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

Seyedhamed Mousavipour a; Hiwa Farughi b, *; Fardin Ahmadizar c

a. Department of industrial engineering, University of Kurdistan, Pasdaran Boulevard, Sanandaj, Iran, Email: hamedmousavi942@gmail.com
b. Corresponding author. Tel.: +988733660073
E-mail address: h.farughi@uok.ac.ir
Department of industrial engineering, University of Kurdistan, Pasdaran Boulevard, Sanandaj, Iran
c. Department of industrial engineering, University of Kurdistan, Pasdaran Boulevard, Sanandaj, Iran,
Email: f.ahmadizar@uok.ac.ir

Abstract

This paper aims at introducing a novel bi-objective model for a Job Shop Scheduling Problem (JSSP) in order to minimize makespan and maximum tardiness simultaneously. Some realistic assumptions, i.e. 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-time based learning effects on duration of maintenance activities have been considered, to provide a more general and practical model for the JSSP. Based on the learning effects, Processing times decrease as a machine performs an operation frequently, and workers gain working skills and experiences. In this paper based on DeJong’s learning effect a novel and modified formulation has been 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 to an auxiliary crisp model, given that model is a possibilistic programming, it is then solved by the TH and ε-constraint methods for small instances, and the results are compared. For medium and large instances, five metaheuristic algorithms, including NSGA-III, PESA-II, SPEA-II, NSGA-II, and MOEA/D are utilized, and the results are finally compared on the basis of three performance metrics.

Keywords: job shop scheduling problem; learning effects; availability constraints; transportation times

1. Introduction
Owing to the theoretical challenges and wide industrial applications, the JSSP is undoubtedly worth research endeavor 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 of makespan and maximum tardiness. Since the model is original, and there are few benchmarks, a number of instances of three different sizes are generated randomly. For solving small-sized instances to obtain optimal Pareto solutions, we utilize Torabi and Hassini’s [1] proposed interactive method of fuzzy multi-objective decision-making (FMODM), called TH method. This method obtains an equivalent auxiliary single-objective crisp model, which is then optimally solved using GAMS. In addition, small instances are optimally solved as well using the ε-constraint method. Given the complexity of the problem, five metaheuristics, i.e. 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 for tackling instances of large and medium sizes. 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 in spite of 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. On the other hand, as far as we know, no study has investigated learning effects on maintenance activities in scheduling problems, and no other research on JSSP has taken all these assumptions into account.

2. Literature review

JSSP as a well-known scheduling problem 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, 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. Job processing times are assumed by most researchers in classical scheduling problems to be constant and known throughout the planning horizon, while, in the vast majority of real situations, they decrease as a machine does an operation frequently, or workers gain working skills and experiences. The above phenomenon is called the “learning effects.” The notion was first introduced into 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. Azzouz et al. [5], in the recent review paper have emphasized that in spite of numerous studies which have been conducted on different types of scheduling environments under the phenomenon of learning, there is no study related to learning effects on JSSP but recently Mousavipour et al. [6] studied a sequence dependent set-up times job shop scheduling problem with availability constraints based on classic position-based learning effects. They also provided a 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 set-up 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 range of operations by introducing a hybrid algorithm. Renna [9] studied flexible job shop scheduling with learning and forgetting effects. Okolowski and Gwiejnowicz [10] introduced a novel learning curve in a parallel machine environment, called DeJong’s Learning Curve, in which they proposed an incompressibility index to separate 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 highly frequent assumption in classical JSSP is that machines are always available. This is not the case in the real world, however, for reasons such as failures and maintenance activities that have been studied in the literature as availability constraints. Schmit [17], Ma and Chu [18]. Have conducted comprehensive reviews on scheduling with availability constraints. Two main types of deterministic availability constraints have been considered: first one is fixed availability constraints in which Start time and duration of the maintenance activity are already certain and fixed. In the second one, start time of maintenance activity is flexible and lie 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 the 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. Benttaeleb et al. [22], have provided two formulation for a two-machine JSSP with availability constraint on one machine. Tamssauet et al. [23], addressed a JSSP when machines are not available always due to fixed periods of maintenance activities. They have 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], where different techniques were proposed 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 the simulated annealing and genetic algorithms.

In current study, multiple fixed periods of preventive maintenance on m machines have been taken into account. The innovation of this paper from availability constraints point of view is that, availability constraints have been formulated based on position-based model of JSSP and under the phenomenon of learning caused by repetition.
As far as we know, in the 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 is responsible for performing preventive maintenance activities. They assumed that the contractor would learn through repetition and preparation how to perform the preventive maintenance operations more efficiently. In this case, the maintenance operational teams are assigned in such a way 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 had to 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. We consider unlimited job-dependent transportation times.

Furthermore, recent researches have increasingly considered the inherent uncertainty in model parameters, and uncertainty seems to be an inseparable part of real-world problems. In this paper, it has been assumed that we have data of approximant values of jobs processing times and due dates, consequently fuzzy parameters have been applied. A large number of papers have taken uncertainty into consideration as fuzzy parameters in different scheduling environments, Refs. [29], [30], [31], and [32], are only a few of them.

The remainder of this paper is organized as follows: The mathematical formulation and validation of proposed model are provided in Section 3. The applied exact and metaheuristic solution techniques are detailed in Section 4 and 5. Section 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 as follows.

1-3. Assumptions

• The jobs cannot be pre-empted.
• Each machine can process each job at most once.

• 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-time based learning effect.

• The processing times include the set-up times.

• Transportation times between machines are considered as job-dependent.

• There is no limitation on transporter and maintenance teams.

2-3. 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 \( k^{th} \) position of machine \( m_i \)
\( P_{ij} \): Normal processing time of job \( J_j \) on machine \( m_i \)
\( d_j \): Due date of job \( J_j \)
\( \alpha_i \): Learning index of job processing on machine \( M_i \) (\( \alpha_i \leq 0 \))
\( \varphi_j \): Learning index of job \( J_j \) (\( \varphi_j \leq 0 \))
\( \lambda_{ij} \): Learning index of job \( J_j \) -on machine \( m_i \)
\( Pm_{iR} \): \( R^{th} \) maintenance activity on machine \( m_i \)
\( t_{iR} \): Normal execution time of \( Pm_{iR} \)
\( t_{iR}^A \): Actual execution time of \( Pm_{iR} \)
\( y_{iRk} \): A binary parameter that is 1 if \( R^{th} \) maintenance activity must be done after processing the job on \( k^{th} \) position of machine \( m_i \) and 0 otherwise

\( r_{ijl} \): A binary parameter that is 1 if the \( l^{th} \) operation of the \( j^{th} \) job is processed on machine \( m_i \), and is 0 otherwise

\( t_{pijl} \): Time needed for transfer of job \( J_j \) from machine \( m_i \) to machine \( m_{ij} \)

3-3. Decision variables

\( C_{ijk} \): Completion time of job \( J_j \) if it is scheduled in the \( k^{th} \) position of machine \( m_i \) and 0 otherwise

\( X_{ijk} \): A binary variable that is 1 if job \( J_j \) is processed in the \( k^{th} \) position of machine \( m_i \), and is 0 otherwise

\( C_{\text{max}} \): makespan

\( L_j \): Lateness of job \( J_j \) = \( C_j - d_j \)

\( T_j \): Tardiness of job \( J_j \) = \( \max \{0, L_j\} \)

\( T_{\text{max}} \): Maximum tardiness of jobs

4-3. Mathematical model

We develop Wagner’s [33] mathematical model, 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 and the variable processing time that is effected by learning from each other based on DeJong’s learning effect. Amirian and Sahraean [13], defined each index separately for each 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, as in Ref. [11], the learning parameter is 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.

\[
p_{ijk} = p_{ij} (M_{ij} + (1-M_{ij}) \times K^{\lambda_{ij}}) \quad \text{and} \quad \lambda_{ij} = \alpha_i + \varphi_j
\]

Here, \( \alpha_i < 0 \) and \( \varphi_j < 0 \) are the learning indices of machine \( m_i \) and job \( J_j \), respectively, \( \lambda_{ij} \) 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 Ref. [12] and [13], multiple fixed preventive maintenance activities on each machine have been 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 has 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 operations with the intervention of the operator, the learning effects are considered, as in Yang and Kou [34], according to the sum of processing times, different in that a separate learning rate has been assumed for each machine.

\[
t_{iR}^A = t_{iR} (1 + \sum_{R=1}^{R-1} t_{iR}^A)^{\alpha_i}
\]
Here, $\omega_i < 0$ is the learning index of maintenance operations on machine $m_i$. Noteworthy is to mention that, in this study, autonomous learning effects have been assumed. In this kind of learning effects, training has no impact on learning rate, and this rate is impressed by repetition of jobs and operator experience. Other kind of learning effects which called induced learning effects, are affected by training. In the real situations these indices have been gained experimentally by using learning curves. 

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

\[ \tilde{p}_{ij} = (p_{ij}, m_{ij}, o_{ij}) \]  
\[ \tilde{d}_j = (p_{dj}, m_{dj}, o_{dj}) \] 

$(p_{ij}, d_j)$ are the pessimistic values, $(m_{ij}, d_j)$ are the probable values, and $(o_{ij}, d_j)$ are 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 as follows based on the assumptions stated above.

\[ \text{Min } Z_1 = C_{\text{max}} \]  
\[ \text{Min } Z_2 = T_{\text{max}} \] 

S.to

\[ \sum_k X_{ijk} = 1 \quad \forall i = 1,...,m, j = 1,...,n \]  
\[ \sum_j X_{ijk} = 1 \quad \forall i = 1,...,m, k = 1,...,n \]  
\[ \sum_j C_{ijk} + \sum_j X_{ijk+1} \times \tilde{p}_{ijk+1} + \sum_{R=1}^{R^*} y_{iRk} \times t_{iR} \leq \sum_j C_{ijk+1} \quad \forall i = 1,...,m, k = 1,...,n-1 \]
\[
\sum_i r_{ijl} \times C_{ijk} + \sum_i r_{ijl+1} \times X_{ijl} \times \tilde{P}_{ijk} + \sum_i \sum_l r_{ijl} \times r_{ijl+1} \times tp_{ijl} \leq V \left(1 - \sum_i r_{ijl} \times X_{ijl} \right) + V \\
\times \left(1 - \sum_l r_{ijl+1} \times X_{ijl} \right) + \sum_l r_{ijl+1} \times C_{ijl} \]

\[C_{ijk} \leq V \times X_{ijk} \quad \forall i = 1, \ldots, m \ , \ j = 1, \ldots, n \ , \ k = 1, \ldots, n \] (10)

\[C_j \geq \sum_k C \quad \forall j = 1, \ldots, n \] (11)

\[\sum_k C_{[l]jk} \geq \sum_k X_{[l]jk} \times P_{[l]jk} \quad \forall j = 1, \ldots, n \] (12)

\[C_j - \tilde{d}_j \leq T_j \quad \forall j = 1, \ldots, n \] (13)

\[C_{\text{max}} \geq C_j \quad \forall j = 1, \ldots, n \] (14)

\[T_{\text{max}} \geq T_j \quad \forall j = 1, \ldots, n \] (15)

\[C_j, T_j \geq 0 \quad \forall j = 1, \ldots, n \] (16)

\[X_{ij} \in \{0, 1\} \quad \forall i = 1, \ldots, m \ , \ j = 1, \ldots, n \] (17)

\[X_{ij} \in \{0, 1\} \quad \forall i = 1, \ldots, m \ , \ j = 1, \ldots, n \] (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 through 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. Equation (11) states the completion time of job $J_j$ if scheduled in the $k^{th}$ 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.

5-3 validity of model

This model is an original model, consequently we have no access to the benchmarks instances. To verify this model, as Figure 1, indicates a Gantt chart has been applied.

A Small problem contains 2 jobs and 3 machines and 1 predetermined preventive maintenance activity for each machine is solved. For simplicity's sake, same learning and incompressibility indices have been considered for all jobs and machines. Furthermore, only makespan is minimized. The crisp processing times of jobs which are affected by learning effects, sequence of jobs processing which are generated randomly, duration of maintenance activities and transportation times are shown in Table 1. Learning indices, incompressibility index and actual values of processing times are illustrated in Table 2.

According to the suggested model, the optimal value of makespan is 185. There are 8 feasible solutions in this problem. These solutions contain decision variables which have values of 1 and corresponding makespan are as follows:
As solutions indicate the optimal makespan gained by suggested model is 185 and this value is approved by Gant chart. As a consequent, Gant chart confirms the validity of suggested model.

4. Solution methodology

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

\[
\sum_j C_{ij} + \sum_j X_{ijk+1} \\
\times \left( W_1 P_{ij}^p + W_2 P_{ij}^m + W_3 P_{ij}^p \right) \\
+ \sum_{R=1}^\infty y_{i} t_{iR} \leq \sum_j C_{ijk+1} \\
\forall i = 1, ..., m \\
k = 1, ..., n - 1
\]
\[
\sum r_{ijl} \times C_{ijk} + \sum r_{ijl+1} \times X_{ijk} \quad \forall j = 1,\ldots,n
\]
\[
\times \left( W_1 P_{ij,\beta}^d + W_2 P_{ij,\beta}^m + W_3 P_{ij,\beta}^o \right)
\]
\[
+ \sum \sum r_{ijl} \times r_{ijl+1} \times tP_{ijl}^l
\]
\[
\leq V \left( 1 - \sum r_{ijl} \times X_{ijk} \right)
\]
\[
+ V \left( 1 - \sum r_{ijl+1} \times X_{ijl+1} \right) + \sum r_{ijl+1} \times C_{ijl+1}
\]
\[
C_j - (W_1 d_{ij,\beta}^p + W_2 d_{ij,\beta}^m + W_3 d_{ij,\beta}^o) \leq T_j \quad \forall j = 1,\ldots,n
\] (21)

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

\[
\beta = 0.5, W_1 = 1/6, W_2 = 4/6, W_3 = 1/6.
\] (22)

Based on these proposed values, the expression of the auxiliary crisp bi-objective mixed-integer linear programming (BOMILP) reads:

\[
\text{Min } z = [Z_1, Z_2]
\] (23)

S.to \( X \in F(X) \) (24)

Where" X" and "F(X)" represent the possible continuous and binary variable solution vector in the original model and the possible area including crisp constraints, respectively.

1-4. TH method

Based on Ref. [32], the stages of TH are as 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:
\[ Z_{1}^{\text{PIS}} = \min C_{\text{max}} \quad \text{S.to} \quad X \in F(X) \quad (25) \]

\[ N_{1}^{\text{NIS}} = \max C_{\text{max}} \quad \text{S.to} \quad X \in F(X) \quad (26) \]

\[ N_{2}^{\text{PIS}} = \min T_{\text{max}} \quad \text{S.to} \quad X \in F(X) \quad (27) \]

\[ N_{2}^{\text{NIS}} = \max T_{\text{max}} \quad \text{S.to} \quad X \in F(X) \quad (28) \]

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

\[
\mu_{Z_{1}}(X) = \begin{cases} 
1 & Z_{1} < Z_{1}^{\text{PIS}} \\
\frac{Z_{1}^{\text{NIS}} - Z_{1}}{Z_{1}^{\text{NIS}} - Z_{1}^{\text{PIS}}} & Z_{1}^{\text{PIS}} \leq Z_{1} \leq Z_{1}^{\text{NIS}} \\
0 & Z_{1} > Z_{1}^{\text{NIS}} 
\end{cases} \]

\[
\mu_{Z_{2}}(X) = \begin{cases} 
1 & Z_{2} < Z_{2}^{\text{PIS}} \\
\frac{Z_{2}^{\text{NIS}} - Z_{2}}{Z_{2}^{\text{NIS}} - Z_{2}^{\text{PIS}}} & Z_{2}^{\text{PIS}} \leq Z_{2} \leq Z_{2}^{\text{NIS}} \\
0 & Z_{2} > Z_{2}^{\text{NIS}} 
\end{cases} \quad (30)
\]

\[
(29)
\]

1. Obtaining the equivalent crisp single-objective model:

\[
\max \lambda(x) = \gamma \lambda_{0} + (1 - \gamma) \sum \theta_{h} \mu_{\lambda_{h}} \quad (31)
\]

\[
\text{S.to} \quad \lambda_{0} \leq \mu_{\lambda_{h}}(X) \quad h = 1, 2 \quad (32)
\]

\[
X \in F(X) \quad (33)
\]

\[
\lambda_{0}, \gamma \in [0, 1] \quad (34)
\]

Here, \( \mu_{\lambda_{h}}(x) \) indicates the degree of satisfaction of the \( h^{\text{th}} \) objective function, and \( \lambda_{0} = \min(\mu_{\lambda_{h}}(X)) \) is its minimum satisfaction degree. \( h \) represents the importance degree of the \( h^{\text{th}} \) objective function, determined by the decision-maker, but the important point is that \( \sum \theta_{h} = 1 \) and \( \theta_{h} > 0 \). The decision-maker determines the \( \theta_{h} \) parameters linguistically. \( \gamma \) is the compensation coefficient, which is a dynamic parameter lying in 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 with the solutions, and otherwise, we will return to the third step changing the controllable values $\gamma, \beta$.

2-4. 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 each 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.

$$\begin{align*}
\text{Min} & \quad f(x) \\
\text{S. to} & \\
& f_i(x) \leq \varepsilon_i \quad \text{for all} \quad i = 1, 2, \ldots
\end{align*}$$

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

5. Metaheuristic algorithms

As mentioned before, we apply five metaheuristic algorithms 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 made on 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 the 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 basis for calculation of the crowding distance between any two neighboring solutions. In SPEA–II, which was introduced by Zitzler et al. [37], an external archive including the last non-dominated solutions is updated following each generation, and a value of strength is calculated for every solution. Each individual’s fitness is obtained based on these computed strength values. Region-based selection is suggested in Corne et al.’s [38] proposal, PESA-II, where selection is made in terms of hyperboxes in the objective space, which are assigned selective fitness. The method aimed at decreasing 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 covering 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.

1-5. Solution representation

JSSP is a 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 for presentation of 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 which should be processed by corresponding machine. Fig.2 and 3 show this procedure.

An initial solution for the problem is generated with a heuristic method that generates the first feasible solution for each machine greedily. In this procedure random numbers between one and number of jobs are generated for each machine. As Figure 4, indicates "ones" determine the initial sequence of jobs processing on machines. 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

For comparison of multi-objective algorithms, it is necessary to introduce criteria that can evaluate the Pareto solutions. In this paper, three criteria are used for evaluation of the Pareto solutions; in these criteria, $n$ denotes Non-Dominated Solutions (NDS) set size, and $f_{ki}$ and $f_{kj}$, $i = j = 1 \ldots n$ and $k = 1, 2$ indicate the values of $k^{th}$ objective function for the $i^{th}$ and $j^{th}$ solutions, respectively.

1-6. Spacing Metric (SM)

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

$$SM = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (d_i - \bar{d})^2}$$  (37)
\[ d_i = \min_j \{ |f_{ui} - f_{uj}| + |f_{2ui} - f_{2uj}| \} \quad i, j = 1, 2, ..., n, \quad i \neq j \] (38)

\[ \bar{d} = \frac{\sum_{i=1}^{n} d_i}{n} \] (39)

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

**2-6. Mean ideal point distance (MID)**

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

\[ MID = \frac{\sum_{i=1}^{n} C_i}{n} \] (40)

\[ C_i = \sqrt{\sum_{k=1}^{k} f_{i2}^2} \] (41)

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

**3-6. Quality Measure (QM)**

This criterion was proposed by Schaffer [44] and considers the number of final Non-Dominated Solutions (NDS) found by each of the algorithms. Clearly, higher performance is offered by an algorithm that enjoys a greater QM value.

**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 sizes exactly, and the results of the TH and \(\varepsilon\)-constraint methods are reported. However, the optimal solutions to those of large and medium sizes cannot be found using GAMS within a logical time; here, the MOEA-D, PESA-II, SPEA-II, NSGA-II, and NSGA-III algorithms are used. Comparison is made between the results according to the criteria introduced in later sections. MATLAB 2015 and a PC with a Core i5 2.5GHz CPU, 3MB cache, and 4GB RAM have been used for coding these algorithms.

**1-7. Generation of mathematical instances**

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

\[ \bar{P}_{ij} = \left( P_{ij} - u_{ij}, P_{ij}, P_{ij} + u_{ij} \right) \] (42)

\[ \bar{d}_j = \left( d_j - u_j, d_j, d_j + u_j \right) \] (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. Each instance is designed as "a. b. c." 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 ≤ 75", medium-sized with "a.b.c ≤ 150" and large-sized with "a.b.c > 150". These ranges are determined based on previous studies in the JSSP. Predetermined sequence of each job on each machine has been generated randomly as well. Noteworthy is to mention that problems are handled by different learning effects, but for simplicity, the same learning rates are considered for all machines and all jobs in each problem.

2-7. Comparison of the exact methods

Discussion is made here on the results obtained for the instances of small sizes. Five types of small-sized sample are taken into account with different job, machine, 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 are solved separately for computation of the values of positive and negative ideal solution (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 and negative ideal solution values obtained for each of the objective functions. The TH optimal Pareto solutions for instances "2" and "3" are presented in Table 6. In addition, Figure 5 and Figure 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 through changes in the controllable parameters, such as "ϒ" 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 on the basis of three performance metrics. According to the findings, ε-constraint provides the decision-maker with another Pareto front that has produced more varied solutions in terms of diversity, which enjoys higher quality. Table 7 shows these results for instances 2 and 3. Figure 7 and Figure 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 given 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 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 the MID and QM metrics.

3-7. Comparison of metaheuristic methods

The exact methods are inapplicable solution techniques since the instances of large scale 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 problems of large scale, the results of the exact and metaheuristic methods for the instances of small sizes are compared. As Table 10, indicates, these methods can find optimal Pareto solutions for the problems of small sizes. On the other hand, five instances of medium sizes and five instances of large sizes are generated randomly. Five metaheuristic algorithms are used for solving them, and the comparison metric calculation results are 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, and next come PESA-II, SPEA-II, and NSGA-II, respectively, and MOEA/D is ranked last. Figure 11, reveals that, the lowest MID belongs to NSGA-III, and then, NSGA-II, MOEA/D, SPEA-II, and PESA-II have 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 based on method of 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 maintenance activities execution times, machine availability constraints, and transportation times. With these assumptions, the JSSP seemed more practical in real-world applications. The objective was to minimize makespan and maximum tardiness simultaneously. The original model was converted to one of the auxiliary single-objective crisp type through application of the TH method, with optimal Pareto solutions obtained for the problems of small sizes. On the other hand, the ε-constraint method was used for solving the bi-objective crisp model exactly for this type of problem. It was revealed through comparison of the results of these two methods based on three performance metrics that the results of the ε-constraint method had provided the decision-maker with an additional Pareto front that had produced more varied solutions in terms of diversity, which enjoyed higher quality. Considering two small instances with learning effects and without learning shows the effects of learning on improvement of objective functions. Five different metaheuristics were applied for the problems of large and medium sizes. The validity of the metaheuristic methods for the large-scale problems was verified through comparison of the results of solving the small-sized instances with these methods to the results of the exact methods. On the other hand, the results obtained for the large and medium problems compared based on the three metrics indicated that although these algorithms were ranked as NSGA-III, PESA-II, SPEA-II, NSGA-II, and MOEA/D in terms of Spacing Metric (SM), they are ranked as NSGA-II, NSGA-III, MOEA/D, SPEA-II, and PESA-II based on the other two metrics (i.e., QM and MID), which show 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; Faruqhi,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, pp. 30: 329-340(2019).

7. Mousavipour, S.H; Faruqhi,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.; and Ramezanian, R." 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.; 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.; 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., and Lu, Y. Y." 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; and Zandieh,M ." 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.; Su, and 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.; Wang, Q.; Liu, A.; and Xiao,J." 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-jobshopschedulingproblemswithavailabilityconstraints".InICAPS, 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; Stéphane Dauzère-Pérès.; and S; 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.; 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.; 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), 808–828 (2016).

29. Sakawa, M.; 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.; 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 Research Logistics, 6 (2), pp.131–40 (1959).

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

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

36. Deb, K.; Pratap, A.; Agarwal, S.; and Meyarivan, T." 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.; and Oates, M.J." 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. ; 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.: 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.; 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
Seyedhamed 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 the University of Kurdistan. He was born in 1976. He received BS and MS degrees in Industrial Engineering from Amir Kabir 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 the 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 industrial Engineering from Iran University of science and Technology in 2009. His research interest topics include Scheduling Theory, Combinatorial Optimization, Supply Chain Optimization and metaheuristic algorithms.

Figures:

Figure 1. Gantt chart of validation sample problem
Figure 2. Solution representation
Figure 3. A feasible solution
Figure 4. Generating an initial solution (Sequence of jobs processed on machine $m_i$)
Figure 5. TH optimal Pareto frontier for instance 3
Figure 6. TH optimal Pareto frontier for instance 2
Figure 7. $\varepsilon$-constraint optimal Pareto frontier for instance 2
Figure 8. $\varepsilon$-constraint optimal Pareto frontier for instance 3
Figure 9. Quality measure of metaheuristic algorithms
Figure 10. Spacing metric of metaheuristic algorithms
Figure 11. Mean ideal point distance of metaheuristic algorithms

Tables:

Table 1. Input data of validation sample problem
Table 2. Learning indices and actual jobs processing times of validation sample problem
Table 3. Data generation for random instances and the TH method
Table 4. Specifications of the small-sized instances
Table 5. Positive and negative ideal solutions for the small-sized instances
Table 6. Results of the TH method for instances 2 and 3
Table 7. Results of $\varepsilon$-constraint for instances 2 and 3 given learning effects
Table 8. Results of $\varepsilon$-constraint for instances 2 and 3 regardless of learning effects
Table 9. Comparison of the TH and $\varepsilon$-constraint methods
Table 10. Comparison of the metaheuristic algorithms for the instances of small sizes
Table 11. Comparison of the metaheuristic algorithms for the instances of large and medium sizes.

| $p_{111}$ | $t_{11}$ | $p_{122}$ | $t_{12}$ | $p_{311}$ | $t_{31}$ | $p_{322}$ |
|----------|---------|----------|---------|----------|---------|----------|
| 60       | 8       | 74.8     | 6       | 30       | 1       | 22       |
| $M_1$    | $M_2$   | $M_3$    |         |          |         |          |
| $p_{221}$ | $t_{21}$ | $p_{212}$ | $t_{21}$ |          |         |          |
| 50       | 10      | 44       | 8       |          |         |          |

Figures:
Figure 2.

\[
\begin{array}{ccc}
3 & 2 & 1 \\
1 & 1 & 2 \\
2 & 3 & 2 \\
3 & 3 & 1
\end{array}
\]

Figure 3.

\[
\begin{array}{c|c|c}
0 & 0 & 1 \\
0 & 1 & 0 \\
1 & 0 & 0
\end{array}
\quad
\begin{array}{c|c|c}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{array}
\quad
\begin{array}{c|c|c}
0 & 1 & 0 \\
0 & 0 & 1 \\
1 & 0 & 0
\end{array}
\]

Figure 4.
Figure 5.

Figure 6.
Tables:
| Normal processing times: | Sequence of jobs processing: |
|-------------------------|-----------------------------|
| \( f_1 \) | \( f_2 \) | \( j_1 : (m_1 - m_2 - m_3) \) | \( j_2 : (m_2 - m_1 - m_3) \) |
| \( m_1 \) | 60 | 85 |
| \( m_2 \) | 50 | 45 |
| \( m_3 \) | 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 | \( m_1 \) | \( m_2 \) | \( m_3 \) |
|-----------------------|----------|----------|----------|
| \( tp_{ij} \)      | \( m_1 - j_1 \) | 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 |

| Learning indices: | Actual processing times (\( p_{ijk} \)): |
|-------------------|---------------------------------------|
| 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 |
| Incompressibility index | \( M_{ij} = 0.25 \) | \( m_2 - j_1 \) | 50 | 44 |
|                      |                                        | \( m_2 - j_2 \) | 45 | 39.6 |
|                      |                                        | \( m_3 - j_1 \) | 30 | 26.4 |
|                      |                                        | \( m_3 - j_2 \) | 25 | 22 |
Table 3.

| Notation | Parameter                                      | Value                                      |
|----------|-----------------------------------------------|--------------------------------------------|
| $m$      | Number of machines                            | $\{3 \ldots 18\}$                         |
| $n$      | Number of jobs                                | $\{2 \ldots 14\}$                         |
| $P_{ij}$ | 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 | $\{1 \ldots 6\}$                         |
| $\alpha_i$ | Learning index of machine $m_i$            | Uniform distribution (-0.9, -0.1)         |
| $\varphi_j$ | Learning index of job $j$                | Uniform distribution (-0.9, -0.1)         |
| $\omega_i$ | 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 $\{1, 2, \ldots, n - 1\}$ |
| $tp_{ijr}$ | Transportation time                        | Uniform distribution (5, 150)             |
| $M_{ij}$ | DeJong’s parameter                           | $\{0.25, 0.5\}$                          |
| $\gamma$ | Coefficient of compensation in the TH method | $\{0, 0.1, \ldots, 1\}$                  |
| $\theta_h$ | Importance level of the $h^{th}$ 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

| Instance No. | Representation | No. of Constrains | No. of decision variables | Number of machines | Number of jobs | Number of maintenance activities for each machine |
|--------------|----------------|-------------------|---------------------------|--------------------|---------------|-----------------------------------------------|
| 1            | $3 \times 2 \times 1$ | 57                | 34                        | 3                  | 2             | 1                                             |
| 2            | $3 \times 3 \times 2$ | 125               | 67                        | 3                  | 3             | 2                                             |
| 3            | $3 \times 4 \times 2$ | 235               | 112                       | 3                  | 4             | 2                                             |
| 4            | $4 \times 4 \times 2$ | 326               | 144                       | 4                  | 4             | 2                                             |
| 5            | $3 \times 5 \times 2$ | 399               | 169                       | 3                  | 5             | 2                                             |
Table 5.

| Instance No. | Representation | Z1 PIS | Z1 NIS | Z2 PIS | Z2 NIS |
|--------------|----------------|--------|--------|--------|--------|
| 1            | 3 × 2 x 1      | 284.64 | 284.64 | 242    | 242    |
| 2            | 3 × 3 x 2      | 164.597| 193.349| 18.752 | 44.597 |
| 3            | 3 × 4 x 2      | 233.52 | 240.89 | 62.89  | 71.26  |
| 4            | 4 × 4 x 2      | 271.43 | 324.5  | 171.18 | 178.83 |
| 5            | 3 × 5 x 2      | 278.74 | 315.53 | 130    | 133.74 |

Table 6.

| Υ  | Z1  | Z2  | μ1  | μ2  | Z1  | Z2  | μ1  | μ2  |
|----|-----|-----|-----|-----|-----|-----|-----|-----|
| 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 |

Table 7.

| 3 × 3 x 2 | 3 × 4 x 2 |
|-----------|-----------|
| Z1        | Z1        |
| 193.349   | 18.752    |
| 193.349   | 21.624    |
| 193.349   | 24.469    |
| 193.349   | 27.367    |
| 186.954   | 27.367    |
| 186.954   | 30.239    |
| 186.954   | 33.111    |
| 186.954   | 35.982    |
| 186.954   | 38.854    |
| 164.597   | 44.579    |
| Z2        | Z2        |
| 240.89    | 62.89     |
| 240.89    | 63.820    |
| 240.89    | 64.750    |
| 240.89    | 65.860    |
| 240.89    | 66.610    |
| 240.89    | 67.540    |
| 240.89    | 68.470    |
| 240.89    | 69.400    |
| 235.81    | 70.330    |
| 233.52    | 71.260    |
Table 8.

|        | $3 \times 3 \times 2$ | $3 \times 4 \times 2$ |
|--------|-----------------------|-----------------------|
| $Z_1$  | $Z_2$                | $Z_1$                | $Z_2$                |
| 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                  |

Table 9

| Representation | $\varepsilon$ – constraint method | TH method |
|----------------|-----------------------------------|-----------|
|                | SM  | MD  | QM | SM  | MD  | QM |
| $3 \times 2 \times 1$ | 7.8 | 518.77 | 1 | 7.8 | 518.77 | 1 |
| $3 \times 3 \times 2$ | 9.72 | 184.61 | 0.75 | 3.2 | 191.58 | 0.75 |
| $3 \times 4 \times 2$ | 3.689 | 246.48 | 0.75 | 4.28 | 246.33 | 0.75 |
| $4 \times 4 \times 2$ | 3.65 | 341.09 | 1 | 0 | 358.5 | 0.75 |
| $3 \times 5 \times 2$ | 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
Table 11

| Representation | NSGA – II | NSGA – III | SPEA – II | PESA – II | MOEA/D |
|----------------|-----------|------------|-----------|-----------|--------|
|                | SM       | MID       | QM | SM       | MID       | QM | SM       | MID       | QM | SM       | MID       | QM | SM       | MID       | QM |
| 3 × 2 × 1      | 7.8      | 518.77    | 1  | 7.8      | 518.77    | 1  | 7.8      | 518.77    | 1  | 7.8      | 518.77    | 1  |
| 3 × 3 × 2      | 9.72     | 184.61    | 1  | 9.72     | 184.61    | 1  | 9.72     | 184.61    | 1  | 9.72     | 184.61    | 1  |
| 3 × 4 × 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 × 2      | 3.65     | 341.09    | 1  | 3.65     | 341.09    | 1  | 3.65     | 341.09    | 1  | 3.65     | 341.09    | 1  |
| 3 × 5 × 2      | 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 |

| Representation | NSGA – II | NSGA – III | SPEA – II | PESA – II | MOEA/D |
|----------------|-----------|------------|-----------|-----------|--------|
|                | SM       | MID       | QM | SM       | MID       | QM | SM       | MID       | QM | SM       | MID       | QM | SM       | MID       | QM |
| 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 |
| 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 |
| 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    |
| 8 × 6 × 3      | 3.5      | 827.02    | 0.25 | 0.5      | 825.0      | 0.5  | 4.8      | 1004.82   | 0.125 | 3.5      | 967.0      | 0.25 | 4        | 904.0      | 0.25 |
| 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     | 0    |
| 10 × 9 × 3     | 31.78    | 2259.98   | 0.75 | 7.6      | 2472.43   | 0    | 1.4      | 2573.0    | 0    | 0        | 2594.6     | 0    | 0        | 2269.17    | 0.25 |
| 12 × 10 × 5    | 20       | 7253.8    | 0    | 0        | 7290.42   | 0    | 0        | 6745.4    | 1    | 0        | 7494.5     | 0    | 0        | 7076.0     | 0    |
| 14 × 8 × 6     | 6.5      | 4752.48   | 0.28 | 15.5     | 4692.30   | 0.28 | 24       | 4624.53   | 0.14 | 30       | 4662.98    | 0.14 | 29.4     | 4673.61    | 0.14 |
| 16 × 12 × 6    | 9.5      | 8124.08   | 0.66 | 0        | 8277.24   | 0.33 | 18       | 8513.43   | 0    | 44       | 8779.0     | 0    | 88.5     | 8759.79    | 0    |
| 18 × 14 × 6    | 0        | 4107.33   | 0    | 3862.33  | 0        | 4702.0    | 0    | 4137.5   | 0.33 | 0        | 4170.3     | 0    | Mean     | 3218.73    | 0.46 |
| Mean           | 11.13    | 3223.18   | 0.33 | 10.87    | 3308.94   | 0.15 | 10.34    | 3510.24   | 0.08 | 15.03    | 3269.0     | 0.17 |