Scientia Iranica E (2022) 29(6), 3418–3433

SCIENTIA
IRANICA

Sharif University of Technology
Scientia Iranica
Transactions E: Industrial Engineering
http://scientiairanica.sharif.edu

A novel bi-objective model for a job shop scheduling problem with consideration of fuzzy parameters, modified learning effects, and multiple preventive maintenance activities

S.H. Mousavipour, H. Farughi*, and F. Ahmadizar

Department of Industrial Engineering, University of Kurdistan, Pasdaran Boulevard, Sanandaj, Iran.

Received 12 October 2019; received in revised form 6 October 2020; accepted 18 January 2021

KEYWORDS
Job shop scheduling problem;
Learning effects;
Availability constraints;
Transportation times.

Abstract. This paper aims to introduce a novel bi-objective model for a Job Shop Scheduling Problem (JSSP) in order to minimize makespan and maximum tardiness simultaneously. Some realistic assumptions namely fuzzy processing times and due dates involving triangular possibility distributions, transportation times, availability constraints, modified position-based learning effects on processing times, and sum-of-processing-times based learning effects on duration of maintenance activities were considered to provide a more general and practical model for the JSSP. Based on the learning effects, processing times decrease as the machine performs an operation frequently and workers gain working skill and experience. In this paper, based on DeJong's learning effect, a novel and modified formulation is proposed for this effect. According to the above-mentioned assumptions, a novel Mixed-Integer Linear Programming (MILP) model for the JSSP is suggested. The proposed model is first converted into an auxiliary crisp model, given that the model is a possibilistic programming and is then solved by TH and ε-constraint methods in the case of small-sized instances. Finally, the results are compared. For medium- and large-sized instances, five metaheuristic algorithms including Non-dominated Sorting Genetic Algorithm (NSGA-III), Pareto Envelope-based Selection Algorithm (PESA-II), Strength Pareto Evolutionary Algorithm (SPEA-II), NSGA-II, and Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) are utilized, and the results are finally compared in terms of three performance metrics.

1. Introduction

Owning to the theoretical challenges and many industrial applications, the Job Shop Scheduling Problem (JSSP) is undoubtedly worth further research and listed among the most important issues in the production planning. Job shop scheduling is significant as it specifies process maps and capabilities for most industries including painting, chemical, pharmaceutical, textile, and automobile manufacturing. The present paper seeks to define a comprehensive version of JSSP covering some of the practical assumptions including fuzzy processing times and fuzzy due dates, predetermined multiple preventive maintenance activities, learning effects on maintenance activities as well as on processing times, and job-dependent transportation times with the objectives of simultaneous minimization

*. *Corresponding author. Tel.: +98 8733660073*
E-mail addresses: hamedmousavi942@gmail.com (S.H. Mousavipour); h.farughi@uok.ac.ir (H. Farughi); f.ahmadizar@uok.ac.ir (F. Ahmadizar)

doi: 10.24200/sci.2021.54614.3839

of makespan and maximum tardiness. Since the model is original and only few benchmarks associated with it exist, instances of three different sizes are generated randomly. For solving small-sized instances to obtain optimal Pareto solutions, we utilize the interactive method of Fuzzy Multi-Objective Decision-Making (FMODM), called TH method, which was proposed by Torabi and Hassini [1]. This method managed to obtain an equivalent auxiliary single-objective crisp model, which was then optimally solved by GAMS. In addition, small instances are optimally solved as well using the ε-constraint method. Given the complexity of the problem, five metaheuristics namely Pareto Envelope-based Selection Algorithm (PESA-II), Strength Pareto Evolutionary Algorithm (SPEA-II), Non-dominated Sorting Genetic Algorithm (NSGA-II and NSGA-III), and Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) are applied to tackling large- and medium-sized instances. Three metrics including Space Metric (SM), Mean Ideal point Distance (MID), and Quality Measure (QM) are used to evaluate the Pareto solutions of the algorithms. It is worth mentioning that despite the numerous works performed on job shop scheduling, only two studies have considered classic learning effects on JSSP. This study provides a modified formulation of learning effects on this problem. Furthermore, to the best of our knowledge, no study has investigated learning effects on maintenance activities in scheduling problems; in addition, no JSSP-related study has taken all the mentioned assumptions into account.

2. Literature review

As a well-known scheduling problem, JSSP was introduced by Muth and Thompson [2]. In standard job shop systems, there must be m machines to process n jobs, each of which is divided into different operations that are processed on a specific set of machines based on a distinct routine. At most, one job can be processed on each machine over a fixed period of time. Most researchers have assumed job processing times in classical scheduling problems to be constant and known throughout the planning horizon, while, in the vast majority of real situations, they are reduced as a machine operates frequently, or workers gain working skills and experiences. The above phenomenon is called "learning effects". The notion was first introduced to scheduling problems by Biskup [3]. In a comprehensive review, Biskup [4] classified learning effects into two main types. In the first type, achievement of learning depends on how many jobs are processed; these are referred to as position-based learning effects. This kind of learning occurs in semi-automatic or fully-automatic operations. In the second type, known as sum-of-processing-time based learning effects, processing time is considered for already-processed jobs, unlike in position-based learning effects. If man is a significant part of job processing, this type of learning effects can be taken into consideration. In a recent review paper, Azzouz et al. [5] emphasized that despite the massive volume of research conducted on different types of scheduling environments under the phenomenon of learning, no study related to learning effects on JSSP exists. However, recently, Mousavipour et al. [6] studied a sequence-dependent setup time job shop scheduling problem with availability constraints based on classic position-based learning effects. They also provided an Mixed-Integer Linear Programming (MILP) model for JSSP based on learning effects and flexible maintenance activities [7]. Tayebi Iraqi et al. [8] considered the classical position-based learning effects on the setup times and the deterioration effects on the processing times in flexible job shop environments. They solved the problem without presenting a mathematical model in order to minimize the number of operations by introducing a hybrid algorithm. Renna [9] studied flexible job shop scheduling with learning and forgetting effects. Okolowski and Gawiejnowicz [10] introduced a novel learning curve in a parallel machine environment, called DeJong's Learning Curve, and they proposed an incompressibility index to make a distinction between the manual and automatic parts of jobs. Lai and Wu [11] proposed a new learning rate based on a combination of machine-dependent and job-dependent learning rates in a flow shop environment. They stated that different machines might exhibit different learning rates in practice. Vahedi Nouri et al. [12], Amirian and Sahraean [13], Behnamian and Zandieh [14], Gao et al. [15], and Mousavi et al. [16] addressed the learning effects in the flow shop environment.

Another widely used assumption in classical JSSP is that machines are always available. This is not the case in the real world due to failure and maintenance activities, which have been studied in the literature as availability constraints. Schmit [17] and Ma and Chu [18] conducted comprehensive reviews on scheduling with availability constraints. Two main types of deterministic availability constraints were considered. The first one is fixed availability constraints in which start time and duration of the maintenance activity are already certain and fixed. In the second constraint, start time of maintenance activity is flexible and lies in a specific time window. Machines may be unavailable because of predetermined multiple preventive maintenance activities; thus, each machine is unavailable once it has processed a given number of jobs. Hsu et al. [19] studied this form of machine unavailability in a single-machine scheduling problem. Xu et al. [20] investigated it in the two-machine flow shop scheduling problem. JSSP with consideration of predetermined preventive maintenance activities was first considered by Aggone [21] for a two-machine problem. It was solved using the B&B algorithm based on disjunctive graphs. Benttaleb et al. [22] provided two formulations for a two-machine JSSP with availability constraint on one machine. Tamssaouet et al. [23] addressed a JSSP when machines were not always available due to fixed periods of maintenance activities. They suggested a mathematical model based on disjunctive graphs. A sequence-dependent JSSP involving preventive maintenance for minimization of makespan was studied by Naderi et al. [24], who proposed different techniques for integration of production scheduling with preventive maintenance. Four metaheuristic methods were proposed by them for solving the problems. Without providing a mathematical model, they based their models on simulated annealing and genetic algorithms.

In the current study, multiple fixed periods of preventive maintenance on m machines are taken into account. The innovation of this paper in terms of availability constraints is that availability constraints have been formulated based on the position-based model of JSSP and under the phenomenon of learning caused by repetition.

To the best of our knowledge, in previous studies where production scheduling and maintenance operations were integrated, learning effects on the maintenance operations were not taken into account. Taraki et al. [25] investigated the learning effects on maintenance outsourcing with the assumption that an external contractor was responsible for performing preventive maintenance activities. They assumed that the contractor would learn through repetition and prepare how to perform the preventive maintenance operations more efficiently. In this case, the maintenance operational teams are assigned such that the learning phenomenon occurs in order to reduce operation time and, consequently, costs.

In the job shop environment, jobs are moving between machines. Given the processing time, a job may wait for processing once the operation before it is completed. Magnitude depends merely on the distance between the two consecutive machines for job-independent transportation time, whereas this factor depends on the distance and the job to be carried out when transportation time is job dependent. Depending on the transporter, there are two types of transportation system: single-transporter (limited) and multi-transporter (unlimited) [26]. A generalized job shop problem was considered by Hurink and Knust [27] in which a single transport robot must transport the jobs between the machines. The objective was to specify a schedule with minimal makespan. Nouri et al. [28] investigated a Job Shop scheduling Problem with Transportation times and Many Robots (JSPT-MR). For solving the problem, a hybrid metaheuristic approach was proposed based on clustered holonic multiagent model. The unlimited job-dependent transportation times were considered.

Furthermore, recent research studies have increasingly considered the inherent uncertainty in model parameters, and uncertainty seems to be an inseparable part of real-world problems. In this paper, the assumption is that we have data of approximant values of job processing times and due dates; consequently, fuzzy parameters are applied. A large number of papers have taken uncertainty into consideration as fuzzy parameters in different scheduling environments; references [29–32] are only a few of them.

The remainder of this paper is organized as follows. The mathematical formulation and validation of the proposed model are provided in Section 3. The applied exact and metaheuristic solution techniques are detailed in Sections 4 and 5. Sections 6 and 7 provide an evaluation and comparison of the results obtained by the algorithms. Section 8 concludes the paper.

3. Model description

In the JSSP, there are m machines that are supposed to process n jobs. Every job J_j contains L operations and has a specific predefined sequence. Each machine m_i has k positions and is capable of processing only one job at once, each of which can be processed by only one machine at once. The jobs cannot be pre-empted, that is, they must be processed without interruption. The assumptions, decision variables, and notations in our model are given below.

Assumptions

- The jobs cannot be pre-empted;
- Each machine can process a job once at most;
- At time zero, all of the jobs and all of the machines are available;
- The due dates and processing times of the jobs are assumed to be fuzzy parameters that have triangular possibility functions; moreover, the processing times are affected by DeJong's learning effect;
- R' predetermined maintenance activities must be applied to each machine, and their durations are affected by the sum-of-processing-times-based learning effect;
- The processing times include the setup times;
- Transportation times between machines are considered to be job dependent;
- There is no limitation on transporter and maintenance teams.

Notations

Indices

j Job index

i	Machine index
R	Maintenance index
k	Position index
l	Operation index

Parameters and scalars

m	Number of machines
n	Number of jobs
R'	Number of maintenance activities on each machine
V	A large positive number
P_{ijk}	Processing time of job J_j on the kth position of machine m_i
P_{ij}	Normal processing time of job J_j on machine m_i
d_j	Due date of job J_j
α_i	Learning index of job processing on machine M_i ($\alpha_i \leq 0$)
φ_j	Learning index of job J_j ($\varphi_j \leq 0$)
λ_{ij}	Learning index of job J_j on machine m_i
Pm_{iR}	Rth maintenance activity on machine m_i
t_{iR}	Normal execution time of Pm_{iR}
t^A_{iR}	Actual execution time of Pm_{iR}
y_{iRk}	A binary parameter that is 1 if the Rth maintenance activity must be done after processing the job on the kth position of machine m_i and 0 otherwise
r_{ijl}	A binary parameter that is 1 if the lth operation of the jth job is processed on machine m_i, and 0 otherwise
$tp_{iji'}$	Time needed for transfer of job J_j from machine m_i to machine $m_{i'}$

Decision variables

C_{ijk}	Completion time of job J_j if it is scheduled in the kth position of machine m_i and 0 otherwise
X_{ijk}	A binary variable that is 1 if job J_j is processed in the kth position of machine m_i and 0 otherwise
$C_{\max}$	makespan
L_j	Lateness of job $J_j = \{C_j - d_j\}$
T_j	Tardiness of job $J_j = \max\{0, L_j\}$
$T_{\max}$	Maximum tardiness of jobs

3.1. Mathematical model

Wagner's [33] mathematical model is developed here such that some practical assumptions like availability constraints, learning effects, and transportation times are taken into account. For modeling the learning effects on the processing times of jobs, given that a semi-automatic environment is assumed, an incompressibility index ($0 < M < 1$) is applied for separating the fixed machine time from the variable processing time which is affected by learning from each other based on DeJong's learning effect. Amirian and Sahraean [13] defined each index separately for every job. In this paper, the incompressibility index for each job is defined on each machine, since a certain portion of manual time may belong to each job on each machine. Furthermore, the learning effect is considered as job- and machine-dependent learning; for example, in [11], the learning parameter was obtained from the sum of the learning parameters for the machines and the learning parameters for the jobs. Accordingly, the modified learning effect on processing times is defined as follows:

$$p_{ijk} = p_{ijk}(M_{ij} + (1 - M_{ij}) \times K^{\lambda_{ij}}),$$

$$\lambda_{ij} = a_i + \varphi_j. \tag{1}$$

Here, $\alpha_i < 0$ and $\varphi_j < 0$ are the learning indices of machine m_i and job J_j, respectively, λ_{ij} denotes the job-machine-dependent learning index for job J_j on machine m_i, M_{ij} is incompressibility index of job J_j on machine m_i, and k is the position of job in the sequence.

Similar to the referenced works [12] and [13], multiple fixed preventive maintenance activities on each machine were considered such that at least one maintenance activity on each machine must be scheduled to minimize unexpected breakdowns and improve lifespan with this difference that learning effects have been considered on maintenance activities. The runtime of outsourcing preventive maintenance activities is assumed to be affected by learning effects from repetition. Since maintenance operations are usually manual ones that are subject to the operator's intervention, the learning effects are considered, as in the work of Yang and Kou [34], according to the sum of processing times; the difference is that a separate learning rate is assumed for each machine.

$$t^A_{iR} = t_{iR}\left(1 + \sum\nolimits_{R=1}^{R'-1} t^A_{iR}\right)^{\omega_i}. \tag{2}$$

Here, $\omega_i < 0$ is the learning index of maintenance operations on machine m_i. Of note, in this study, autonomous learning effects were assumed. In this type of learning effects, training has no impact on learning rate and this rate is subject to the repetition of jobs and operator's experience. Other types of learning effects called induced learning effects are affected by training. In real situations, the values of these indices are obtained experimentally via learning curves.

Furthermore, the due dates and processing times are assumed to be fuzzy numbers having the triangular possibility distribution:

$$\tilde{p}_{ij} = (\overset{p}{p_{ij}}, \overset{m}{p_{ij}}, \overset{o}{p_{ij}}), \tag{3}$$

$$\tilde{d}_j = (\overset{p}{d_j}, \overset{m}{d_j}, \overset{o}{d_j}), \tag{4}$$

where $\begin{pmatrix} p & p \\ p_{ij} & d_j \end{pmatrix}$ are the pessimistic values, $\begin{pmatrix} m & m \\ p_{ij}, & d_j \end{pmatrix}$ the probable values, and $\begin{pmatrix} o & o \\ p_{ij}, & d_j \end{pmatrix}$ the optimistic values of processing time and due date, which are determined by the decision-maker.

A novel bi-objective possibilistic mixed-integer linear programming model is presented below based on the assumptions stated above:

$$\text{Min } Z_1 = C_{\max}, \tag{5}$$

$$\text{Min } Z_2 = T_{\max}, \tag{6}$$

s.t.:

$$\sum_k X_{ijk} = 1 \qquad \forall i = 1, ..., m \ , \ j = 1, ..., n, \tag{7}$$

$$\sum_j X_{ijk} = 1 \qquad \forall i = 1, ..., m \ , \ k = 1, ..., n, \tag{8}$$

$$\sum_j C_{ijk} + \sum_j X_{ijk+1} \times \tilde{p}_{ijk+1} + \sum_{R=1}^{R'} y_{iRk} \times t^A_{iR} \leq \sum_j C_{jjk+1} \qquad \forall i = 1, ..., m \quad k = 1, ..., n-1, \tag{9}$$

$$\sum_i r_{ijl} \times C_{ijk} + \sum_{i'} r_{i'jl+1} \times X_{i'jk} \times \tilde{p}_{i'jk} + \sum_i \sum_{i'} r_{ijl} \times r_{i'jl+1} \times tp_{iji'} \leq V \left(1 - \sum_i r_{ijl} \times X_{ijk}\right) + V \times \left(1 - \sum_{i'} r_{i'jl+1} \times X_{i'jk}\right) + \sum_{i'} r_{i'jl+1} \times C_{i'Jk}$$
$$\forall j = 1, ..., n, \quad l = 1, ..., m-1, \quad k = 1, ..., n, \tag{10}$$

$$C_{ijk} \leq V \times X_{ijk} \qquad \forall i = 1, ..., m, \quad j = 1, ..., n, \quad k = 1, ...n, \tag{11}$$

$$C_j \geq \sum_k C_{[last\ machine]_{jk}} \qquad \forall j = 1, ..., n, \tag{12}$$

$$\sum_k C_{[1]jk} \geq \sum_k X_{[1]jk} \times P_{[1]jk} \qquad \forall j = 1, ..., n, \tag{13}$$

$$C_j - \tilde{d}_j \leq T_j \qquad \forall j = 1, ..., n, \tag{14}$$

$$C_{\max} \geq C_j \qquad \forall j = 1, ..., n, \tag{15}$$

$$T_{\max} \geq T_j \qquad \forall j = 1, ..., n, \tag{16}$$

$$C_j, T_j \ \geq 0 \qquad \forall j = 1, ..., n, \tag{17}$$

$$X_{ij} \in \{0, 1\} \qquad \forall i = 1, ..., m \ , \quad j = 1, ..., n. \tag{18}$$

In this model, the first equation represents the first objective function aimed at minimizing makespan. The second equation represents the second objective, consisting of minimizing maximum tardiness. It is worth mentioning that even though $T_{\max}$ and $C_{\max}$ are naturally similar, $C_{\max}$ depends on the characteristics of the last job on each machine and $T_{\max}$ may be obtained by a job other than the last job; consequently, decreasing/increasing $C_{\max}$ does not necessarily decrease/increase $T_{\max}$.

According to Constraints (7) and (8), each job may be assigned to only one position on each machine, and each position can involve only one job. The completion times of a job on different machines are calculated by Constraint (9) based on their actual processing times and the actual operation times of maintenance activities. Constraint (10) is the precedence constraint. It ensures that a job has all its operations executed in the given order and considers transportation times. Eq. (11) states the completion time of job J_j if scheduled in the kth position on machine m_i, and it is 0 otherwise. The completion times of different jobs are computed by Constraints (12) and (13). Constraints (14)–(16) calculate $C_{\max}$ and $T_{\max}$; Constrains (17) and (18) define positive and binary variables.

3.2. Validity of model

This model is an original model; consequently, we have no access to the benchmark instances. To verify this model, a Gantt chart was applied, as shown in Figure 1.

A small problem comprises two jobs and three machines, and one predetermined preventive maintenance activity for each machine is managed. For simplicity, the same learning and incompressibility indices were considered for all jobs and machines. Furthermore, only makespan is minimized. The crisp processing times of jobs affected by learning effects, sequence of jobs processing generated randomly, durations of maintenance activities, and transportation times are shown in Table 1. Learning indices, incompressibility index, and actual values of processing times are given in Table 2.

Table 1. Input data of validation sample problem.

Normal processing times:			**Sequence of jobs processing:**
	j_1	j_2	
m_1	60	85	$j_1 : (m_1 - m_2 - m_3)$
m_2	50	45	$j_2 : (m_2 - m_1 - m_3)$
m_3	30	25	

Predetermined maintenance activities:	**Duration of maintenance activities**
$y_{111} = 1$	$t_{11} = 8$
$y_{211} = 1$	$t_{21} = 10$
$y_{311} = 1$	$t_{31} = 15$

Transportation matrix		$\boldsymbol{m_1}$	$\boldsymbol{m_2}$	$\boldsymbol{m_3}$
$tp_{iji'}$	$m_1 - j_1$	0	6	10
	$m_1 - j_2$	0	4	2
	$m_2 - j_1$	5	0	8
	$m_2 - j_2$	12	0	9
	$m_3 - j_1$	11	3	0
	$m_3 - j_2$	7	10	0

Table 2. Learning indices and actual jobs processing times of validation sample problem.

Learning indices:		**Actual processing times ($\boldsymbol{p_{ijk}}$):**		
			Position 1	**Position 2**
Learning index of machine m_i	$a_i = -0.15$	$m_1 - j_1$	60	52.8
Learning index of job j_j	$\varphi_j = -0.10$	$m_1 - j_2$	85	74.8
		$m_2 - j_1$	50	44
Incompressibility index	$M_{ij} = 0.25$	$m_2 - j_2$	45	39.6
		$m_3 - j_1$	30	26.4
		$m_3 - j_2$	25	22

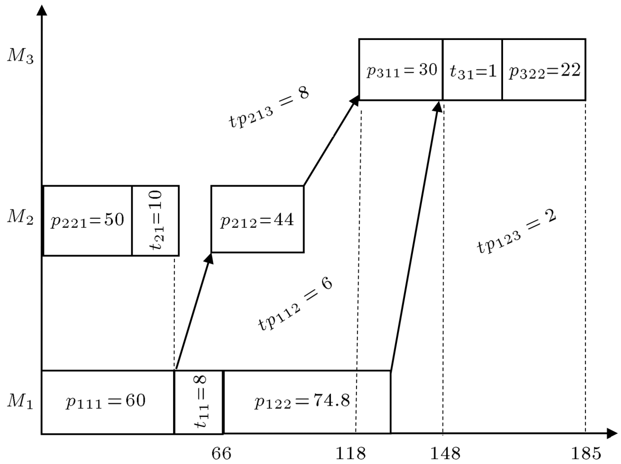


Figure 1. Gantt chart of validation sample problem.

According to the suggested model, the optimal value of makespan is 185. There are 8 feasible solutions to this problem. These solutions contain decision variables with a value of 1 and the corresponding makespan is as follows:

x_{111}, x_{122}	x_{221}, x_{212}	x_{311}, x_{322}	$C_{\max} = 185$
x_{111}, x_{122}	x_{221}, x_{212}	x_{321}, x_{312}	$C_{\max} = 209.2$
x_{121}, x_{112}	x_{211}, x_{222}	x_{321}, x_{312}	$C_{\max} = 239.2$
x_{111}, x_{122}	x_{211}, x_{222}	x_{311}, x_{322}	$C_{\max} = 274.4$
x_{121}, x_{112}	x_{211}, x_{222}	x_{311}, x_{322}	$C_{\max} = 279.8$
x_{121}, x_{112}	x_{221}, x_{212}	x_{321}, x_{312}	$C_{\max} = 294.4$
x_{111}, x_{122}	x_{211}, x_{222}	x_{321}, x_{312}	$C_{\max} = 320.8$
x_{121}, x_{112}	x_{221}, x_{212}	x_{311}, x_{322}	$C_{\max} = 335$

The solutions indicate that the optimal makespan determined by the suggested model is 185 and this value is approved by Gant chart. In consequence, Gant chart confirms the validity of the suggested model.

4. Solution methodology

To achieve the optimal Pareto front for small-sized instances, the TH and ε-constraint methods were applied. The proposed model is first converted into an auxiliary crisp model given that it is a possibilistic programming model and is then solved by the TH method. The ε-constraint method is used further to solve the crisp bi-objective model, and a comparison is made between the results. The fuzzy parameters undergo defuzzification using the weighted average method based on which those on the left sides of Constraints (9), (10), and (14) are converted into crisp numbers, and the corresponding auxiliary crisp constraints are given below:

$$\sum_j C_{ijk} + \sum_j X_{ijk+1} \times \left(W_1 P^p_{ijk+1,\beta} + W_2 P^m_{ijk+1,\beta} + W_3 P^o_{ijk+1,\beta}\right) + \sum_{R=1}^{R'} y_{iRk} \times t^A_{iR} \le \sum_j C_{ijk+1}$$

$$\forall i = 1, ..., m, \quad k = 1, ..., n-1, \tag{19}$$

$$\sum_i r_{ijl} \times C_{ijk} + \sum_i r_{ijl+1} \times X_{ijk} \times \left(W_1 P^p_{ijk,\beta} + W_2 P^m_{ijk,\beta} + W_3 P^o_{ijk,\beta}\right) + \sum_i \sum_{i'} r_{ijl} \times r_{i'jl+1} \times tP_{iji'} \le V\left(1 - \sum_i r_{ijl} \times X_{ijk}\right) + V\left(1 - \sum_{i'} r_{i'jl+1} \times X_{i'jk}\right) + \sum_{i'} r_{i'jl+1} \times C_{i'jk}$$

$$\forall j = 1, ..., n, \quad l = 1, ..., m-1, \quad k = 1, ..., n, \tag{20}$$

$$C_j - (W_1 d^p_{j.\beta} + W_2 d^m_{j.\beta} + W_3 d^o_{j.\beta}) \le T_j$$

$$\forall j = 1, ..., n. \tag{21}$$

Decision-makers usually decide on the value of β as the minimal acceptable possibility. It is worth mentioning that events with a possibility equal to or greater than β are acceptable. W_1 is the most pessimistic, W_2 the most possible, and W_3 the most optimistic fuzzy parameter. Lay and Huang [35] suggested the values of these parameters as follows:

$$\beta = 0.5, \quad W_1 = 1/6, \quad W_2 = 4/6, W_3 = 1/6. \tag{22}$$

Based on these proposed values, the expression of the auxiliary crisp Bi-Objective Mixed-Integer Linear Programming (BOMILP) is as follows:

$$\text{Min } z = [Z_1, Z_2]. \tag{23}$$

s.t.:

$$X \in F(X), \tag{24}$$

where X and $F(X)$ represent the possible continuous and binary variable solution vectors in the original model and the possible area including crisp constraints, respectively.

4.1. TH method

Based on [32], the stages of TH are listed below:

1. Developing a BOMILP for the problem by determining the triangular possibility distribution for fuzzy due times and fuzzy processing times.
2. Obtaining the auxiliary crisp BOMILP model.
3. Calculating the Positive and Negative Ideal Solutions (PIS and NIS, respectively) for each of the objective functions as follows:

$$Z_1^{PIS} = \text{Min } C_{\max} \quad \text{s.t.:} \quad X \in F(X), \tag{25}$$

$$N_1^{NIS} = \text{Max } C_{\max} \quad \text{s.t.:} \quad X \in F(X), \tag{26}$$

$$N_2^{PIS} = \text{Min } T_{\max} \quad \text{s.t.:} \quad X \in F(X), \tag{27}$$

$$N_2^{NIS} = \text{Max } T_{\max} \quad \text{s.t.:} \quad X \in F(X). \tag{28}$$

4. Assigning a linear membership function to each objective function as follows:

$$\mu_{Z_1}(X) = \begin{cases} 1 & Z_1 < Z_1^{PIS} \\ \frac{Z_1^{NIS} - Z_1}{Z_1^{NIS} - Z_1^{PIS}} & Z_1^{PIS} \le Z_1 \le Z_1^{NIS} \\ 0 & Z_1 > Z_1^{NIS} \end{cases} \tag{29}$$

$$\mu_{Z_2}(X) = \begin{cases} 1 & Z_2 < Z_2^{PIS} \\ \frac{Z_2^{NIS} - Z_2}{Z_2^{NIS} - Z_2^{PIS}} & Z_2^{PIS} \le Z_2 \le Z_2^{NIS} \\ 0 & Z_2 > Z_2^{NIS} \end{cases} \tag{30}$$

5. Obtaining the equivalent crisp single-objective model:

$$\text{Max} \quad \lambda(x) = \gamma\lambda_0 + (1-\gamma)\sum \theta_h \mu_{z_h}, \tag{31}$$

$$\text{s.t.} \quad \lambda_0 \le \mu_{z_h}(X) \qquad h = 1, 2, \tag{32}$$

$$X \in F(X), \tag{33}$$

$$\lambda_0, \gamma \in [0, 1]. \tag{34}$$

Here, $\mu_{z_h}(x)$ indicates the degree of satisfaction of the hth objective function, and $\lambda_0 = \min(\mu_{Z_h}(X))$ is its minimum satisfaction degree. h represents the importance degree of the hth objective function, determined by the decision-maker, but the important point is that $\sum_h \theta_h = 1$ and $\theta_h > 0$. The decision-maker determines the values of θ_h parameters linguistically. Υ is the compensation coefficient, which is a dynamic parameter lying at the interval [0,1].

6. Determining the compensation coefficient and the significance with the objectives and solving a single-objective model. After solving this model, we will stop if the decision-maker agrees on the solutions; otherwise, we will return to the third step changing the controllable values γ, and β.

4.2. Epsilon constraint method

To solve multi-objective problems exactly, the epsilon constraint method was applied, as well. In this method, the problem is first solved considering every objective function separately for obtaining the lower and upper bounds of the Pareto frontier. Then, one of the objective functions is considered as a constraint as follows:

$$\text{Min} \quad f(x), \tag{35}$$

s.t.:

$$f_i\,(x) \leq \varepsilon_i \qquad \text{for all} \quad i = 1, 2, ... \tag{36}$$

The first and second objective functions are considered as the main objective and a constraint, respectively. ε lies between two values of the second objective function: the one when the first objective function is optimized and the optimal one.

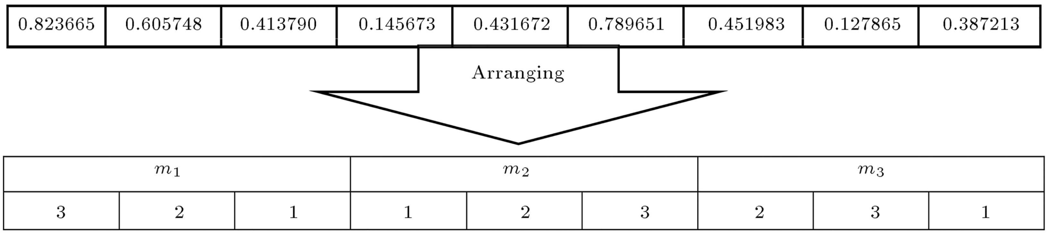
5. Metaheuristic algorithms

As mentioned before, five metaheuristic algorithms are applied to solve instances of large and medium sizes. These algorithms are briefly investigated in the present section. NSGA-II was introduced by Deb [36]. It uses an elitism-based non-dominated sorting method to rank and sort the individuals, preserving the diversity between the Pareto optimal solutions that have been obtained by applying a crowding distance approach in the section operator. Evaluation is first applied to the objective functions for each of the solutions, and different non-domination levels are formed as the entire population is then sorted. The population is sorted in ascending order in the second step of non-dominated sorting, and the lowest and highest values of each objective function are then chosen as its boundary values, which are assigned an infinite value of crowding distance. Next, the normalized difference in the value of the objective function is used as a basis for calculation of the crowding distance between any two neighboring solutions. In SPEA-II introduced by Zitzler et al. [37], an external archive including the last non-dominated solutions was updated following each generation and the value of strength was calculated for every solution. The fitness of each individual was obtained based on these computed strength values. Region-based selection was suggested in line with the proposal of Corne et al. [38], i.e., PESA-II, in which selection is made in terms of hyperboxes in the objective space that are assigned selective fitness. The method aims to reduce computational cost in Pareto ranking [39]. NSGA-III is one of the most powerful techniques among the present non-dominated solutions introduced by Deb and Jain [40] for overcoming the shortcomings of NSGA-II, including the lack of a lateral diversity-preserving operator and uniform diversity. A multi-objective evolutionary algorithm based on decomposition (MOEA/D) was proposed by Zhang and Li [41]; this method decomposes a multi-objective optimization problem into sub-problems, optimizing it at the same time.

5.1. Solution representation

JSSP is an NP-hard problem [42]. Exact methods exhibit impractical computational times for medium and large instances; therefore, it is inevitable to apply proper and efficient solution methods for solving these types of problems. A string of decimal numbers is applied to present the solution. The solution representation should represent only one solution to the problem. The string is as long as "*the number of machines*" × "*the number of jobs*". Each string is divided into machines, and each section is arranged in ascending order. In each section, order of arranged numbers determines the sequence of jobs that should be processed by the corresponding machine. Figures 2 and 3 show this procedure.

An initial solution to the problem is generated via a heuristic method which generates the first feasible



0.823665	0.605748	0.413790	0.145673	0.431672	0.789651	0.451983	0.127865	0.387213

m_1			m_2			m_3		
3	2	1	1	2	3	2	3	1

Figure 2. Solution representation.

x_{1jk}

0	0	1
0	1	0
1	0	0

x_{2jk}

1	0	0
0	1	0
0	0	1

x_{3jk}

0	1	0
0	0	1
1	0	0

Figure 3. A feasible solution.

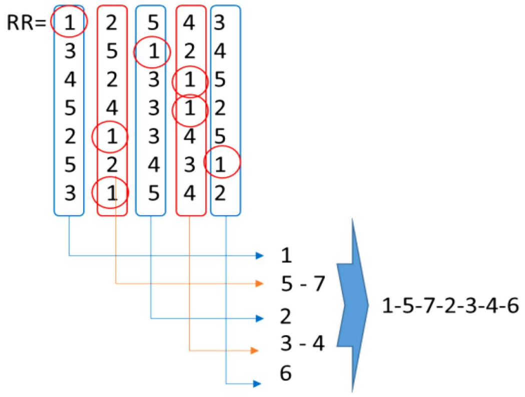


Figure 4. Generating an initial solution (sequence of jobs processed on machine m_i).

solution for each machine greedily. In this procedure, random numbers between one and number of jobs are generated for each machine; "ones" determine the initial sequence of jobs being processed on machines, as shown in Figure 4. Problem solutions should be simulated for gaining the objective function values of the problem; therefore, the value of competition time of each job is computed with consideration of effects of learning on processing times of jobs and duration of maintenance activities besides transportation times. The value of tardiness for each job is also computed with regard to the due dates of jobs.

6. Comparison criteria

To make a comparison between multi-objective algorithms, it is necessary to introduce certain criteria that can evaluate the Pareto solutions. In this paper, three criteria are used for evaluating the Pareto solutions; in these criteria, n denotes Non-Dominated Solutions (NDS) set size, and f_{ki} and f_{kj}, $I = j = 1...n$, and $k = 1, 2$ indicate the values of the kth objective function for the ith and jth solutions, respectively.

6.1. Spacing Metric (SM)

Srinivas and Deb [43] applied this type of metric, indicating the dispersion of the Pareto solutions. SM is defined as follows:

$$SM = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(d_i - \bar{d}\,\right)^2}, \tag{37}$$

$$d_i = \min_j \left\{|f_{1i} - f_{1j}| + |f_{2i} - f_{2j}|\right\},$$

$$i, j = 1, 2, ..., n, \quad i \neq j, \tag{38}$$

$$\bar{d} = \frac{\sum_{i=1}^{n} d_i}{n}. \tag{39}$$

In fact, lower values of SM are more suitable and represent lower Pareto dispersion.

6.2. Mean Ideal point Distance (MID)

The distance from the ideal point (0, 0) to the obtained Pareto solutions is indicated by MID. The definition of the metric is shown below:

$$MID = \frac{\sum_{i=1}^{n} C_i}{n}, \tag{40}$$

$$C_i = \sqrt{\sum\nolimits_{k=1}^{k} {f_{ki}}^2}. \tag{41}$$

A Pareto front with a lower value of MID is more appropriate.

6.3. Quality Measure (QM)

This criterion, proposed by Schaffer [44], considers the number of final NDS determined by each of the algorithms. Clearly, the algorithm that enjoys a greater QM value exhibits higher performance.

7. Computational results

For testing and analyzing the efficiency and validity of the proposed mathematical model, several small-, medium-, and large-sized instances are created randomly. CPLEX solver of GAMS 24.8.5 is used for solving the instances of small size exactly, and the results of the TH and ε-constraint methods are reported. However, the optimal solutions to problems of large and medium sizes cannot be found using GAMS within a logical time span; here, the MOEA-D, PESA-II, SPEA-II, NSGA-II, and NSGA-III algorithms are used. A comparison is made between the results based on the criteria to be introduced in the next sections. MATLAB 2015 and a PC with a Core i5 2.5GHz CPU, 3MB cache, and 4GB RAM were used for coding these algorithms.

7.1. Generation of mathematical instances

Due dates and processing times are assumed to be fuzzy parameters, and the symmetric triangular possibility distributions are shown below:

$$\tilde{P}_{ij} = (P_{ij} - u_{ij}\,,\, P_{ij}\,, P_{ij} + u_{ij}\,), \tag{42}$$

$$\tilde{d}_j = (d_j - u_j\,,\, d_j\,, d_j + u_j\,), \tag{43}$$

where P_{ij} and d_j are the probable due date and processing time values, and u_{ij} and u_j are the values of extension of these fuzzy numbers. Table 3 briefly shows the data generation method for random instances.

Table 3. Data generation for random instances and the TH method.

Notation	Parameter	Value
m	Number of machines	$\{3, ..., 18\}$
n	Number of jobs	$\{2, ..., 14\}$
P_{ij}	Normal processing time of job J_j on machine m_i	Uniform distribution (5, 200)
d_j	Due date of job J_j	Uniform distribution (250, 750)
R'	Number of maintenance activities performed on each machine	$\{1, ..., 6\}$
α_i	Learning index of machine m_i	Uniform distribution $(-0.9, -0.1)$
φ_j	Learning index of job j	Uniform distribution $(-0.9, -0.1)$
ω_i	Learning index of pm_i	Uniform distribution $(-0.9, -0.1)$
t_{iR}	Normal maintenance execution time	Uniform distribution (5, 50)
y_{iRk}	Maintenance position	Uniform distribution $\{1, 2, ..., n-1\}$
$tp_{iji'}$	Transportation time	Uniform distribution (5, 150)
M_{ij}	DeJong's parameter	$\{0.25,\ 0.5\}$
γ	Coefficient of compensation in the TH method	$\{0, 0.1, ..., 1\}$
θ_h	Importance level of the hth objective function in the TH method	$\theta_1 = \theta_2 = 0.5$
u_{ij}, u_j	Extension values of these fuzzy numbers	$U[4, 10]$

Table 4. Specifications of the small-sized instances.

Instance no.	Representation	No. of constrains	No. of decision variables	No. of machines	No. of jobs	No. of maintenance activities for each machine
1	3 × 2 × 1	57	34	3	2	1
2	3 × 3 × 2	125	67	3	3	2
3	3 × 4 × 2	235	112	3	4	2
4	4 × 4 × 2	326	144	4	4	2
5	3 × 5 × 2	399	169	3	5	2

Table 5. Positive and negative ideal solutions for the small-sized instances.

Instance no.	Representation	$\mathbf{Z1_{PIS}}$	$\mathbf{Z1_{NIS}}$	$\mathbf{Z2_{PIS}}$	$\mathbf{Z2_{NIS}}$
1	3 × 2 × 1	284.64	284.64	242	242
2	3 × 3 × 2	164.597	193.349	18.752	44.597
3	3 × 4 × 2	233.52	240.89	62.89	71.26
4	4 × 4 × 2	271.43	324.5	171.18	178.83
5	3 × 5 × 2	278.74	315.53	130	133.74

Each instance is designed as "$a.b.c$", in which "a" indicates the number of machines, "b" is the number of jobs, and "c" is the number of maintenance activities. Instances are divided into three groups: small-sized instances with "$a.b.c \leq 75$", medium-sized instances with "$a.b.c \leq 150$", and large-sized instances with "$a.b.c > 150$". These ranges are determined based on previous studies in the JSSP. Predetermined sequence of each job on every machine is generated randomly, as well. Of note, problems are handled by different learning effects, but for the sake of simplicity, the same learning rates are considered for all machines and all jobs in each problem.

7.2. *Comparison of the exact methods*

The results obtained for the instances of small sizes are discussed. Five types of the small-sized sample are taken into account with different jobs, machines, and maintenance activity combinations. A numerical instance is generated randomly for each sample. The specifications of each instance are presented in Table 4.

MILP model is solved separately for computation of the values of positive and negative ideal solutions (PIS and NIS, respectively) for each instance, as illustrated in Table 5.

GAMS is used for solving the TH mathematical model for each of the instances, given the positive

Table 6. Results of the TH method for Instances 2 and 3.

	$3 \times 4 \times 2$				$3 \times 3 \times 2$			
Υ	Z_1	Z_2	μ_1	μ_2	Z_1	Z_2	μ_1	μ_2
0	240.89	62.89	0	1	193.349	18.752	0	1
0.1	233.52	71.26	1	0	193.349	18.752	0	1
0.2	233.52	71.26	1	0	186.954	27.212	0.222	0.673
0.3	233.52	71.26	1	0	186.954	27.212	0.222	0.673
0.4	235.81	69.89	0.689	0.164	186.954	27.212	0.222	0.673
0.5	235.81	69.89	0.689	0.164	186.954	27.212	0.222	0.673
0.6	235.81	69.89	0.689	0.164	186.954	27.212	0.222	0.673
0.7	235.81	69.89	0.689	0.164	186.954	27.212	0.222	0.673
0.8	235.81	69.89	0.689	0.164	186.954	27.212	0.222	0.673
0.9	235.81	69.89	0.689	0.164	186.954	27.212	0.222	0.673
1	235.81	69.89	0.689	0.164	186.954	38.848	0.222	0.222

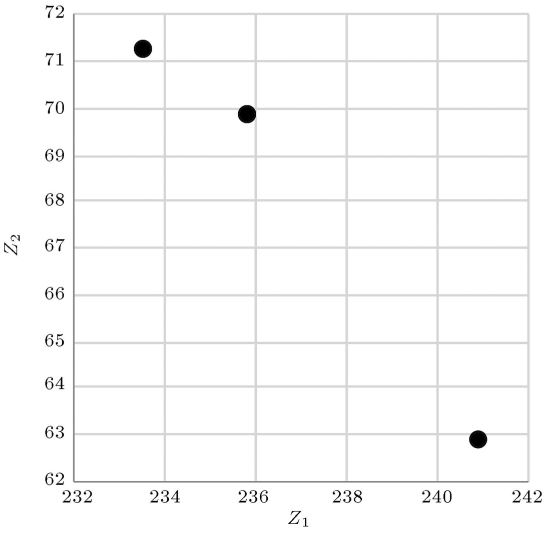


Figure 5. TH optimal Pareto frontier for Instance 3.

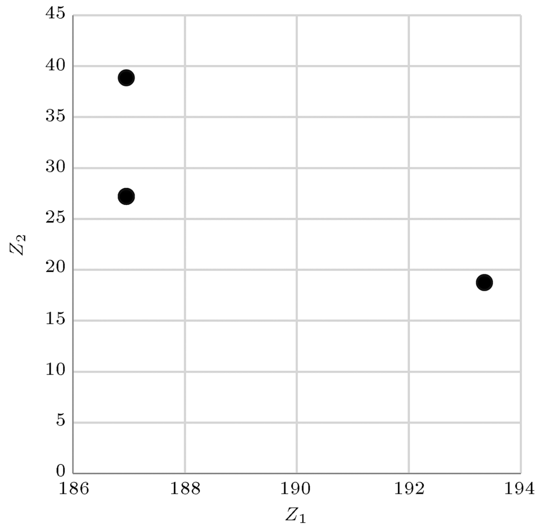


Figure 6. TH optimal Pareto frontier for Instance 2.

and negative ideal solution values obtained for every objective function. The TH optimal Pareto solutions to Instances 2 and 3 are presented in Table 6. In addition, Figures 5 and 6 depict these points.

Given the results obtained by the TH method, when the importance levels are equal, it can be observed that the values obtained for the objective functions are the same in many cases, i.e., the solutions obtained based on TH are almost balanced and compromised. It should be noted that two or three Pareto fronts can be obtained upon making changes to the controllable parameters, such as "γ" and "θ". The ε-constraint method is also used for solving the BOMILP model. Comparison is made between the results of these two exact methods in terms of three performance metrics. According to the findings, ε-constraint provides decision-makers with another Pareto front that has produced more solutions in terms of diversity, enjoying higher quality. Table 7 shows these results for Instances 2 and 3. Figures 7 and 8 depict these points. For indicating the learning effects on the objective functions, Table 8 illustrates their values for Instances 2 and 3 without regard to learning effects. The compared objective function values with respect to the learning effects (Table 7) and regardless of learning effects (Table 8) indicate the effects of learning on improvement of the objective functions.

The results of these two methods are compared through the application of comparison metrics. As can be seen from Table 9, in spite of the higher diversity of the results of ε-constraint, it exhibits better performance for MID and QM metrics.

Table 7. Results of ε-constraint for Instances 2 and 3 given learning effects.

3 × 3 × 2		**3 × 4 × 2**	
Z_1	Z_2	Z_1	Z_2
193.349	18.752	240.89	62.89
193.349	21.624	240.89	63.820
193.349	24.469	240.89	64. 750
193.349	27.367	240.89	65.680
186.954	27.367	240.89	66.610
186.954	30.239	240.89	67.540
186.954	33.111	240.89	68.470
186.954	35.982	240.89	69.400
186.954	38.854	235.81	70.330
164.597	44.579	233.52	71.260

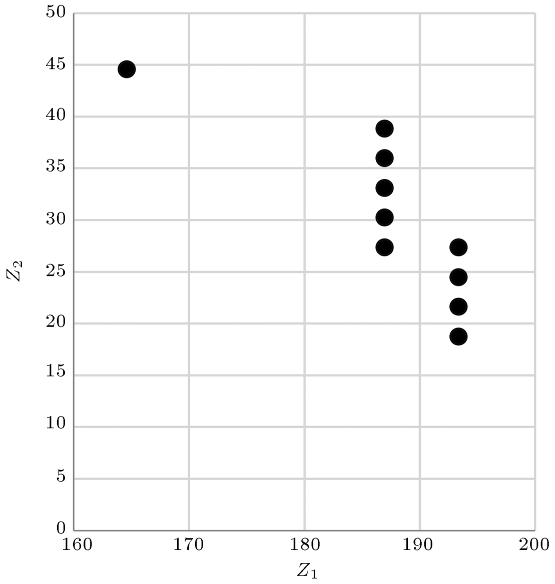


Figure 7. ε-constraint optimal Pareto frontier for Instance 2.

7.3. Comparison of metaheuristic methods

The exact methods are inapplicable solution techniques since the large-scale instances are highly complex. Consequently, five metaheuristic methods are applied to tackle the instances of large and medium sizes. For validation of the metaheuristic methods for the large-scale problems, the results of the exact and metaheuristic methods for the instances of small size are compared. As shown in Table 10, these methods can find optimal Pareto solutions to small-sized problems.

On the other hand, five instances of medium size and five instances of large size are generated randomly. Five metaheuristic algorithms are used for solving them, and the comparison of metric calculation results is reported. Table 11 illustrates these results. As can be seen from Figure 9, with regard to Quality Measure, NSGA-II obtains higher NDS, followed by NSGA-III, MOEA/D, SPEA-II, and PESA-II. As for Spacing Metric, Figure 10 clarifies that NSGA-III exhibits the best uniformity in the spread of the points, followed by PESA-II, SPEA-II, and NSGA-II, orderly. MOEA/D was ranked last. Figure 11 reveals that the lowest MID

Table 8. Results of ε-constraint for Instances 2 and 3 regardless of learning effects.

3 × 3 × 2		**3 × 4 × 2**	
Z_1	Z_2	Z_1	Z_2
222.36	49.5	277	85.50
222.36	52.567	277	89.333
222.36	58.035	277	93.167
222.36	62.434	277	97
222.36	66.986	277	100.833
222.36	70.033	277	104.667
222.36	73.564	277	108.5
222.36	75.162	277	112.333
222.36	80.350	277	116.167
204.435	85.423	277	120

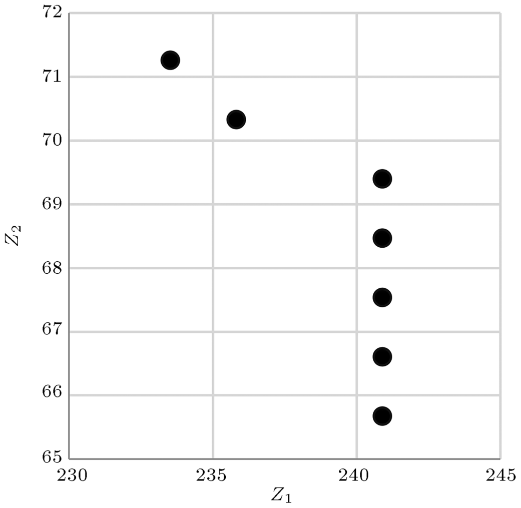


Figure 8. ε-constraint optimal Pareto frontier for Instance 3.

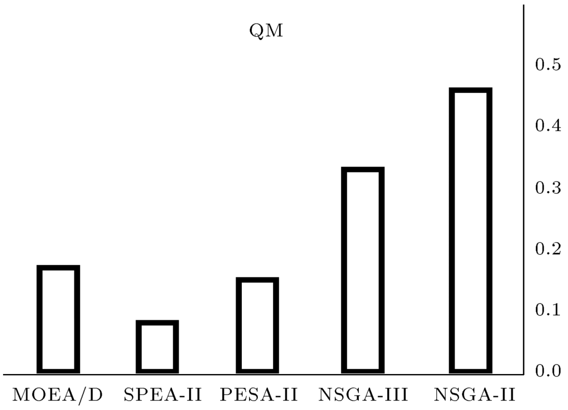


Figure 9. Quality measure of metaheuristic algorithms.

Table 9. Comparison of the TH and ε-constraint methods.

	ε-constraint method			**TH method**		
Representation	**SM**	**MD**	**QM**	**SM**	**MD**	**QM**
$\mathbf{3 \times 2 \times 1}$	7.8	518.77	1	7.8	518.77	1
$\mathbf{3 \times 3 \times 2}$	9.72	184.61	0.75	3.2	191.58	0.75
$\mathbf{3 \times 4 \times 2}$	3.689	246.48	0.75	4.28	246.33	0.75
$\mathbf{4 \times 4 \times 2}$	3.65	341.09	1	0	358.5	0.75
$\mathbf{3 \times 5 \times 2}$	1.868	319.66	1	0	330.17	0.75
Mean	**5.34**	**322.13**	**0.9**	**3.05**	**329.07**	**0.8**

Table 10. Comparison of the metaheuristic algorithms for the instances of small sizes (the metaheuristics values are the best values in 10 runs).

		NSGA-II			**NSGA-III**			**SPEA-II**			**PESA-II**			**MOEA/D**		
Instance	**Representation**	**SM**	**MID**	**QM**	**SM**	**MID**	**QM**	**SM**	**MID**	**QM**	**SM**	**MID**	**QM**	**SM**	**MID**	**QM**
1	$3 \times 2 \times 1$	7.8	518.77	1	7.8	518.77	1	7.8	518.77	1	7.8	518.77	1	7.8	518.77	1
2	$3 \times 3 \times 2$	9.72	184.61	1	9.72	184.61	1	9.72	184.61	1	9.72	184.61	1	9.72	184.61	1
3	$3 \times 4 \times 2$	3.68	246.55	0.75	2.54	247.45	0.75	3.68	246.55	0.75	3.68	246.55	0.75	2.72	247.46	0.33
4	$4 \times 4 \times 2$	3.65	341.09	1	3.65	341.09	1	3.65	341.09	1	3.65	341.09	1	3.65	341.09	1
5	$3 \times 5 \times 2$	1.87	319.66	1	1.87	319.66	1	1.87	319.66	1	1.87	319.66	1	1.87	319.66	1
	Mean	5.344	322.13	0.95	5.15	322.19	0.95	5.37	322.13	0.95	5.37	322.13	0.95	5.18	322.31	0.86

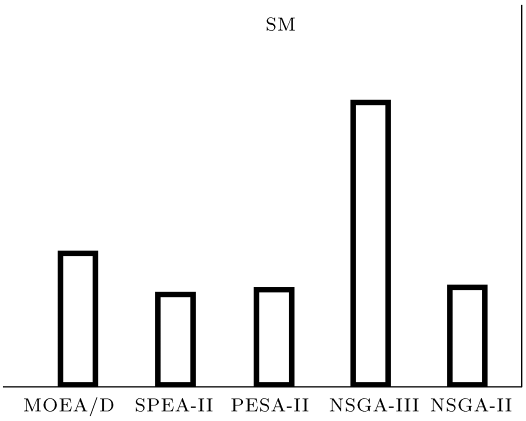


Figure 10. Spacing metric of metaheuristic algorithms.

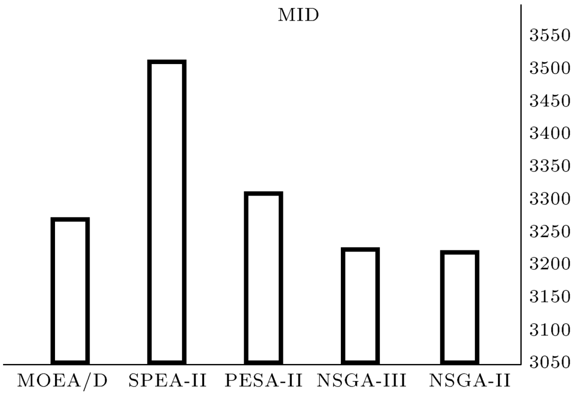


Figure 11. Mean ideal point distance of metaheuristic algorithms.

belongs to NSGA-III and then, NSGA-II, MOEA/D, SPEA-II, and PESA-II are characterized by high to low performance. According to this relationship, NSGA-II, NSGA-III, and MOEA/D exhibit better performance and obtain higher NDS for the medium- and large-sized instances. It is necessary to mention that parameters of algorithms have been tuned through trial and error.

8. Conclusion

The present paper proposed a new bi-objective model for the job shop scheduling problem given fuzzy due dates and processing times, modified DeJong's learning effect on job processing times, the sum-of-processing-time learning effect on execution times of maintenance activities, machine availability constraints, and transportation times. With these assumptions, the Job Shop Scheduling Problem (JSSP) seemed more practical in real-world applications. The objective was to minimize makespan and maximum tardiness simultaneously. The original model was converted into one of the auxiliary single-objective crisp types through the application of the TH method, with optimal Pareto solutions obtained for the small-sized problems. On the other hand, the ε-constraint method was used for solving the bi-objective crisp model and this type of problem exactly. Based on the comparison of the results of these two methods in terms of three performance metrics, the results of the ε-constraint

Table 11. Comparison of the metaheuristic algorithms for the instances of large and medium sizes.

		NSGA-II			NSGA-III			SPEA-II			PESA-II			MOEA/D		
Instance	Representation	SM	MID	QM	SM	MID	QM	SM	MID	QM	SM	MID	QM	SM	MID	QM
6	5 × 6 × 3	6.6	522.07	0.83	6.6	522.07	0.83	4	522.27	0.33	22	526.27	0.16	4	522.27	0.33
7	6 × 6 × 3	4.2	717.75	0.5	2.18	718.19	0.5	0.15	735.9	0	0.15	769.31	0	0.15	722.1	0.25
8	7 × 7 × 3	0.3	1071.07	0.75	0	1102.3	0.33	22	1194.93	0	3.82	1259.96	0	1	1208.39	0
9	8 × 6 × 3	3.5	827	0.25	0.5	825	0.5	4.8	1004.82	0.125	3.5	967	0.25	4	904	0.25
10	8 × 9 × 3	29	2552.08	0.25	0	2469.81	0.25	23.5	2503.69	0	0	2672.33	0	7.1	2390	0.5
11	10 × 9 × 3	31.78	2259.98	0.75	7.6	2472.43	0	1.4	2573	0	0	2594.6	0	0	2269.17	0.25
12	12 × 10 × 5	20	7253.8	0	0	7290.42	0	0	6714.4	1	0	7494.5	0	0	7076	0
13	14 × 8 × 6	6.5	4752.48	0.28	15.5	4692.38	0.28	24	4624.53	0.14	30	4662.96	0.14	29.4	4673.61	0.14
14	16 × 12 × 6	9.5	8124.08	0.66	0	8277.24	0.33	18	8513.43	0	44	8779	0	88.5	8759.79	0
15	18 × 14 × 6	0	4107	0.33	0	3862	0.33	0	4702	0	0	4137.5	0.33	0	4170.3	0
	Mean	11.13	3218.73	0.46	3.23	3223.18	0.33	10.87	3308.94	0.15	10.34	3510.24	0.08	15.03	3269	0.17

method provided the decision-maker with an additional Pareto front, which produced more solutions in terms of diversity and enjoyed higher quality. Considering two small instances with and without learning effects shows the effects of learning on improvement of objective functions. Five different metaheuristics were applied to the problems of large and medium sizes. The validity of the metaheuristic methods for the large-scale problems was verified by comparing the results of solving small-sized instances using these methods with those of the exact methods. On the other hand, the results obtained for the large and medium problems compared in terms of the three metrics indicated that although these algorithms were ranked as NSGA-III (Non-dominated Sorting Genetic Algorithm), PESA-II (Pareto Envelope-based Selection Algorithm), SPEA-II (Strength Pareto Evolutionary Algorithm), NSGA-II, and MOEA/D (Multi-Objective Evolutionary Algorithm based on Decomposition) in terms of Spacing Metric (SM), they were sorted as NSGA-II, NSGA-III, MOEA/D, SPEA-II, and PESA-II based on the other two metrics (i.e., QM and MID), thus showing the performance of the algorithms. It is suggested for future works that similar models be defined in the environment of a multi-objective flexible job shop or open shop or to consider constraints like flexible maintenance activities and sequence-dependent setup times.

References

1. Torabi, S.A. and Hassini, E. “An interactive possibilistic programming approach for multiple objective supply chain master planning”, *Fuzzy Sets and Systems*, **159**(2), pp. 193–214 (2008).
2. Muth, J.F. and Thompson, G.L., *Industrial Scheduling*, Englewood Cliffs, New Jersey (1963).
3. Biskup, D. “Single-machine scheduling with learning considerations”, *European Journal of Operational Research*, **115**(1), pp. 173–178 (1999).
4. Biskup, D. “A state-of-the-art review on scheduling with learning effect”, *European Journal of Operational Research*, **188**(2), pp. 188–315 (2008).
5. Azzouz, A., Ennigrou, M., and Ben Said, L. “Scheduling problems under learning effects: classification and cartography”, *International Journal of Production Research*, **56**(4), pp. 1642–1661 (2018).
6. Mousavipour, S.H., Farughi, H., and Ahmadizar, F. “A job shop scheduling problem with sequence-dependent setup times considering position-based learning effects and availability constraints”, *International Journal of Industrial Engineering* & *Production Research*, **30**(3), pp. 329–340 (2019).
7. Mousavipour, S.H., Farughi, H., and Ahmadizar, F. “Job shop scheduling problem based on learning effects, flexible maintenance activities and transportation times”, *Journal of Industrial and Systems Engineering*, **12**(3), pp. 107–119 (2019).
8. Tayebi Araghi, M.E., Jolai, F., and Rabiee, M. “Incorporating learning effect and deterioration for solving a SDST flexible job-shop scheduling problem with a hybrid meta-heuristic approach”, *International Journal of Computer Integrated Manufacturing*, **27**(8), pp. 733–746 (2014).
9. Renna, P. “Flexible job-shop scheduling with learning and forgetting effect by Multi-Agent System”, *International Journal of Industrial Engineering Computations*, **10**, pp. 521–534 (2019).
10. Okolowski, D. and Gawiejnowicz, S. “Exact and heuristic algorithms for Parallel-machine scheduling with DeJong's learning effect”, *Computers* & *Industrial Engineering*, **59**(2), pp. 272–279 (2010).
11. Lai, P.J. and Wu, H.C. “Using heuristic algorithms to solve the scheduling problems with job-dependent and machine-dependent learning effects”, *J Intell Manuf*, **26**, pp. 691–701 (2015).
12. Vahedi-Nouri, B., Fattahi, P., Tavakkoli-Moghaddam, R., et al. “A general flow shop scheduling problem

with consideration of position-based learning effect and multiple availability constraints", *International Journal of Advanced Manufacturing Technology*, **73**(5), pp. 601–611 (2014).

13. Amirian, H. and Sahraeian, R. "Augmented ε-constraint method in multi-objective flow shop problem with past sequence set-up times and a modified learning effect", *International Journal of Production Research*, **53**(19), pp. 1–15 (2015).

14. Behnamian, J. and Zandieh, M. "Earliness and tardiness minimizing on a realistic hybrid flowshop scheduling with learning effect by advanced metaheuristic", *Arab J Sci Eng*, **38**, pp. 1229–1242 (2013).

15. Gao, F., Liu, M., Wang, J.J., et al. "No-wait two-machine permutation flow shop scheduling problem with learning effect, common due date and controllable job processing times", *International Journal of Production Research*, **56**(6), pp. 2361–2369 (2018).

16. Mousavi, S.M., Mahdavi, L., Rezaeian, J., et al. "Bi-objective scheduling for the re-entrant hybrid flow shop with learning effect and setup times", *Scientia Iranica E*, **25**(4), pp. 2233–2253 (2018).

17. Schmidt, G. "Scheduling with limited machine availability", *European Journal of Operational Research*, **121**(1), pp. 1–15 (2000).

18. Ma, Y., Chu, C., and Zuo, C. "A survey of scheduling with deterministic machine availability constraints", *Computers & Industrial Engineering*, **58**(2), pp. 199–211 (2010).

19. Hsu, C.J., Low, C., and Su, C.T. "A single-machine scheduling problem with maintenance activities to minimize Make Span", *Applied Mathematics and Computation,* **215**(11), pp. 29–35 (2010).

20. Xu, D., Xu, Z., He, J., et al. "Mixed integer programming formulations for two-machine flow shop scheduling with an availability constraint", *Arab J Sci Eng*, **43**, pp. 777–788 (2018).

21. Aggoune, R. "Two-job shop scheduling problems with availability constraints", *14th International Conference on Automated Planning and Scheduling*, pp. 253–259 (2004).

22. Benttaleb, M., Hnaien, F., and Yalaoui, F. "Two-machine job shop problem under availability constraints on one machine: Makespan minimization", *Computers & Industrial Engineering*, **117**, pp. 138–151 (2018).

23. Tamssaouet, K., Dauzère-Pérès, S., and Yugma, C. "Metaheuristics for the job-shop scheduling problem with machine availability constraints", *Computers & Industrial Engineering*, **125**, pp. 1–8 (2018).

24. Naderi, B., Zandieh, M., and Fatemi Ghomi, S.M.T. "Scheduling sequence-dependent setup time job shops with preventive maintenance", *Int J Adv Manuf Technol*, **43**(1), pp. 170–181 (2009).

25. Taraki, H., Tang, K., and Teyarachakul, S. "Learning effects on maintenance outsourcing", *European Journal of Operational Research*, **192**(1), pp. 138–150 (2009).

26. Ahmadizar, F. and Shahmaleki, P. "Group-shop scheduling with sequence-dependent set-up and transportation times", *Applied Mathematical Modelling*, **38**(21), pp. 5080–5091 (2014).

27. Hurink, J. and Knust, S. "Tabu search algorithms for job-shop problems with a single transport robot", *European Journal of Operational Research*, **162**(1), pp. 99–111 (2005).

28. Nouri, H.E., Driss, O.B., and Ghédira, K. "Hybrid metaheuristics for scheduling of machines and transport robots in job shop environment", *Appl. Intell*, **45**(3), pp. 808–828 (2016).

29. Sakawa, M. and Kubota, R. "Two-objective fuzzy job shop scheduling through genetic algorithm", *Electronics and Communication in Japan (Part III: Fundamental Electronic Science)*, **84**(4), pp. 60–68 (2001).

30. Lei, D. "Co-evolutionary genetic algorithm for fuzzy flexible job shop scheduling", *Applied Soft Computing*, **12**(8), pp. 2237–2245 (2012).

31. Ahmadizar, F. and Zarei, A. "Minimizing Make Span in a group shop with fuzzy release dates and processing times", *Int J Adv Manuf Technol*, **66**(9–12), pp. 206–219 (2013).

32. Noori-Darvisha, S., Mahdavi, I., and Mahdavi-Amir, N. "A bi-objective possibilistic programming model for open shop scheduling problems with sequence-dependent setup times, fuzzy processing times, and fuzzy due dates", *Applied Soft Computing*, **12**(4), pp. 1399–1416 (2012).

33. Wagner, H.M. "An integer linear-programming model for machine scheduling", *Naval Rest Logis Q*, **6**(2), pp. 131–140 (1959).

34. Yang, D.L. and Kuo, W.H. "Single-machine scheduling with an actual time-dependent learning effect", *Journal of the Operational Research Society*, **58**(10), pp. 1348–1353 (2007).

35. Lai, Y.J. and Hwang, C.L. "A new approach to some possibilistic linear programming problems", *Fuzzy Sets and Systems*, **49**(2), pp. 121–133 (1992).

36. Deb, K., Pratap, A., Agarwal, S., et al. "A fast and elitist multi-objective genetic algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation*, **6**(2), pp. 182–97 (2002).

37. Zitzler, E., Laumanns, M., and Thiele, L. "SPEA2: Improving the strength Pareto evolutionary algorithm", Computer Engineering and Networks Laboratory (TIK), Department of Electrical Engineering Swiss Federal Institute of Technology (ETH), Zurich (2001).

38. Corne, D.W., Jerram, N.R., Knowles, J.D., et al. "PESA-II: Region-based selection in evolutionary

multi-objective optimization", *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 283–290 (2001).

39. Gadhvia, B., Savsania, B., and Vivek, P. "Multi-objective optimization of vehicle passive suspension system using NSGA-II, SPEA2 and PESA-II", *Procedia Technology*, **23**, pp. 361–368 (2016).

40. Deb, K. and Jain, H. "An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, part I: Solving problems with box constraints", *IEEE International Transactions in Operational Research*, **18**(4), pp. 577–601 (2014).

41. Zhang, Q. and Li, H. "MOEA/D: A multi-objective evolutionary algorithm based on decomposition", *IEEE Transaction on Evolutionary Computation*, **11**(6), pp. 712–731 (2007).

42. Poursabzi, O., Mohammadi, M., and Naderi, B. "An improved model and heuristic for capacitated lot-sizing and scheduling in job shop problems", *Scientia Iranica E*, **25**(6), pp. 3667–3684 (2018).

43. Srinivas, N. and Deb, K. "Multi-objective optimization using non-dominated sorting in genetic algorithms", *Evolutionary Computation*, **2**(3), pp. 221–248 (1994).

44. Schaffer, J.D. "Multiple objective optimizations with vector evaluated genetic algorithms and their applications", *Proceedings of the First International Conference on Genetic Algorithms*, Lawrence Erlbaum, Hillsdale, USA, pp. 93–100 (1985).

Biographies

Seyed Hamed Mousavipour is a PhD Candidate of Industrial Engineering at University of Kurdistan. He was born in 1983. He received BS and MS degrees in Industrial Engineering from Islamic Azad University in 2006 and 2009, respectively. His research interest topics include scheduling theory, combinatorial optimization, and production planning.

Hiwa Farughi is an Associate Professor at University of Kurdistan. He was born in 1976. He received BS and MS degrees in Industrial Engineering from Amirkabir University of Technology in 1998 and 2000, respectively. He received his PhD degree in Industrial Engineering from Iran University of Science and Technology in 2012. His research interest topics include operations research applications in health care, production planning, reliability, and quality control.

Fardin Ahmadizar is an Associate Professor of Industrial Engineering at University of Kurdistan. He received BS and MS degrees in Industrial Engineering from Amirkabir University of Technology and Isfahan University of Technology in 1996 and 1998, respectively. He received his PhD degree in the same field of study from Iran University of science and Technology in 2009. His research interest topics include scheduling theory, combinatorial optimization, supply chain optimization, and metaheuristic algorithms.