International Journal of Industrial Engineering Computations 13 (2022) 543–556

Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

More effective heuristics for a two-machine no-wait flowshop to minimize maximum lateness

Harun Aydilek[a], Asiye Aydilek[b], Muberra Allahverdi[c] and Ali Allahverdi[d*]

[a]*Mathematics & Natural Sciences Dept., Gulf University for Science and Technology, Kuwait*
[b]*Economics & Finance Dept., Gulf University for Science and Technology, Kuwait*
[c]*Department of Mathematics, Faculty of Engineering and Natural Sciences, Ankara Yıldırım Beyazıt University, Ankara, Turkey*
[d]*Industrial and Management Systems Engineering Dept., Kuwait University, Kuwait*

C H R O N I C L E

Article history:
Received March 30 2022
Received in Revised Format May 20 2022
Accepted July 20 2022
Available online July, 20 2022

Keywords:
Flowshop scheduling
Uncertain setup times
No-wait
Maximum lateness
Dominance relations
Heuristics

A B S T R A C T

We address a manufacturing environment with the no-wait constraint which is common in industries such as metal, plastic, and semiconductor. Setup times are modelled as uncertain with the objective of minimizing maximum lateness which is an important performance measure for customer satisfaction. This problem has been addressed in scheduling literature for the two-machine no-wait flowshop where dominance relations were presented. Recently, another dominance relation was presented and shown to be about 90% more efficient than the earlier ones. In the current paper, we propose two new dominance relations, which are less restrictive than the earlier ones in the literature. The new dominance relations are shown to be 140% more efficient than the most recent one in the literature. As the level of uncertainty increases, the newly proposed dominance relation performs better, which is another strength of the newly proposed dominance relation. Moreover, we also propose constructive heuristics and show that the best of the newly proposed heuristics is 95% more efficient than the existing one in the literature under the same CPU time.

1. Introduction

The no-wait flowshop is a type of flowshop where consecutive operations are carried out with no delay. This is vital for certain settings in which waiting causes undue difficulty in manufacturing. A typical example is when temperature is involved, and operations must be completed while the material is still hot. Reducing work-in-process is another advantage of the no-wait flowshop, Macchiaroli et al. (1999). Scheduling of patients (Hsu et al., 2003), aircraft landing (Kim et al.; 2009), trains (Liu and Kozan, 2011), and bakery production (Hecker et al., 2014) are some examples where such a flowshop would be necessary. It is also crucial in the pharmaceutical industry, chemical industry, and plastic industry, Allahverdi (2016), and Hall and Sriskandarajah (1996).

The amount of time required to set up a resource for production is called setup time. Since setup times clearly affect the completion time, it is necessary to consider them for scheduling problems. Nonetheless, a surprising percentage of scheduling research (at least 90%) ignores setup times, Allahverdi (2015). Among those that do not ignore, setup times are considered as known values, e.g., Dileepan (2004), yet there are many manufacturing settings when this is not the case. In fact, far from being known, they may change due to many factors such as unforeseen breakdowns and shortage of equipment, Kim and Bobrowski (1997). For such manufacturing settings, ignoring set up times or considering them as known values will greatly impact the maximum lateness, resulting in an unnecessarily poor efficiency.

* Corresponding author
E-mail: ali.allahverdi@ku.edu.kw (A. Allahverdi)

2022 Growing Science Ltd.
doi: 10.5267/j.ijiec.2022.7.002

In the scheduling literature, the assumption of a deterministic environment (where setup and/or processing times are fixed and known in advance) is commonly utilized, Seidgar et al. (2014) and Keshavarz and Salmasi (2013). On the other hand, manufacturing environments in industries are frequently subject to a wide range of uncertainties, Allahverdi (2022a), Wang and Choi (2012), Gonzalez-Neira et al. (2017). Hence, managers face substantial uncertainty in setup times. Moreover, assuming certain probability distributions for setup times may not be valid for some environments, e.g., Kouvelis and Yu (1997). Thus, it is essential to model setup times as uncertain.

Let $UBs_{i,k}$ and $LBs_{i,k}$ be the upper and lower bounds of setup time $s_{i,k}$ of job i on machine k, respectively. Dominance relations are provided by Allahverdi et al. (2003) and Allahverdi (2005) for the problems $F2|LBs_{i,k}\leq s_{i,k}\leq UBs_{i,k}|C_{max}$, $\sum C_j$ and $F2|LBs_{i,k}\leq s_{i,k}\leq UBs_{i,k}|C_{max}$, respectively, where C_{max} and $\sum C_j$ symbolize the makespan and completion time. Additionally, dominance relations are provided for the problem $F2|LBs_{i,k}\leq s_{i,k}\leq UBs_{i,k}|C_{max}$ by Aydilek et al. (2013) assuming fixed processing times and by Aydilek et al. (2015) relaxing the assumption of fixed processing times, i.e., processing times are modelled as uncertain. Allahverdi (2022b) provided and algorithm for the problem of $F2|no\text{-}wait,LBs_{i,k}\leq s_{i,k}\leq UBs_{i,k}|L_{max}$ and showed that the algorithm performs much better than the earlier existing algorithms. Other papers studying uncertain environments include Braun et al. (2002), Sotskov et al. (2009), Matsveichuk et al. (2011), Sotskov and Lai (2012), and Sotskov and Matsveichuk (2012), where set up/processing times are also assumed to be within certain upper and lower bounds.

Allahverdi and Allahverdi (2018) address the problem $F2|no\text{-}wait,LBs_{i,k}\leq s_{i,k}\leq UBs_{i,k}|L_{max}$ which aims to minimize maximum lateness assuming setup times to be uncertain. Another dominance relation for the same problem is provided by Allahverdi et al. (2021), which is proven to be of higher efficiency. In the current paper, we present a new dominance relation and demonstrate that it is much more efficient than the one provided by Allahverdi et al. (2021). Moreover, we propose new heuristics for this problem and demonstrate that they are more effective than the previous heuristics in the literature while maintaining the same computational time.

Problem definition is provided in the next section. The new dominance relations are presented in Section 3 while their evaluation is given in Section 4. The new heuristics are described in Section 5 while their performance compared with the recent heuristic in the literature is given in Section 6. Conclusion remarks are presented in Section 7.

2. Problem Definition

We address the problem of minimizing maximum lateness in a two-machine no-wait flowshop scheduling environment with separate setup times. A typical two-machine no-wait flowshop problem with three jobs can be seen in a Gantt chart in Figure 1. We model setup times as uncertain and bounded, within lower and upper bounds. We assume that all jobs are available at time zero and every job has a positive processing time on each machine. A machine can process at most one job and a job can be processed on at most one machine at any given time. Let $s_{h,m}$ and $t_{h,m}$ denote the setup and processing times of job h on machine m (m=1, 2), respectively. Also let the lateness, completion time, and due date of job h be denoted by L_h, C_h and d_h, respectively, where $L_h = C_h - d_h$. Uncertain setup times satisfy the inequality $LBs_{h,m} \leq s_{h,m} \leq UBs_{h,m}$ where $UBs_{h,m}$ and $LBs_{h,m}$ represent upper bound and lower bound of the setup time $s_{h,m}$, respectively. Closed brackets ([ ]) are used to represent the position of a job in a given sequence, e.g, $d_{[h]}$ denotes the due date of the job in position h.
For the considered problem, the completion time of the job in position h is

$$C_{[h]} = \sum_{r=1}^{h} \max\{s_{[r,2]},\ s_{[r,1]}+t_{[r,1]} - t_{[r-1,2]}\} + \sum_{r=1}^{h} t_{[r,2]}$$

where $t_{[0,2]}=0$. Therefore, lateness of the job in position h is given by

$$L_{[h]} = \sum_{r=1}^{h} \max\{s_{[r,2]},\ s_{[r,1]}+t_{[r,1]} - t_{[r-1,2]}\} + \sum_{r=1}^{h} t_{[r,2]} - d_{[h]} \tag{1}$$

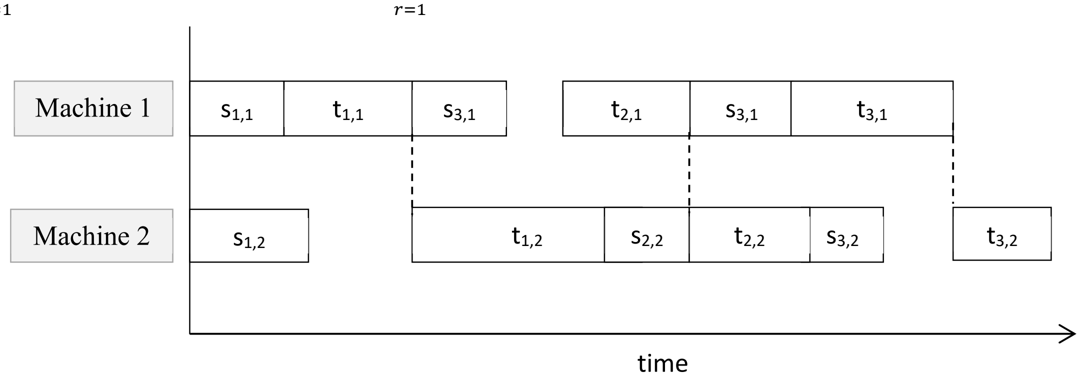

Fig. 1. A three-job two-machine example

3. More Efficient Dominance Relations

A dominance relation was presented by Allahverdi and Allahverdi (2018) for the problem. Recently, Allahverdi et al. (2021) presented another dominance relation and showed that their dominance relation is more effective than the dominance relation presented by Allahverdi and Allahverdi (2018). In this section, we propose two new dominance relations (Theorems 2 and 4) and show that the new dominance relations are much more efficient than that of Allahverdi et al. (2021). Let π_1 be a sequence such that job j, in position τ, immediately precedes job i, i.e., they are adjacent. Also let π_2 be another sequence obtained from π_1 by interchanging the jobs i and j; job i precedes job j in sequence π_2. Moreover, let the bold $\boldsymbol{S_{h,k}}$ be a random variable denoting the setup time of job h on machine k, which satisfies

$$LBs_{h,k} \leq \boldsymbol{S_{h,k}} \leq UBs_{h,k} \tag{2}$$

Let

$$\Delta = \sum_{r=1}^{\tau-1} \max\{s_{[r,2]},\ s_{[r,1]} + t_{[r,1]} - t_{[r-1,2]}\} + \sum_{r=1}^{\tau-1} t_{[r,2]}$$

Then, the lateness of jobs i and j in the two sequences π_1 and π_2 for jobs in positions τ, τ+1, and τ+2 are given by

$$L_{[\tau]}(\pi_1) = \Delta + \max\{\boldsymbol{S_{j,2}},\ \boldsymbol{S_{j,1}} + t_{j,1} - t_{[\tau-1,2]}\} + t_{j,2} - d_j \tag{3}$$
$$L_{[\tau]}(\pi_2) = \Delta + \max\{\boldsymbol{S_{i,2}},\ \boldsymbol{S_{i,1}} + t_{i,1} - t_{[\tau-1,2]}\} + t_{i,2} - d_i \tag{4}$$
$$L_{[\tau+1]}(\pi_1) = \Delta + \max\{\boldsymbol{S_{j,2}},\ \boldsymbol{S_{j,1}} + t_{j,1} - t_{[\tau-1,2]}\} + t_{j,2} + \max\{\boldsymbol{S_{i,2}},\ \boldsymbol{S_{i,1}} + t_{i,1} - t_{j,2}\} + t_{i,2} - d_i \tag{5}$$
$$L_{[\tau+1]}(\pi_2) = \Delta + \max\{\boldsymbol{S_{i,2}},\ \boldsymbol{S_{i,1}} + t_{i,1} - t_{[\tau-1,2]}\} + t_{i,2} + \max\{\boldsymbol{S_{j,2}},\ \boldsymbol{S_{j,1}} + t_{j,1} - t_{i,2}\} + t_{j,2} - d_j \tag{6}$$
$$\begin{aligned} L_{[\tau+2]}(\pi_1) = \Delta &+ \max\{\boldsymbol{S_{j,2}},\ \boldsymbol{S_{j,1}} + t_{j,1} - t_{[\tau-1,2]}\} + t_{j,2} + \max\{\boldsymbol{S_{i,2}}, \boldsymbol{S_{i,1}} + t_{i,1} - t_{j,2}\} + t_{i,2} \\ &+ \max\{\boldsymbol{S_{[\tau+2,2]}}, \boldsymbol{S_{[\tau+2,1]}} + t_{[\tau+2,1]} - t_{i,2}\} + t_{[\tau+2,2]} - d_{[\tau+2]} \end{aligned} \tag{7}$$
$$\begin{aligned} L_{[\tau+2]}(\pi_2) = \Delta &+ \max\{\boldsymbol{S_{i,2}},\ \boldsymbol{S_{i,1}} + t_{i,1} - t_{[\tau-1,2]}\} + t_{i,2} + \max\{\boldsymbol{S_{j,2}}, \boldsymbol{S_{j,1}} + t_{j,1} - t_{i,2}\} + t_{j,2} \\ &+ \max\{\boldsymbol{S_{[\tau+2,2]}},\ \boldsymbol{S_{[\tau+2,1]}} + t_{[\tau+2,1]} - t_{j,2}\} + t_{[\tau+2,2]} - d_{[\tau+2]} \end{aligned} \tag{8}$$

Lemma 1: For ρ = 1,2, …, τ-1

$$L_{[\rho]}(\pi_2) = L_{[\rho]}(\pi_1)$$

Proof: The proof follows from the fact that both sequences π_1 and π_2 have the same jobs in those positions.

Theorem 1: Let jobs i and j be two adjacent jobs in a given sequence. If the following conditions hold,

a) $d_i \leq d_j$ and $t_{i,2} \leq t_{j,2}$
b) $UBs_{i,1} + t_{i,1} \leq \min_{k\in\{1,\dots,n\}} t_{k,2} + LBs_{i,2}$
c) $UBs_{j,1} + t_{j,1} \leq \min_{k\in\{1,\dots,n\}} t_{k,2} + LBs_{j,2}$

then, $\max\{L_{[\tau]}(\pi_2), L_{[\tau+1]}(\pi_2), L_{[\tau+2]}(\pi_2)\} \leq \max\{L_{[\tau]}(\pi_1), L_{[\tau+1]}(\pi_1), L_{[\tau+2]}(\pi_1)\}$.

Proof: It follows from Eq. (4) and Eq. (5) that

$$\begin{aligned} L_{[\tau]}(\pi_2) - L_{[\tau+1]}(\pi_1) = &\max\{\boldsymbol{S_{i,2}}, \boldsymbol{S_{i,1}} + t_{i,1} - t_{[\tau-1,2]}\} - \max\{\boldsymbol{S_{j,2}},\ \boldsymbol{S_{j,1}} + t_{j,1} - t_{[\tau-1,2]}\} \\ &- \max\{\boldsymbol{S_{i,2}},\ \boldsymbol{S_{i,1}} + t_{i,1} - t_{j,2}\} - t_{j,2} \end{aligned} \tag{9}$$

By Eq. (2) and conditions b) and c) of the theorem, Eq. (9) reduces to

$$L_{[\tau]}(\pi_2) - L_{[\tau+1]}(\pi_1) = -\boldsymbol{S_{j,2}} - t_{j,2}$$

Thus,

$$L_{[\tau]}(\pi_2) \leq L_{[\tau+1]}(\pi_1). \tag{10}$$

Furthermore, by Eq. (5) and Eq. (6)

$$L_{[\tau+1]}(\pi_2) - L_{[\tau+1]}(\pi_1) = \max\{\boldsymbol{S}_{i,2}\,,\ \boldsymbol{S}_{i,1} + t_{i,1} - t_{[\tau-1,2]}\} + \max\{\boldsymbol{S}_{j,2}\,, \boldsymbol{S}_{j,1} + t_{j,1} - t_{i,2}\} - d_j \\ - \max\{\boldsymbol{S}_{j,2}\,,\ \boldsymbol{S}_{j,1} + t_{j,1} - t_{[\tau-1,2]}\} - \max\{\boldsymbol{S}_{i,2}\,,\ \boldsymbol{S}_{i,1} + t_{i,1} - t_{j,2}\} + d_i \tag{11}$$

By Eq. (2) and conditions b) and c) of the theorem, Eq. (11) reduces to

$$L_{[\tau+1]}(\pi_2) - L_{[\tau+1]}(\pi_1) = d_i - d_j$$

which implies

$$L_{[\tau+1]}(\pi_2) \le L_{[\tau+1]}(\pi_1) \tag{12}$$

by condition a) of the theorem. Therefore, it follows from Eq. (10) and Eq. (12) that

$$\max\{L_{[\tau]}(\pi_2), L_{[\tau+1]}(\pi_2)\} \le L_{[\tau+1]}(\pi_1)$$

which implies

$$\max\{L_{[\tau]}(\pi_2), L_{[\tau+1]}(\pi_2)\} \le \max\{L_{[\tau]}(\pi_1), L_{[\tau+1]}(\pi_1)\} \tag{13}$$

By Eq. (7) and Eq. (8), and conditions a), b), and c) of the theorem, it follows that

$$L_{[\tau+2]}(\pi_2) \le L_{[\tau+2]}(\pi_1) \tag{14}$$

It follows from Eq. (13) and Eq. (14) that

$$\max\{L_{[\tau]}(\pi_2), L_{[\tau+1]}(\pi_2), L_{[\tau+2]}(\pi_2)\} \le \max\{L_{[\tau]}(\pi_1), L_{[\tau+1]}(\pi_1), L_{[\tau+2]}(\pi_1)\}.$$

Lemma 2: For $\rho = \tau+3, \ldots, n$

$$L_{[\rho]}(\pi_2) \le L_{[\rho]}(\pi_1)$$

Proof: It follows by equation (1) that for $\rho = \tau+3, \ldots, n$

$$L_{[\rho]}(\pi_1) = \Delta + \max\{\boldsymbol{S}_{j,2}\,,\ \boldsymbol{S}_{j,1} + t_{j,1} - t_{[\tau-1,2]}\} + t_{j,2} + \max\{\boldsymbol{S}_{i,2}\,, \boldsymbol{S}_{i,1} + t_{i,1} - t_{j,2}\} + t_{i,2} \\ + \max\{\boldsymbol{S}_{[\tau+2,2]}, \boldsymbol{S}_{[\tau+2,1]} + t_{[\tau+2,1]} - t_{i,2}\} + t_{[\tau+2,2]} + \sum_{r=\tau+3}^{\rho} \max\{\boldsymbol{S}_{[r,2]},\quad \boldsymbol{S}_{[r,1]} + t_{[r,1]} - t_{[r-1,2]}\} + \sum_{r=\tau+3}^{\rho} t_{[r,2]} - d_{[\rho]} \tag{15}$$

$$L_{[\rho]}(\pi_2) = \Delta + \max\{\boldsymbol{S}_{i,2}\,,\ \boldsymbol{S}_{i,1} + t_{i,1} - t_{[\tau-1,2]}\} + t_{i,2} + \max\{\boldsymbol{S}_{j,2}\,, \boldsymbol{S}_{j,1} + t_{j,1} - t_{i,2}\} \\ + t_{j,2} + \max\{\boldsymbol{S}_{[\tau+2,2]}, \boldsymbol{S}_{[\tau+2,1]} + t_{[\tau+2,1]} - t_{j,2}\} + t_{[\tau+2,2]} \\ + \sum_{r=\tau+3}^{\rho} \max\{\boldsymbol{S}_{[r,2]},\quad \boldsymbol{S}_{[r,1]} + t_{[r,1]} - t_{[r-1,2]}\} + \sum_{r=\tau+3}^{\rho} t_{[r,2]} - d_{[\rho]} \tag{16}$$

From Eq. (2), Eq. (15), and Eq. (16) and conditions a), b), and c) of the theorem, we see that

$L_{[\rho]}(\pi_2) \le L_{[\rho]}(\pi_1)$ for $\rho = \tau+3, \ldots, n$

Theorem 2: Let jobs i and j be two adjacent jobs in a given sequence. If the following conditions hold,

a) $d_i \le d_j$ and $t_{i,2} \le t_{j,2}$

b) $UBs_{i,1} + t_{i,1} \le \min_{k\in\{1,\ldots,n\}} t_{k,2} + LBs_{i,2}$

c) $UBs_{j,1} + t_{j,1} \le \min_{k\in\{1,\ldots,n\}} t_{k,2} + LBs_{j,2}$

then, $L_{max}(\pi_2) \le L_{max}(\pi_1)$, i.e. job i should precede job j.

Proof: The proof directly follows from Lemma 1, Lemma 2, and Theorem 1.

Corollary 1: Let jobs *i* and *j* be two adjacent jobs in a given sequence. If the following conditions hold, for a deterministic problem where setup times are known for certain,

a) $d_i \leq d_j$ and $t_{i,2} \leq t_{j,2}$
b) $s_{i,1} + t_{i,1} \leq \min_{k\in\{1,\ldots,n\}} t_{k,2} + s_{i,2}$
c) $s_{j,1} + t_{j,1} \leq \min_{k\in\{1,\ldots,n\}} t_{k,2} + s_{j,2}$

then, $L_{max}(\pi_2) \leq L_{max}(\pi_1)$, i.e. job *i* precedes job *j*.

Proof: Since the problem is deterministic, Eq. (2) reduces to

$$LBs_{h,k} = \boldsymbol{S_{h,k}} = UBs_{h,k}$$

Moreover, since setup times are known values (denoted by $s_{h,k}$), the last equation is equivalent to

$$LBs_{h,k} = s_{h,k} = UBs_{h,k}$$

Hence, the proof follows from that of Theorem 2.

Theorem 3: Let jobs *i* and *j* be two adjacent jobs in a given sequence. If the following conditions hold,

a) $t_{j,2} \leq t_{i,2}$ and $d_i + t_{j,2} \leq d_j + t_{i,2}$
b) $UBs_{i,2} + \max_{k\in\{1,\ldots,n\}} t_{k,2} \leq t_{i,1} + LBs_{i,1}$
c) $UBs_{j,2} + \max_{k\in\{1,\ldots,n\}} t_{k,2} \leq t_{j,1} + LBs_{j,1}$

then, $\max\{L_{[\tau]}(\pi_2), L_{[\tau+1]}(\pi_2), L_{[\tau+2]}(\pi_2)\} \leq \max\{L_{[\tau]}(\pi_1), L_{[\tau+1]}(\pi_1), L_{[\tau+2]}(\pi_1)\}$.

Proof: By Eq. (2) and conditions b) and c) of theorem 3, Eq. (9) reduces to

$$L_{[\tau]}(\pi_2) - L_{[\tau+1]}(\pi_1) = -\boldsymbol{S_{j,1}} - t_{j,1}$$

which implies that

$$L_{[\tau]}(\pi_2) \leq L_{[\tau+1]}(\pi_1). \tag{17}$$

By Eq. (2) and conditions b) and c) of Theorem (3), Eq. (11) reduces to

$$L_{[\tau+1]}(\pi_2) - L_{[\tau+1]}(\pi_1) = -t_{i,2} - d_j + t_{j,2} + d_i$$

thus, by condition a) of Theorem 3

$$L_{[\tau+1]}(\pi_2) \leq L_{[\tau+1]}(\pi_1) \tag{18}$$

From Eq. (7) and Eq. (8), and the conditions of Theorem 3, we obtain

$$L_{[\tau+2]}(\pi_2) \leq L_{[\tau+2]}(\pi_1) \tag{19}$$

It follows from Eq. (17), Eq. (18), and Eq. (19) that

$$\max\{L_{[\tau]}(\pi_2), L_{[\tau+1]}(\pi_2), L_{[\tau+2]}(\pi_2)\} \leq \max\{L_{[\tau]}(\pi_1), L_{[\tau+1]}(\pi_1), L_{[\tau+2]}(\pi_1)\}.$$

Theorem 4: Let jobs *i* and *j* be two adjacent jobs in a given sequence. If the following conditions hold,

a) $t_{j,2} \leq t_{i,2}$ and $d_i + t_{j,2} \leq d_j + t_{i,2}$
b) $UBs_{i,2} + \max_{k\in\{1,\ldots,n\}} t_{k,2} \leq t_{i,1} + LBs_{i,1}$
c) $UBs_{j,2} + \max_{k\in\{1,\ldots,n\}} t_{k,2} \leq t_{j,1} + LBs_{j,1}$

then, $L_{max}(\pi_2) \leq L_{max}(\pi_1)$, i.e. job *i* should precede job *j*.

Proof: Note that Lemma 2 holds under the conditions of Theorem 3. Therefore, the proof follows from Lemma 1, Lemma 2, and Theorem 3.

Corollary 2: Let jobs *i* and *j* be two adjacent jobs in a given sequence. If the following conditions hold, for a deterministic problem where setup times are known for certain,

a) $t_{j,2} \leq t_{i,2}$ and $d_i + t_{j,2} \leq d_j + t_{i,2}$

b) $s_{i,2} + \max_{k \in \{1,\ldots,n\}} t_{k,2} \leq t_{i,1} + s_{i,1}$

c) $s_{j,2} + \max_{k \in \{1,\ldots,n\}} t_{k,2} \leq t_{j,1} + s_{j,1}$

then, $L_{max}(\pi_2) \leq L_{max}(\pi_1)$, i.e. job *i* should precede job *j*.

Proof: For the same reasons stated in the proof of Corollary 1, it is true that

$LBs_{h,k} = s_{h,k} = UBs_{h,k}$

Thus, the proof follows from that of Theorem 4.

4. Numerical Results for Dominance Relations

Allahverdi et al. (2021) presented a dominance relation for our problem and showed that their dominance relation was about 90% more efficient than the earlier existing ones in the literature. In this paper, we develop two new dominance relations (Theorems 2 and 4) for the problem and show that our newly developed dominance relations are more efficient than that of Allahverdi et al. (2021). The dominance relation developed in Theorem 2 is denoted by *DR2* and the one in Theorem 4 is represented by *DR4*. It should be noted that *DR2* and *DR4* are not overlapping since the conditions stated in Theorems 2 and 4 cannot hold at the same time. In other words, these two theorems are concerned with different cases. Therefore, both theorems can be applied for a given sequence. Hence, we denote the combination of *DR2* and *DR4* by *DRC*. Let *n* denote the number of jobs, *R* denote the range factor and *T* denote tardiness factor. Also, let Δ denote the uncertainty range between the lower and upper bounds of setup times. For a fair comparison, we use the same computational settings of Allahverdi et al. (2021). Specifically,

- $n \in \{30, 40, 50, 60, 70\}$
- $R \in \{0.25, 0.50, 0.75\}$
- $T \in \{0.25, 0.50, 0.75\}$
- $\Delta \in \{5, 10, 20\}$
- $t_{i,k}$ follows a uniform distribution, $U(1, 100)$
- $Us_{i,k}$ follows a uniform distribution, $U(50, 100)$,
- $Ls_{i,k} = Us_{i,j} - \Delta$
- d_i follows a uniform distribution, *U(LB-Cmax(1-T-R/2), LB-Cmax(1-T-R/2))* where *LB-Cmax* is a lower bound value on makespan.

Hall and Posner (2001) suggest generating processing times from the uniform distribution U(1,100) since its variance is very large. On the other hand, it is common to generate due dates using a uniform distribution between *LB-Cmax(1-T-R/2)* and *LB-Cmax(1-T-R/2)* where *R* and *T* are usually taken between 0 and 1, e.g., Kim (1993), and Vallada and Ruiz (2010), and *LB-Cmax* is a lower bound on makespan. The following *LB-Cmax* is used.

$$LB\text{-}Cmax = \sum_{i=1}^{n} LBs_{i,2} + \sum_{i=1}^{n} t_{i,2}.$$

Tables 1-3 present computational results for Δ=5, Δ=10, and Δ=20, respectively. The numbers in the tables are the average of 250 replications. The first column in the tables show the *R* values and the second column indicates n values. The next three columns present the results for the percentage improvement of *DR2*, *DR4*, and *DRC* (combined dominance relations), respectively for *T*=0.25. The next three columns give the results for the percentage improvements for *T*=0.5, and the following three columns are the results for *T*=0.75. For example, the bold numbers **67.7, 66.9**, and **134.6** for *R*=0.5, *n*=40, *T*=0.25 in Table 1 show that the percentage of improvement of *DRC* is 134.6% where 67.7% of this improvement comes from *DR2* while 66.9% of the improvement comes from *DR4*. This specific result shows that the newly developed dominance relation *DRC* is 134.6% more efficient than that of the best existing dominance relation in the literature. It should be noted that the overall percentage improvements of newly developed dominance relations with respect to all the parameters of *n, R, T*, Δ, and 250 replications, which results in 33,750 runs (5×3×3×3×250), are 67.7, 69.6, 137.3 for *DR2*, *DR4*, and *DRC*, respectively. Figs. 2-5 summarize the results in Tables 1-3 with respect to *n, R, T*, and Δ values, respectively, where the y-axis denotes the percentage improvement for dominance relations. Fig. 2 shows the percentage improvements of *DR2*, *DR4*, and *DRC* with respect to the number of jobs. It shows that the percentage slightly decreases as the number of jobs increases. However, the percentage of improvement is at least 120 for the *DRC*. Results with respect to the *R* values are given in Fig. 3 where the percentage improvements of the dominance relations are independent of the *R* values. In other words, the improvement of dominance relations is similar for different due date ranges. A similar effect can be seen from Fig. 4 with respect to tardiness factor *T*. However, the percentage improvement of the dominance relations increases as the Δ (Delta) values increase. In other words, as the uncertainty level increases, the newly developed dominance relations perform better, which is another strength of the newly developed dominance relations.

Table 1

Percentage Improvement of Dominance Relations (Δ = 5)

		T=0.25			*T*=0.5			*T*=0.75		
		Dominance Relations			**Dominance Relations**			**Dominance Relations**		
R	***n***	**DR2**	**DR4**	**DRC**	**DR2**	**DR4**	**DRC**	**DR2**	**DR4**	**DRC**
0.25	30	79.9	76.4	156.3	56.3	63.9	120.2	66.5	71.9	138.3
	40	66.9	67.8	134.7	59.0	55.0	114.0	70.3	67.8	138.1
	50	55.8	60.9	116.7	56.2	56.6	112.8	65.1	70.3	135.3
	60	57.9	52.8	110.8	55.7	64.4	120.1	57.0	61.8	118.8
	70	69.3	61.8	131.1	56.9	65.7	122.6	51.2	50.4	101.6
0.5	30	61.8	70.8	132.5	63.0	65.7	128.8	83.9	97.7	181.6
	40	**67.7**	**66.9**	**134.6**	55.0	57.0	112.0	66.7	64.6	131.3
	50	67.6	64.3	131.9	49.6	53.4	103.0	49.8	59.8	109.6
	60	54.9	55.0	109.8	53.2	57.1	110.3	65.8	60.0	125.9
	70	49.5	49.8	99.3	58.2	58.4	116.6	61.3	60.3	121.6
0.75	30	47.7	54.0	101.7	56.4	66.7	123.1	69.3	65.8	135.1
	40	64.5	59.9	124.3	54.6	59.3	113.9	62.8	56.5	119.3
	50	44.8	49.4	94.2	63.4	58.6	122.0	55.5	56.5	112.0
	60	72.1	71.0	143.2	53.8	51.6	105.4	57.8	61.8	119.5
	70	49.0	50.3	99.3	53.3	57.6	110.8	57.0	59.1	116.1
Avg	30	63.1	67.1	130.2	58.6	65.4	124.0	73.2	78.5	151.7
	40	66.3	64.8	131.2	56.2	57.1	113.3	66.6	63.0	129.6
	50	56.1	58.2	114.3	56.4	56.2	112.6	56.8	62.2	119.0
	60	61.6	59.6	121.3	54.2	57.7	111.9	60.2	61.2	121.4
	70	56.0	54.0	109.9	56.1	60.5	116.7	56.5	56.6	113.1

Table 2

Percentage Improvement of Dominance Relations (Δ = 10)

		T=0.25			*T*=0.5			*T*=0.75		
		Dominance Relations			**Dominance Relations**			**Dominance Relations**		
R	***n***	**DR2**	**DR4**	**DRC**	**DR2**	**DR4**	**DRC**	**DR2**	**DR4**	**DRC**
0.25	30	74.9	83.6	158.5	57.5	71.6	129.1	58.7	67.2	126.0
	40	71.9	86.3	158.2	68.3	70.2	138.5	68.8	70.2	138.9
	50	63.0	62.2	125.2	107.3	113.3	220.6	62.1	63.1	125.1
	60	63.5	62.2	125.7	50.2	58.6	108.8	47.0	58.8	105.8
	70	54.4	54.4	108.8	59.0	62.1	121.1	50.2	49.9	100.1
0.5	30	88.7	88.1	176.9	62.9	64.2	127.1	80.3	94.2	174.5
	40	61.6	64.5	126.1	56.4	54.7	111.1	66.8	66.0	132.8
	50	63.8	63.6	127.4	57.4	67.6	125.0	63.2	61.7	125.0
	60	67.7	70.2	137.8	66.5	65.8	132.3	55.3	65.9	121.2
	70	62.1	66.6	128.7	51.4	56.7	108.1	57.1	61.3	118.4
0.75	30	81.6	70.2	151.7	63.5	70.2	133.7	60.5	55.6	116.1
	40	58.1	56.8	114.9	58.4	55.1	113.6	57.7	59.6	117.3
	50	47.1	50.2	97.3	53.9	57.1	111.0	72.3	75.1	147.3
	60	59.7	66.0	125.7	71.8	68.9	140.7	63.6	63.5	127.1
	70	57.0	62.2	119.1	70.0	69.9	139.8	66.2	65.6	131.8
Avg	30	81.7	80.6	162.4	61.3	68.6	129.9	66.5	72.3	138.9
	40	63.8	69.2	133.0	61.0	60.0	121.0	64.4	65.3	129.7
	50	58.0	58.6	116.6	72.9	79.3	152.2	65.9	66.6	132.5
	60	63.6	66.1	129.7	62.8	64.4	127.3	55.3	62.7	118.0
	70	57.8	61.1	118.9	60.1	62.9	123.0	57.8	59.0	116.8

Table 3

Percentage Improvement of Dominance Relations (Δ = 20)

		T=0.25			*T*=0.5			*T*=0.75		
		Dominance Relations			**Dominance Relations**			**Dominance Relations**		
R	***n***	**DR2**	**DR4**	**DRC**	**DR2**	**DR4**	**DRC**	**DR2**	**DR4**	**DRC**
0.25	30	92.8	77.9	170.7	75.5	87.7	163.2	71.8	73.1	144.9
	40	66.1	72.5	138.5	66.1	68.6	134.7	93.1	114.6	207.7
	50	97.2	106.7	204.0	96.1	99.3	195.5	94.4	79.3	173.7
	60	75.0	69.1	144.1	58.8	60.1	118.9	84.8	88.7	173.6
	70	68.9	63.0	131.9	65.6	73.9	139.5	54.7	52.9	107.6
0.5	30	81.8	78.6	160.4	107.2	89.9	197.0	81.5	84.0	165.5
	40	80.8	100.8	181.6	70.5	90.0	160.5	92.9	88.7	181.6
	50	39.6	40.6	80.2	81.2	86.7	167.9	97.6	107.7	205.4
	60	69.1	72.3	141.4	117.4	111.4	228.8	96.8	93.4	190.3
	70	59.0	63.0	121.9	79.0	64.0	143.1	72.0	93.8	165.8
0.75	30	93.5	74.6	168.1	91.4	117.9	209.3	89.5	66.2	155.7
	40	73.9	64.7	138.7	73.0	68.1	141.1	69.6	79.7	149.3
	50	87.4	88.3	175.7	77.8	68.9	146.7	48.4	59.1	107.5
	60	73.1	82.2	155.3	82.1	74.6	156.7	85.2	81.7	166.9
	70	81.8	91.3	173.2	107.1	99.6	206.7	63.9	68.5	132.4
Avg	30	89.4	77.0	166.4	91.4	98.5	189.9	80.9	74.4	155.3
	40	73.6	79.3	152.9	69.9	75.6	145.4	85.2	94.3	179.5
	50	74.8	78.5	153.3	85.0	85.0	170.0	80.1	82.0	162.2
	60	72.4	74.5	146.9	86.1	82.0	168.1	89.0	88.0	176.9
	70	69.9	72.4	142.3	83.9	79.2	163.1	63.5	71.7	135.3

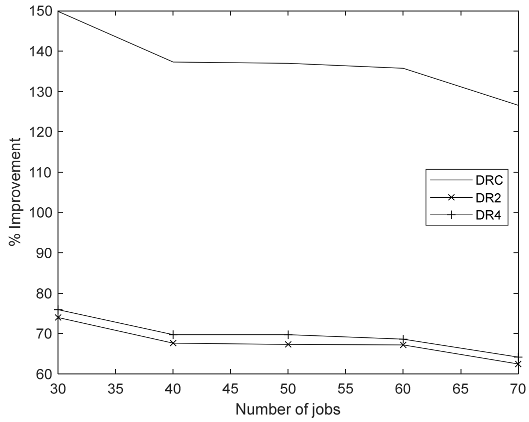

Fig. 2. Percentage Improvement versus number of jobs

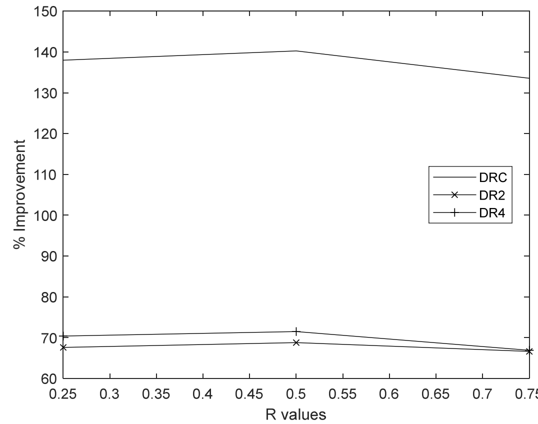

Fig. 3. Percentage Improvement versus *R* values

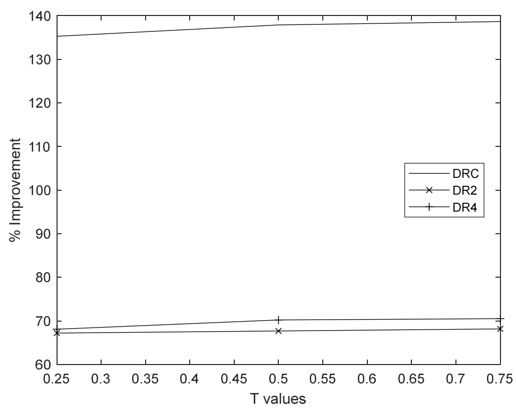

Fig. 4. Percentage Improvement versus *T* values

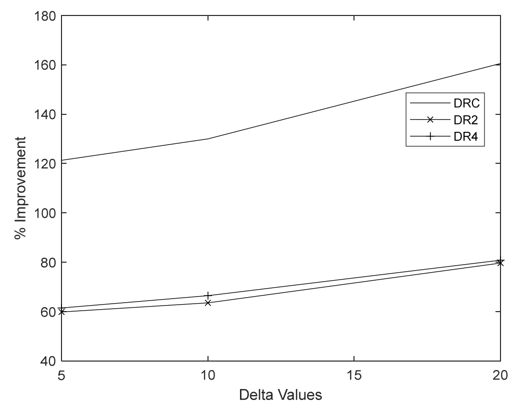

Fig. 5. Percentage Improvement versus Delta values

5. Heuristics

Allahverdi et al. (2021) recently presented a constructive heuristic, based on lower and upper bounds of setup times, with five versions for the considered problem. They demonstrated that one version of the heuristics, called *CH4*, performed better than the rest. In this section, we propose new heuristics for the problem and show in the next section that they are much more efficient than that of Allahverdi et al. (2021) with the same computational time. Let π denote a partial sequence of scheduled jobs and σ the set of jobs which are going to be scheduled. Moreover, let A be the set of all jobs and *nw* the number of jobs to be scheduled. Also, let $L_{\max}(\pi)$ represent the maximum lateness of the partial sequence π.

Proposed Heuristics – *PH(α, β)*

Step 1: Initialize the set of jobs to be scheduled to the set of all the jobs ($\sigma \leftarrow$A) and the partial sequence π to the empty sequence ($\pi \leftarrow \emptyset$)
Step 2: Initialize the number of unscheduled jobs to the number of all jobs (nw $\leftarrow$ n)
Step 3: Start scheduling the first job, $js \leftarrow 1$.
Step 4: Choose $\alpha \in [0, N]$ and $\beta \in [0, N]$ to be used to construct the setup time
Step 5: Construct a setup time, s_{ik}, between the lower (LBs_{ik}) and upper bounds (UBs_{ik})

$$s_{i1} \leftarrow [\alpha UBs_{i1} + (1 - \alpha) LBs_{i1}]/N \text{ and } s_{i2} \leftarrow [\beta UBs_{i2} + (1 - \beta) LBs_{i2}]/N$$

Step 6: Among the unscheduled jobs, choose the job $t_1 \in \sigma$ with the smallest due date.
Step 7: Schedule the job t_1 as the first job ($\pi \leftarrow \{t_1\}$)
Step 8: Take the job t_1 out of the set of unscheduled jobs ($\sigma \leftarrow \sigma \setminus \{t_1\}$)
Step 9: Reduce the number of unscheduled jobs by 1 ($nw \leftarrow nw - 1$)
Step 10: Start the scheduling of the next job ($js \leftarrow js + 1$)
Step 11: For each job w from σ

a) Form a partial sequence π_w by scheduling the job $w \in \sigma$ as the job to be processed after the jobs in the sequence π ($\pi_w = \pi \cup \{\text{job } w\}$). In other words, place the job w in position js of π_w.
b) Compute the maximum lateness, $L_{\max}(\pi_w)$, of the partial sequence π_w as below:

$L_{\max} = \max\ \{C_{[k]} - d_{[k]}, k = 1,\ldots,js\}$,

where for the job in position k, $L_{[k]}$ stands for the lateness, $C_{[k]}$ represents the completion time, $d_{[i]}$ stands for the due date.

Step 12: Among the calculated *nw* values in Step 11, choose the job w^* which has the smallest maximum lateness. In other words, $L_{\max}(\pi_{w^*}) = \max\ \{L_{\max}(\pi_w), w \in \sigma\}$

Step 13: Form *js* sequences ($seq_1,\ldots,seq_{js}$) by inserting the job w^* in all possible positions (positions 1 up to *js*) and compute $L_{\max}$ of each sequence.

For example, place the job w^* in the first position and shift all the jobs in the sequence π by one position to the right to produce seq_1.

$seq_1 \leftarrow \{\text{job } w^*\} \cup \pi$

Similarly, job w^* is placed in the 2nd position of μ and jobs in positions greater than or equal to two are shifted by one position to the right (if $j \geq 2$) in seq_2.

Step 14: Among the *js* sequences formed in Step 13, choose the sequence (seq_{j^*}) which has the largest maximum completion time.

$L_{\max}(seq_{j^*}) = \max\ \{L_{\max}(seq_1),\ldots, L_{\max}(seq_{js})\}$

Step 15: Update the partial sequence of scheduled jobs π by assigning it to seq_{j^*}. ($\pi \leftarrow seq_{j^*}$)

Step 16: Increase the number of scheduled jobs by 1 ($js \leftarrow js + 1$).

Step 17: Decrease the number of unscheduled jobs by 1 ($nw \leftarrow nw - 1$).

Step 18: Take the recently scheduled job w^* out of the set of unscheduled jobs ($\sigma \leftarrow \sigma \setminus \{w^*\}$).

Step 19: Go to Step 11 if there are jobs waiting to be scheduled (If $js < n$)

Stop if all of the jobs are scheduled (If $js = n$)

For our proposed heuristics, we set the parameter N to 4 after computational experiments. Hence, we set α and β to be integer values between 0 and 4, specifically $\alpha \in \{0, 1, 2, 3, 4\}$ and $\beta \in \{0, 1, 2, 3, 4\}$. This results in 25 heuristics. For example, α=2 and β=3 in the proposed heuristic is denoted by *PH(2, 3).*

6. Computational Results for Heuristics

We evaluate the performance of the proposed 25 heuristics, *PH(0,0)* to *PH(4,4),* and the best existing heuristic *CH4* in the literature. Experimental computations are conducted using the software Matlab on a PC with Intel(R) Core(TM) i7-8550U CPU processor of 1.99 GHz with 8 GB RAM. We set the parameters to the values used in Section 4. To compare the performances of the proposed heuristics and the best one in the literature, the realized setup times (between the lower and upper bounds) are necessary. Since the actual setup times are uncertain and only the lower and upper bounds are known, we considered different distributions to generate actual setup times. These distributions, given below, include both skewed (positive and negative) and symmetric distributions. Details and justifications for using these distributions are given in Allahverdi et al. (2021).

It should be noted that 1,200 replications are conducted for each combination of *n, R, T,* Δ, and distributions. More specifically, n changes from 30 to 70 with the increment of 10, R and T take values of 0.25, 0.50, and 0.75, Δ takes values of 5, 10, and 20. Six different distributions are used for generating realized setup times, i.e., normal distribution, uniform distribution, positive and negative linear distributions, positive and negative exponential distributions. These six distributions are the distributions that were used in Allahverdi et al. (2021). This results in 8100 combinations, and for each combination, 1200 replications are generated. Thus, the number of problems considered are 97,200 problems. The relative error (*RE*) of the heuristic *PH(i, j)* and the benchmark heuristic *CH4* are computed as:

$$RE(X) = 100 \times [\ L_{max}(X) - L_{max}(best)\ ] \ /\ L_{max}(best)$$

where "*X*" denotes one of the heuristics of *PH(i, j)* and *CH4*, and "*best"* refers to the best of these heuristics. There are 25 proposed heuristics and the benchmark heuristic *CH4*, in total 26 heuristics, to compare with each other. The computational experiments were conducted for all the 26 heuristics. However, reporting all the computational results for all the heuristics is cumbersome and infeasible. Therefore, we present the results of the two best-proposed heuristics, *PH(4, 3)* and PH(5, 4), and the two worst proposed heuristics, *PH(1, 1)* and *PH(2, 2)*, along with the benchmark heuristic *CH4*, in total 5 heuristics.

The RE values for these 5 heuristics are presented in Table 4, Table 5, and Table 6, for Δ=5, Δ=10, and Δ=20, respectively. An entry in a table is the average error regarding the considered distributions with 1200 runs. Standard deviations were small as a result of a large number of replications, and hence, they are not reported. The results in Tables 4-6 are summarized in Figs. 6-10. Fig. 6 indicates the RE values of the five heuristics with respect to the number of jobs, *n*. The figure clearly shows that the proposed heuristics *PH(4, 3), PH(5, 4), PH(1, 1)* and *PH(2, 2)* perform much better than the benchmark heuristic of *CH4*. Moreover, even though the performance of the proposed heuristics gets better as *n* increases, as can better be seen in Fig. 7, however, that of the benchmark heuristic *CH4* deteriorates. It is seen in Figure 6 that the performances of the proposed heuristics *PH(4, 3), PH(5, 4), PH(1, 1)* and *PH(2, 2)* look much better than the heuristic *CH4*. Furthermore, it is clear from Fig. 7 (excluding *CH4*) that the difference between the best and the worst proposed heuristics is about 40%. The results for *R, T*, and Δ values are presented in Figures 8, 9, and 10, respectively, by excluding *CH4* for a vivid comparison of the proposed heuristics. Fig. 8 shows that the performances of *PH(4, 3)* and *PH(5, 4)* get slightly better as *R* increases. Moreover, Figure 9 indicates that the performances of the proposed heuristics get better as *T* increases. As expected, as Δ values increase, the *RE* values of the proposed heuristics slightly increase. The overall errors of the proposed heuristics *PH(4, 3), PH(5, 4), PH(1, 1), PH(2, 2),* and the benchmark heuristic *CH4* are 1.38, 1.37, 2.03, 1.90, 27.56, respectively. This shows that the worst of the proposed heuristics, *PH(1, 1),* reduces the error of the benchmark heuristic (the best existing heuristic in the literature), *CH4*, by 92%. Moreover, the error of the best proposed heuristic, *PH(5, 4)*, is about 32% better than the heuristic *PH(1, 1).* Finally, it should be noted that the CPU times of the proposed heuristics and that of the benchmark heuristic *CH4* were similar, and it was less than one second for the largest problem size.

Table 4
Heuristic *RE* (Δ = 5)

		T=0.25			***T***=0.5			***T***=0.75		
			R			***R***			***R***	
n	Heuristic	0.25	0.5	0.75	0.25	0.5	0.75	0.25	0.5	0.75
	PH(5,4)	2.39	1.72	1.25	1.81	1.36	0.35	1.05	1.01	0.64
	PH(4,3)	2.23	1.68	1.56	1.79	1.35	0.60	1.01	0.95	0.76
30	*PH(1,1)*	3.04	2.07	2.52	2.40	1.99	1.25	1.36	1.19	0.89
	PH(2,2)	2.95	2.01	2.44	2.33	1.81	1.11	1.34	1.19	0.97
	CH4	15.78	20.38	83.74	14.77	8.77	28.97	19.82	16.15	20.56
	PH(5,4)	2.00	1.54	0.85	1.81	0.89	0.69	1.03	0.88	0.62
	PH(4,3)	1.99	1.56	1.41	1.77	0.86	0.69	1.01	0.89	0.58
40	*PH(1,1)*	2.79	2.12	4.47	2.23	1.12	0.84	1.41	1.11	0.97
	PH(2,2)	2.78	2.18	4.31	2.26	1.10	0.95	1.32	1.07	0.76
	CH4	17.18	27.67	93.21	13.69	11.45	27.24	20.19	15.13	25.31
	PH(5,4)	2.25	2.43	2.23	1.38	1.26	0.75	1.05	0.75	0.44
	PH(4,3)	2.40	2.39	2.36	1.34	1.18	0.88	1.01	0.79	0.51
50	*PH(1,1)*	3.07	3.44	3.64	1.78	1.70	1.30	1.39	0.95	0.85
	PH(2,2)	3.02	3.22	3.27	1.73	1.54	1.27	1.37	0.93	0.54
	CH4	16.27	21.13	83.05	14.12	11.59	30.24	20.25	15.55	29.42
	PH(5,4)	1.97	1.90	1.35	1.40	1.12	0.75	1.08	0.68	0.50
	PH(4,3)	2.00	2.07	1.39	1.49	1.06	0.85	1.08	0.75	0.50
60	*PH(1,1)*	2.47	2.59	2.64	2.08	1.45	1.22	1.37	1.00	0.72
	PH(2,2)	2.50	2.60	2.99	2.05	1.42	1.13	1.38	0.90	0.65
	CH4	16.43	20.23	85.54	14.76	13.69	30.57	19.89	13.69	26.15
	PH(5,4)	2.37	1.66	2.37	1.31	1.17	0.64	0.78	0.58	0.32
	PH(4,3)	2.36	1.64	2.32	1.22	1.23	0.64	0.80	0.59	0.45
70	*PH(1,1)*	3.16	2.07	3.15	1.63	1.76	1.12	1.07	0.85	0.67
	PH(2,2)	3.15	1.99	3.05	1.60	1.56	0.91	1.06	0.79	0.71
	CH4	15.68	23.49	92.41	15.92	15.45	36.88	20.02	17.10	28.34

Table 5
Heuristic *RE* (Δ = 10)

		T=0.25			***T***=0.5			***T***=0.75		
			R			***R***			***R***	
n	Heuristic	0.25	0.5	0.75	0.25	0.5	0.75	0.25	0.5	0.75
	PH(5,4)	3.09	2.50	2.76	1.66	1.17	1.38	1.22	1.22	0.66
	PH(4,3)	3.14	2.49	3.76	1.57	1.27	1.51	1.25	1.21	0.76
30	*PH(1,1)*	4.13	3.53	4.79	2.17	1.66	2.27	1.76	1.59	1.18
	PH(2,2)	3.99	3.13	4.96	2.15	1.59	2.42	1.75	1.47	1.17
	CH4	16.75	20.67	84.54	15.49	12.25	22.21	20.06	12.71	22.99
	PH(5,4)	2.55	2.36	1.82	1.43	1.49	0.90	1.30	0.71	0.73
	PH(4,3)	2.53	2.28	1.78	1.42	1.52	0.80	1.26	0.84	0.66
40	*PH(1,1)*	3.41	2.91	3.73	1.84	1.92	1.58	1.67	1.11	0.98
	PH(2,2)	3.39	2.92	3.43	1.70	1.74	1.30	1.70	1.17	0.82
	CH4	16.91	20.50	60.97	14.69	10.11	28.03	19.27	14.80	24.81
	PH(5,4)	2.17	2.62	2.85	1.71	1.21	1.11	0.98	1.00	0.56
	PH(4,3)	2.02	2.61	2.81	1.66	1.03	1.20	0.97	1.00	0.75
50	*PH(1,1)*	2.65	3.66	4.33	2.20	1.31	1.63	1.33	1.41	1.25
	PH(2,2)	2.47	3.39	4.10	2.27	1.14	1.52	1.27	1.33	1.21
	CH4	17.22	23.31	72.61	13.90	14.20	27.99	20.44	13.65	23.48
	PH(5,4)	1.80	1.76	1.91	1.25	1.11	0.96	1.14	0.79	0.60
	PH(4,3)	1.77	1.78	2.07	1.38	1.02	0.94	1.11	0.72	0.53
60	*PH(1,1)*	2.43	2.79	3.03	1.70	1.37	1.98	1.52	1.18	1.13
	PH(2,2)	2.30	2.47	2.70	1.79	1.41	1.52	1.55	1.12	1.17
	CH4	18.81	23.19	64.79	16.15	13.35	28.40	20.42	15.43	28.41
	PH(5,4)	2.17	1.75	1.92	1.31	1.18	1.05	1.23	0.57	0.73
	PH(4,3)	2.16	1.82	2.14	1.33	1.17	1.10	1.15	0.58	0.73
70	*PH(1,1)*	2.97	2.84	2.78	1.92	1.79	1.78	1.49	0.98	1.23
	PH(2,2)	3.18	2.60	2.81	1.71	1.75	1.48	1.48	0.85	1.12
	CH4	18.83	22.10	68.34	13.49	13.06	31.82	19.95	16.81	25.50

Table 6
Heuristic RE (Δ = 20)

		T=0.25			***T***=0.5			***T***=0.75		
			R			***R***			***R***	
n	Heuristic	0.25	0.5	0.75	0.25	0.5	0.75	0.25	0.5	0.75
30	*PH(5,4)*	2.69	2.54	1.95	1.67	1.17	1.19	1.31	1.02	0.94
	PH(4,3)	2.75	2.08	2.43	1.64	1.19	1.31	1.22	0.94	0.82
	PH(1,1)	3.91	2.82	4.59	2.10	2.00	2.59	1.66	1.28	1.75
	PH(2,2)	3.48	2.69	3.69	1.92	1.85	2.03	1.72	1.24	1.31
	CH4	22.22	21.89	47.39	21.25	17.91	29.41	25.93	20.99	28.38
40	*PH(5,4)*	2.57	1.88	2.94	1.39	1.31	1.31	1.30	1.06	0.99
	PH(4,3)	2.45	1.83	2.99	1.38	1.44	1.25	1.11	1.03	0.92
	PH(1,1)	3.08	2.61	4.60	1.98	1.97	2.10	1.57	1.38	1.29
	PH(2,2)	2.95	2.50	4.20	1.78	1.81	1.90	1.49	1.37	1.20
	CH4	22.47	25.20	68.78	20.49	18.37	28.68	26.03	19.80	29.78
50	*PH(5,4)*	2.33	1.70	2.88	1.42	1.14	1.29	1.07	0.94	0.99
	PH(4,3)	2.05	1.61	3.63	1.34	1.18	1.41	1.02	0.94	1.01
	PH(1,1)	2.82	2.32	4.98	1.75	1.57	2.72	1.53	1.52	1.39
	PH(2,2)	2.84	2.23	5.76	1.72	1.50	2.07	1.42	1.37	1.28
	CH4	23.28	22.10	65.67	20.57	18.35	31.98	26.66	20.04	29.89
60	*PH(5,4)*	1.76	1.92	2.33	1.30	0.91	1.63	1.21	0.81	1.01
	PH(4,3)	1.59	1.97	2.40	1.10	1.06	1.53	1.12	0.91	0.87
	PH(1,1)	2.64	2.12	4.22	1.50	1.38	2.02	1.38	1.49	1.48
	PH(2,2)	2.31	1.97	3.15	1.40	1.33	1.86	1.29	1.23	1.18
	CH4	22.97	24.88	71.62	21.96	20.35	32.27	26.37	21.55	31.20
70	*PH(5,4)*	1.80	1.94	3.25	1.53	0.96	1.24	1.05	0.82	0.73
	PH(4,3)	1.70	1.74	3.16	1.46	0.91	1.39	1.00	0.80	0.87
	PH(1,1)	2.58	2.70	4.64	2.06	1.64	2.32	1.47	1.16	1.47
	PH(2,2)	2.31	2.42	4.90	1.90	1.27	2.09	1.28	1.07	1.26
	CH4	21.45	28.13	63.87	20.52	18.56	35.81	25.58	21.93	31.40

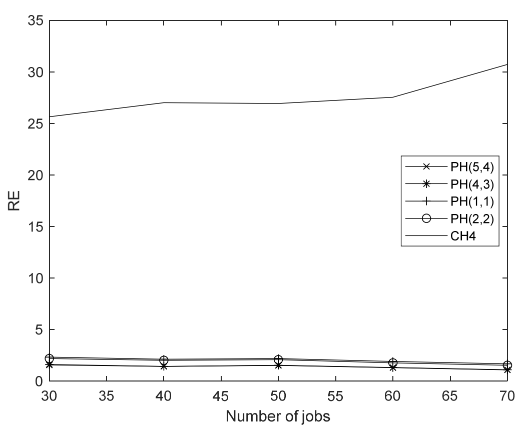

Fig. 6. Heuristic comparisions with respect to number of jobs

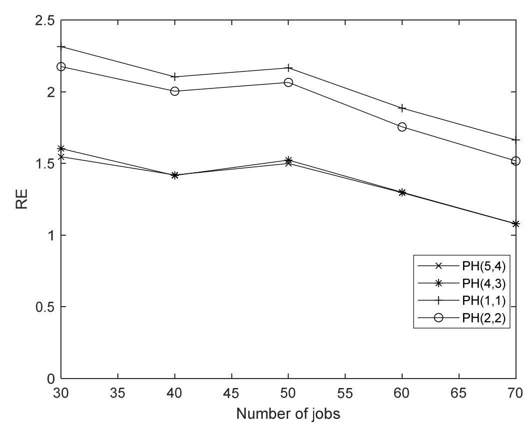

Fig. 7. Heuristic comparisions with respect to number of jobs without *CH4*

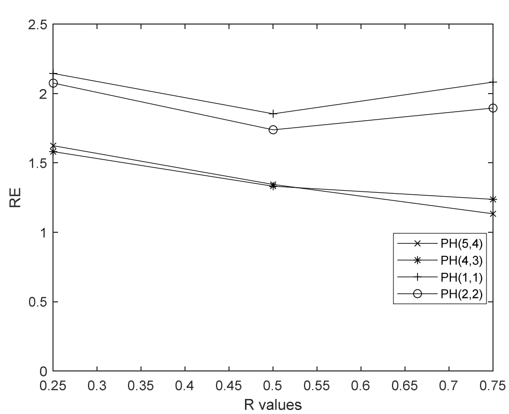

Fig. 8. Heuristic comparisions with respect to *R* values

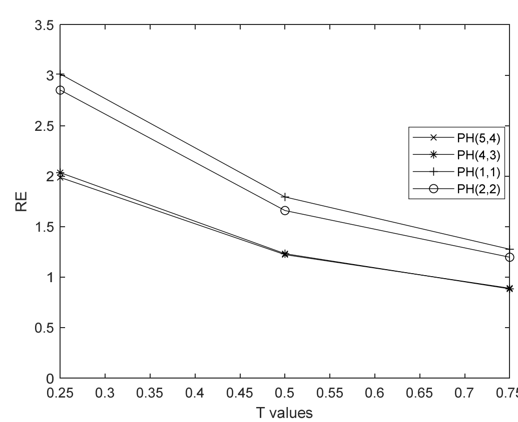

Fig. 9. Heuristic comparisions with respect to *T* values

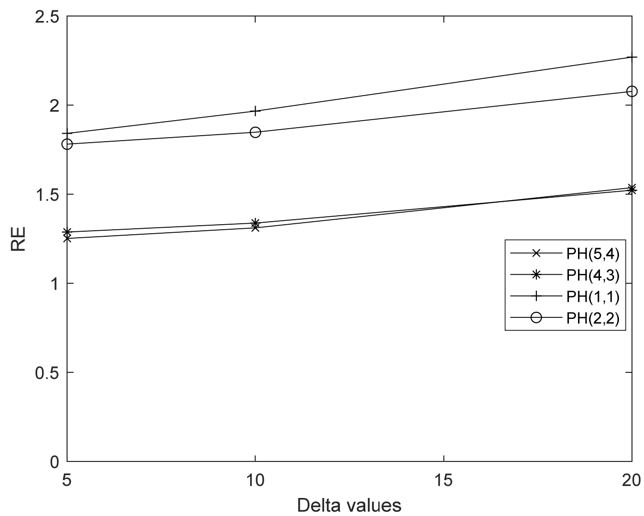

Fig. 10. Heuristic comparisions with respect to Delta values

7. Concluding Remarks

The two-machine no-wait flowshop scheduling problem is addressed with the objective of minimizing maximum lateness where setup times are modelled as uncertain with given lower and upper bounds. The problem was addressed earlier in the literature and dominance relations along with heuristics were presented. In this paper, we develop new dominance relations and show that they are at least 130% more efficient than the best existing one in the literature. Moreover, we propose new heuristics and show that the worst of the newly proposed heuristics is at least 90% more efficient than the existing one under the same CPU time. Given that the newly developed dominance relations are less restrictive, and the proposed heuristics are more efficient and they are computationally fast, they can easily be implemented for solving real life application problems. The addressed problem can be extended to other performance measures such as the number of tardy jobs, e.g., Aydilek et al. (2017). Furthermore, setup times are considered as sequence independent in the current paper. However, since setup times may be sequence dependent in some real-life applications (Guevara-Guevara et al. 2022), the results of this paper may also be extended to the case of sequence dependent setup times. Yet another extension to the problem is to consider total tardiness performance measure since this is a commonly considered performance measure, e.g., Braga-Santos et al. (20220, Rosssit et el. (2021), González-Neira and Montoya-Torres (2019) and Akande et al. (2018).

References

Akande, S., Oluleye, A., & Oyetunji, E. (2018). Effective heuristics for solving dynamic variant of single processor total tardiness problems. *Journal of Project Management*, *3*(1), 13-22.

Allahverdi, A., & Allahverdi, M. (2018). Two-machine no-wait flowshop scheduling problem with uncertain setup times to minimize maximum lateness. *Computational and Applied Mathematics*, *37*(5), 6774-6794.

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. (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. (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. (2022a). A survey of scheduling problems with uncertain interval/bounded processing/setup times. *Journal of Project Management*, *7*(4), 255-264.

Allahverdi, M. (2022b). An improved algorithm to minimize the total completion time in a two-machine no-wait flowshop with uncertain setup times. *Journal of Project Management*, *7*(1), 1-12.

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

Aydilek, A., Aydilek, H., & Allahverdi, A. (2013). Increasing the profitability and competitiveness in a production environment with random and bounded setup times. *International Journal of Production Research*, *51*(1), 106-117.

Aydilek, A., Aydilek, H., & Allahverdi, A. (2015). Production in a two-machine flowshop scheduling environment with uncertain processing and setup times to minimize makespan. *International Journal of Production Research*, *53*(9), 2803-2819.

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.

Braga-Santos, S., Barroso, G., & Prata, B. (2022). A size-reduction algorithm for the order scheduling problem with total tardiness minimization. *Journal of Project Management*, *7*(3), 167-176.

Braun, O., Lai, T. C., Schmidt, G., & Sotskov, Y. N. (2002). Stability of Johnson's schedule with respect to limited machine availability. *International Journal of Production Research*, *40*(17), 4381-4400.

Dileepan, P. (2004). A note on minimizing maximum lateness in a two-machine no-wait flowshop. *Computers & Operations Research*, *31*(12), 2111-2115.

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.

González-Neira, E., & Montoya-Torres, J. (2019). A simheuristic for bi-objective stochastic permutation flow shop scheduling problem. *Journal of Project Management*, *4*(2), 57-80.

Guevara-Guevara, A., Gómez-Fuentes, V., Posos-Rodríguez, L., Remolina-Gómez, N., & González-Neira, E. (2022). Earliness/tardiness minimization in a no-wait flow shop with sequence-dependent setup times. *Journal of Project Management*, *7*(3), 177-190.

Hall, N. G., & Posner, M. E. (2001). Generating experimental data for computational testing with machine scheduling applications. *Operations Research*, *49*(6), 854-865.

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.

Hecker, F. T., Stanke, M., Becker, T., & Hitzmann, B. (2014). Application of a modified GA, ACO and a random search procedure to solve the production scheduling of a case study bakery. *Expert systems with applications*, *41*(13), 5882-5891.

Hsu, V. N., De Matta, R., & Lee, C. Y. (2003). Scheduling patients in an ambulatory surgical center. *Naval Research Logistics (NRL)*, *50*(3), 218-238.

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.

Kim, S. C., & Bobrowski, P. M. (1997). Scheduling jobs with uncertain setup times and sequence dependency. *Omega*, *25*(4), 437-447.

Kim, J., Kröller, A., & Mitchell, J. (2009). Scheduling aircraft to reduce controller workload. In *9th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS'09)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Kouvelis, P., & Yu, G., (1997). Robust discrete optimization and its applications. *Kluwer Academic Publishers*, Boston.

Liu, S. Q., & Kozan, E. (2011). Scheduling trains with priorities: a no-wait blocking parallel-machine job-shop scheduling model. *Transportation Science*, *45*(2), 175-198.

Macchiaroli, R., Mole, S., & Riemma, S. (1999). Modelling and optimization of industrial manufacturing processes subject to no-wait constraints. *International Journal of Production Research*, *37*(11), 2585-2607.

Matsveichuk, N. M., Sotskov, Y. N., & Werner, F. (2011). The dominance digraph as a solution to the two-machine flow-shop problem with interval processing times. *Optimization*, *60*(12), 1493-1517.

Rossit, D.A., Toncovich, A., Rossit, D.G., & Nesmachnow, S. (2021). Solving a flow shop scheduling problem with missing operations in an Industry 4.0 production environment. *Journal of Project Management*, *6*, 33-44.

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.

Sotskov, Y. N. (2012). Measure of uncertainty for Bellman-Johnson problem with interval data/Yu. N. Sotskov, NM Matsveichuk. *Cybernetics and System Analysis*, *48*(5), 641-652.

Sotskov, Y. N., Egorova, N. G., & Lai, T. C. (2009). Minimizing total weighted flow time of a set of jobs with interval processing times. *Mathematical and Computer Modelling*, *50*(3-4), 556-573.

Sotskov, Y. N., & Lai, T. C. (2012). Minimizing total weighted flow time under uncertainty using dominance and a stability box. *Computers & Operations Research*, *39*(6), 1271-1289.

Vallada, E., & Ruiz, R. (2010). Genetic algorithms with path relinking for the minimum tardiness permutation flowshop problem. *Omega*, *38*(1-2), 57-67.

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.