Shortest Processing Time (SPT) is optimal with respect to total completion time performance measure when jobs have known processing times, *i.e.*, $LBt_{i,1} = UBt_{i,1}$, for a single machine scheduling problem. We propose an algorithm with ten scenarios, which are based on the idea of converting the four-machine flowshop problem to a single machine problem by giving different weights to the lower and upper bounds of processing times on different machines. Steps of the algorithm are given next.

Steps of the Algorithm with Scenarios AL1–AL10

Step 1: Select n (given).

Step 2: Select values $LBt_{i,k}$ and $UBt_{i,k}$ (given) for $i = 1, \dots, n$ and $k = 1, \dots, 4$.

Step 3: Let $r = 1$.

Step 4: $t_r(1) = \sum_{j=1}^4 (0.25 * LBt_{r,j} + 0.75 * UBt_{r,j})$.

Step 5: $t_r(2) = \sum_{j=1}^4 (0.75 * LBt_{r,j} + 0.25 * UBt_{r,j})$.

Step 6: $t_r(3) = 0.25 * \left(\frac{LBt_{r,1} + UBt_{r,1}}{2} \right) + 0.25 * \left(\frac{LBt_{r,2} + UBt_{r,2}}{2} \right) + 0.25 * \left(\frac{LBt_{r,3} + UBt_{r,3}}{2} \right) + 0.25 * \left(\frac{LBt_{r,4} + UBt_{r,4}}{2} \right)$.

Step 7: $t_r(4) = 0.7 * \left(\frac{LBt_{r,1} + UBt_{r,1}}{2} \right) + 0.1 * \left(\frac{LBt_{r,2} + UBt_{r,2}}{2} \right) + 0.1 * \left(\frac{LBt_{r,3} + UBt_{r,3}}{2} \right) + 0.1 * \left(\frac{LBt_{r,4} + UBt_{r,4}}{2} \right)$.

Step 8: $t_r(5) = 0.1 * \left(\frac{LBt_{r,1} + UBt_{r,1}}{2} \right) + 0.7 * \left(\frac{LBt_{r,2} + UBt_{r,2}}{2} \right) + 0.1 * \left(\frac{LBt_{r,3} + UBt_{r,3}}{2} \right) + 0.1 * \left(\frac{LBt_{r,4} + UBt_{r,4}}{2} \right)$.

Step 9: $t_r(6) = 0.1 * \left(\frac{LBt_{r,1} + UBt_{r,1}}{2} \right) + 0.1 * \left(\frac{LBt_{r,2} + UBt_{r,2}}{2} \right) + 0.7 * \left(\frac{LBt_{r,3} + UBt_{r,3}}{2} \right) + 0.1 * \left(\frac{LBt_{r,4} + UBt_{r,4}}{2} \right)$.

Step 10: $t_r(7) = 0.1 * \left(\frac{LBt_{r,1} + UBt_{r,1}}{2} \right) + 0.1 * \left(\frac{LBt_{r,2} + UBt_{r,2}}{2} \right) + 0.1 * \left(\frac{LBt_{r,3} + UBt_{r,3}}{2} \right) + 0.7 * \left(\frac{LBt_{r,4} + UBt_{r,4}}{2} \right)$.

Step 11: $t_r(8) = 0.3 * \left(\frac{LBt_{r,1} + UBt_{r,1}}{2} \right) + 0.2 * \left(\frac{LBt_{r,2} + UBt_{r,2}}{2} \right) + 0.2 * \left(\frac{LBt_{r,3} + UBt_{r,3}}{2} \right) + 0.3 * \left(\frac{LBt_{r,4} + UBt_{r,4}}{2} \right)$.

Step 12: $t_r(9) = 0.35 * \left(\frac{LBt_{r,1} + UBt_{r,1}}{2} \right) + 0.15 * \left(\frac{LBt_{r,2} + UBt_{r,2}}{2} \right) + 0.15 * \left(\frac{LBt_{r,3} + UBt_{r,3}}{2} \right) + 0.35 * \left(\frac{LBt_{r,4} + UBt_{r,4}}{2} \right)$.

Step 13: $t_r(10) = 0.4 * \left(\frac{LBt_{r,1} + UBt_{r,1}}{2} \right) + 0.1 * \left(\frac{LBt_{r,2} + UBt_{r,2}}{2} \right) + 0.1 * \left(\frac{LBt_{r,3} + UBt_{r,3}}{2} \right) + 0.4 * \left(\frac{LBt_{r,4} + UBt_{r,4}}{2} \right)$.

Step 14: Let $r = r + 1$.

Step 15: If $r < n$, go to Step 4, else go to Step 16.

Step 16: Let $p = 1$.

Step 17: Sequence the jobs according to SPT based on $t_r(p)$ and let the resulting sequence be called πp .

Step 18: Let $p = p + 1$.

Step 19: If $p < 10$, go to Step 17, else go to Step 20.

Step 20: Set $b = 1$ and $z = 1$.

Step 21: Set $t_{h,k} = t_{[b,k]}(\pi z)$ for $k = 1, \dots, 4$ and $t_{g,k} = t_{[b+1,k]}(\pi z)$ for $k = 1, \dots, 4$.

Step 22: If $t_{h,k}$ and $t_{g,k}$ satisfy any of the conditions of Theorem 2.8, swap the jobs in positions b and $b + 1$ of the sequence πz .

Step 23: Set $b = b + 1$.

Step 24: If $b < n$, go to Step 21.

Step 25: Let $z = z + 1$.

Step 26: If $z < 10$, go to Step 21, else go to Step 27.

Step 27: The sequence πp is the solution of the algorithm with scenario ALp.

4. COMPUTATIONAL EXPERIMENTS

The performances of proposed algorithm with scenarios AL1 to AL10, with respect to total completion time performance measure, are compared with each other based on randomly generated data. First the upper bounds on processing times are generated where $UBt_{j,m}$ is generated from the uniform distribution of $U(1, 100)$. This uniform distribution has a large variance, and hence, it is usually used in the scheduling literature for generating processing times. On the other hand, $LBt_{j,m}$ is generated from the uniform distribution of $U(\max(1, UBt_{j,m} - \Delta), UBt_{j,m})$, where Δ is set at three values of 10, 20, and 30. Generating $LBt_{j,m}$ and $UBt_{j,m}$ in this way is commonly used in the literature for uncertain processing times, *e.g.*, Aydilek *et al.* [15].

Once $LBt_{j,m}$ and $UBt_{j,m}$ are generated, $t_{j,m}$ ($LBt_{j,m} \leq t_{j,m} \leq UBt_{j,m}$) needs to be generated for an instance of processing time, between the lower and upper bounds, in order to compare the performance of the proposed algorithm with different scenarios. Assuming a specific distribution for generating processing times (between the lower and upper bounds) is not appropriate since we do not know its distribution. Therefore, we consider four different distributions which are uniform, positive linear, negative linear, and normal. The considered distributions are representative of many distributions as they are representatives of symmetric, positively skewed, and negatively skewed distributions. These are the distributions which Aydilek *et al.* [15] also used to evaluate the performance of their algorithms.

We now describe the four distributions used for generating processing times. For the uniform distribution, $t_{i,k}$ is generated from $U(LBt_{i,k}, UBt_{i,k})$. For the normal distribution, mean μ is set to the average of the lower and upper bounds, *i.e.*, $(LBt_{i,k} + UBt_{i,k})/2$, and the standard deviation σ is set to $(UBt_{i,k} - LBt_{i,k})/6$. Generating $t_{i,k}$ from a normal distribution with this mean and standard deviation may result in a value which is outside $LBt_{i,k}$ and $UBt_{i,k}$ with a small probability, in which case we regenerate $t_{i,k}$ such that it falls between $LBt_{i,k}$ and $UBt_{i,k}$. That is to say, the normal distribution used to generate $t_{i,k}$ is truncated. For positive linear and negative linear distributions, the probability density functions are $f_p(t_{i,k}) = 2(t_{i,k} - LBt_{i,k})/x(UBt_{i,k} - LBt_{i,k})^2$ and

TABLE 1. Computational results for normal distribution.

Scenario	Δ	N					
		200	300	400	500	600	Avg.
AL1	10	0.82	0.53	0.45	0.39	0.32	0.50
AL2	10	0.78	0.56	0.46	0.37	0.32	0.50
AL3	10	0.78	0.53	0.46	0.36	0.32	0.49
AL4	10	10.24	10.91	11.37	11.58	11.80	11.18
AL5	10	15.65	16.85	17.61	18.21	18.46	17.36
AL6	10	17.81	18.60	19.23	19.71	19.95	19.06
AL7	10	19.55	20.30	20.68	20.98	21.15	20.53
AL8	10	2.49	2.92	2.88	3.25	3.29	2.97
AL9	10	6.76	7.65	7.84	8.36	8.52	7.83
AL10	10	11.01	12.08	12.46	13.00	13.29	12.37
AL1	20	0.94	0.68	0.55	0.50	0.44	0.62
AL2	20	0.90	0.64	0.50	0.49	0.41	0.59
AL3	20	0.87	0.62	0.49	0.49	0.39	0.57
AL4	20	10.79	11.55	11.84	12.14	12.30	11.72
AL5	20	16.46	17.66	18.14	18.70	19.15	18.02
AL6	20	18.51	19.26	19.94	20.40	20.64	19.75
AL7	20	20.31	20.89	21.37	21.73	21.91	21.24
AL8	20	2.63	2.97	3.24	3.39	3.56	3.16
AL9	20	7.06	7.86	8.34	8.68	8.97	8.18
AL10	20	11.49	12.43	13.10	13.52	13.90	12.89
AL1	30	1.04	0.84	0.67	0.60	0.55	0.74
AL2	30	0.98	0.75	0.61	0.52	0.48	0.67
AL3	30	0.91	0.68	0.58	0.48	0.45	0.62
AL4	30	11.07	11.85	12.22	12.65	12.76	12.11
AL5	30	16.68	17.69	18.48	19.21	19.57	18.33
AL6	30	18.77	19.69	20.38	20.73	21.13	20.14
AL7	30	20.77	21.48	21.88	22.20	22.40	21.75
AL8	30	2.75	3.10	3.31	3.55	3.60	3.26
AL9	30	7.29	8.11	8.60	9.02	9.18	8.44
AL10	30	11.83	12.89	13.54	13.99	14.19	13.29

$f_n(t_{i,k}) = 2(\text{UB}t_{i,k} - t_{i,k})/x(\text{UB}t_{i,k} - \text{LB}t_{i,k})^2$ for $t_{i,k} \in (\text{LB}t_{i,k}, \text{UB}t_{i,k})$, respectively. Figure 1 shows all four distributions for generating processing times between the lower and upper bounds.

Performances of the proposed algorithm with scenarios AL1 to AL10 are compared with each other based on two performance measures: average percentage error (Error) and standard deviation (Std). The percentage error is defined as

$$\text{Error (AL}i\text{)} = 100 * \frac{\text{TCT (AL}i\text{)} - \min\{\text{TCT (AL1)}, \dots, \text{TCT (AL10)}\}}{\min\{\text{TCT (AL1)}, \dots, \text{TCT (AL10)}\}}.$$

We consider five different values for n , which are 200, 300, 400, 500, and 600. Furthermore, we consider three values of 10, 20, and 30 for Δ , and four distributions (positive, negative, uniform, and normal). A total of 60 ($5 * 3 * 4$) combinations are considered. Since for each combination, 1000 replications are generated, a total of 60 000 problems are generated.

The computational results are given in Tables 1, 2, 3, and 4 for the normal, uniform, positive linear, and negative linear distributions, respectively. In Tables 1–4, the first column denotes the scenario, the second column shows the Δ value, the next five columns indicate the number of jobs. Each value in the tables is the average of 1000 replications. The final column in the Tables 1–4 is the average over different number of jobs.

TABLE 2. Computational results for uniform distribution.

Scenario	Δ	N					
		200	300	400	500	600	Avg.
AL1	10	0.75	0.57	0.44	0.36	0.33	0.49
AL2	10	0.74	0.55	0.43	0.39	0.35	0.49
AL3	10	0.70	0.54	0.44	0.38	0.34	0.48
AL4	10	10.39	10.94	11.38	11.59	11.94	11.25
AL5	10	15.76	16.58	17.54	18.05	18.56	17.30
AL6	10	17.72	18.59	19.28	19.64	20.09	19.06
AL7	10	19.71	20.16	20.67	20.86	21.27	20.53
AL8	10	2.56	2.85	3.05	3.11	3.36	2.99
AL9	10	6.83	7.52	8.00	8.24	8.62	7.84
AL10	10	11.12	11.94	12.59	12.92	13.32	12.38
AL1	20	0.94	0.70	0.60	0.56	0.45	0.65
AL2	20	0.88	0.71	0.54	0.44	0.43	0.60
AL3	20	0.91	0.66	0.54	0.47	0.43	0.60
AL4	20	10.70	11.44	11.77	12.05	12.47	11.69
AL5	20	16.23	17.45	18.25	18.65	19.16	17.95
AL6	20	18.40	19.27	19.91	20.17	20.72	19.69
AL7	20	20.31	21.09	21.49	21.51	21.88	21.26
AL8	20	2.68	3.05	3.22	3.37	3.70	3.20
AL9	20	7.07	7.86	8.35	8.60	9.10	8.20
AL10	20	11.45	12.45	13.06	13.41	13.97	12.87
AL1	30	1.07	0.86	0.67	0.61	0.59	0.76
AL2	30	0.96	0.70	0.59	0.51	0.46	0.64
AL3	30	0.96	0.71	0.59	0.52	0.43	0.64
AL4	30	10.94	11.84	12.25	12.33	12.70	12.01
AL5	30	16.52	17.74	18.55	18.95	19.53	18.26
AL6	30	18.79	19.65	20.34	20.65	21.03	20.09
AL7	30	20.84	21.52	21.78	21.88	22.37	21.68
AL8	30	2.96	3.08	3.30	3.52	3.68	3.31
AL9	30	7.39	8.07	8.54	8.89	9.20	8.42
AL10	30	11.84	12.85	13.43	13.83	14.18	13.23

Average errors of the algorithm with the ten scenarios for the 1000 replications are shown in Figure 2 with respect to the four distributions while the standard errors of the replications are given in Figure 3. As can be seen from the figures, the errors and standard deviations of the scenarios are in general proportional. This is due to a large number of replications.

As seen from Figures 2 and 3, the best performing scenarios are AL1, AL2, and AL3 among the considered ten scenarios. To show that these three scenarios are the best statistically, 95% confidence intervals of the average errors for the scenarios are constructed which are given in Table 5.

It is clear from Table 5 that the confidence intervals for errors of scenarios AL1, AL2, and AL3 do not overlap with those of the others and are much smaller compared to them. Statistically this shows that, the algorithm with scenarios AL1, AL2, and AL3 perform the best among the considered ten scenarios. However, confidence errors of the three scenarios AL1, AL2, and AL3 overlap and it is difficult to statistically state that one performs better than the rest.

Since the best performing scenarios are AL1, AL2, and AL3 and since their performances are so close to each other, it is interesting to know why they are performing well. The main difference between these three scenarios and the other scenarios is that the same weight is given to the processing times of all the four machines in

TABLE 3. Computational results for positive linear distribution.

Scenario	Δ	N					
		200	300	400	500	600	Avg.
AL1	10	0.77	0.53	0.46	0.37	0.30	0.49
AL2	10	0.76	0.51	0.44	0.36	0.29	0.47
AL3	10	0.74	0.51	0.43	0.35	0.29	0.46
AL4	10	10.00	10.67	11.23	11.46	11.66	11.00
AL5	10	15.39	16.34	17.21	17.82	18.24	17.00
AL6	10	17.54	18.29	18.97	19.37	19.73	18.78
AL7	10	19.37	19.95	20.22	20.70	20.92	20.23
AL8	10	2.40	2.72	2.95	3.16	3.31	2.91
AL9	10	6.46	7.36	7.84	8.19	8.45	7.66
AL10	10	10.60	11.68	12.33	12.76	13.12	12.10
AL1	20	0.93	0.61	0.55	0.46	0.41	0.59
AL2	20	0.87	0.70	0.57	0.46	0.42	0.60
AL3	20	0.89	0.62	0.56	0.45	0.38	0.58
AL4	20	10.33	11.14	11.43	11.71	12.05	11.33
AL5	20	15.86	16.93	17.74	18.12	18.66	17.46
AL6	20	17.95	18.72	19.45	19.69	20.24	19.21
AL7	20	19.74	20.41	20.86	21.07	21.42	20.70
AL8	20	2.64	2.90	3.07	3.29	3.41	3.06
AL9	20	6.85	7.64	8.03	8.39	8.69	7.92
AL10	20	11.07	12.05	12.62	13.03	13.47	12.45
AL1	30	0.93	0.70	0.62	0.52	0.46	0.65
AL2	30	0.99	0.80	0.67	0.59	0.54	0.72
AL3	30	0.88	0.68	0.56	0.51	0.43	0.61
AL4	30	10.55	11.25	11.71	11.97	12.24	11.54
AL5	30	16.04	17.26	17.89	18.39	18.82	17.68
AL6	30	18.08	19.03	19.55	19.99	20.31	19.39
AL7	30	19.97	20.70	21.06	21.28	21.61	20.92
AL8	30	2.70	2.92	3.14	3.41	3.56	3.15
AL9	30	7.00	7.70	8.28	8.55	8.84	8.07
AL10	30	11.30	12.25	12.97	13.29	13.68	12.70

these three scenarios while different weights are given to the processing time of the four machines in the other scenarios.

Figure 4 shows the errors of scenarios with respect to the three values of Δ . As expected the errors of all scenarios slightly increase as Δ increases. This is because as Δ increases, the uncertainty, *i.e.*, the gap between the lower and upper bounds of the processing times increases.

Figure 5 indicates errors of the scenarios with respect the number of jobs, *i.e.*, the size of the problem. It is interesting to note that while errors of scenarios AL4–AL10 increase as n increases, those of the best performing scenarios AL1, AL2, and AL3 decrease. This is yet another advantage of the best performing three scenarios.

The effect of the developed dominance relation (Thm. 2.8) in Step 22 of the scenarios was also explored. It was observed that the improvement, on the average, was less than 10%.

Finally, the computation time (CPU) of the algorithm was also investigated. It was observed that there is no difference among different scenarios of the algorithm with respect to the CPU time. It was also observed that the CPU time does not change with respect to value either. Furthermore, the CPU time does not change with respect to different distributions used to generate processing times. However, the CPU time changes as the number of jobs n changes. The change of CPU time (in seconds) with respect to n is given in Table 6.

TABLE 4. Computational results for negative linear distribution.

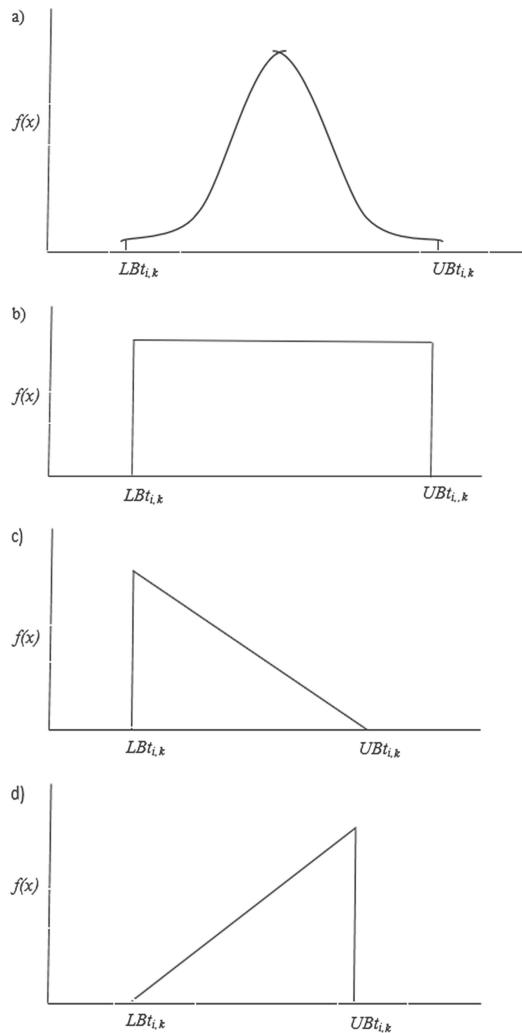
Scenario	Δ	N					
		200	300	400	500	600	Avg.
AL1	10	0.78	0.58	0.45	0.38	0.33	0.50
AL2	10	0.75	0.55	0.43	0.38	0.31	0.48
AL3	10	0.78	0.57	0.44	0.36	0.33	0.50
AL4	10	10.38	11.06	11.53	11.86	12.20	11.41
AL5	10	15.91	17.07	17.81	18.40	18.82	17.60
AL6	10	18.00	19.02	19.47	19.92	20.30	19.34
AL7	10	20.02	20.76	20.91	21.35	21.61	20.93
AL8	10	2.54	2.67	3.08	3.26	3.43	3.00
AL9	10	6.81	7.55	8.14	8.48	8.72	7.94
AL10	10	11.15	12.14	12.79	13.20	13.52	12.56
AL1	20	0.97	0.80	0.68	0.55	0.50	0.70
AL2	20	0.84	0.69	0.54	0.44	0.38	0.58
AL3	20	0.86	0.71	0.59	0.48	0.38	0.60
AL4	20	11.09	11.77	12.27	12.63	12.60	12.07
AL5	20	16.78	17.98	18.80	19.35	19.87	18.56
AL6	20	18.91	19.99	20.47	20.96	21.31	20.33
AL7	20	20.87	21.72	22.06	22.41	22.62	21.94
AL8	20	2.72	2.96	3.30	3.51	3.47	3.19
AL9	20	7.26	7.93	8.58	9.01	9.08	8.37
AL10	20	11.83	12.75	13.47	14.00	14.16	13.24
AL1	30	1.22	0.97	0.84	0.77	0.71	0.90
AL2	30	0.88	0.68	0.61	0.47	0.37	0.60
AL3	30	1.01	0.77	0.64	0.54	0.49	0.69
AL4	30	11.69	12.17	12.78	13.20	13.46	12.66
AL5	30	17.41	18.42	19.33	19.86	20.33	19.07
AL6	30	19.64	20.58	21.17	21.70	21.95	21.01
AL7	30	21.67	22.26	22.82	23.00	23.22	22.59
AL8	30	3.01	3.09	3.46	3.79	3.95	3.46
AL9	30	7.83	8.23	8.97	9.46	9.73	8.84
AL10	30	12.55	13.15	14.10	14.60	14.96	13.87

TABLE 5. Confidence intervals.

Scenario	Avg. error	95% Confidence interval on the avg. error
AL1	0.63	(0.59–0.68)
AL2	0.58	(0.53–0.63)
AL3	0.57	(0.52–0.61)
AL4	11.66	(11.52–11.81)
AL5	17.88	(17.74–18.03)
AL6	19.66	(19.50–19.81)
AL7	21.19	(21.02–21.36)
AL8	3.14	(3.01–3.26)
AL9	8.14	(7.99–8.30)
AL10	12.83	(12.66–12.99)

TABLE 6. CPU time of the algorithm.

<u>n</u>	<u>CPU time (s)</u>
200	12.24
300	21.41
400	27.77
500	36.60
600	47.11

FIGURE 1. The distributions used for generating $t_{i,k}$ between $LBt_{i,k}$ and $UBt_{i,k}$. (a) Normal. (b) Uniform. (c) Negative Linear. (d) Positive Linear.

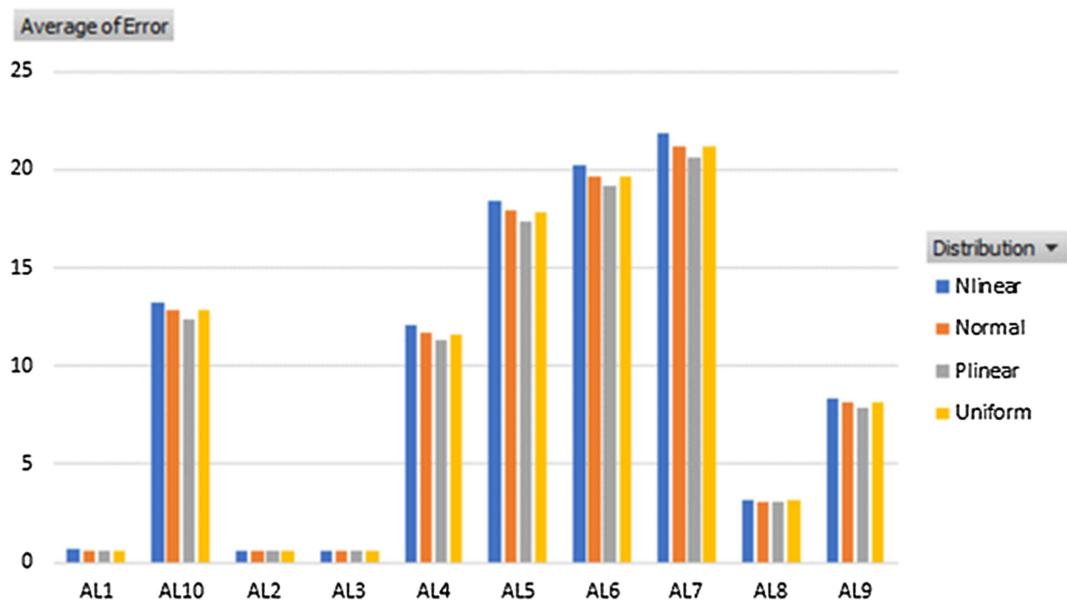


FIGURE 2. Average of error of the scenarios for different distributions.

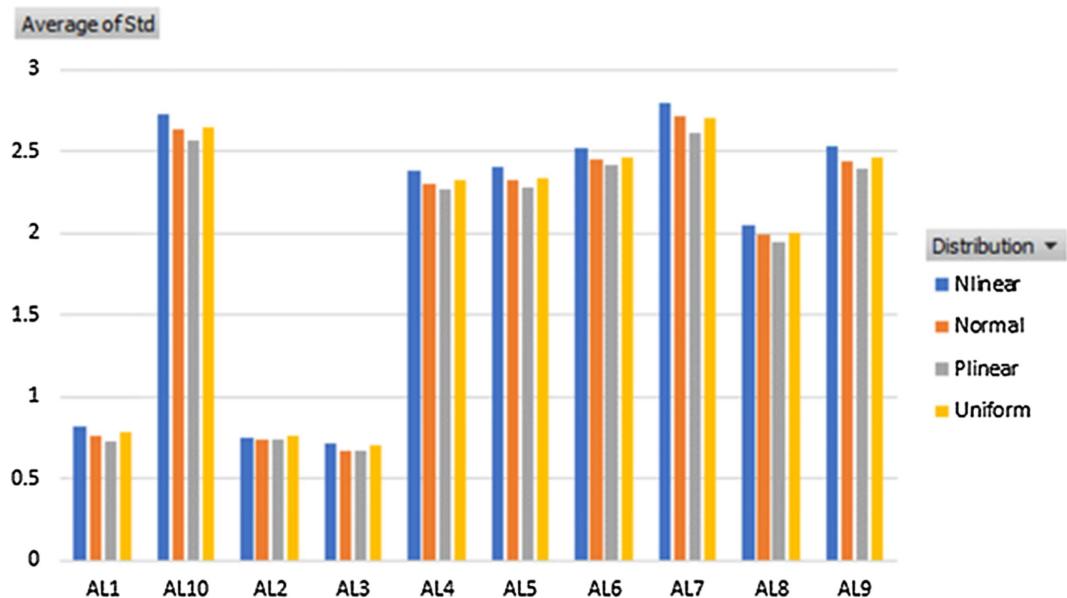
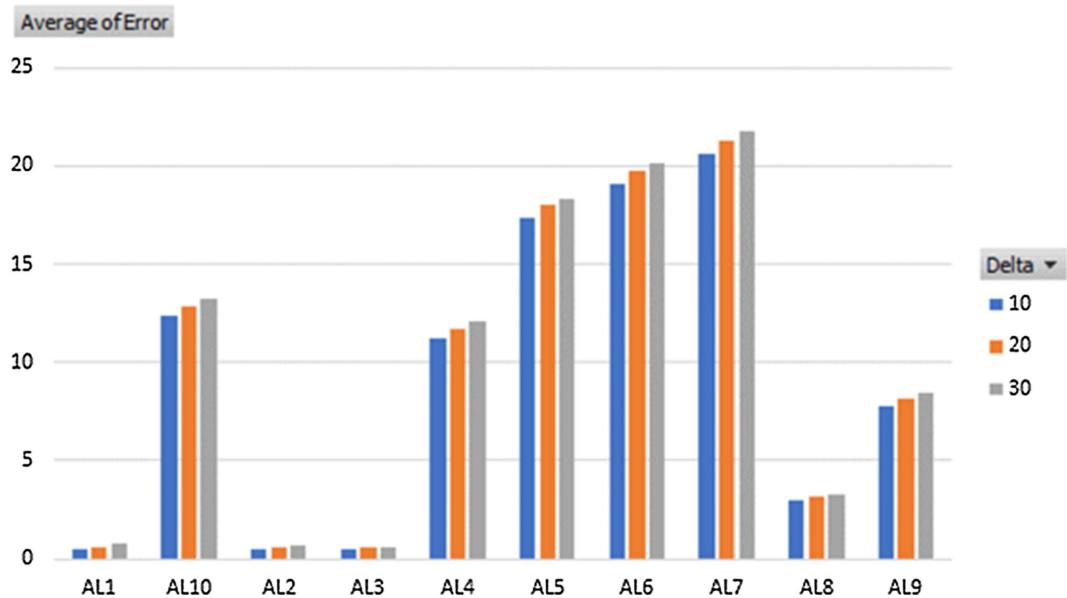
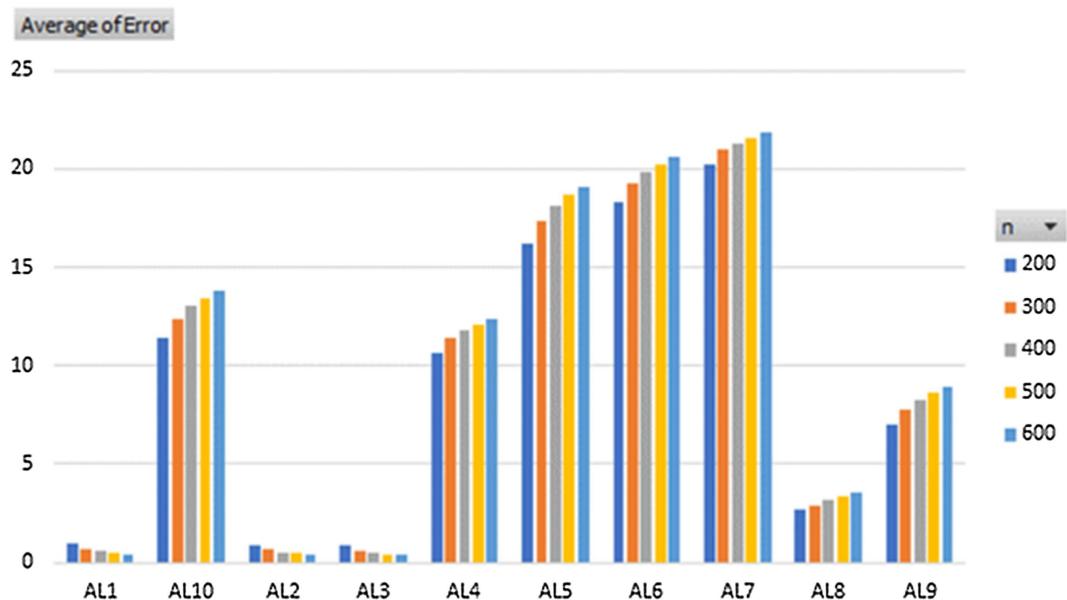


FIGURE 3. Average of std of the scenarios for different distributions.

FIGURE 4. Average of error of the scenarios with respect to Δ .FIGURE 5. Average of error of the scenarios with respect to n .

These values are the average over all parameters. It should be noted that even for the extreme case of $n = 600$, the CPU time is less than a minute. Therefore, the CPU time of the algorithm is not an issue.

5. CONCLUSIONS

In this paper, the four-machine flowshop scheduling problem is addressed with respect to the performance measure of total completion time where processing times are uncertain. Only the lower and upper bounds of processing times are known. It is known that the SPT sequence provides an optimal solution for the single machine scheduling problem with the performance measure of total completion time. Therefore, the four-machine flowshop problem is converted into a single machine problem where different SPT solutions (scenarios) are obtained by using different weights to the lower and upper bounds of processing times on different machines. Moreover, the scenarios are enhanced by using dominance relations that are developed in this paper.

The proposed algorithm with ten scenarios is evaluated through computational experiments. The computational experiments reveal that three of the scenarios (AL1, AL2, and AL3) are statistically performing better than the other seven scenarios (AL4–AL10) for different distributions of job processing times and different gaps between the lower and upper bounds of the processing times. Moreover, computational experiments also reveal that errors of the best performing scenarios are decreasing as the size of the problem gets larger. It is important to note that the common factor in the scenarios of AL1, AL2, and AL3 is that a similar weight is given to the processing times of all the four machines.

Setup times are ignored in this paper. Ignoring setup times may be valid for some manufacturing problems when setup times are very small compared to processing times. Allahverdi and Soroush [8] indicated that ignoring setup times may not be valid for some other manufacturing environments. Recent research on scheduling problems with separate setup times is summarized [3]. He stated that in order to improve resource utilization, increase productivity, and eliminate waste, setup times have to be considered as separate from processing times. Hence, an extension to the current research is to address the four-machine flowshop scheduling problem to minimize total completion time where setup times are separate from processing times and where processing times are uncertain and within some intervals.

In regular flowshop scheduling environments, we can have delay between successive operations. There are many real-life problems where this is true. However, in some manufacturing environments, it is necessary to have successive operations following one another without delay. Such flowshops are known as no-wait flowshops. No-wait flowshops are applicable in industries such as pharmaceutical, chemical, and plastic [4]. In the literature, although some work has been conducted for no-wait flowshops with uncertain data, *e.g.*, [5, 10, 11] for the two-machine case, the analysis conducted in this paper can be extended to no-wait flowshops with four machines.

REFERENCES

- [1] A. Allahverdi, The tricriteria two-machine flowshop scheduling problem. *Int. Trans. Oper. Res.* **8** (2001) 403–425.
- [2] A. Allahverdi, A new heuristic for m -machine flowshop scheduling problem with bicriteria of makespan and maximum tardiness. *Comput. Oper. Res.* **31** (2004) 157–180.
- [3] A. Allahverdi, The third comprehensive survey on scheduling problems with setup times/costs. *Eur. J. Oper. Res.* **246** (2015) 345–378.
- [4] A. Allahverdi, A survey of scheduling problems with no-wait in process. *Eur. J. Oper. Res.* **255** (2016) 665–686.
- [5] A. Allahverdi and M. Allahverdi, Two-machine no-wait flowshop scheduling problem with uncertain setup times to minimize maximum lateness. *Comput. Appl. Math.* **37** (2018) 6774–6794.
- [6] A. Allahverdi and H. Aydilek, Heuristics for two-machine flowshop scheduling problem to minimize makespan with bounded processing times. *Int. J. Prod. Res.* **48** (2010) 6367–6385.
- [7] A. Allahverdi and H. Aydilek, Heuristics for two-machine flowshop scheduling problem to minimize maximum lateness with bounded processing times. *Comput. Math. Appl.* **60** (2010) 1374–1384.
- [8] A. Allahverdi and H.M. Soroush, The significance of reducing setup times/setup costs. *Eur. J. Oper. Res.* **187** (2008) 978–984.
- [9] M. Allahverdi and A. Allahverdi, Algorithms for four-machine flowshop scheduling problem with uncertain processing times to minimize makespan. *RAIRO: OR* **54** (2020) 529–553.
- [10] M. Allahverdi and A. Allahverdi, Minimizing total completion time in a two-machine no-wait flowshop with uncertain and bounded setup times. *J. Ind. Manage. Optim.* **16** (2020) 2439–2457.

- [11] 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. To appear in: *J. Ind. Manage. Optim.* (2020). doi: [10.3934/jimo.2020054](https://doi.org/10.3934/jimo.2020054)
- [12] A. Allahverdi and Y.N. Sotskov, Two-machine flowshop minimum length scheduling problem with random and bounded processing times. *Int. Trans. Oper. Res.* **10** (2003) 65–76.
- [13] A. Aydilek, H. Aydilek and A. Allahverdi, Increasing the profitability and competitiveness in a production environment with random and bounded setup times. *Int. J. Prod. Res.* **51** (2013) 106–117.
- [14] A. Aydilek, H. Aydilek and A. Allahverdi, Production in a two-machine flowshop scheduling environment with uncertain processing and setup times to minimize makespan. *Int. J. Prod. Res.* **53** (2015) 2803–2819.
- [15] 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. Model.* **45** (2017) 982–996.
- [16] H. Aydilek and A. Allahverdi, Two-machine flowshop scheduling problem with bounded processing times to minimize total completion time. *Comput. Math. Appl.* **59** (2010) 684–693.
- [17] J.F. Chen, Unrelated parallel-machine scheduling to minimize total weighted completion time. *J. Intel. Manuf.* **26** (2015) 1099–1112.
- [18] J.M. Framinan and P. Perez-Gonzalez, The 2-stage assembly flowshop scheduling problem with total completion time: efficient constructive heuristic and metaheuristic. *Comput. Oper. Res.* **88** (2017) 237–246.
- [19] J.M. Framinan and P. Perez-Gonzalez, New approximate algorithms for the customer order scheduling problem with total completion time objective. *Comput. Oper. Res.* **78** (2017) 181–192.
- [20] H.Y. Fuchigami and S. Rangel, A survey of case studies in production scheduling: Analysis and perspectives. *J. Comput. Sci.* **25** (2018) 425–436.
- [21] 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. Model. Pract. Theory* **79** (2017) 23–36.
- [22] T. Keshavarz and N. Salmasi, Makespan minimization in flexible flowshop sequence-dependent group scheduling problem. *Int. J. Prod. Res.* **51** (2013) 6182–6193.
- [23] P. Kouvelis and G. Yu, Robust Discrete Optimization and its Applications. Kluwer Academic Publisher, Dordrecht (1997).
- [24] T.C. Lai and Y.N. Sotskov, Sequencing with uncertain numerical data for makespan minimization. *J. Oper. Res. Soc.* **50** (1999) 230–243.
- [25] T.C. Lai, Y.N. Sotskov, N.Y. Sotskova and F. Werner, Optimal makespan scheduling with given bounds of processing times. *Math. Comput. Model.* **26** (1997) 67–86.
- [26] M. Pinedo, Scheduling Theory, Algorithms, and Systems. Prentice Hall, Englewood Cliffs, NJ (1995) 349.
- [27] 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.
- [28] Y.N. Sotskov, A. Allahverdi and T.C. Lai, Flowshop scheduling problem to minimize total completion time with random and bounded processing times. *J. Oper. Res. Soc.* **55** (2004) 277–286.
- [29] 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.
- [30] 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.