International Journal of Engineering

Journal Homepage: www.ije.ir

Minimizing Makespan with Start Time-Dependent Jobs in a Two-Machine Flow Shop

A. A. Jafari[a], H. Khademi Zare[a*], M. M. Lotfi[a], R. Tavakkoli-Moghaddam[b]

[a] *Department of Industrial Engineering, Faculty of Engineering, Yazd University, Yazd, Iran*
[b] *School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran*

PAPER INFO

Paper history:
Received 17 February 2016
Received in revised form 03 May 2016
Accepted 03 May 2016

Keywords:
Linear Deteriorating Jobs
Start Time-Dependent
Flow Shop
Makespan
Branch and Bound
Heuristics

ABSTRACT

The purpose of this paper is to consider the problem of scheduling a set of start time-dependent jobs in a two-machine flow shop, in which the actual processing times of jobs increase linearly according to their starting time. The objective of this problem is to minimize the makespan. The problem is known to be NP-hardness; therefore, there is no polynomial-time algorithm to solve it optimally in a reasonable time. So, a branch-and-bound algorithm is proposed to find the optimal solution by means of dominance rules, upper and lower bounds. Several easy heuristic procedures are also proposed to derive near-optimal solutions. To evaluate the performance of the proposed algorithms, the computational experiments are extracted based on the recent literature. Deteriorating jobs lead to an increase in the makespan of the problems; therefore, it is important to obtain the optimal or near-optimal solution. Considering the complexity of the problem, the branch-and-bound algorithm is capable of solving problems of up to 26 jobs. Additionally, the average error percentage of heuristic algorithms is less than 1.37%; therefore, the best one is recommended to obtain a near-optimal solution for large-scale problems.

***doi**: 10.5829/idosi.ije.2016.29.06c.07*

Parameters

$N=\{J_1,J_2,...,J_n\}$	Set of jobs to be processed that n is number of jobs	S_{ij}	Starting time of J_i on M_j,
M_1, M_2	two available machines	$S_{[k]j}$	Starting time of the job located in the kth position on M_j
$J_{[k]}$	the job located in the kth position	$C_{ij}(S)$	Completion time of J_i on M_j under schedule S
P_{ij}	Actual processing time of J_i on M_j,	$C_{[k]j}(S)$	Completion time of the job located in the kth position on M_j, under schedule S
P'_{i1}	Unreal processing time of J_i on M_j,	$C_{Max}=C_{[n]2}(S)$	Makespan
a_{ij}	Normal processing time of J_i on M_j,	δ	Partial sequence including scheduled jobs
b_i	Deteriorating rate of J_i	δ'	Unscheduled jobs or complementary of δ

1. INTRODUCTION

In the deterministic production and operations scheduling, most of the researchers assume that the operation or processing times are fixed once they are given [1]. In many real-life situations, the actual processing time of a job increase according to the start time of the job, so that a job processed later consumes more time than the same job when it is processed earlier. For example, a drop in the temperature of an ingot while waiting to be processed in steel rolling, a delay in fire fighting, or a medical procedure under any delay. This type of problem is known as scheduling problem with start time-dependent processing times, in which it has received increasing attention in recent years [1-13].

*Corresponding Author's Email: *hkhademiz@yazd.ac.ir* (H. Khademi Zare)

Most models consider the problems in which the actual processing time of a job is defined as a linear or piecewise linear function of its starting time. The linear model is one of the most popular ones that are presented as $P_i = a_i + b_i S$, $P_i = a_i + bS$, $P_i = b_i S$, that is called general linear, linear and simple linear deteriorating jobs, respectively. We provide a critical review on the related literature for them.

Browne and Yechiali [2] studied a single machine scheduling problem with a general linear deteriorating jobs to minimize the makespan and proved that the optimal sequence is based on a non-decreasing rate of a_i / b_i. Bahalke at el. [4] proposed a tabu search and genetic algorithm for a scheduling problem of minimizing makespan on a single machine under general linear deteriorating jobs and sequence-dependent setup times. Hamta at el. [5] introduced a single machine scheduling problem with precedence constraints and general linear deterioration and developed a mathematical model based on binary integer programming. Lee at el. [6] addressed a general linear model in M machine flow shop and developed a B&B and two metaheuristic algorithms to minimize the total tardiness.

Jafari and Moslehi [1] showed that the scheduling problem on a single machine with linear deteriorating jobs to minimize the number of tardy jobs is NP-hard; hence, a B&B procedure and a heuristic as an upper bound was proposed. Lee et al. [7] investigated this problem with release times to minimize makespan and presented a heuristic and B&B algorithm to solve it. Wang and Wang [8] provided a B&B algorithm and two heuristic algorithms for the problem of minimizing the makespan in three machine flow shop with linear deterioration. Lee et al. [9] developed a B&B algorithm and several heuristics to minimize the makespan in two-machine flow shop with linear deterioration. Ng et al. [10] continued this problem to minimize the total completion time and proposed a B&B algorithm to solve it.

Wang et al. [14] used the B&B and heuristic algorithm to obtain the optimal solution in three machine flow shop with simple linear deterioration to minimize makespan. Wang et al. [15] showed that two machine flow shop scheduling problem under simple linear deteriorating jobs to minimize total completion time is NP-hard; they proposed a B&B algorithm able to handle 14 jobs. Yang and Wang [16] continued this problem to minimize total weighted completion time and developed a B&B and heuristic algorithm to solve the problem.

Some authors supposed that setup time of each job is not constant and it is similar to processing time as a simple linear function of start time. Lee et al. [17] investigated a single machine to minimize the number of tardy jobs with simple linear function for processing and setup time of jobs. They proposed a B&B algorithm for the problem which could solve the instances up to 1000 jobs in the reasonable times. Cheng et al. [18] and Lee and Lu [19] presented a B&B algorithm for this problem to minimize maximum tardiness and weighted number of tardy jobs, respectively.

There is a growing interest in the literature to study the Scheduling problems on flow shop environment. Researchers considered this problem with various assumptions, e.g., flow shop scheduling problem with modified learning effect [20], two-stage hybrid flow shop scheduling problem with serial batching [21], no-wait reentrant flow shop scheduling problem [22], and deteriorating jobs [6-9,14-16].

Scheduling problems with linear deteriorating jobs have received a sufficient attention in the recent years especially in the flow shop environment; but, as seen, the existing researches addressed linear deteriorating jobs scheduling problem with special deterioration functions as $P_i = a_i + bS$, $P_i = b_i S$. In many realistic situations however, jobs have the different deterioration rates e.g., $P_i = a_i + b_i S$; so, it is necessary to investigate the problem in a more realistic manner. Based on the mentioned gaps in the literature, in this paper we consider general linear deteriorating jobs in a two machine flow shop to minimize the makespan where the deterioration rates is different and the actual processing time of each job is calculated based on $P_i = a_i + b_i S$.

The rest of the paper is organized in the following manner: Section 2 is dedicated to describing the problem and its complexity. The exact algorithm and several heuristic procedures are established in Sections 3 and 4, respectively. In Section 5, the computational experiments are provided to assess the efficiency of the proposed algorithms. Finally in Section 6, conclusions are mentioned.

2. PROBLEM DESCRIPTION

In this section, we establish general linear deteriorating jobs scheduling problem in a two machine flow shop environment. The set of N consisting n jobs is processed first on M_1 and next on M_2. Two machines are assumed to be available all the times and each one can handle one job at a moment and each job can only be processed on one machine at a time. All the jobs are available for processing at time $T_0 \geq 0$ and will be processed without interruption or preemption. Also, it is assumed that P_{ij} is calculated according to a general linear function of start time S_{ij}, as Equation (1).

$$P_{ij} = a_{ij} + b_i S_{ij} \tag{1}$$

The objective is to find an optimal schedule S^* so that for any schedule S, we have:

$Max\left\{C_{12}(S^*),C_{22}(S^*),...,C_{n2}(S^*)\right\} \le Max\left\{C_{12}(S),C_{22}(S),...,C_{n2}(S)\right\}$ or $C_{[n]2}(S^*) \le C_{[n]2}(S)$.

We analyze the problem complexity at first, because no other research is observed so far. Lee *et al.* [9] proved that the linear deteriorating jobs scheduling problem in a two machine flow shop where the deterioration rates are equal belongs to the group of NP-hard problems. Since in our research, the deterioration rate of jobs is different, therefore this problem is known to be NP-hard, as well.

According to the complexity of the problem, a B&B algorithm will be presented to find the optimal solution. In the proposed B&B algorithm, appropriate lower bounds and dominance rules are adopted so that branches can be fathomed more rapidly and the optimal solution can be obtained in a reasonable time. In addition, several heuristic techniques are provided to find a near optimal solution as an upper bound for the B&B algorithm. In the coming sections, the B&B and heuristic algorithms are presented.

3. B&B ALGORITHM

In order to find the optimal solution for the described problem, a B&B algorithm using back tracking approach is proposed which requires very little memory. This algorithm assigns the jobs in a forward manner starting from the first position. Each time, a branch is chosen and systematically worked down the searching tree until either it is fathomed by dominance rules and/or lower bounds, or the final node is reached that is used as a substitute for the initial solution if the makespan is decreased. Therefore, an upper bound, some dominance rules and seven lower bounds are presented in Subsections 3.1, 3.2 and 3.3, respectively, to increase the efficiency of the B&B algorithm.

3. 1. Upper Bound

In this paper, the best heuristic algorithm, described in the next section, is considered as the upper bound for the problem and its resulting sequence will be the base of generation of searching tree.

3. 2. Dominance Rules

In order to facilitate the B&B algorithm, some properties of dominated sequences are required for node elimination. Suppose that schedule δ_1 has two adjacent jobs J_i and J_j with J_i immediately preceding J_j. Now, a pairwise interchange of J_i and J_j is performed to derive a new sequence δ_2. That is, $\delta_1 = (\pi, J_i, J_j, \pi')$ and $\delta_2 = (\pi, J_j, J_i, \pi')$ where π and π' are partial sequences. Let S_1 and S_2 denote the completion time of the last job in π on M_1 and M_2, respectively. It is obvious that S_1 and S_2 are equal in δ_1 and δ_2.

$(A): Either\ \ (1+b_j)(a_{i1}+a_{i1}b_j-b_ia_{j1}) \le a_{i2}+b_ia_{j2}$

$or\ \ (1+b_j)(a_{j1}+(1+b_j)(a_{i1}+S_1+b_iS_1)) \le a_{i2}+(1+b_i)(1+b_j)S_2+b_ia_{j2}$

$or\ \ a_{j2}+a_{i1}(2b_j+b_j^2-b_i)+a_{j1}(b_j-b_i^2-2b_i) \le a_{i2}+S_1(b_i-b_j)(1+b_i+b_j+b_ib_j)$

$(B): Either\ \ b_ja_{i2} \le b_ia_{j2}$

$or\ \ a_{j2}+(1+b_j)(1+b_i)(S_2-(1+b_i)S_1)+b_ja_{i2} \le (1+b_i)(a_{i1}+(1+b_i)a_{j1})$

$or\ \ (1+b_j)(1+b_i)(S_2-a_{j1}-S_1-b_jS_1) \le b_ia_{j2}-b_ja_{i2}$

$(C): Either\ \ a_{j2}+b_ja_{i2} \le (1+b_i)(a_{j1}+b_ia_{j1}-b_ja_{i1})$

$or\ \ (1+b_j)(1+b_i)(a_{i1}+S_1+b_iS_1)+b_ja_{i2} \le (1+b_i)(1+b_j)S_2+b_ia_{j2}$

$or\ \ (1+b_j)(1+b_i)(a_{i1}+b_iS_1)+b_ja_{i2} \le (1+b_i)(1+b_j)(a_{j1}+b_jS_1)+b_ia_{j2}$

If $Max\left\{C_{k1}(\delta_1),C_{j2}(\delta_1)\right\} \le Max\left\{C_{k1}(\delta_2),C_{i2}(\delta_2)\right\}$ is satisfy where J_k is the first job in π', then the makespan in δ_1 is less than that in δ_2. Based on this observation, δ_1 dominates δ_2. Suppose J_i and J_j satisfy $a_{i1}b_j \le a_{j1}b_i$ and the following conditions.

The relation $a_{i1}b_j \le a_{j1}b_i$ along with exactly one equation from the cases *A*, *B* and *C* will make a dominance rule. Consequently, we will have 27 dominance rules that now we only explain one of them as follows.

Property 1: If $(1+b_j)(a_{i1}+a_{i1}b_j-b_ia_{j1}) \le a_{i2}+b_ia_{j2}$, $b_ja_{i2} \le b_ia_{j2}$, $a_{j2}+b_ja_{i2} \le (1+b_i)(a_{j1}+b_ia_{j1}-b_ja_{i1})$ and $a_{i1}b_j \le a_{j1}b_i$, then δ_1 dominates δ_2.

Proof. The completion times of J_i and J_j on M_2 in δ_2 and δ_1 are, respectively, determined as follows:

$$\begin{aligned}
&C_{i2}(\delta_2) = a_{i2}+(1+b_i)Max\left\{a_{j2}+(1+b_j)(a_{j1}+(1+b_j)S_1),\right.\\
&\left.a_{j2}+(1+b_j)S_2, a_{i1}+(1+b_i)(a_{j1}+(1+b_j)S_1)\right\} =\\
&Max\left\{a_{i2}+(1+b_i)(a_{j2}+(1+b_j)(a_{j1}+S_1+b_jS_1)),\right.\\
&a_{i2}+(1+b_i)(a_{j2}+(1+b_j)S_2),\\
&\left.a_{i2}+(1+b_i)(a_{i1}+(1+b_i)(a_{j1}+S_1+b_jS_1))\right\}
\end{aligned} \tag{2}$$

and

$$\begin{aligned}
&C_{j2}(\delta_1) = a_{j2}+(1+b_j)Max\left\{a_{j1}+(1+b_j)(a_{i1}+(1+b_i)S_1),\right.\\
&\left.a_{i2}+(1+b_i)S_2, a_{i2}+(1+b_i)(a_{i1}+(1+b_i)S_1)\right\} =\\
&Max\left\{a_{j2}+(1+b_j)(a_{j1}+(1+b_j)(a_{i1}+S_1+b_iS_1)),\right.\\
&a_{j2}+(1+b_j)(a_{i2}+(1+b_i)S_2),\\
&\left.a_{j2}+(1+b_j)(a_{i2}+(1+b_i)(a_{i1}+S_1+b_iS_1))\right\}
\end{aligned} \tag{3}$$

Because of the relation $(1+b_j)(a_{i1}+a_{i1}b_j-b_ia_{j1}) \le a_{i2}+b_ia_{j2}$, first term in (3)≤ first term in (2).

Due to the relation $b_ja_{i2} \le b_ia_{j2}$, second term in (3)≤ second term in (2).

Owing to the relation $a_{j2}+b_j a_{i2} \le (1+b_i)(a_{j1}+b_i a_{j1}-b_j a_{i1})$, third term in (3)≤ third term in (2).
Since ith term in (3)≤ ith term in (2), Hence, $C_{j2}(\delta_1)\le C_{i2}(\delta_2)$.

The completion times of J_k on M_1 in δ_1 and δ_2 are, respectively, calculated as follows:

$$C_{k1}(\delta_1)=a_{k1}+(1+b_k)(a_{j1}+(1+b_j)(a_{i1}+S_1+b_i S_1)) \tag{4}$$

$$C_{k1}(\delta_2)=a_{k1}+(1+b_k)(a_{i1}+(1+b_i)(a_{j1}+S_1+b_j S_1)) \tag{5}$$

from $a_{i1}b_j \le a_{j1}b_i$ it implies that $C_{k1}(\delta_1)\le C_{k1}(\delta_2)$.

Because of establishing the relations $C_{k1}(\delta_1)\le C_{k1}(\delta_2)$ and $C_{j2}(\delta_1)\le C_{i2}(\delta_2)$, the relation $Max\left\{C_{k1}(\delta_1),C_{j2}(\delta_1)\right\}\le Max\left\{C_{k1}(\delta_2),C_{i2}(\delta_2)\right\}$ is satisfied; so, δ_1 dominates δ_2.

Notably, the other properties and proofs of them are omitted since they are similar to that of property 1.

3. 3. Lower Bounds

The efficiency of the branch and bound algorithm largely depends on the lower bounds. In this subsection, we establish five lower according to M_1 and M_2. Suppose that δ denotes a partial sequence including k scheduled jobs and δ' is the set of r remaining unscheduled jobs so that $k+r=n$. In addition, S_1 and S_2 are the completion time of the last job in δ on M_1 and M_2, respectively. In each node of searching tree, the objective function of partial sequence δ is $C_{\max}(\delta)$ and its lower bound is shown by LB*. In order to calculate LB*, the following theorems are utilized.

The well-known Johnson's rule (JR) give the optimal solution for the classical two-machine makespan scheduling problem where deterioration is not taken into consideration. If we consider deterioration value at the time S_1 on M_1 and $Max\left\{S_2,S_1+Min\left\{P'_{i1}\right\}\right\}$ on M_2 for each job in the set δ', then the processing time of the jobs in the set δ' will be constant. Therefore, the following theorem can be incorporated.

Theorem 1. In a partial sequence δ, the lower bound (LB_1) is obtained by using JR algorithm if the processing time of J_i on M_j in the set δ' is calculated by:

$$P'_{i1}=a_{i1}+b_i S_1 \tag{6}$$

$$P'_{i2}=a_{i2}+b_i Max\left\{S_2,S_1+Min\left\{P'_{i1}\right\}\right\} \tag{7}$$

Proof:
It is obvious that the processing time of jobs with the assumption of deterioration at times S_1 and $Max\left\{S_2,S_1+Min\left\{P'_{i1}\right\}\right\}$ on M_1 and M_2 is not more than the state of real deterioration. Since JR optimizes makespan in the classical two machine flow shop scheduling problem, so with relaxation of the real deterioration and utilization of deterioration value at times S_1 and $Max\left\{S_2,S_1+Min\left\{P'_{i1}\right\}\right\}$ for all the jobs in the set δ' , the makespan of each complete sequence will not be less than LB_1.

Browne and Yechiali [2] demonstrated in their research that the scheduling problem under deteriorating jobs on single machine to minimize makespan can be optimized via the created sequence based on the non-decreasing order of a_i/b_i ratio. So, if jobs in the set δ' are scheduled from the time S_1 using this rule on M_1 and its makespan as $C_{\max}(a_{i1}/b_i)$ is added to the least processing time of jobs in the set δ' on M_2, then theorem 2 can be defined as follows.

Theorem 2. In a partial sequence δ, the lower bound is calculated by the following relation.

$$LB_2=C_{\max}(a_{i1}/b_i)+Min\left\{a_{i2}+b_i C_{\max}(a_{i1}/b_i)\right\} \tag{8}$$

where $C_{\max}(a_{i1}/b_i)$ is the makespan of the resulting sequence based on increasing ratio of a_{i1}/b_i on M_1.

Proof:
As previously discussed, $C_{\max}(a_{i1}/b_i)$ is the least possible makespan on M_1. Now if the least processing time of jobs in the set δ' is calculated on M_2 at time $C_{\max}(a_{i1}/b_i)$, then by adding it to $C_{\max}(a_{i1}/b_i)$ the lower bound will be derived.

In Theorem 2, if jobs in the set δ' are scheduled from the time $S_{[k+1]2}=Max\left\{S_1+Min\left\{a_{i1}+b_i S_1\right\},S_2\right\}$ on M_2 according to non-decreasing of a_{i2}/b_i, and $C_{\max}(a_{i2}/b_i)$ be its makespan, then the theorem can be defined as follows.

Theorem 3. In a partial sequence δ, the lower bound of the problem is calculated by:

$$LB_3=C_{\max}(a_{i2}/b_i) \tag{9}$$

where $C_{\max}(a_{i2}/b_i)$ is obtained according to arrangement jobs in the set δ' based on increasing ratio of a_{i2}/b_i and schedule them from the least start time of position k+1 on M_2 i.e., $Max\left\{S_1+Min\left\{a_{i1}+b_i S_1\right\},S_2\right\}$.

Proof:
The start time of position k+1 on M_2 is as follows:
$S_{[k+1]2}=Max\left\{C_{[k+1]1},C_{[k]2}\right\}=Max\left\{C_{[k+1]1},S_2\right\}=$
$Max\left\{S_1+a_{[k+1]1}+b_{[k+1]}S_1,S_2\right\}\ge Max\left\{S_1+Min\left\{a_{i1}+b_i S_1\right\},S_2\right\}$
If we schedule the resulting sequence of increasing ratio of a_{i2}/b_i on M_2 from the start time $Max\left\{S_1+Min\left\{a_{i1}+b_i S_1\right\},S_2\right\}$, then will obtain $C_{\max}(a_{i2}/b_i)$.

Because of optimally of the created sequence based on the non-decreasing order of a_i/b_i ratio on single machine and scheduling it from the least start time on M_2, hence the amount of objective function of any complete sequence will not be less than LB_3.

If we calculate the processing time of jobs in the set δ' by considering the deterioration value at the time S_1 on M_1, then T_1 will be obtained as the makespan of M_1. Therefore, the following theorem can be used.

Theorem 4. In a partial sequence δ, the lower bound is calculated by:

$$LB_4 = T_1 + Min\left\{a_{i2} + b_i T_1\right\} \tag{10}$$

where

$$T_1 = S_1 + \sum_{i \in \delta'} (a_{i1} + b_i S_1) \tag{11}$$

and $Min\{a_{i2} + b_i T_1\}$ is the least processing time of the jobs of the set δ' at the time T on M_2.

Proof:
Based on Equation (1), the actual processing time of J_i on M_j is $P_{ij} = a_{ij} + b_i S_{ij}$. With assumption of deterioration amount at the point S_1 for each J_i from δ', the following relation is valid:

$P_{i1} \geq a_{i1} + b_i S_1$

therefore, $C_{[n]1} = S_1 + \sum_{i \in \delta'} P_{i1} \geq S_1 + \sum_{i \in \delta'} (a_{i1} + b_i S_1) = T_1$

from $S_{[n]2} = Max\left\{C_{[n]1}, C_{[n-1]2}\right\}$, we have

$$P_{[n]2} = a_{[n]2} + b_{[n]} S_{[n]2} = a_{[n]2} + b_{[n]} Max\left\{C_{[n]1}, C_{[n-1]2}\right\}$$
$$\geq a_{[n]2} + b_{[n]} C_{[n]1} \geq a_{[n]2} + b_{[n]} T_1$$

According to the definition of makespan, we have:

$$C_{\max} = C_{[n]2} = P_{[n]2} + Max\left\{C_{[n]1}, C_{[n-1]2}\right\} \geq P_{[n]2} + C_{[n]1} \geq P_{[n]2} + T_1$$
$$\geq Min\left\{P_{[n]2}\right\} + T_1 \geq Min\left\{a_{i2} + b_i T_1\right\} + T_1 = LB_4$$

If we consider the processing time of jobs of the set δ' at the time $Max\left\{S_1 + Min\{a_{i1} + b_i S_1\}, S_2\right\}$ on M_2, then the following theorem can be used.

Theorem 5. In a partial sequence δ, the lower bound is calculated by following relation:

$$LB_5 = T_2 + \sum_{i \in \delta'} (a_{i2} + b_i T_2) \tag{12}$$

Where

$$T_2 = Max\left\{S_1 + Min\left\{a_{i1} + b_i S_1\right\}, S_2\right\} \tag{13}$$

Proof:
As in theorem 3 showed, the least start time of position k+1 on M_2 is $Max\left\{S_1 + Min\{a_{i1} + b_i S_1\}, S_2\right\}$. Similarly based on Equation (1), the actual processing time of each J_i from δ' on M_2 is:

$$P_{i2} \geq a_{i2} + b_i S_{[k+1]2} \geq a_{i2} + b_i Max\left\{S_1 + Min\left\{a_{i1} + b_i S_1\right\}, S_2\right\}$$
$$= a_{i2} + b_i T_2$$

According to the definition of makespan, we have:

$$C_{\max} = C_{[n]2} \geq S_{[k+1]2} + \sum_{i \in \delta'} P_{i2} \geq Max\left\{S_1 + Min\left\{a_{i1} + b_i S_1\right\}, S_2\right\} + \sum_{i \in \delta'} P_{i2}$$
$$\geq T_2 + \sum_{i \in \delta'} (a_{i2} + b_i T_2) = LB_5$$

In order to make the lower bound tighter, we choose the maximum value of the lower bound that is,

$$LB^* = Max\left\{LB_1, LB_2, LB_3, LB_4, LB_5\right\} \tag{14}$$

4. HEURISTIC ALGORITHM

An alternative approach to the NP-hard problem is to provide a heuristic algorithm although they do not necessarily present the optimal solution. Thus, several easy heuristic methods are proposed for finding a near-optimal solution to the problem described above.

JR gives the optimal solution for the classical two machine makespan scheduling problem where deterioration is not taken into consideration. Though JR fails to provide the optimal answer for the proposed problem, we use it as the first heuristic algorithm. Since the deterioration rates of jobs are unequal, the lowest deterioration rate rule (LDR) and the highest deterioration rate rule (HDR) may be considered as the second and third heuristic methods, respectively. Also, the sequences formed based on the non-decreasing rates of $\frac{a_{i1}}{b_i}$, $\frac{a_{i2}}{b_i}$ and $\frac{a_{i1} + a_{i2}}{b_i}$ are proposed as the fourth, fifth and sixth heuristic methods denoted as Ratio$_1$, Ratio$_2$ and Ratio$_{1+2}$, respectively.

The main idea in the seventh one is to lessen the idle times in M_2; therefore, the jobs are arranged based on the shortest normal processing times on M_1 (SNPT$_1$). The eighth method focuses on reducing the queuing time of jobs in the queue of M_2 (SNPT$_2$). By applying SNPT to sum of the normal processing times on M_1 and M_2, the ninth heuristic approach (SNPT$_{1+2}$) is formed.

Moreover, the efficiency of all the mentioned heuristic rules is increased using a pairwise interchange (PI) method. The procedures of the heuristic rules are demonstrated as follows:

Algorithm *JR*

Step 1. Set k=1, l=n, and $N = \{J_1, J_2, ..., J_n\}$.

Step 2. Choose J_i with the shortest normal processing time from N and name it as a_{min}.

Step 3. If a_{min} is on M_1, then schedule J_i in the k^{th} position. Omit J_i from N, set k=k+1, and go to **Step 4**. Otherwise, schedule J_i in the l^{th} position. Delete J_i from N, set l=l-1, and go to **Step 4**.

Step 4. If N is not empty, go to **Step 2**. Otherwise, stop.

Algorithm LDR

Step 1. Set k=1 and $N = \{J_1, J_2, ..., J_n\}$.

Step 2. Choose J_i with the lowest b_i from N and schedule it in the k^{th} position. Delete J_i from N.

Step 3. If $k<n$, set $k=k+1$ and go to **Step 2**. Otherwise, stop.

Algorithm HDR

To creat algorithm HDR, we should replace lowest b_i by highest b_i in step 2 in algorithm LDR.

Algorithm Ratio$_1$

Step 1. Set k=1 and $N=\{J_1,J_2,...,J_n\}$.

Step 2. Choose J_i with the highest $\frac{a_{i1}}{b_i}$ from N and schedule it in the k^{th} position. Delete J_i from N.

Step 3. If $k<n$, set $k=k+1$ and go to **Step 2**. Otherwise, stop.

Algorithm Ratio$_2$

To creat this algorithm, we should replace $\frac{a_{i1}}{b_i}$ by $\frac{a_{i2}}{b_i}$ in step 2 in algorithm Ratio$_1$.

Algorithm Ratio$_{1+2}$

To creat this algorithm, we should replace $\frac{a_{i1}}{b_i}$ by $\frac{a_{i1}+a_{i2}}{b_i}$ in step 2 in algorithm Ratio$_1$.

Algorithm SNPT$_1$

Step 1. Set k=1 and $N=\{J_1,J_2,...,J_n\}$.

Step 2. Choose J_i with the smallest a_{i1} from N and schedule it in the k^{th} position. Delete J_i from N.

Step 3. If $k<n$, set $k=k+1$ and go to **Step 2**. Otherwise, stop.

Algorithm SNPT$_2$

To creat this algorithm, we should replace a_{i1} by a_{i2} in step 2 in algorithm SNPT$_1$.

Algorithm SNPT$_{1+2}$

To creat this algorithm, we should replace a_{i1} by $a_{i1}+a_{i2}$ in step 2 in algorithm SNPT$_1$.

Algorithm PI

Step 1. Implement one of the above algorithms to find an initial sequence S_0. Set k=1 and l=1.

Step 2. If $l<n$, set $k=k+1$, Otherwise, stop.

Step 3. Create a new sequence S_1 by interchanging jobs in the l^{th} and k^{th} positions in S_0. Replace S_0 by S_1 if the makespan of S_1 is smaller than that of S_0.

Step 4. If $k<n$, set $k=k+1$ and go to **Step 3**. Otherwise, $l=l+1$ and go to **Step 2**.

5. COMPUTATIONAL EXPERIMENTS

In this section, in order to assess the performance of the B&B algorithm and the heuristic algorithms, computational experiments are considered. All the algorithms are coded in *C++* and run on a Pentium 4 PC with 2.53 GHz CPU and a RAM size of 3G. The parameters are generated according to related studies [9, 11, 23] in the literature. The normal processing time of J_i on M_j is randomly generated [9] from a discrete uniform distribution with interval (0, 10].

The computational analyses consist of two experiments. The first experiment is designed in order to study the effect of deterioration rate on the performance of proposed algorithms; the results are summarized in Table 1. The job size [9] was fixed at n=11 and the value of deterioration rate b was segmented to ten groups (small to large) and have been generated randomly from continuous uniform distributions over (0, 0.1], (0.1, 0.2], (0.2, 0.3], (0.3, 0.4], (0.4, 0.5], (0.5, 0.6], (0.6, 0.7], (0.7, 0.8], (0.8, 0.9] and (0.9, 1).

Evaluating the efficiency of the B&B algorithm and the performance of the best heuristic algorithm are as the goals of the second experiment; the results are given in Table 2. In this experiment, the algorithms are tested with nine different job sizes $n = 8,10,12,14,16,18,20,22\, and\ 24$ and the deterioration rates are generated [11] from a continuous uniform distribution from interval (0, 1).

For any condition, 20 replications were randomly generated; since, 10 experimental conditions were examined in the first experiment and 9 conditions in the second one; therefore, 380 problems (i.e., 19×20) are generated, totally. In the B&B method, we dedicated 4000 seconds as time constraint to each problem and if a problem cannot get the optimal solution in this constraint, then the solution procedure will stop for that.

From Table 1 and Figure 1, the number of nodes of B&B algorithm is increased when the deterioration rate is increased especially from small to medium group. This trend is also observed from the mean and maximum CPU time according to Figure 2. The main reason is that the lower bounds become very tight when the deterioration rate is low or high; though, the trend of increment becomes less obvious when the deterioration rate is higher. This is due to the easier increment of the completion time of jobs according to the lower bounds.

The performance of the heuristic algorithms is presented in Table 1 by a number of the best solution and average error percentage that algorithm *PI* is utilized in all of them. The error percentage of the solution produced by a heuristic algorithm is calculated as follows:

$$\%\text{Error} = \frac{(Z-Z^*)}{Z^*}\times 100\ \%$$

where Z is the makespan of the solution generated by the heuristic method, and Z^* is the makespan of the optimal schedule.

Noteworthily, some heuristic algorithms (e.g., Algorithm LDR and JR) are not shown in Table 1 due to a high error or lack of a better solution than the others. The results demonstrate that among the heuristic algorithms, the performances of Ratio$_{1+2}$ and Ratio$_1$ with PI are the best in terms of the number of best solutions and the average error percentage. Although JR provides the optimal solution for the traditional two machine flow shop, it has the worst performance among the algorithms in Table 1. As shown also, the average error percentages are increased as the deterioration rates are increased.

For the B&B algorithm, the average CPU time, the number of optimal solutions and the efficiency are reported in Table 2. The efficiency of B&B algorithm is calculated by comparing the number of nodes explored to the total number of nodes. As previously mentioned, if a problem could not get the optimal solution in 4000 seconds, then B&B procedure will be stopped and its result will be removed. From Table 2**,** it is found that the B&B algorithm is able to solve most problems of job sizes less than or equal 26 in a reasonable amount of time. The average CPU time is dramatically increased as the job sizes are increased since it is an NP-hard problem; however, this increment is not related to job sizes 24 and 26, because the B&B algorithm cannot solve all the problems in these job sizes in 4000 seconds. Number of problems solved by B&B in 4000 seconds was shown as a number of optimum samples in Table 2, for example, the B&B algorithm can solve 16 and 12 problems (from 20 problems) with job sizes 24 and 26, respectively.
The efficiency of the B&B algorithm increases as the job size is increased as presented in Table 2. This index is calculated according to following relation.

$$efficiency = 1 - \frac{number\ of\ \ considered\ nodes}{(n!)} \qquad (15)$$

In this index, the number of considered nodes is compared with the total number of nodes (n!). Based on Figure 3, the index increase as the job size increase and it even reaches 1.

It means that the performance of the B&B algorithm increase. The same trend is also observed from the average percentage of entire fathomed nodes, in which it is calculated based on the sum of the average percentage of fathomed nodes by each of the dominance rules, lower bounds, lemmas, etc. These indexes prove a fantastic performance of the proposed B&B method.

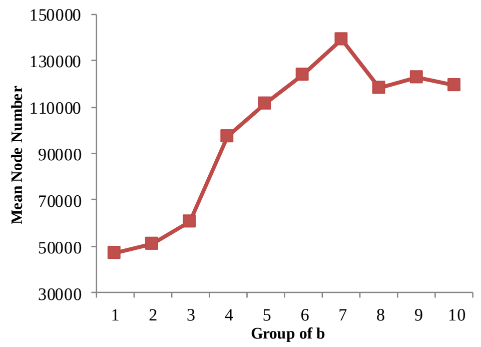

Figure 1. Mean node number of the B&B algorithm for n=11 in the first experiment

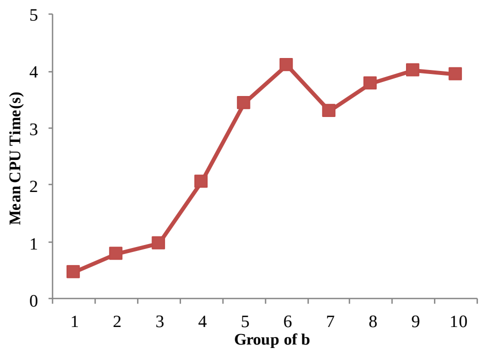

Figure 2. Mean CPU time of the B&B algorithm for n=11 in the first experiment.

TABLE 1. Computational results for the B&B and heuristic algorithms for n=11

Group of b	Number of best solution (with PI)			Average error percentage of heuristic (with PI)			CPU time (s)		Node number	
	Ratio$_1$	HDR	Ratio$_{1+2}$	Ratio$_1$	HDR	Ratio$_{1+2}$	Mean	Max	Mean	Max
1	11	2	7	0.87	1.03	0.92	0.46	0.93	46728.88	75381
2	13	0	7	0.67	0.92	0.88	0.78	1.38	51039.24	120840
3	9	2	9	1.02	1.63	1.45	0.97	1.84	60422.60	202500
4	8	4	8	0.95	2.77	0.98	2.05	4.66	97571.91	253290
5	6	2	12	1.20	2.05	1.55	3.45	7.22	111480.01	295381
6	10	3	7	1.43	1.35	1.01	4.12	6.71	123748.72	287810
7	7	0	13	0.81	2.48	1.23	3.30	8.19	139058.63	300180
8	14	2	4	1.36	1.81	1.69	3.78	9.06	118408.19	303471
9	7	2	11	1.51	1.93	0.75	4.02	7.32	122730.87	293001
10	9	0	11	0.97	1.94	1.44	3.94	8.14	119320.92	285583

TABLE 2. Computational results for the B&B and heuristic algorithms for various *n* values.

n	Number of best solution (with PI)			Average error percentage of best heuristic	number of optimum samples		average CPU time of B&B (s)	Efficiency of B&B	Average percentage of entire fathomed nodes
	Ratio$_1$	*HDR*	*Ratio*$_{1+2}$		B&B	H			
8	8	3	9	0.33	20	5	0.02	9.93E-01	81.67
10	12	1	8	0.49	20	7	0.13	9.96E-01	75.11
12	6	3	11	1.09	20	7	3.42	9.99E-01	79.28
14	9	3	8	0.70	20	4	22.88	9.99E-01	82.01
16	8	0	12	1.37	20	5	155.76	1.00E+00	91.74
18	11	4	5	0.82	20	2	638.64	1.00E+00	93.85
20	9	1	10	0.96	20	4	1439.40	1.00E+00	87.15
22	10	1	9	1.14	20	1	2181.32	1.00E+00	93.06
24	11	1	8	1.06	16	2	1552.85	1.00E+00	86.44
26	9	4	7	1.26	12	2	1686.19	1.00E+00	94.07

In addition, for the heuristic algorithms, number of optimum solutions, number of the best solutions and the average error percentage of the best heuristic algorithm are noted in Table 2; algorithm PI is utilized for all of them.

The number of optimum solutions indicates that the heuristic algorithms are able to optimally solve some problems showing the difficulty and complexity of the studied problem in this paper. The number of the best solutions shows that the heuristic algorithms Ratio$_{1+2}$ and Ratio$_1$ with PI are as the best. Moreover, the average error percentage of the best heuristic algorithm (which is mostly Ratio$_{1+2}$ and Ratio$_1$ with PI) is less than 1.37% as illustrated in Figure 4. Thus, they are recommended to obtain a near-optimal solution for the described problem. As previously discussed, JR has the worst performance among the algorithms.

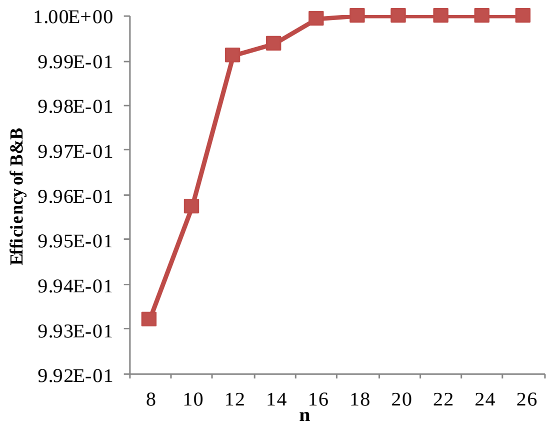

Figure 3. Average error percentage of the best heuristic algorithm for various *n*

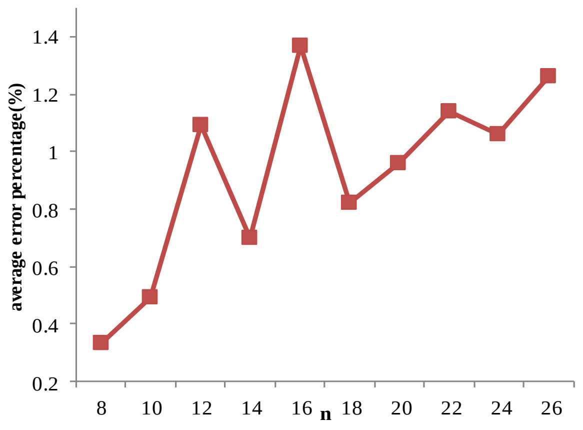

Figure 4. Average error percentage of the best heuristic algorithm for various *n*

6. CONCLUSION

In this paper, a two-machine flow shop scheduling problem with start time-dependent jobs was considered that the actual processing time of each job was calculated based on a linear function of its start time. The objective was to minimize the makespan. A B&B algorithm was proposed to solve the problem optimally. Several easy heuristic methods were also presented to obtain the near optimal solutions. The experimental results showed a high performance of the proposed B&B algorithm as it could solve most of the problems in a reasonable time when the job sizes were less than or equal to 26. Furthermore, it was shown that the average error percentage of heuristic approaches was less than 1.37%. The future studies may be focused on the cases with multiple machines or the other objective functions. Also, a deterioration function can be considered in various types and forms, such as piecewise linear and the exponential. Other interesting assumptions in this problem can be the addition of a release time or machine availability constraint, which are applicable in many real-word problems.

7. REFERENCES

1. Jafari, A. and Moslehi, G., "Scheduling linear deteriorating jobs to minimize the number of tardy jobs", ***Journal of Global Optimization***, Vol. 54, No. 2, (2012), 389-404.
2. Browne, S. and Yechiali, U., "Scheduling deteriorating jobs on a single processor", ***Operations Research***, Vol. 38, No. 3, (1990), 495-498.
3. Hsu, Y. and Lin, B., "Minimization of maximum lateness under linear deterioration", ***Omega***, Vol. 31, No. 6, (2003), 459-469.
4. Bahalke, U., Yolmeh, A., Hajizade, A. and Bahalke, A., "Genetic and tabu search algorithms for the single machine scheduling problem with sequence-dependent set-up times and deteriorating jobs", ***International Journal of Engineering-Transactions A: Basics***, Vol. 23, No. 3&4, (2010), 227-233.
5. Hamta, N., Ghomi, S., Bahalke, U. and Golpaigani, H., "Single machine scheduling problem with precedence constraints and deteriorating jobs", ***International Journal of Engineering-Transactions A: Basics***, Vol. 24, No. 2, (2011), 115-121.
6. Lee, W.-C., Yeh, W.-C. and Chung, Y.-H., "Total tardiness minimization in permutation flowshop with deterioration consideration", ***Applied Mathematical Modelling***, Vol. 38, No. 13, (2014), 3081-3092.
7. Lee, W.-C., Wu, C.-C. and Chung, Y.-H., "Scheduling deteriorating jobs on a single machine with release times", ***Computers & Industrial Engineering***, Vol. 54, No. 3, (2008), 441-452.
8. Wang, J.-B. and Wang, M.-Z., "Minimizing makespan in three-machine flow shops with deteriorating jobs", ***Computers & Operations Research***, Vol. 40, No. 2, (2013), 547-557.
9. Lee, W.-C., Wu, C.-C., Wen, C.-C. and Chung, Y.-H., "A two-machine flowshop makespan scheduling problem with deteriorating jobs", ***Computers & Industrial Engineering***, Vol. 54, No. 4, (2008), 737-749.
10. Ng, C., Wang, J.-B., Cheng, T.E. and Liu, L., "A branch-and-bound algorithm for solving a two-machine flow shop problem with deteriorating jobs", ***Computers & Operations Research***, Vol. 37, No. 1, (2010), 83-90.
11. Mosheiov, G., Sarig, A. and Sidney, J., "The browne–yechiali single-machine sequence is optimal for flow-shops", ***Computers & Operations Research***, Vol. 37, No. 11, (2010), 1965-1967.
12. Mosheiov, G., "Scheduling jobs under simple linear deterioration", ***Computers & Operations Research***, Vol. 21, No. 6, (1994), 653-659.
13. Chung, Y.-H., Liu, H.-C., Wu, C.-C. and Lee, W.-C., "A deteriorating jobs problem with quadratic function of job lateness", ***Computers & Industrial Engineering***, Vol. 57, No. 4, (2009), 1182-1186.
14. Wang, L., Sun, L.-Y., Sun, L.-H. and Wang, J.-B., "On three-machine flow shop scheduling with deteriorating jobs", ***International Journal of Production Economics***, Vol. 125, No. 1, (2010), 185-189.
15. Wang, J.-B., Ng, C.D., Cheng, T.E. and Liu, L.-L., "Minimizing total completion time in a two-machine flow shop with deteriorating jobs", ***Applied Mathematics and Computation***, Vol. 180, No. 1, (2006), 185-193.
16. Yang, S.-H. and Wang, J.-B., "Minimizing total weighted completion time in a two-machine flow shop scheduling under simple linear deterioration", ***Applied Mathematics and Computation***, Vol. 217, No. 9, (2011), 4819-4826.
17. Lee, W.-C., Lin, J.-B. and Shiau, Y.-R., "Deteriorating job scheduling to minimize the number of late jobs with setup times", ***Computers & Industrial Engineering***, Vol. 61, No. 3, (2011), 782-787.
18. Cheng, T.E., Hsu, C.-J., Huang, Y.-C. and Lee, W.-C., "Single-machine scheduling with deteriorating jobs and setup times to minimize the maximum tardiness", ***Computers & Operations Research***, Vol. 38, No. 12, (2011), 1760-1765.
19. Lee, W.-C. and Lu, Z.-S., "Group scheduling with deteriorating jobs to minimize the total weighted number of late jobs", ***Applied Mathematics and Computation***, Vol. 218, No. 17, (2012), 8750-8757.
20. Amirian, H. and Sahraeian, R., "Multi-objective differential evolution for the flow shop scheduling problem with a modified learning effect", ***International Journal of Engineering-Transactions C: Aspects***, Vol. 27, No. 9, (2014), 1395-1404.
21. Ghafari, E. and Sahraeian, R., "Appling metaheuristic algorithms on a two stage hybrid flowshop scheduling problem with serial batching (research note)", ***International Journal of Engineering-Transactions C: Aspects***, Vol. 27, No. 6, (2013), 899-910.
22. Hassanpour, S.T., Naseri, M.A. and Nahavandi, N., "Solving re-entrant no-wait flow shop scheduling problem", ***International Journal of Engineering-Transactions C: Aspects***, Vol. 28, No. 6, (2015), 903-912.
23. Moslehi, G. and Jafari, A., "Minimizing the number of tardy jobs under piecewise-linear deterioration", ***Computers & Industrial Engineering***, Vol. 59, No. 4, (2010), 573-584.

Minimizing Makespan with Start Time-Dependent Jobs in a Two-Machine Flow Shop

A. A. Jafari[a], H. Khademi Zare[a], M. M. Lotfi[a], R. Tavakkoli-Moghaddam[b]

[a] *Department of Industrial Engineering, Faculty of Engineering, Yazd University, Yazd, Iran*
[b] *School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran*

PAPER INFO

Paper history:
Received 17 February 2016
Received in revised form 03 May 2016
Accepted 03 May 2016

Keywords:
Linear Deteriorating Jobs
Start Time-Dependent
Flow Shop
Makespan
Branch and Bound
Heuristics

چکیده

هدف این مقاله بررسی مسئله زمان‌بندی مجموعه‌ای از کارهای وابسته به زمان شروع در یک محیط فلوشاپ دو ماشین می‌باشد که زمان پردازش واقعی کارها بر اساس زمان شروع آن‌ها به صورت خطی افزایش می‌یابد. هدف این مسئله حداقل کردن دامنه عملیات است. مسئله از نوع NP-hard می‌باشد بنابراین هیچ الگوریتم با زمان چندجمله‌ای برای حل آن وجود ندارد. در نتیجه یک الگوریتم شاخه‌وکران با در نظر گرفتن اصول غلبه، حدود پایین و بالا جهت به دست آوردن جواب بهینه ارائه شده است. همچنین چندین الگوریتم ابتکاری به منظور یافتن جواب‌های نزدیک به بهینه ارائه شده است. جهت ارزیابی الگوریتم‌های پیشنهاد شده، آزمایشات محاسباتی بر اساس ادبیات موضوع ارائه شده است. فعالیت‌های روبه‌زوال موجب افزایش در دامنه عملیات مسائل می‌شود؛ بنابراین به دست آوردن جواب بهینه و یا نزدیک به بهینه مهم می‌باشد. با در نظر گرفتن پیچیدگی مسئله، الگوریتم شاخه‌وکران قادر به حل مسائل با ابعاد ۲۰ فعالیت است. علاوه بر این، متوسط درصد خطای الگوریتم‌های ابتکاری کمتر از ٪۱/٦٧ است، بنابراین بهترین الگوریتم برای به دست آوردن جواب نزدیک به بهینه در مسائل با ابعاد بزرگ پیشنهاد می‌گردد.

***doi**: 10.5829/idosi.ije.2016.29.06c.07*