Journal of Industrial
Engineering International
a SpringerOpen Journal

ORIGINAL RESEARCH   **Open Access**

Minimizing the total tardiness and makespan in an open shop scheduling problem with sequence-dependent setup times

Samaneh Noori-Darvish[1] and Reza Tavakkoli-Moghaddam[2*]

Abstract

We consider an open shop scheduling problem with setup and processing times separately such that not only the setup times are dependent on the machines, but also they are dependent on the sequence of jobs that should be processed on a machine. A novel bi-objective mathematical programming is designed in order to minimize the total tardiness and the makespan. Among several multi-objective decision making (MODM) methods, an interactive one, called the TH method is applied for solving small-sized instances optimally and obtaining Pareto-optimal solutions by the Lingo software. To achieve Pareto-optimal sets for medium to large-sized problems, an improved non-dominated sorting genetic algorithm II (NSGA-II) is presented that consists of a heuristic method for obtaining a good initial population. In addition, by using the design of experiments (DOE), the efficiency of the proposed improved NSGA-II is compared with the efficiency of a well-known multi-objective genetic algorithm, namely SPEA-II. Finally, the performance of the improved NSGA-II is examined in a comparison with the performance of the traditional NSGA-II.

Keywords: Open shop scheduling, Total tardiness, Makespan, Sequence-dependent setup times, NSGA-II, SPEA-II

Background

An open shop scheduling problem (OSSP) is a kind of shop scheduling such that the operations can be executed in any order. The open shop allows much flexibility in scheduling, but it is difficult to develop rules that give an optimum sequence for every problem (Sule 1997). This problem is a class of NP-hard ones (Gonzalez and Sahni 1976. In this paper, we consider a special feature in OSSPs, called sequence-dependent setup time. The process of preparing machines between jobs is considered as a setup. In fact, setup times affect on the completion time of each job. As a result, they also affect on tardiness, earliness and other important criteria. Allahverdi *et al.* (2008) surveyed the literature of setup times or costs in scheduling problems. They classified scheduling problems into those with batching and non-batching considerations as well as sequence-independent and sequence-dependent setup times. According to the

technology and the kind of machines used in a work environment and variety of products, setup times can be dependent on both machines and the sequence of jobs that should be processed on a machine.

In many practical production systems (e.g., chemical, printing, pharmaceutical and automobile manufacturing), the setup tasks (i.e., cleaning up and changing tools) are sequence-dependent (Zandieh *et al.*, 2006 and Roshanaei *et al.*, 2009). Low and Yeh (2008) addressed an open shop scheduling problem as a 0–1 integer programming model with the objective of minimizing the total job tardiness along with some assumptions, such as sequence-independent setup and sequence-dependent removal times. They proposed some hybrid genetic-based heuristics to solve the problem in an acceptable computing time. Mosheiov and Oron (2008) addressed batch scheduling problems on an open-shop with *m* machines and *n* jobs. Identical processing time jobs, machine-independent and sequence-independent setup times are the main assumptions of their problems. The objectives are to minimize the makespan and minimize

* Correspondence: tavakoli@ut.ac.ir
[2]Department of Industrial Engineering, College of Engineering, University of Tehran, PC: 14399-57131, Tehran, Iran
Full list of author information is available at the end of the article

Springer

flow time. They proposed an $O(n)$ time algorithm for the flow time minimization problem.

To achieve Pareto-optimal sets for medium to large-sized open shop problems using efficient meta-heuristic methods can be necessary and helpful. Naderi *et al.* (2011) considered an open shop with a set of parallel machines at each stage to minimize the total completion times. They proposed a mixed-integer linear programming (MILP) model for this problem. Moreover, they applied a memetic algorithm to solve the problem. Ahmadizar et al. (2010) addressed a stochastic group shop scheduling problem with known distributions for random release dates and random processing times. They formulated a stochastic programming problem and solved it by the use of an approach being a hybrid of an ant colony optimization (ACO) algorithm and a heuristic algorithm to minimize the expected makespan. Zhang and van de Velde (2010) considered an on-line two-machine open shop scheduling problem with time lags between the completion time and the start time of two consecutive operations of any job. They developed and analyzed the performance of a greedy algorithm to minimize the makespan. Mastrolilli et al. (2010) dealt with a concurrent open shop and proposed a primal–dual 2-approximation algorithm to minimize the total weighted completion times. They also considered several natural linear programming relaxations for the problem.

Fei et al. (2010) considered a weekly surgery schedule in an operating theatre. The objectives are to maximize the utilization of the operating rooms, minimize the overtime cost in the operating theatre, and minimize the unexpected idle time between surgical cases. They proposed a column-generation-based heuristic (CGBH) procedure and a hybrid genetic algorithm (HGA) for solving the planning problem as a set-partitioning integer-programming model and the daily scheduling problem as a 2-stage hybrid flow-shop problem, respectively. Matta (2009) developed two original mixed-integer programming (MIP) models (i.e., time-based model and sequence-based model) for the proportionate multiprocessor open shop scheduling problem and proposed a genetic algorithm (GA) to schedule the shop with the objective of the makespan minimization. Seraj and Tavakkoli-Moghaddam (2009) proposed a TS method to solve a new bi-objective mixed-integer mathematical programming model for an OSSP. This model seeks to minimize the mean tardiness and the mean completion time. Panahi and Tavakkoli-Moghaddam (2011) proposed a hybrid method based on multi-objective simulated annealing (SA) and ant colony optimization (ACO) to solve an open shop scheduling problem that minimizes bi-objectives, namely makespan and total tardiness. They

compared their computational results with a well-known multi-objective genetic algorithm, namely NSGA II.

By considering the previous studies, we can conclude that there is only one paper considered the sequence-dependent setup time as an assumption for a single-objective OSSP without any mathematical model and written by Roshanaei *et al.* (2009). Therefore, in this paper, a bi-objective mixed-integer linear programming (MILP) model is designed with the sequence-dependent setup time as a constraint for the OSSP. To solve medium to large-sized problems, the well-known NSGA-II proposed by Deb *et al.* (2002) is improved by using a heuristic method in order to achieve good approximate Pareto-optimal frontiers.

The rest of this paper is organized as follows. The mathematical programming is presented in Section 2. The interactive multi-objective decision making approach is presented in Section 3. Section 4 elaborates the proposed improved NSGA-II. The numerical examples, computational results and performance analysis are given in Section 5. Finally, Section 6 includes the conclusion remark.

Mathematical Programming

The OSSP considered in this study includes n jobs to be processed on at most m machines. We extend the mathematical model proposed by Low and Yeh (2008) by changing it to a bi-objective model with sequence-dependent setup times.

Problem assumptions

The main assumptions of the presented model are as follows.

- Each job is to be processed on at most m machines.
- The processing sequence of each job is immaterial.
- A job is not processed by more than one machine simultaneously.
- Each machine processes at most one operation at a time.
- All jobs are available at time 0.
- No preemption is allowed.
- Machine-dependent and sequence-dependent setup times are considered for each operation.

Notations

The indices, parameters and decision variables used to formulate the mathematical model are introduced below.

Indices

N Set of jobs to be processed, $N=\{1, 2, \ldots, n\}; |N| = n$
L Set of machines, $L = \{1, 2, \ldots, m\}; |L| = m$.

i,k,g Job indices, $(i,k,g= 0,1,\ldots,n)$, where job 0 is a dummy job.
j,h Machine indices, $(j,h=1,2,\ldots,m)$.

Parameters

M A large positive number.
O_{ij} Operation of job i on machine j ;$\forall i \in N; \forall j \in L$.
S_{kij} Setup time of job i on machine j immediately after job k.
p_{ij} Processing time of job i on machine j.
d_i Due date of job i.
H_j Dependent set up time matrix of machine j.

$$H_j = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \cdots \\ n \end{array} \begin{array}{c} 0\,1\,2\ldots n \\ \begin{bmatrix} - & S_{01j} & S_{02j} & \ldots & S_{0nj} \\ - & - & S_{12j} & \ldots & S_{1nj} \\ - & S_{21j} & - & \ldots & S_{2nj} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ - & S_{n1j} & S_{n2j} & \cdots & - \end{bmatrix} \end{array}.$$

Decision variables

Ts_{ij} Starting time of a setup task for operation O_{ij}.
T_i Tardiness of job i.
C_{max} Makespan.

$$Y_{ijh} = \begin{cases} 1 & if \ 0_{ij} \ precedes \ 0_{ih} \ for \ job i \\ 0 & Otherwise \end{cases}$$
$$\forall i \in N; \forall j, h \in L, j \neq h$$

$$X_{kij} = \begin{cases} 1 & if \ 0_{kj} \ precedes \ 0_{ij} \ on \ machine \ j \\ & Otherwise \end{cases}$$
$$\forall k \in N \cup \{0\}; \forall i \in N, i \neq k; \forall j \in L$$

$$Z_{kij} = \begin{cases} 1 & if \ 0_{kj} \ precedes \ 0_{ij} \ on \ machine \ j \\ 0 & Otherwise \end{cases}$$
$$\forall k \in N \cup \{0\}; \forall i \in N, i \neq k; \forall j \in L$$

Mathematical model

As mentioned at the beginning of Section 2, the model designed in this paper is developed by extending and modifying the mathematical model proposed by Low and Yeh (2008). In this new model, the sequence-dependent setup time is used instead of the sequence-independent setup time considered in the traditional model. The makespan is added to the model as another criterion along with the total tardiness for the minimization purpose. According to these changes and for the validity of the extended model, all of the existing constraints are changed and four new constraints are added to the model. Thus, the bi-objective MILP (BOMILP) model is formulated. It should be noted that S_{0ij} is the setup time of job i on machine j when i is the first job on machine j. Moreover, if job i is the

first job on machine j, then $Z_{0ij} = 1$. The proposed model is as follows:

$$\text{Min } Z_1 = \sum_{i=1} T_i \tag{1}$$

$$\text{Min } Z_2 = C_{max} \tag{2}$$

s.t.

$$Ts_{ij} + \sum_{k=0,k \neq i}^{n} (S_{kij} \times Z_{kij}) + p_{ij} \leq C_{max} ; \forall i \in N; \forall j \in L \tag{3}$$

$$Ts_{ij} + \sum_{k=0,k \neq 0}^{n} (S_{kij} \times Z_{kij}) + p_{ij} - M(1 - Y_{ijh} \leq Ts_{ih});$$
$$\forall i \in N; \forall j, h \in L, j \neq h \tag{4}$$

$$Ts_{ih} + \sum_{k=0,k \neq i}^{n} (S_{kih} \times Z_{kih}) + p_{ih} - M \times Y_{ijh} \leq Ts_{ij};$$
$$\forall i \in N; \forall j, h \in L, j \neq h \tag{5}$$

$$Ts_{ij} + \sum_{k=0,k \neq i \neq g}^{n} (S_{kij} \times Z_{kij}) + p_{ij} - M(1 - X_{igj}) \leq Ts_{gj};$$
$$\forall i,g \in N, i \neq g; \forall j \in L \tag{6}$$

$$Ts_{gj} + \sum_{k=0,k \neq i \neq g}^{n} (S_{kij} \times Z_{kij}) + p_{gj} - M \times X_{igj} \leq Ts_{ij};$$
$$\forall i,g \in N, i \neq g; \forall j \in L \tag{7}$$

$$Ts_{ij} + \sum_{k=0,k \neq i}^{n} (S_{kij} \times Z_{kij}) + p_{ij} - T_i; \forall_i \in N; \forall j \in L \tag{8}$$

$$Y_{ijh} + Y_{ihj} = 1 ; \forall i \in N; \forall j, h \in L, j \neq h \tag{9}$$

$$X_{igj} + X_{gij} = 1; \forall i, g \in N, i \neq g; \forall j \in L \tag{10}$$

$$X_{kij} - Z_{kij} \geq 0 ; \forall k \in N \cup \{0\}; \forall i \in N, i \neq k; \forall j \in L \tag{11}$$

$$X_{kij} + Z_{ikj} \leq 1 ; \forall i, k \in N, i \neq k; \forall j \in L \tag{12}$$

$$\sum_{i=1,i \neq k}^{n} Z_{kij} \leq 1 ; \forall k \in N \cup \{0\}; \forall j \in L \tag{13}$$

$$\sum_{k=0,k \neq i}^{n} Z_{kij} = 1 ; \forall i \in N; \forall j \in L \tag{14}$$

$$Ts_{ij} \geq 0 ; \forall i \in N; \forall j \in L \tag{15}$$

$$T_i \geq 0 ; C_{max} \geq 0 ; \forall i \in N \tag{16}$$

$$X_{kij}, Z_{kij} \in \{0,1\}; \forall k \in N \cup \{0\}; \forall i \in N, i \neq k; \forall j \in L \tag{17}$$

$$Y_{ijh} \in \{0,1\}; \forall i \in N; \forall j, h \in L, j \neq h \tag{18}$$

Two objective functions (i.e., total tardiness and makespan) are shown by Eqs. (1) and (2).Constraint (3) describes the makespan. Constraints (4) and (5) express the relationship between two operations of job i that do not require being consecutive. The starting time for setup task of operation O_{ih} is greater than or equal to the completion time of operation O_{ij}. Constraints (6) and (7) state the operational sequence of the operations which are processed on the same machine and do not require to be consecutive. The setup task of operation O_{gj} cannot be started until machine j has finished the processing task of operation O_{ij}. Constraint (8) describes the tardiness for job i. Constraint (9) expresses the order of any two operations of a job; if $Y_{ijh} = 1$ then $Y_{ihj} = 0$; otherwise, $Y_{ijh} = 0$ and $Y_{ihj} = 1$. Constraint (10)expresses the order of any operation pairs (O_{ij}, O_{gj}) on the same machine j; if $X_{igj} = 1$ then $X_{gij} = 0$, otherwise, $X_{igj} = 0$ and $X_{gij} = 1$. Constraints (11) and (12) describe the relationship between X_{kij} and Z_{kij}; if $X_{kij} = 1$, then $Z_{kij} = 1$ or 0; otherwise, $X_{kij} = 0$ and $Z_{kij} = 0$, by considering the dummy job 0. If $X_{kij} = 1$ then $Z_{ikj} = 0$; otherwise, $X_{kij} = 0$ and $Z_{ikj} = 1$ or 0; in this case, the dummy job 0 is not considered. Because the dummy job cannot be located after any job, it is only used for characterizing the first job on each machine to apply the corresponding relative setup time. Constraint (13) indicates that there is at most one job which can be processed immediately after job k on machine j. if job k is the last job on machine j, then $Z_{kij} = 0$. Constraint (14) indicates that there is only one job that can be processed immediately before job i on machine j by considering the dummy job 0. Constraint (15) expresses that all jobs should be available for scheduling at time 0.Constraints (16) to (18) define the continuous and binary decision variables, respectively.

Interactive MODM Approach

To solve the original bi-objective decision making (BODM) problem, an interactive fuzzy programming solution method, called the TH method proposed by Torabi and Hassini (2008), is used. This method is applied to achieve the Pareto-optimal solutions of the presented bi-objective crisp model. Torabi and Hassini (2008) proved that the TH method obtains efficient solutions for the original multi-objective model. According to the characteristics of the given problems, the steps of the TH method are as follows.

Algorithm1: TH method
Step 1
Calculate the positive ideal solution (PIS) and the negative ideal solution (NIS) for each objective function by solving the corresponding MILP model given below.

$$Z_1^{PIS} = \min \sum_{i=1}^{n} T_i \qquad \text{s.t. } X \in F_{(x)} \tag{19}$$

$$Z_1^{NIS} = \max \sum_{i=1}^{n} T_i \qquad \text{s.t. } X \in F_{(x)} \tag{20}$$

$$Z_2^{PIS} = \min C_{max} \qquad \text{s.t. } X \in F_{(x)} \tag{21}$$

$$Z_2^{NIS} = \max C_{max} \qquad \text{s.t. } X \in F_{(x)} \tag{22}$$

where X is a feasible solution vector consists of all of the continuous and binary variables in the original model and $F_{(x)}$ denotes the feasible area consists of Constraints (3) to (18).

Obtainment of the above ideal solutions requires solving four mixed-integer linear programs. In order to reduce the computational time and complexity, we can calculate the PIS for each objective function by solving the corresponding MILP models given in Eqs. (19) and (21). Then, the negative ideal solutions can be determined by the use of the following heuristic rule:

$$Z_h^{NIS} = Max(Z_h(x_k^*)); \ h = 1, 2, k = 1, 2 \tag{23}$$

It is noted that x_h^* and $Z_h(x_h^*)$ denote the decision vector associated with the PIS of the h-th objective function and the corresponding value of the h-th objective function, respectively (Torabi and Hassini, 2008). The related results are shown in Table 1.

Step 2
For each objective function, determine a linear membership function by:

$$\mu_{Z_1}(X) = \begin{cases} 1, & Z_1 < Z_1^{PIS} \\ \dfrac{Z_1^{NIS} - Z_1}{Z_1^{Nis} - Z_1^{PIS}}, & Z_1^{PIS} \leq Z_1 \leq Z_1^{NIS} \\ 0, & Z_1 > Z_1^{NIS} \end{cases} \tag{24}$$

$$\mu_{Z_2}(X) = \begin{cases} 1, & Z_2 < Z_2^{PIS} \\ \dfrac{Z_2^{NIS} - Z_2}{Z_2^{Nis} - Z_2^{PIS}}, & Z_2^{PIS} \leq Z_2 \leq Z_2^{NIS} \\ 0, & Z_2 > Z_2^{NIS} \end{cases} \tag{25}$$

Figure 1 illustrates the graph of these membership functions.

Table 1 Payoff results

	Z_1	Z_2
X_1^*	Z_1^{PIS}	Z_2^{NIS}
X_2^*	Z_1^{NIS}	Z_2^{PIS}

Step 3

Transform the BOMILP model into an equivalent single-objective MILP using the following auxiliary formulation.

$$\text{Max } \lambda_{(X)} = \gamma\lambda_0 + (1 - \gamma)\sum_h \theta_h \mu_{Z_h}(X) \qquad (26)$$

s.t.

$$\lambda_0 \le \mu_{Z_h}(X), \quad h = 1, 2 \qquad (27)$$

$$X \in F_{(X)} \qquad (28)$$

$$\lambda_0, \gamma \in [0, 1] \qquad (29)$$

According to the two objective functions considered in the problem, Constraint (27) can be written by:

$$Z_1^{NIS} - \sum_{i=1}^{n} T_i \ge \lambda_0(Z_1^{NIS} - Z_1^{PIS}) \qquad (30)$$

$$Z_2^{NIS} - C_{max} \ge \lambda_0(Z_2^{NIS} - Z_2^{PIS}) \qquad (31)$$

where $\mu_{Z_h}(X)$ is the satisfaction degree of the h-th objective function and $\lambda_0 = \min_h\{\mu_{Z_h}(X)\}$ is the minimum satisfaction degree of objectives. Also, θ_h denotes the importance level of the h-th objective function such that $\sum_h \theta_h = 1, \theta_h > 0$. The θ_h parameters are determined linguistically by the decision maker based on her preference. Moreover, γ is the coefficient of compensation. By changing the values of this parameter in the interval [0,1], the TH method can obtain both unbalanced and balanced compromised solutions. It means that for higher values of γ, the solution method results bigger

Figure 1 Linear membership function for Z_1 (Z_2).

lower bound for the satisfaction degrees of objectives (λ_0) for a given sample example. These solutions are balanced compromised solutions. These kinds of solution can be more appropriate when the importance levels of all objective functions are equal. On the other hand, for lower values of γ, the solution method resulting solutions with bigger satisfaction degrees for some objectives with higher importance levels than others. These solutions are unbalanced compromised solutions. These kinds of solutions can be more appropriate when the importance levels of the objective functions are different.

Step 4: Solve equivalent single-objective MILP model using the given coefficients (θ_h, γ). If the decision maker is satisfied with obtained efficient compromised solution, then stop; otherwise, change the value of some controllable parameters and then go to Step 2.

A set of non-dominated solutions as the Pareto-optimal solutions can be obtained for the BOMILP model by changing the values of controllable parameters of the MILP model given in Eqs. (26) to (31). As mentioned at the beginning of Section 3, it is proved that the TH method achieves efficient solutions for multi-objective optimization problems. It means that each optimal solution of the single-objective model given in Eqs. (26) to (31) is a Pareto-optimal solution of the original BOMILP model.

Proposed Method

The OSSP studied in this paper is the NP-hard problem. Thus, to solve medium to large-sized problems considered in this paper, an improved non-dominated sorting genetic algorithm II (NSGA-II) is proposed. Noori-Darvish and Tavakkoli-Moghaddam (2011) used the original NSGA-II for an OSSP. NSGA-II belongs to a class of multi-objective evolutionary algorithms (MOEAs). In most problems, it is able to find much better spread of solutions and better convergence near the true Pareto-optimal front compared to two other elitist MOEAs, namely Pareto-archived evolution strategy (PAES) and strength-Pareto evolutionary algorithm (SPEA), which pay special attention to creating a diverse Pareto-optimal front (Deb *et al.*, 2002).

Solution Representation

In our proposed method, a chromosome is an operation-based array or permutation list that has been a typical solution representation studied in OSSPs. As illustrated in Figure 2, the permutation list is a single-row array consisting of $n \times m$ operations. By using this encoding scheme, the basic assumptions of OSSPs are satisfied, in which the processing sequence of each job is immaterial and a job is not processed more than once by one machine. In this representation, operations are
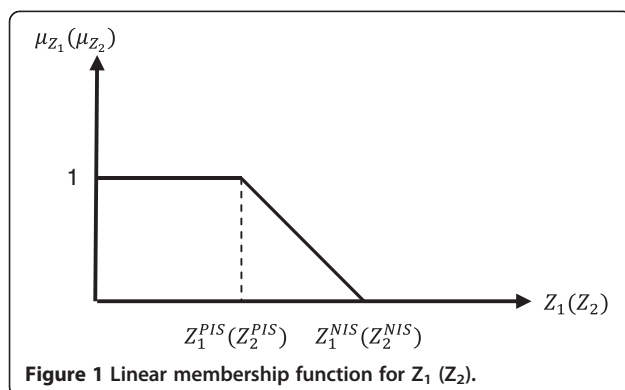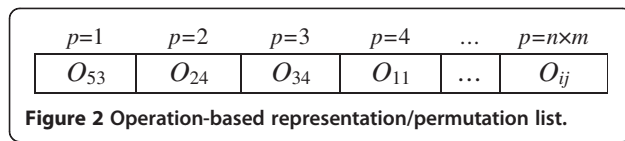
$p=1$	$p=2$	$p=3$	$p=4$	...	$p=n\times m$
O_{53}	O_{24}	O_{34}	O_{11}	...	O_{ij}

Figure 2 Operation-based representation/permutation list.

listed in the relative order by which they are scheduled. p denotes the position of an operation in the list.
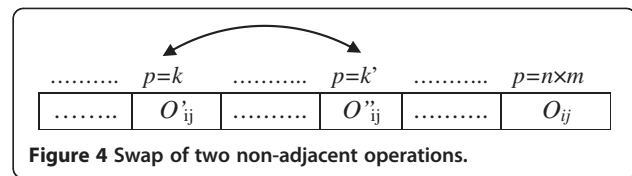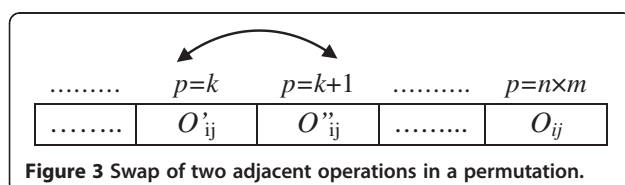
To decode the list presented in Figure 2 and obtain its equivalent solution in the solution space, we start from the beginning of the list and schedule the first operation. Then, according to the corresponding order of operations in the list, we schedule other operations one by one. It means, job 5 is first scheduled on machine 3. Then, job 2 is scheduled on machine 4 and so on. According to the rule that each operation is scheduled at the earliest time, there are two variables, namely 1) Max $\{C_{ij}\}$ for all i as the maximum completion time of jobs on machine j and 2) Max $\{C_{ij}\}$ for all j as the maximum completion time of job i. The starting time and completion time of each operation are computed by:

$$Ts_{ij} = \text{Max}\{0, \underset{\forall i}{\text{Max}}\{C_{ij}\}, \underset{\forall j}{\text{Max}}\{C_{ij}\}\} \qquad (32)$$

$$C_{ij} = Ts_{ij} + \sum_{k=0,k\neq i}^{n}(S_{kij} \times Z_{kij}) + p_{ij} \qquad (33)$$

Initialization

A good initial population can help an evolutionary algorithm to start multi-objective optimization from good solutions in the solution space and have a better performance that will be discussed in the next section. In our algorithm, a heuristic method is applied to generate the initial set of solutions or population. It first constructs a set of random sequences of operations as chromosomes (as many as the population size). By using the swapping of two adjacent or non-adjacent operations (see Figures 3 and 4), it generates all the feasible sequences in the neighborhood of each permutation list. For each chromosome, the best feasible solution obtained by these neighborhood searches is a member of the initial population. According to the dominance concept, the best solution is an efficient solution of the Pareto-optimal frontier. The set of Pareto-optimal solutions dominate any other solutions in the feasible area and improve all the objective functions simultaneously.

....	$p=k$	....	$p=k+1$	....	$p=n\times m$
....	O'_{ij}	O''_{ij}	....		O_{ij}

Figure 3 Swap of two adjacent operations in a permutation.

....	$p=k$	....	$p=k'$	....	$p=n\times m$
....	O'_{ij}	....	O''_{ij}	....	O_{ij}

Figure 4 Swap of two non-adjacent operations.

It should be noted that $k,k' = 1,2,\ldots, n\times m$, as shown in Figures 3 and 4.

Crossover

A procedure proposed by Low and Yeh (2008) is used for the crossover operator.

Algorithm 2: Crossover operator

Step 1 Select two cut points randomly along the positions of the strings for each pair of parent strings (A and B). So, each permutation list is divided into three sections, called substring 1, substring 2 and substring 3.

Step 2 Exchange substring 1 of parent string A and substring 3 of parent string B. Similarly, exchange substring 3 of parent string A and substring 1 of parent string B. Do not change the elements of substring 2 of each parent string. Therefore, two proto-children are generated which are named A_0 and B_0.

Step 3 Legalize two generated offsprings A_0 and B_0. In order to do this, remove the elements of substring 1 and substring 3 of offspring string A_0, which are the same as substring 2, and then replace them with corresponding elements of substring 2 of offspring string B_0. Also, do the same procedure for offspring string B_0. Thus, two offspring strings (i.e., A_0 and B_0) are produced.

Mutation

As depicted in Figure 5, the insertion operator is applied for the mutation operator. In the permutation shown in this figure, $k,k'= 1,2,\ldots,n\times m$.

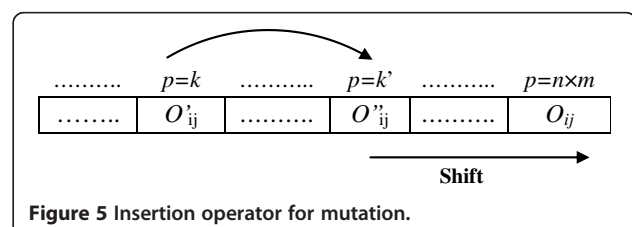|	$p=k$	....	$p=k'$	....	$p=n\times m$
....	O'_{ij}	....	O''_{ij}	....	O_{ij}

Shift

Figure 5 Insertion operator for mutation.

Table 2 Parameters of the NSGA-II

Population size	Maximum number of generations	Crossover rate	Mutation rate
100	300	0.95	0.02

Elitism and selection operator

Deb et al. (2002) designed an elitist NSGA including "Fast non-dominated sorting approach" and "Diversity Preservation" for solving multi-objective problems. In order to rank solutions (individuals) and sort them into different non-dominated frontiers, an approach called "Fast non-dominated sorting" is applied. By the use of this approach, a domination count (n_p) is calculated for each solution to specify its non-dominated frontier. The domination count of all solutions in the first non-dominated frontier ($i_{rank}=1$) is equal to zero. At the end of the multi-objective optimization process, solutions with $n_p=0$, dominate all other solutions in the solution space. It means they are the best global solutions. This approach results better convergence near the true Pareto-optimal frontier.

Along with the convergence to the Pareto-optimal set, it is also desired that an EA maintains a good spread of solutions in the obtained set of solutions (Deb et al., 2002). An operator called crowded-comparison operator is proposed for "Diversity Preservation". By considering the value of crowding distance of each solution ($i_{distance}$) in its frontier, this operator guides the selection process at the various stages of the algorithm toward a uniformly spread-out Pareto-optimal frontier. When the improved NSGA-II is iterated as many as the pre-specified iterations (i.e., maximum number of generations), the multi-objective optimization process is terminated (Noori-Darvish and Tavakkoli-Moghaddam, 2011).

Results and discussion

Several numerical examples in small to large sizes are generated randomly by the use of a classic approach of the literature in order to examine and analyze the validity and efficiency of the mathematical model presented in Section 2, the TH method presented in Section 3, and the performance of the improved NSGA-II presented in Section 4. The small-sized problems are solved exactly by the use of the Lingo 9 software and the results of the TH method are analyzed. If the number of jobs and the number of machines are more than 4, the sizes of the problems are medium to large. In

Table 3 Parameters of the SPEA-II

Population size	Maximum number of generations	Selection rate	Crossover rate	Mutation rate
100	400	0.8	0.8	0.2

Table 4 Values of parameters for numerical instances

Numerical instances	*R*	*T*
a	1	0.8
b	0.6	0.6
c	0.2	0.4

these conditions, even after several hours of running, Lingo cannot yield an optimal solution in a reasonable time. Thus, the improved NSGA-II is applied to solve these kinds of problems. Finally, the efficiency of this algorithm is first compared with the efficiency of a well-known multi-objective genetic algorithm, namely SPEA-II, by using the design of experiments (DOE). Then, the proposed algorithm is compared with the traditional NSGA-II. These algorithms are coded in Turbo C++4.5 on a PC (Main board P4, CPU E5200 2.5GHz, 6M Cache, RAM 2GB BUS 800).

SPEA-II

The strength Pareto evolutionary algorithm II (SPEA-II) proposed by Zitzler, et al. (2001) is specially designed for multi-objective optimization problems. This algorithm includes some special features (i.e., fitness assignment strategy) considering a number of solutions that dominate each individual and a number of solutions that each individual dominates the "nearest neighbor density estimation technique" that yields a density value for each solution, and the "archive truncation method" that preserves a specified number of solutions in the external non-dominated archive.

In our SPEAII, the solution representation is the permutation list and the initial population is generated randomly. Moreover, the binary tournament selection procedure is applied. When the SPEA-II is iterated as many as the pre-specified number of iterations (i.e., maximum number of generations), the multi-objective optimization process is terminated.

Generating numerical examples

In this section, we use a classical approach in the literature proposed by Loukil et al. (2005) to generate several numerical examples of small to large sizes

Table 5 Values of positive and negative ideal solutions

Z_2		Z_1		Sample
NIS	PIS	NIS	PIS	example
119	95	193	130	4×3-a
103	99	114	107	4×3-b
112	104	52	26	4×3-c
141	119	383	283	4×4-a
151	146	310	196	4×4-b
145	135	249	136	4×4-c

randomly. The processing times and due dates are uniformly distributed in the intervals [0,100] and $\left[P\left(1-T-\frac{R}{2}\right), P\left(1-T+\frac{R}{2}\right)\right]$, respectively.

where $P = (m+n-1)\bar{P}$ and $\bar{P} = \sum_{i=1}^{n}\sum_{j=1}^{m}p_{ij}/(n \times m)$ which is the mean values of the processing times. Parameters R and T take their values in the sets {0.2,0.6,1}, {0.4,0.6,0.8},respectively. Also, the setup times are random variables between 10 and 50.

Parameter setting

Various sets of controllable parameters and different sizes of problems are considered. Then, many experiments are designed and run by using them. The effective values in terms of solution quality are determined. The parameter of the TH method are as follows. Parameter γ takes its values in the set {0,0.1,. . .,1}. Also, it is assumed that the preference information corresponding to the importance levels of the objective functions are specified linguistically by the decision maker as: $\theta_1 = \theta_2$ and $\theta_1 > \theta_2$. So, the values of these parameters in the first condition are $\theta_1 = \theta_2 = 0.5$, and in the second condition are $\theta_1 = 0.8$ and $\theta_2 = 0.2$. It should be mentioned that the values of controllable parameters (i.e., R and T) of the generating instances approach are given in the following subsections. Tables 2 and 3 depict the values are set for the parameters of the NSGA-II and SPEA-II, respectively. Moreover, each example is solved 10 times independently.

Solving small-sized problems

Two kinds of sample examples in small sizes with 4 jobs and 3 machines (i.e., 4×3), and 4 jobs and 4 machines (i.e., 4×4) are considered. For each type of these examples, three numerical instances are generated randomly. Tables 4 and 5 represent the values of controllable
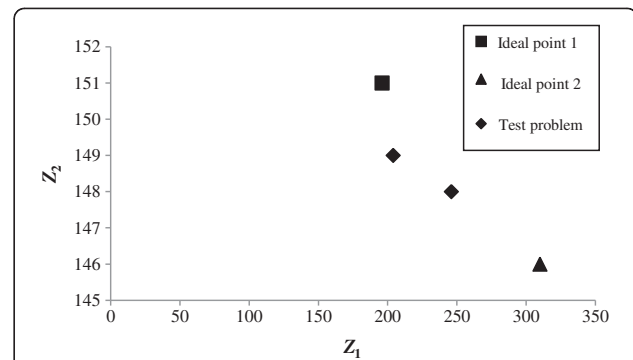

Figure 6 Pareto-optimal frontier of example 4×4-b.

parameters and the values of positive ideal solutions (Z_i^{PIS}) and negative ideal solutions (Z_i^{NIS}) for each numerical instances, respectively. Using the obtained values of the PIS and the NIS of each objective function, the final MILP model is exactly solved for each numerical instances by the Lingo 9 software. Table 6 illustrates the results of example 4×4-b.

The obtained resuts are analyzed as follows.

- In more cases with different importance levels of objective functions, the TH method performs correctly and efficiently. It means according to the importance levels , the solutions found by this method are unbalanced compromised solutions. However, in some cases, the TH method does not perform well and the satisfaction degree of the objective function with lower importance is more than that of the objective function with higher importance.

- When the importance levels are equal, the satisfaction degrees of objective functions are very close to each other, in some cases. Thus, the method approximately finds balanced compromised solutions.

- According to the discussion in Section 3, each optimal solution of the final MILP model is an efficient solution to the BOMILP model. Thus, in all cases, by changing the values of controllable parameters (i.e. γ and θ) two Pareto-optimal solutions are found by the TH method. Figure 6 indicates the Pareto-optimal frontier of example 4×4-b.

Table 6 Computational results of example 4×4-b

μ_{Z_2}	μ_{Z_1}	Z_2	Z_1	μ_{Z_2}	μ_{Z_1}	Z_2	Z_1	γ
	θ_1= 0.8 , θ_2= 0.2				θ_1= 0.5 , θ_2= 0.5			
0.4000	0.9298	149	204	0.4000	0.9298	149	204	0
0.4000	0.9298	149	204	0.6000	0.5614	148	246	0.1
0.4000	0.9298	149	204	0.6000	0.5614	148	246	0.2
0.4000	0.9298	149	204	0.6000	0.5614	148	246	0.3
0.4000	0.9298	149	204	0.6000	0.5614	148	246	0.4
0.6000	0.5614	148	246	0.6000	0.5614	148	246	0.5
0.6000	0.5614	148	246	0.6000	0.5614	148	246	0.6
0.6000	0.5614	148	246	0.6000	0.5614	148	246	0.7
0.6000	0.5614	148	246	0.6000	0.5614	148	246	0.8
0.6000	0.5614	148	246	0.6000	0.5614	148	246	0.9
0.6000	0.5614	148	246	0.6000	0.5614	148	246	1
			T= 0.6		R= 0.6			

Table 7 Characteristic of medium to large-sized examples

Number of machines	Number of jobs	Representation	Sample example
5	10	10×5	1
7	14	14×7	2
9	20	20×9	3

Table 8 Computational results of the quality metric

Sample	Improved NSGAII										SPEAII									
examples	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
10×5-a	3	4	5	3	5	3	4	4	4	3	3	3	2	3	2	1	3	2	3	0
10×5-b	4	4	4	3	3	5	5	4	3	2	2	1	3	2	4	2	3	3	0	3
10×5-c	5	3	5	5	1	2	2	4	4	5	1	3	3	2	2	0	1	1	4	2
14×7-a	5	5	6	4	7	5	6	4	5	6	4	1	2	3	1	4	5	0	3	3
14×7-b	6	4	1	2	6	5	5	4	6	7	1	1	4	5	2	3	1	3	4	4
14×7-c	4	4	7	7	6	4	4	1	3	3	0	4	6	5	2	4	3	1	1	5
20×9-a	7	8	6	6	8	4	4	3	5	5	5	5	4	2	2	1	2	6	7
20×9-b	1	5	4	4	6	8	3	4	3	6	2	1	3	2	3	5	4	4	1	5
20×9-c	6	8	5	7	7	4	5	8	6	5	5	6	2	2	3	5	1	4	4	1

Solving medium to large-sized problems

In this section, three types of sample examples in medium to large sizes with different combinations of jobs and machines are designed. For each kind of these examples, three numerical instances are generated randomly. The values of controllable parameters for each instance are considered as shown in Table 4 and the characteristic of sample examples are presented in Table 7. All of these instances are solved independently by the proposed improved NSGA-II with the features described in Section 4 and SPEA-II. The performance of the improved NSGA-II is compared with the performance of the SPEA-II by using the design of experiments (DOE) based on three comparison metrics. Meanwhile, the instance 20×9-c is solved by traditional NSGA-II in order to compare the efficiency of the improved NSGA-II with the efficiency of the traditional NSGA-II.

Performance evaluation metrics

There are various metrics in the literature for evaluating the performance of multi-objective metaheuristics. Here we use three common and valid metrics to evaluate the performance of the improved NSGA-II, traditional NSGA-II and SPEA-II. These comparison metrics are defined as follows:

- *Quality Metric* (*QM*): This kind of the metric was applied by Schaffer (1985). According to this metric, each algorithm that finds more Pareto solutions is considered to have a higher quality. However, some Pareto solutions of an algorithm may dominate those of another algorithm. Thus, a number of the final non-dominated solutions found by each algorithm are counted.
- *Diversity Metric* (*DM*): This metric was applied by Zitzler (1999). To calculate the spread of the solutions in the final Pareto frontier found by each algorithm, the diversity metric is used and computed by:

$$D = \sqrt{\sum_{i=1}^{n} \max(x_i - y_i; \rightarrow x, \rightarrow y \in F)} \quad (34)$$

where, F is the set of obtained Pareto solutions, $\bar{x}$ and $\bar{y}$ are two solution vectors of Pareto frontier, and n denotes the dimension of the solution space that is equal to the number of objective functions. In this paper, n is equal to 2.

Table 9 Computational results of the diversity metric

Sample	Improved NSGA-II										SPEA-II									
examples	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
10×5-a	342.1	200.2	95.2	109.4	287.3	311.5	364	215.1	390.6	373.8	105.4	125.8	67.3	258.3	304.1	111.5	279.9	299.4	178.2	214.9
10×5-b	312.2	422.8	255	361.7	288.7	346.3	342.9	476.5	400.8	312.4	222.8	315.4	321.1	213.3	223.1	85.7	301	158.9	400.1	112
10×5-c	488.8	405.6	346.1	369	287	241.3	211.9	247.3	376.9	398.6	223	302.8	210.9	117.3	241.8	92.1	355	403.6	323.3	398.9
14×7-a	412.1	532.3	574	316.6	348.9	499	296.5	488.7	366.6	505.8	528.1	398.1	145.6	503.3	202.9	403.8	365.5	244.3	307.1	413.2
14×7-b	546.8	569	346.1	310.2	445.5	563.1	549.9	411	343.3	406.3	401.3	321.8	303.9	286	442.2	431.1	514.6	102.8	536.4	432.3
14×7-c	455.2	512.2	414.4	326	263.9	343.7	417.3	501.1	333.3	470.8	378.9	225.3	328.5	312.5	436.1	337.1	203.3	220	241.1	236.7
20×9-a	512.2	479.4	585.5	568.7	385.8	531.4	473.8	508.2	434.9	621.5	474.2	99.1	333.3	326.2	344.5	228.9	466.1	389	371.4	352.1
20×9-b	612.5	418.7	565.5	673.8	597.9	507.2	659.3	677.2	510.1	564	569.3	347.5	512.9	319	546.1	463.9	564.1	385.7	236.8	510.9
20×9-c	491.6	638.4	594.7	736.8	500.3	595.4	670.6	617.2	644.6	591.4	533.3	358.4	478.1	419.3	516.4	320.2	596.9	488.4	349.1	573.6

Table 10 Computational results of the spacing metric

Sample examples	Improved NSGA-II										SPEA-II									
	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
10×5-a	0.25	0.04	1.33	2.01	3	0.90	0.41	3	0.64	1.88	0.95	1.85	0.95	3	2	0.63	1.44	0.72	1.76	3
10×5-b	0.41	2	0.05	0.14	2	0.18	0.99	3	0.33	1.11	0.07	0.58	0.49	2.64	0.88	2	3	3	0.66	0.51
10×5-c	0.69	2	3	0.95	0.67	2.35	1.13	0.19	0.85	1.44	3	2.25	3	3	2	0.71	0.58	3	2.41	0.26
14×7-a	0.19	0.05	0.03	0.26	1.29	1.35	2	0.14	0.96	0.90	0.66	0.73	0.46	2.27	0.43	3	3	1.69	2	1.74
14×7-b	0.03	0.16	0.33	0.97	1.64	2.87	1.41	0.77	0.46	1.36	0.26	0.48	2.33	2.14	0.51	3	2	1.26	3	3
14×7-c	1.40	0.16	1.36	2	0.39	0.87	0.58	0.97	1.23	0.46	2.14	0.82	0.27	0.33	1.65	2	3	2	2.88	1.92
20×9-a	2.59	0.90	0.26	0.99	3	1.21	0.03	0.60	0.08	0.01	2.51	2.39	1.23	0.48	3	3	0.12	2	1.01	3
20×9-b	0.12	0.84	1.61	0.43	0.80	2.41	1.02	0.32	0.63	0.11	1.20	1.37	0.29	0.80	0.86	3	3	2	0.72	2.11
20×9-c	0.19	0.01	0.29	0.94	1.04	0	1.17	0.55	0.39	0.82	1.82	2	3	2	0.85	0.32	1.66	0.25	0.06	0.34

- *Spacing Metric* (*SM*): This kind of the metric was applied by Srinivas and Deb (1994). This metric is used for estimating the uniformity of the spread of the points in the final Pareto solution frontier found by each algorithm and is calculated by:

$$S = \frac{\Sigma_{i=1}^{N-1}\left|d_i - \bar{d}\right|}{(N-1)\bar{d}} \quad (35)$$

where, d_i represents the Euclidean distance between the consecutive solutions of the obtained Pareto solution set, $\bar{d}$ is the mean value of all Euclidean distances, and N denotes the number of the final obtained Pareto solutions. For this metric, a higher quality value is a lower value and, of course, the best value is equal to 0.

Comparative results of the improved NSGAII and SPEAII by DOE

All of the medium to large-sized problems defined in Tables 4 and 7 are solved by the improved NSGA-II and the SPEA-II. In order to analyze the results, two-factor factorial experiments are designed to examine the effect of the two solution methods and nine test problems on each of three comparison metrics with 10 times executions of each algorithm for each test problem. This approach is similar to the approach given in, (Noori-Darvish et al. 2011). The computational results of three comparison metrics, which are calculated for the sets of Pareto solutions obtained by the improved NSGA-II and

the SPEA-II in each time of running, are illustrated in Tables 8, 9 to 10. The following statistical linear model (Montgomery and Design and analysis of experiments, 2001) can represent the results of these tables.

$$y_{ijk} = \mu + \tau_i + \beta_j + (\tau\beta)_{ij} + \epsilon_{ijk} \begin{cases} i = 1, 2, \ldots, 9 \\ j = 1, 2 \\ k = 1, 2, \ldots, 10 \end{cases} \quad (36)$$

where μ is a common effect for the whole experiment, τ_i is the effect of the i-th test problem, β_j is the effect of the j-th solution method, $(\tau\beta)_{ij}$ is the interaction effect of the i-th test problem and the j-th solution method, and ϵ_{ijk} is the random error.

It should be noted that the adequacy of factorial designs is checked formerly. The hypothesis tests are considered as follows. The row treatment (i.e., test problems) effects are equal to 0. The column treatment (i.e., solution methods) effects are equal to 0. The interactions are equal to

Table 11 ANOVA result for the quality metric

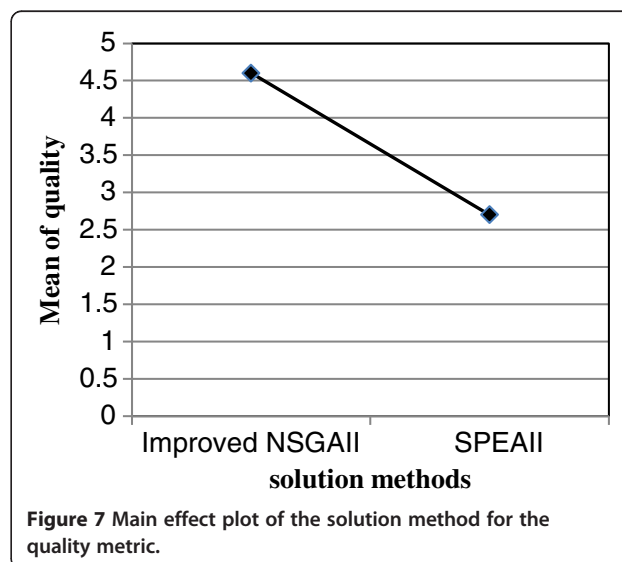| P-Value	F₀	MS	SS	DF	Source of Variation
0.000	4.28	10.118	80.944	8	Test problem
0.000	62.45	147.606	147.606	1	Solution Method
0.714	0.67	1.593	12.744	8	Interaction
		2.364	382.900	162	Error
			624.194	179	Total

Figure 7 Main effect plot of the solution method for the quality metric.

Table 12 ANOVA result for the diversity metric

P-Value	F_0	MS	SS	DF	Source of Variation
0.000	19.86	200573	1604583	8	Test problem
0.000	56.35	568958	568958	1	Solution Method
0.455	0.99	10013	80101	8	Interaction
		10097	1635771	162	Error
			3889413	179	Total

Table 13 ANOVA result for the spacing metric

P-Value	F_0	MS	SS	DF	Source of Variation
0.310	1.19	1.0399	8.319	8	Test problem
0.000	23.44	20.5436	20.544	1	Solution Method
0.976	0.26	0.2318	1.854	8	Interaction
		0.8764	141.979	162	Error
			172.696	179	Total

0. In addition, the significance level (α) is set as 0.05.

$$H_0 : \tau_1 = \tau_2 = \cdots = \tau_9 = 0$$
$$H_1 : at \ least \ one \ \tau_i \neq 0 \tag{37}$$

$$H_0 : \beta_1 = \beta_2 = 0$$
$$H_1 : \beta_1 \neq \beta_2 \tag{38}$$

$$H_0 : (\tau\beta)_{ij} = 0$$
$$H_1 : at \ least \ one \ (\tau\beta)_{ij} \neq 0 \tag{39}$$

Table 11 depicts the ANOVA result for the QM. Considering the results, the *p*-value for the solution method and the test problem main effects are less than α=0.05. Thus, the effects of the solution method and the test problem are significant. It means that there is a significant difference between the mean values for the two solution methods and there is a significant difference between the mean values for the nine test problems. However, the interaction is not significant. Figure 7 indicates that more Pareto solutions are found by the improved NSGA-II, and this method performs better than the SPEA-II.

Table 12 shows the ANOVA result for the DM. Considering the results, the *p*-value for the main effects of the solution method and the test problem are less than α=0.05. Therefore, the method effects of the solution

method and the test problem are significant. However, the interaction is not significant. The diversity of solutions found by the improved NSGA-II is more than the SPEAII. Thus, the improved NSGA-II method performs better than the SPEA-II, as shown in Figure 8.

The ANOVA result for the SM, which are illustrated in Table 13, shows that the *p*-value for the main effect of the solution method is less than α=0.05. Therefore, the effects of the solution method is significant. Moreover, the main effect of the test problem and the interaction are not significant. The spacing metric value for solutions found by the improved NSGA-II is less than the SPEA-II, as shown in Figure 9. Thus, the performance of the improved NSGA-II is better than the SPEA-II.

Considering the above descriptions, all of the three hypothesis tests indicate the significant effect and better performance of the improved NSGA-II than the SPEA-II based on three performance metrics.

Comparative results of the improved and the traditional NSGAII

As discussed in Subsection 5.5, in order to examine if our improved NSGA-II performs better than the original NSGA-II, one of the large-sized instances (i.e., 20×9-c) is solved for 10 times independently by using the original NSGA-II. Then, three-mentioned comparison metrics
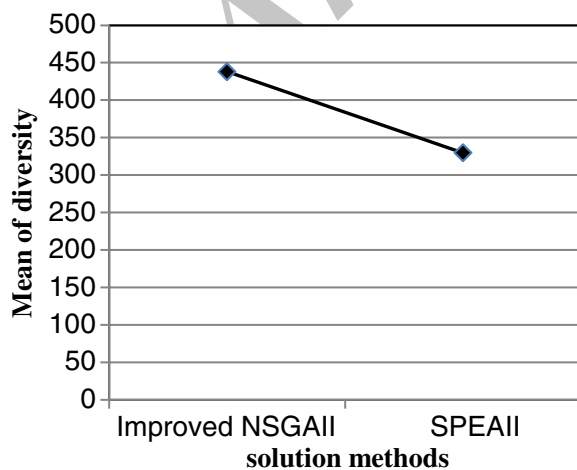

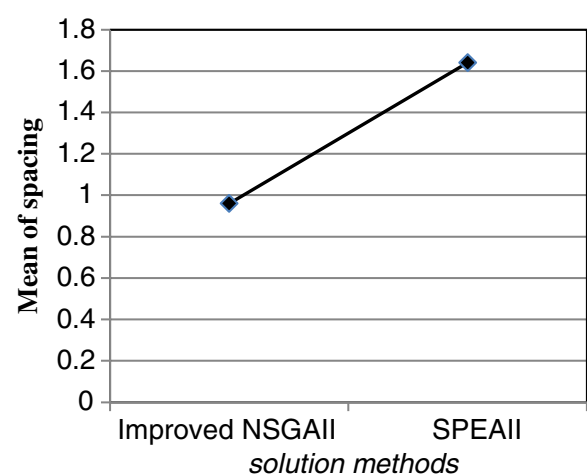**Figure 8** Main effect plot of the solution method for the diversity metric.

Figure 9 Main effect plot of the solution method for the spacing metric.

Table 14 Computational results of the QM, DM and SM of example 20×9-c solved by the improved NSGA-II

Comparison metrics	Improved NSGA-II										Mean value
	1	2	3	4	5	6	7	8	9	10	
QM	6	8	5	7	7	4	5	8	6	5	6.1
DM	491.6	638.4	594.7	736.8	500.3	595.4	670.6	617.2	644.6	591.4	608.1
SM	0.19	0.01	0.29	0.94	1.04	0	1.17	0.55	0.39	0.82	0.54

(i.e., QM, DM and SM) are calculated based on the set of Pareto solutions achieved by this algorithm in each time of running. These results are compared with the results of solving the instance 20×9-c by the improved NSGA-II, as shown in Tables 8 to 10. The computational results and the mean values of these three performance metrics are illustrated in Tables 14 and 15.

According to the obtained results, the mean values of the quality and diversity metrics calculated for the sets of Pareto solutions, which are found by the improved NSGA-II, are more than those of the original NSGA-II. In addition, the mean value of the spacing metric calculated for the sets of Pareto solutions obtained by improved NSGA-II are less than those of the traditional NSGA-II. Therefore, considering the above descriptions, all results indicate the better performance of the improved NSGA-II than the traditional NSGAII based on three performance metrics.

Conclusion

In this paper, an open shop scheduling problem with sequence-dependent setup times was examined. A novel bi-objective mathematical programming was designed in order to minimize the total tardiness and the makespan. An interactive multi-objective decision making (MODM) approach proposed by Torabi and Hassini (2008) was applied for solving small-sized instances optimally and obtaining Pareto-optimal solutions. Considering the results in more cases, the TH method has performed well according to the importance levels of the objective functions, and two Pareto-optimal solutions have been found. In order to achieve Pareto-optimal sets for medium to large-sized problems, an improved non-dominated sorting genetic algorithm II (NSGA-II) was presented by embedding a heuristic method for obtaining the good initial population. Finally, by using the design of experiments (DOE), the efficiency of the proposed improved NSGA-II

was compared with the efficiency of the SPEA-II. Moreover, the performance of the improved NSGA-II was compared with the performance of the traditional NSGA-II. The results have indicated the better performance of the improved NSGA-II based on three performance metrics.

Authors' contributions
SND studied the literature review and proposed the problem solving methods. She analyzed the computational results and drafted the manuscript. RTM presented the mathematical model and was responsible for revising the whole manuscript. All authors read and approved the final manuscript.

Author details
[1]Department of Industrial Engineering, Allame Mohades Noori University, PC: 46415-451, Noor, Iran. [2]Department of Industrial Engineering, College of Engineering, University of Tehran, PC: 14399-57131, Tehran, Iran.

References
Ahmadizar F, Ghazanfari M, Fatemi Ghomi SMT (2010) Group shops scheduling with makespan criterion subject to random release dates and processing times. Comput Oper Res 37:152–162
Allahverdi A, Ng CT, Cheng TCE, Kovalyov MY (2008) A survey of scheduling problems with setup times or costs. European Journal Of Operational Research 187:985–1032
Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6:182–197
Fei H, Meskens N, Chu C (2010) A planning and scheduling problem for an operating theatre using an open scheduling strategy. Comput Ind Eng 58:221–230
Gonzalez T, Sahni S (1976) Open shop scheduling to minimize finish time. J ACM 23(4):665–679
Loukil T, Teghem J, Tuyttens D (2005) Solving multi-objective production scheduling problems using metaheuristics. Eur J Oper Res 161:42–61
Low C, Yeh Y (2008) Genetic algorithm-based heuristics for an open shop scheduling problem with setup, processing, and removal times separated. Robotics and Computer-Integrated Manufacturing 25:314–322
Mastrolilli M, Queyranne M, Schulz AS, Svensson O, Uhan NA (2010) Minimizing the sum of weighted completion times in a concurrent open shop. Oper Res Lett 38(5):390–395
Matta ME (2009) A genetic algorithm for the proportionate multiprocessor open shop. Comput Oper Res 36:2601–2618

Table 15 Computational results of the QM, DM and SM of example 20×9-c solved by the traditional NSGA-II

Comparison metrics	Traditional NSGA-II										Mean value
	1	2	3	4	5	6	7	8	9	10	
QM	6	5	7	7	6	4	6	4	7	6	5.8
DM	675.2	624.01	425.6	579.8	621.1	573.2	566.1	599.7	601.4	605.1	587.1
SM	0.28	0.39	0.26	0.47	0.31	1.01	0.7	1.23	0.15	0.92	0.57

Montgomery DC (2001) Design and analysis of experiments, 5th edn. John Wiley & Sons, INC, New York

Mosheiov G, Oron D (2008) Open-shop batch scheduling with identical jobs. Eur J Oper Res 187:1282–1292

Naderi B, FatemiGhomi SMT, Aminnayeri M, Zandieh M (2011) Scheduling open shops with parallel machines to minimize total completion time. J Comput Appl Math 235:1275–1287

Noori-Darvish S, Tavakkoli-Moghaddam R (2011) Solving a bi-objective open shop scheduling problem with fuzzy parameters. Journal of Applied Operational Research 3(2):59–74

Noori-Darvish S, Tavakkoli-Moghaddam R, Javadian N (2011) A multi-objective particle swarm optimization algorithm for possibilistic open shop problems with weighted mean tardiness and weighted mean completion times criterions. Iranian Journal of Operations Research (IJOR), Accepted for publication

Panahi H, Tavakkoli-Moghaddam R (2011) Solving a multi objective open shop scheduling problem by a novel hybrid ant colony optimization. Expert Syst Appl 38(3):2817–2822

Roshanaei V, SeyyedEsfehani MM, Zandieh M (2009) Integrating non-preemptive open shops scheduling with sequence-dependent setup times using advanced metaheuristic. Expert Syst Appl 37:259–266

Schaffer JD (1985) Multiple objective optimizations with vector evaluated genetic algorithms. In: Schaffer JD (ed) Genetic algorithms and their applications: Proceedings of the first international conference on genetic algorithms. Lawrence Erlbaum, Hillsdale, New Jersey, pp 93–100

Seraj O, Tavakkoli-Moghaddam R (2009) A tabu search method for a new bi-objective open shop scheduling problem by a fuzzy multi-objective decision making approach. Int J of Engineering, Transaction B: Application 22:269–282

Srinivas N, Deb K (1994) Multi-objective optimization using non-dominated sorting in genetic algorithms. Evol Comput 2:221–248

Sule DR (1997) Industrial scheduling. PWS Publishing Company, Boston, MA.

Torabi SA, Hassini E (2008) An interactive possibilistic programming approach for multiple objective supply chain master planning. Fuzzy Set Syst 159:193–214

Zandieh M, Fatemi Ghomi SMT, Moattar Husseini SM (2006) An immune algorithm approach to hybrid flowshops scheduling with sequence-dependent setup times. Appl Math Comput 180:111–127

Zhang X, van de Velde S (2010) On-line two machine open shop scheduling with time lags. European Journal Of Operational Research 204:14–19

Zitzler E (1999) Evolutionary algorithm for multi-objective optimization: methods and applications. Ph.D. Dissertation, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland

Zitzler E, Laumanns M, Thiele L (2001) Improving the Strength Pareto Evolutionary Algorithm. In: Giannakoglou K, Tsahalis D, Periaux J, Papailou P, Fogarty T (eds) EUROGEN 2001, Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems., Athens, Greece, 19-21 September 2011, Published by International Center for Numerical Methods in Engineering (Cmine), Barcelona, Spain, ISBN: 84-89925-97-6, pp 95–100

www.SID.ir