Oper Res Int J (2018) 18:123–158
https://doi.org/10.1007/s12351-016-0257-6

ORIGINAL PAPER

An efficient bi-objective algorithm to solve re-entrant hybrid flow shop scheduling with learning effect and setup times

S. M. Mousavi[1] · I. Mahdavi[1] · J. Rezaeian[1] · M. Zandieh[2]

Received: 26 November 2015 / Revised: 16 May 2016 / Accepted: 12 July 2016 / Published online: 19 July 2016

Abstract This paper deals with a bi-objective hybrid flow shop scheduling problem minimizing the maximum completion time (makespan) and total tardiness, in which we consider re-entrant lines, setup times and position-dependent learning effects. The solution method based on genetic algorithm is proposed to solve the problem approximately, which belongs to non-deterministic polynomial-time (NP)-hard class. The solution procedure is categorized through methods where various solutions are found and then, the decision-makers select the most adequate (a posteriori approach). Taguchi method is applied to set the parameters of proposed algorithm. To demonstrate the validation of proposed algorithm, the full enumeration algorithm is used to find the Pareto-optimal front for special small problems. To show the efficiency and effectiveness of the proposed algorithm in comparison with other efficient algorithm in the literature (namely MLPGA) on our problem, the experiments were conducted on three dimensions of problems: small, medium and large. Computational results are expressed in terms of standard multi-objective metrics. The results show that the proposed algorithm is able to obtain more diversified and competitive Pareto sets than the MLPGA.

Keywords Re-entrant hybrid flow shop · Setup times · Learning effect · Multi-objective problems · A posteriori approach

✉ S. M. Mousavi
mousavi@iauns.ac.ir; mostafa_mousavi85@yahoo.com

[1] Department of Industrial Engineering, Mazandaran University of Science and Technology, Babol, Iran

[2] Department of Industrial Management, Management and Accounting Faculty, Shahid Beheshti University, G. C., Tehran, Iran

1 Introduction

Scheduling is an important tool for manufacturing and engineering, where it can have a major impact on the productivity of a process. Production scheduling is management and allocation of resources, events and processes to create goods and services. Production scheduling aims to maximize the efficiency of the operation and to minimize the production time and costs, by telling a production facility when to make, with which staff, and on which equipment.

In the literature, the notion of hybrid flow shop (HFS) scheduling problem has emerged in 1970s (Arthanary and Ramaswamy 1971). The HFS can be found in various types of industries. The most representative one is in electronics, such as semiconductor wafer fabrication, printed circuit board (PCB) manufacturing, thin film transistor-liquid crystal display (TFT-LCD) manufacturing, etc. In addition, various traditional industries, such as food, oil, pharmaceutical, tobacco, textile, chemical, steel, paper, and metallurgical industry, have various HFSs (or can be modeled as a HFS). Ruiz and Rodríguez (2010) described the HFS problem in its “standard” form. This paper investigates an HFS problem in standard form with additional features, including: setup times, re-entrant flows, and position-dependent learning effects.

The literature of HFS is filled with different applied industrial assumptions to better represent the real nature of scheduling environments. One of the most prevailing and extremely favored assumptions by many researchers in real scheduling configurations is the integration of sequence-dependent setup times into different shop scheduling environments. The importance and applications of scheduling models with explicit considerations of setup times (costs) have been discussed in several studies (i.e. Chang et al. 2003; Andrés et al. 2005). One of the underlying assumptions in this paper is to consider setup times in scheduling configurations. The setup times considered in this problem are classified into two types: (1) sequence-independent setup time (SIST); and (2) sequence-dependent setup times (SDST). In the former, setup depends only on the job to be processed. In the latter, setup depends on both the job to be processed and the immediate preceding job. Allahverdi et al. (1999, 2008) provided a comprehensive review of the literature on scheduling problems involving setup times (costs). The HFS problem with setup times has been investigated in several studies (Jungwattanakit et al. 2008, 2009; Behnamian et al. 2009; Davoudpour and Ashrafi 2009; Naderi et al. 2009a, b; Rashidi et al. 2010; Karimi et al. 2010; Mousavi et al. 2011a, b, 2012a; b; Hakimzadeh Abyaneh and Zandieh 2012; Pargar and Zandieh 2012; Behnamian and Zandieh 2013; Fadaei and Zandieh 2013; Jolai et al. 2013; Attar et al. 2014; Wang and Liu 2014). For example, Jungwattanakit et al. (2008, 2009) considered the flexible flow shop with unrelated parallel machines and sequence/machine dependent setup times, release date and due date constraints. Davoudpour and Ashrafi (2009) focused on the SDST HFS problems with identical parallel machines, and release date. Rashidi et al. (2010) investigated the HFS problems with unrelated parallel machines, SDST and blocking processor. Mousavi et al. (2011a, b, 2012a, b) studied the problem of scheduling n independent jobs in

HFS environment with SDST. Fadaei and Zandieh (2013) investigated group scheduling in the problem of HFS scheduling within the area of sequence-dependent family setup times. Wang and Liu (2014) also found an integrated bi-objective optimization problem with non-resumable jobs for production scheduling and preventive maintenance in a two-stage HFS. They considered the SDST and preventive maintenance on the first stage machine.

HFSs can be classified into two types according to product flows: (1) those with unidirectional flows; and (2) those with re-entrant flows. The unidirectional flows imply that each job starts at the first stage and finishes at the last stage. On the other hand, in the reentrant flows, each job may visit a serial stage twice or more. Therefore, the re-entrant HFS (RHFS) means that there are n jobs to be processed on g stages and every job must be processed on stages in the order of stage 1, stage 2,…, stage g for l times (l is the number of repetition of jobs on the sequence of stages). Lin and Lee (2011) provided a comprehensive review of the literature on scheduling problems involving re-entrant flows. Also, Dugardin et al. (2010), Cho et al. (2011) and Ying et al. (2014) considered the multi-objective HFS problem with re-entrant flow.

The last underlying assumption in this paper is the consideration of learning effects in scheduling configurations. In classical scheduling, job processing and setup times are assumed to be constant from the first job to be processed until the last job to be completed. Despite the effect of learning in a production environment, the processing and setup times of a given job are shorter if it is scheduled later in the production sequence. The learning effects considered in scheduling environments are classified into two types: (1) position-based learning; and (2) the sum of processing time. Regarding the last underlying assumption in this paper, processing and setup times of a given job depend on its position in the sequence arrived to each stage which means the learning effects are position-based learning. Biskup (2008) provided a comprehensive review of the literature on scheduling problems involving learning effects. The HFS problem with learning effects has been investigated in several studies. For example, Pargar and Zandieh (2012) and Behnamian and Zandieh (2013) investigated the HFS problems with SDST and position-dependent learning effects.

The present study investigates scheduling problem with learning considerations, using the learning curve introduced by Biskup (1999). The learning curve assumed by Biskup (1999) reflects decrease in production time as a function of number of repetitions. As shown in Biskup (1999), we assume that the processing time of job j at stage t of layer l if scheduled in position r, is given by Eq. (1).

$$P_{jrl}^{t} = P_{jl}^{t} \times (r_{jl}^{t})^{(a_{jl}^{t})} \quad \forall i, j, t, r, l \tag{1}$$

where $-1 \leq a_{jl}^{t} \leq 0$ is a constant learning index, given as the logarithm to the base 2 of the learning rate (LR). In this paper, we assume that all machines and jobs in each stage and layer have the same learning rate $(a_{jl}^{t} = a)$. Similarly, the setup time of job i to job j if scheduled in position r at stage t of layer l, is given by Eq. (2).

$$S_{ijrl}^{t} = S_{ijl}^{t} \times (r_{jl}^{t})^{(a_{jl}^{t})} \quad \forall i, j, t, r, l \tag{2}$$

P^t_{jl} actual processing time for job j at stage t of layer l, S^t_{ijl} actual setup time between job j and job i at stage t of layer l while job j is scheduled immediately after job i, P^t_{jrl} the processing time for job j in position r at stage t of layer l, S^t_{ijrl} the setup time of job i to job j, scheduled in position r at stage t of layer l.

Nowadays, because of the intensely competitive markets and limited resources, many manufacturers have come to appreciate the importance of scheduling by attempting to reduce their production expenses and the final costs of their products. The scheduling objective in such industries may vary. Three types of decision-making goals are prevalent in scheduling: (1) efficient utilization of resources; (2) rapid response to demands, or minimizing the work-in-progress; and (3) close conformance to prescribed deadlines (Pinedo 2008). According to just-in-time (JIT) concept, production managers should consider more than one criterion in scheduling problems. Therefore, simultaneous minimization of two conflicted objective functions that are makespan and total tardiness. In fact, minimizing the makespan causes internal efficiency and maintains the work-in-process inventory at a low level. Minimizing the total tardiness causes external efficiency and reduces the penalties incurred for late jobs.

According to the best of our knowledge, bi-objective RHFS with SDST and learning effect problem have never been investigated in the scheduling problems. Therefore, the aim of this paper is to develop a solution method for the proposed problem that search a set of non-dominated solutions.

It has been shown in several studies that some of scheduling problems are belong to NP-hard class; for example, a single machine SDST scheduling problem is equivalent to a traveling salesman problem (TSP) and is NP-hard (Pinedo 1995). The HFS problem is significantly more complex than the regular single machine scheduling. On the other hand, Gupta (1988) showed the flow shop with multiple processors (FSMP) problem with only two stages to be NP-hard even when one of the two stages contains a single machine. The FSMP problem can be considered as a specific case of the HFS. Also, the re-entrant permutation flowshop scheduling problem for minimizing makespan has already been proven to be NP-hard (Wang et al. 1997). According to the research presented, we can easily conclude that our proposed problem is also an NP-hard problem which is not easy to solve by a traditional mathematical model. The exact methods are unable to render feasible solutions even for small instances of this problem in a reasonable computational time. Therefore, this inability justifies the need for employment of a variety of heuristics and meta-heuristics to solve these problems to optimality or near optimality. In this paper, we are going to use a meta-heuristic algorithm to solve scheduling problem. The proposed meta-heuristic and the details of it are explained in Sect. 2. The rest of the paper is organized as follows: Sect. 3 presents the computational results and numerical comparisons. Finally, Sect. 4 is devoted to conclusion and future works.

2 The proposed algorithm

In this paper, a genetic algorithm is proposed for solving bi-objective optimization problem. The main reason for using this approach is that the problem under study is NP-hard and genetic algorithm approach has been demonstrated cost- effective for solving this kind of problem.

The term “genetic algorithm”, almost universally abbreviated to GA, was first used by Holland (1975). Different approaches of GA appear in the literature of multi-objective optimization problem (MOP), some of them are Vector Evaluated Genetic Algorithm (VEGA) (Schaffer 1985), Multi-Objective Genetic Algorithm (MOGA) (Fonseca and Fleming 1993), Niched Pareto Genetic Algorithm (NPGA) (Horn et al. 1994), Non-dominated Sorting Genetic Algorithm (NSGA & NSGA-II) (Srinivas and Deb 1995; Deb et al. 2002), Pareto Stratum-Niche Cubicle Genetic Algorithm (PS-NC GA) (Hyun et al. 1998), Multiple Objective Genetic Local Search (MOGLS) (Ishibuchi and Murata 1998), Strength Pareto Evolutionary Algorithm (SPEA & SPEA-II) (Zitzler and Thiele 1999; Zitzler et al. 2001), Pareto Archive Evolution Strategy (PAES) (Knowles and Corne 1999), Elitist Non-dominated Sorting Genetic Algorithm (ENGA) (Bagchi 1999), The Pareto Envelope based Selection Algorithm (PESA & PESA-II) (Corne et al. 2000, 2001), and so on.

Although many studies have provided valuable developments and applications for GA, improvements still can be made in designing GA for MOPs. In this paper, the proposed algorithm is somewhat similar to the NSGA-II. The NSGA-II algorithm has been developed by Deb et al. (2002) as a fast and efficient multi-objective genetic algorithm. The aim of this algorithm is to find a set of non-dominated solutions based on the Pareto dominance relationship. The main concept of the NSGA-II is the creation of an initial population, the selection of parents, the creation of children and the finding of non-dominated solutions. In the following, we extensively describe the structure and details of the proposed algorithm.

2.1 The structure of the proposed algorithm

The steps of proposed algorithm are shown below.

Step 1: Encoding The application of an algorithm requires the representation of a solution. We apply a scheme using integers that shows the number of job. In this kind of representation, a single row array of the size equals to the number of the jobs to be scheduled. The value of the first element of the array shows which job is scheduled first. The second value shows a job which is scheduled secondly and so on. For example, consider a problem with five jobs ($n = 5$), two stages ($g = 2$), two machines at stage one ($m^1 = 2$), and three machines at stage two ($m^2 = 3$). Suppose a solution is generated according to integer coding as [3 1 4 2 5]. It is known that the machines in parallel are identical in capability and processing rate. Therefore, job 3 is process on machine 1 and job 1 is process on machine 2 at stage one. Then, the job 4 is assigned to the machine of which the completion time is smaller than other machines. This process continues like this, until all jobs are assigned to the first stage machines. For determining the order of jobs in the second stage, first in first out (FIFO) rule is used.

Step 2: Initialization

(i) Initialize the number of initial population (N_{pop}), Probability of crossover (P_c), Probability of mutation (P_m), Number of generation (N_g).
(ii) Initialize an initial population (P_0) randomly.

Step 3: Non-dominated sorting and ranking procedure First, the objective values of all chromosomes in the current population (P_i) is evaluated, then the population of solutions is classified into successive non-dominated fronts (primary rank), F_1 is the set of non-dominated solutions in P_i which is the first frontier and F_2 is the set of non-dominated solutions in $P_i \backslash F_1$ which is the second frontier and so on, F_t is the set of non-dominated solutions in the last frontier, and finally, the crowding distance of each solution with respect to every other solution on the same front (secondary rank) will computed.

Step 4: Dividing the population The current population is divided into 3 categories: (1) Set 'A' contains the non-dominated solutions in F_1, and the size of A ($|F_1|$) is N_A. (2) Set 'B' contains the non-dominated solutions in F_1, F_2, …, and F_q, and the size of B ($|F_1| + |F_2| + \cdots + |F_q|$) is N_B. (3) Set 'C' contains the non-dominated solutions in F_{q+1}, F_{q+2}, …, F_t, and the size of C ($|F_{q+1}| + |F_{q+2}| + \cdots + |F_t|$) is N_C. If t fronts are obtained by primary rank, then q is half of fronts. It is known that the number of fronts (t) is different in each generation.

Step 5: Selection scheme The binary-tournament selection is employed at the selection operation to reproduce the next generation. According to this selection scheme, between two solutions with differing non-domination ranks (primary rank), we prefer the solution with the lower rank. Otherwise, if both solutions belong to the same front, then the solution located in a lesser crowded region is preferred (secondary rank).

Step 6: Neighborhood operator First, a neighborhood relation on the search space is defined, and then k-neighborhood solutions ($k = 2$) of each solution in set 'A' are generated. Inversion, swap, shift and k-exchange moves are applied in this paper. In each generation, neighborhood operator is selected randomly among introduced operators.

Step 7: Crossover operator Select $(N_{pop} - (k + 1) \times N_A) \times P_c$ pairs of parents from set 'B' based on the binary-tournament selection, and perform crossover on the parents. Order crossover (OX) is applied in this paper.

Step 8: Mutation operator Select $(N_{pop} - (k + 1) \times N_A) \times P_m$ parents from set 'C' based on the binary-tournament selection, and perform mutation on the parents. Inversion, swap, and k-exchange moves are applied in this paper. In each generation, mutation operator is selected randomly among introduced operators.

Step 9: Replacement Set A (include F_1) with solutions obtained from the previous steps (include steps 6, 7 and 8) are combined as new population (P_{i+1}).

Step 10: Stopping Rule If there is not improvement in F_1 obtained of two successive generations (P_i and P_{i+1}), then U count increases by one. If counter U equals to the pre-specified number (N_g) then stop, otherwise go to step 3. Data envelopment analysis (DEA) is used to design the stopping criterion.

Graphically, the proposed algorithm so-called NSGA-DEA can be presented as in Fig. 1. In the following subsections, we describe the details of the computation of crowding distance and stopping criterion.

2.2 Crowding distance

The introduced method by Pasupathy et al. (2007) is used for the computation of crowding distance for a bi-objective problem. The crowding distance of the ith solution in its front, called cd_i is computed as given in Eq. (3). In order to compute the cd_i first the "normalized Euclidean-distance based on crowding distance" (NEDCD) between solutions i and j, called D_{ij}, is computed as given in Eq. (4).

$$cd_i = \sum_{\substack{j \in \{F_{f_i}\} \\ i \neq j}} D_{ij} \tag{3}$$

$$D_{ij} = \sqrt{\sum_{b=1}^{2} \left(\frac{Z_{bi} - Z_{bj}}{\max\{Z_{bi'}\} - \min\{Z_{bi'}\}} \right)^2} \qquad i' \in \left[F_{f_{i'}}\right] \tag{4}$$

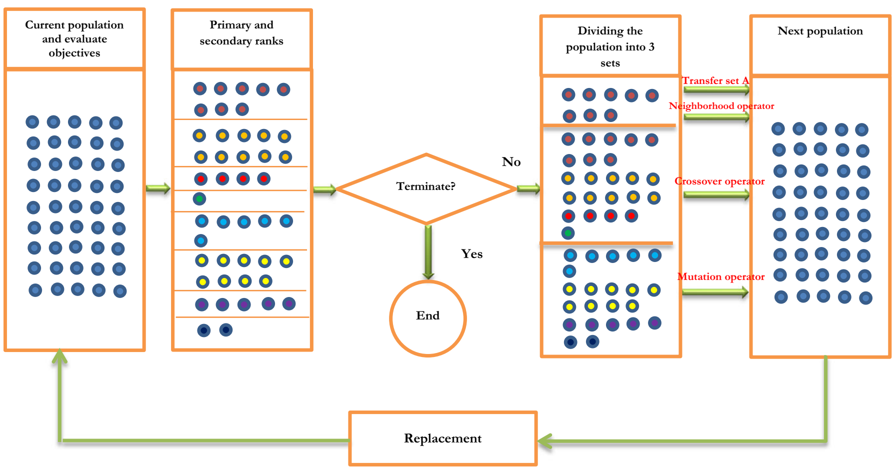

Fig. 1 Flowchart of proposed algorithm

In Eq. (3), f_i denotes the front on which solution i lies, and $[F_{f_i}]$ denotes the set of solutions that lie on the same front as that of solution i. In Eq. (4), Z_{bi} and Z_{bj} denote the values with respect to the bth objective function for solutions i and j, respectively. The solution with the largest value of crowding distance is assigned with the highest secondary rank (i.e., rank 1) in comparison to other solutions on the same front.

2.3 Proposed stop criterion

Suppose F_1 and F'_1 are the first frontier of two successive generations. Two cases can be expected of comparison between F_1 and F'_1 which are as follows: (1) F'_1 is better than F_1, (2) F'_1 and F_1 are the same. In order to compare two sets of F_1 and F'_1 quantitatively, the introduced method by Ruiz-Torres and Lopez (2004) is applied. They used free disposal hull (FDH) formulation that is a particular case of DEA. In this subsection, only the stages of method are briefly described. It is noted that each scheduling solution is a decision making unit (DMU), and inputs are the makespan and the total tardiness.

1. The scheduling solutions (or DMUs) of F_1 and F'_1 are combined in one single data set to generate an 'FDH problem' set.
2. In this stage, each DMU is comparable to the other DMUs on a one-to-one basis. DMUs which dominate the others but do not dominate themselves are efficient DMUs that are collected in a set so-called T. Efficient DMUs (or scheduling solutions) always have a degree of efficiency equal to one. Now, we want to calculate the degree of efficiency for DMU G, which is inefficient.

$F_1$and F'_1	Set of DMUs provided by two successive generations
$\lvert F_1\rvert$ and $\lvert F'_1\rvert$	Number of DMUs in sets F_1 and F'_1
C^s_{max}	Makespan corresponding to DMU S
$\bar{T}^s$	Total tardiness corresponding to DMU S
w_i	Relative weight of criteria i, with $\sum_{i=1}^{2} w_i = 1$
T	Set of all efficient DMUs
Q^G	Set of DMUs in T that make schedule G inefficient
E^G	Degree of efficiency of DMU G

The degree of efficiency of any DMU (efficient or inefficient) is computed as given in Eq. (5).

$$E^G = \begin{cases} 1 & \text{if G is efficient} \\ Max_{S\in Q^G}\left\{ w_1\left(\frac{C^S_{\max}}{C^G_{\max}}\right) + w_2\left(\frac{\bar{T}^S}{\bar{T}^G}\right)\right\} & \text{if G is inefficient} \end{cases} \tag{5}$$

3. The ‘efficiency’ of F_1 and F'_1 are calculated as given in Eq. (6) (called A_1 and A'_1 respectively).

$$A_1 = \frac{\sum_{G \in F_1} E^G}{|F_1|} \quad \text{and} \quad A'_1 = \frac{\sum_{G \in F'_1} E^G}{|F'_1|} \tag{6}$$

Two cases can be expected of comparison between A_1 and A'_1 which are as follows: (1) A'_1 value is bigger than A_1. (2) A'_1 value is equal to A_1. In the first case, improvement has been observed in the process. In the second, this result shows that no improvement has been reported. Therefore, U count increases by one. The stopping criterion for algorithm is terminated while counter U is equal to N_g (pre-specified number).

2.4 Discussion on the structure of algorithm

In this subsection, ideas used in the structure of algorithm are discussed in the form of questions and answers. The most important questions can be expressed as follows: (1) Why is a new operator so-called neighborhood operator added to search the neighborhood first frontier (F_1) in each generation? (2) Why are several neighborhood and mutation operators introduced to select among alternatives randomly in each generation? (3) Why is crossover operator on solutions in high level fronts applied? (4) Why is mutation operator on solutions in low level fronts applied? (5) Why is stop condition as stated in Sect. 2.3 considered?

Now we will respond to questions. In response to the first question, solutions belonging to the best non-dominant set F_1 are of best solutions in the population and must be emphasized more than any other solutions in the population. In fact, set F_1 is the closest front to the Pareto-optimal front. We make slight changes in solutions of set F_1 with the hope to find better results (closest to the Pareto-optimal set). These slight changes are done through neighborhood operator. This corresponds to the concept of the exploitation.

In response to the second question, the main reason for using this approach is that the algorithm is able to guide the search to another promising region through different types of moves. Therefore, the performance of algorithm with cited characteristic can be better. This corresponds to the concept of the diversification.

Concerning the third question, genetic algorithms have a recombination operation so-called crossover which is probably closest to the natural paragon. The crossover operator is used to mimic biological recombination between two single chromosome organisms. Therefore, offspring has the information from two parents. According to nature, competition among individuals for scanty resources results in the fittest individuals dominating over the weaker ones. Therefore, the population is modified with the natural law. We are going to modify population more quickly. In order to increase the selection probability of fittest individuals, crossover operator is applied to the solutions in high level fronts.

Regarding the fourth question, the main concept of the mutation is the changing of the structure of a gene, resulting in a variant form which may be transmitted to subsequent generations, caused by the alteration of single base units in DNA, or the deletion, insertion, or rearrangement of larger sections of genes or chromosomes. In the nature, mutation is usually employed to eliminate defects occurred in individuals. The defects here are considered to be as equivalent solutions in low level fronts. Therefore, mutation operator is applied on the solutions in low-level fronts.

In response to the last question, suppose our aim is to solve a single-objective minimization problem with GA. The objective value of the best solution in each generation will improve/remain constant when the algorithm is running. The relative difference (Δ) between the best solutions in two successive generations is simply calculated. Two cases can occur as follows: (1) $\Delta = 0$, (2) $\Delta < 0$. In the first, the best solutions remain constant in two successive generations (improvement has not been observed). In the second case, the algorithm has found a better solution for the next generation (improvement has been observed). The Δ value can be used to design two stop conditions as follows: (1) algorithm has reached a plateau such that successive iterations no longer produce better results (i.e. $\Delta = 0$ in successive iterations for the pre-specified number), (2) in non-consecutive U-generation, improvement has not been observed (i.e. $\Delta = 0$ in non-consecutive U iterations for the pre-specified number). Now, our aim is to solve a MOP with GA. In a MOP, non-dominated solutions in each generation will improve/remain constant as the algorithm is running. However, there is no straightforward manner such as Δ. Therefore, the terminating condition is proposed based on DEA to express the improvement or lack of improvement in the non-dominated solutions of two successive generations. The proposed approach is similar to Δ. The proposed stop condition is a new approach in the design of the stop condition.

3 Computational experiments

This section contains the method of generating data sets, performance criteria, the parameter setting with Taguchi method, validation of the proposed algorithm, running data sets by proposed algorithm and algorithm in the literature, and then expresses the results of the comparisons.

3.1 Test problems

The numerical data should be created to test the performance of the algorithm. Data required for a problem consists of number of jobs, number of stages, number of machines per stage, number of re-entrants, processing times, setup times, due dates, and learning indices. Note that, the largest number of machines in a stage must be less than the number of jobs $(n > max\{m^t; t \in g\})$. Designing range of levels of each factor is illustrated in Table 1. The number of machines, processing times and setup times (20–40 % of the mean of the processing time) are randomly generated from a discrete uniform distribution as described in Table 1. This table is divided into four categories: (1) special small problems, (2) small problems, (3) medium

Table 1 Factors and their levels

Factor	Levels			
	Special small	Small	Medium	Large
Number of jobs (n)	5; 7; and10	10; 15; and 20	25; 30; and 35	40; 50; and 60
Number of stages (g)	1; and 2	5; 7; and 10	10; 12; and 15	15; 17; and 20
Number of re-entrants (l)	1; and 2	1; and 2	2; and 3	3; and 4
Number of machines (m^t)	1; and 3	Uniform (1, 3)	Uniform (1, 6)	Uniform (1, 9)
Processing times (p)	Uniform (10, 20)	Uniform (10, 20)	Uniform (10, 40)	Uniform (10, 100)
Setup times (s)	Uniform (3, 6)	Uniform (3, 6)	Uniform (5, 10)	Uniform (11, 22)

problems, and (4) large problems. Special small problems are designed to assess the validity of the proposed algorithm. The term "specific" is given to these problems because they cover small problems of single machine, parallel machine, flow shop, and two-stage HFS. To demonstrate the effectiveness of proposed NSGA-DEA compared to algorithm in the literature, the experiments were conducted on three sizes of problems: small, medium and large. Due to levels of factor, the twenty-four problems are produced for the special small problems. The eighteen problems of multiplication levels of factors ($n \times g \times L$) are produced for the small, medium, and large problems. Learning indices -0.152 and -0.514 were selected with respect to the learning curve of 90 and 70 %, respectively. Also, we consider RHFS with SDST scheduling problems with no learning effect ($a_{jl}^t = 0$). In general, all problems are tested with regard to the level of learning indices. To generate due dates of all n jobs, we proposed the following steps:

Compute total processing time of each job on all g stages.

$$P_j = \sum_{l=1}^{L} \sum_{t=1}^{g} P_{jl}^t \quad \forall j \in n \tag{7}$$

Compute average setup time for all possible subsequent jobs and sum it for all g stages.

$$S_j = \sum_{l=1}^{L} \sum_{t=1}^{g} \left(\frac{\sum_{k=1}^{n} S_{kjl}^t}{n} \right) \quad \forall j \in n \tag{8}$$

Determine a due date for each job.

$$d_j = (P_j + S_j) \times \left(\frac{\max\limits_{t \in g} (m^t)}{g} \right) \times (1 + random \times 3) \quad \forall j \in n \tag{9}$$

where *random* is a random number from a uniform distribution over range (0, 1).

3.2 Performance criteria

In the literature of multi-objective optimization problems, several performance criteria have been presented to evaluate the quality of the obtained non-dominated front and to assess the performance of multi-objective optimizers. Each criterion has its advantages and disadvantages. There is no agreement as to which criteria should be used. In general, the quality of the approximated sets must be measured by qualitative and quantitative criteria. The following performance criteria are applied to compare the results of multi-objective algorithms quantitatively.

A. *Qualitative metrics*

1. Number of Pareto solutions (NPS): This performance criterion presents the number of non-dominated solutions obtained from each algorithm. The larger the number, the better the performance of the algorithm will be.
2. Quality metric 1 (QM1): To calculate the value of this criterion, first, the net non-dominated solutions (NDS) are generated by a set of all non-dominated solutions obtained from all algorithms (whose members should be also non-dominated in relation to one another) and then the percentage of non-dominated solutions of each algorithm in NDS to NPS is calculated. The larger the number, the better the performance of the algorithm will be.
3. Quality metric 2 (QM2): To calculate the value of this criterion, the percentage of non-dominated solutions of each algorithm in NDS to the number of NDS is calculated. The third metric signifies the percentage of the solutions in the net non-dominated Pareto set obtained by a certain algorithm. The larger the number, the better the performance of the algorithm will be.

B. *Quantitative metrics*

1. Mean ideal distance (MID): This measure presents the closeness between Pareto solution and ideal point (0, 0) which can be shown as Eq. (10).

$$MID = \frac{\sum_{i=1}^{n} c_i}{n} \tag{10}$$

   where n is the number of non-dominated set and $c_i = \sqrt{z_{1i}^2 + z_{2i}^2}$. The lower value of MID, the better of solutions quality we have.
2. Spread of non-dominated solution (SNS): This metric indicates the measure of diversity of Pareto-solutions, and more diversity of solutions is desirable. The value of SNS is measured as Eq. (11).

$$SNS = \sqrt{\frac{\sum_{i=1}^{n} (MID - c_i)^2}{n-1}} \tag{11}$$

3. Triangle method (TM): This measure presents the area under linear regression curve which can be calculated as Eq. (12) (Mousavi et al. 2012).

$$z_1 = b + a \times z_2 \tag{12}$$

where a and b are calculated as Eqs. (13) and (14):

$$a = \bar{z}_1 - b \times \bar{z}_2 \tag{13}$$

$$b = \frac{\sum_{i=1}^{n} (z_{2i} - \bar{z}_2)(z_{1i} - \bar{z}_1)}{\sum_{i=1}^{n} (z_{2i} - \bar{z}_2)} \tag{14}$$

Therefore, a smaller value of this metric for an algorithm proves to be better.

(B-4) Free Disposal Hull approach (FDH): In order to calculate this metric, all non-dominated solutions obtained by the algorithms are combined and the efficiency of these points is obtained by the FDH approach, which was proposed by Ruiz-Torres and Lopez (2004).

3.3 Parameter setting

Algorithm parameter values vary depending on different problem types when applying algorithm to achieve efficient solutions, so appropriate value selection has significant impact on the efficiency of algorithm. In this paper, the existing parameters in algorithm are determined by Taguchi method. Taguchi's method applies the quality loss function to evaluate product quality along with an orthogonal array to reduce the number of experiments.

In parameter setting, you first choose control factors and their levels and choose an orthogonal array appropriate for these control factors. The control factors comprise the inner array. In this paper, the parameters and their levels are shown in Table 2. The square matrix with 4 parameters in 3 levels used in the Taguchi method is L_9, which is given in Table 3.

The experiment is carried out by running several times each combination of control factor settings. The response data from each run in the outer array are usually aligned in a row, next to the factors settings for that run of the control factors in the inner array. Then, measured values are transferred in the form of S/N value.

Now, let us confirm the research characteristic-anticipating minimizing the makespan and total tardiness of jobs; namely, the smaller the cost values the better. Therefore, S/N ratio must be calculated using lower-is-better formula as Eq. (15).

$$S/N = -10 \log\left(\frac{\sum_{i=1}^{n} Y_i^2}{n}\right) \tag{15}$$

Table 2 Algorithm parameters and their levels

Level	Controllable factors			
	A (N_{pop})	B (N_g)	C (P_c)	D (P_m)
1	60	0.60	0.85	0.05
2	80	0.70	0.95	0.10
3	100	0.80	1	0.15

Table 3 The orthogonal array L_9

Experiments	A	B	C	D
1	A (1)	B (1)	C (1)	D (1)
2	A (1)	B (2)	C (2)	D (2)
3	A (1)	B (3)	C (3)	D (3)
4	A (2)	B (1)	C (2)	D (3)
5	A (2)	B (2)	C (3)	D (1)
6	A (2)	B (3)	C (1)	D (2)
7	A (3)	B (1)	C (3)	D (2)
8	A (3)	B (2)	C (1)	D (3)
9	A (3)	B (3)	C (2)	D (1)

Table 4 S/N ratio response table

Level	A (N_{pop})	B (N_g)	C (P_c)	D (P_m)
1	−80.5326	−80.5091	−80.5090	−80.4138
2	−80.4959	−80.5199	−80.3834	−80.5029
3	−80.3966	−80.3961	−80.5327	−80.5084
Delta	0.1360	0.1238	0.1492	0.0947
Rank	2	3	1	4

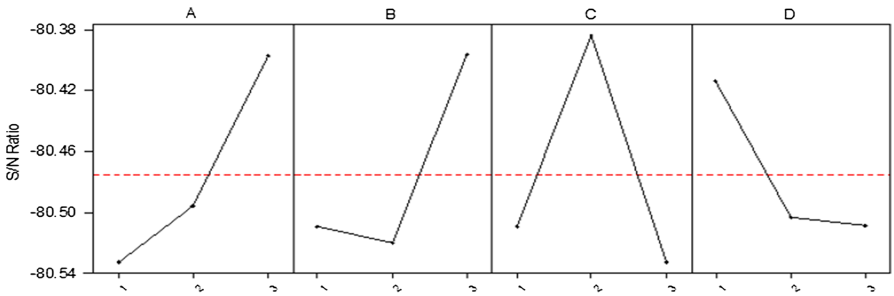

Fig. 2 Diagram of mean effect of the S/N ratio

The problem is that the response data from each run in MOPs are usually as set (non-dominated solutions). Since the Taguchi function should be assessed by one criterion, then a function which has shown the combination of all indexes is defined. The introduced utility function by Jolai et al. (2013) is used as follows (Eq. 16):

$$\text{Utility function} = \sqrt[11]{(NPS)^1 + (QM1)^1 + (QM2)^1 + (MID)^2 + (SNS)^2 + (TM)^2 + (FDH)^2} \tag{16}$$

This utility function is comprised of three qualitative and four quantitative criteria. The weight of 1 is allocated to qualitative criteria and the weight of 2 is allocated to quantitative criteria. This function has the role of a variable Y in Eq. (15).

Table 5 Details special small problems and results of the full enumeration and proposed algorithm

Test problem	$n \times g \times L$	m^t	a^t	Number of Pareto optimal solutions	NPS	NPS in Pareto optimal	CPU time (s)	
							Enumeration algorithm	NSGA-DEA
TSS1	$5 \times 1 \times 1$	1	−0.152	3	3	3	0.0936	26.4266
			−0.514	6	6	6		
TSS2	$7 \times 1 \times 1$	1	−0.152	6	6	6	1.3572	27.7370
			−0.514	5	5	5		
TSS3	$10 \times 1 \times 1$	1	−0.152	4	4	4	1051.8679	28.9382
			−0.514	12	12	12		
TSS4	$5 \times 1 \times 2$	1	−0.152	2	2	2	0.0936	29.3126
			−0.514	5	5	5		
TSS5	$7 \times 1 \times 2$	1	−0.152	5	5	5	1.7784	33.4154
			−0.514	2	2	2		
TSS6	$10 \times 1 \times 2$	1	−0.152	6	4	3	1450.0448	38.0018
			−0.514	3	2	1		
TSS7	$5 \times 1 \times 1$	3	−0.152	2	2	2	0.0780	33.8210
			−0.514	1	1	1		
TSS8	$7 \times 1 \times 1$	3	−0.152	5	5	5	1.4664	30.1394
			−0.514	4	4	4		
TSS9	$10 \times 1 \times 1$	3	−0.152	12	11	11	1154.5790	30.5450
			−0.514	6	6	6		
TSS10	$5 \times 1 \times 2$	3	−0.152	1	1	1	0.1092	36.2546
			−0.514	1	1	1		
TSS11	$7 \times 1 \times 2$	3	−0.152	5	5	5	2.0748	31.2158
			−0.514	3	3	3		
TSS12	$10 \times 1 \times 2$	3	−0.152	5	5	5	1654.4686	34.5074
			−0.514	4	4	4		
TSS13	$5 \times 2 \times 1$	1–1	−0.152	12	12	12	0.0936	28.4234
			−0.514	5	5	5		
TSS14	$7 \times 2 \times 1$	1–1	−0.152	3	3	3	1.8720	31.4342
			−0.514	4	4	4		
TSS15	$10 \times 2 \times 1$	1–1	−0.152	10	8	7	1458.6249	34.3046
			−0.514	9	7	7		
TSS16	$5 \times 2 \times 2$	1–1	−0.152	2	2	2	0.1248	32.7602
			−0.514	3	3	3		
TSS17	$7 \times 2 \times 2$	1–1	−0.152	4	4	4	2.6988	39.8895
			−0.514	1	1	1		
TSS18	$10 \times 2 \times 2$	1–1	−0.152	2	2	2	2238.1931	39.4839
			−0.514	5	4	4		
TSS19	$5 \times 2 \times 1$	3–3	−0.152	3	3	3	0.1092	29.1324
			−0.514	3	3	3		
TSS20	$7 \times 2 \times 1$	3–3	−0.152	5	5	5	2.0592	31.5122
			−0.514	3	3	3		

Table 5 continued

Test problem	$n \times g \times L$	m^t	a^t	Number of Pareto optimal solutions	NPS	NPS in Pareto optimal	CPU time (s)	
							Enumeration algorithm	NSGA-DEA
TSS21	$10 \times 2 \times 1$	3–3	−0.152	7	6	6	1687.0104	34.7726
			−0.514	5	4	4		
TSS22	$5 \times 2 \times 2$	3–3	−0.152	1	1	1	0.1248	39.2187
			−0.514	1	1	1		
TSS23	$7 \times 2 \times 2$	3–3	−0.152	8	8	8	3.0576	35.3186
			−0.514	3	3	3		
TSS24	$10 \times 2 \times 2$	3–3	−0.152	13	12	12	2713.5281	43.2903
			−0.514	7	7	6		

Table 6 The results of qualitative metrics for small problems

Test problem	a^t	NSGA-DEA				MLPGA				No. NDS	No. subscriber solutions in NDS
		NPS	NDS	QM1 (%)	QM2 (%)	NPS	NDS	QM1 (%)	QM2 (%)		
TS1	0	9	9	100	100	8	4	50	44.4	9	4
	−0.152	12	12	100	100	9	8	88.8	66.6	12	8
	−0.514	7	7	100	100	7	7	100	100	7	7
TS2	0	7	3	42.8	60	3	2	66.6	40	5	0
	−0.152	6	6	100	100	4	0	0	0	6	0
	−0.514	13	12	92.3	92.3	10	5	50	38.4	13	4
TS3	0	3	3	100	100	3	0	0	0	3	0
	−0.152	14	14	100	93.3	9	1	11.1	6.6	15	0
	−0.514	7	6	85.7	66.6	6	3	50	33.3	9	0
TS4	0	5	5	100	100	3	2	66.6	40	5	2
	−0.152	12	12	100	100	11	9	81.8	75	12	9
	−0.514	11	11	100	100	8	5	62.5	45.4	11	5
TS5	0	3	2	66.6	66.6	2	1	50	33.3	3	0
	−0.152	5	5	100	100	9	0	0	0	5	0
	−0.514	16	16	100	94.1	12	3	25	17.6	17	2
TS6	0	5	4	80	80	2	1	50	20	5	0
	−0.152	4	4	100	80	1	1	100	20	5	0
	−0.514	8	7	87.5	77.7	5	2	40	22.2	9	0
TS7	0	2	2	100	100	2	1	50	50	2	1
	−0.152	3	3	100	100	3	3	100	100	3	3
	−0.514	14	14	100	100	14	13	92.8	92.8	14	13
TS8	0	5	5	100	100	3	1	33.3	20	5	1
	−0.152	11	10	90.9	90.9	4	2	50	18.1	11	1
	−0.514	16	13	81.2	86.6	10	3	30	20	15	1

Table 6 continued

Test problem	a^t	NSGA-DEA				MLPGA				No. NDS	No. subscriber solutions in NDS
		NPS	NDS	QM1 (%)	QM2 (%)	NPS	NDS	QM1 (%)	QM2 (%)		
TS9	0	7	7	100	100	3	0	0	0	7	0
	−0.152	3	1	33.3	33.3	3	2	66.6	66.6	3	0
	−0.514	8	4	50	40	9	6	66.6	60	10	0
TS10	0	5	3	60	75	2	1	50	25	4	0
	−0.152	14	14	100	100	14	14	100	100	14	14
	−0.514	13	10	76.9	90.9	10	8	80	72.7	11	7
TS11	0	3	0	0	0	6	6	100	100	6	0
	−0.152	6	6	66.6	100	4	0	0	0	6	0
	−0.514	20	18	90	78.2	14	7	50	30.4	23	2
TS12	0	6	4	66.6	80	1	1	100	20	5	0
	−0.152	1	1	100	100	2	0	0	0	1	0
	−0.514	2	2	100	100	5	0	0	0	2	0
TS13	0	6	6	100	100	6	6	100	100	6	6
	−0.152	9	9	100	100	9	9	100	100	9	9
	−0.514	11	9	81.8	90	9	9	100	90	10	8
TS14	0	9	9	100	90	5	1	20	10	10	0
	−0.152	6	3	50	60	6	2	33.3	40	5	0
	−0.514	10	9	90	64.2	12	12	100	85.7	14	7
TS15	0	7	7	100	100	6	0	0	0	7	0
	−0.152	3	0	0	0	2	2	100	100	2	0
	−0.514	4	1	25	20	8	4	50	80	5	0
TS16	0	7	7	100	100	6	5	83.3	71.4	7	5
	−0.152	3	3	100	100	2	2	100	66.6	3	2
	−0.514	5	5	100	100	5	5	100	100	5	5
TS17	0	7	0	0	0	5	5	100	100	5	0
	−0.152	4	4	100	100	4	0	0	0	4	0
	−0.514	6	6	100	100	5	1	20	16.6	6	1
TS18	0	5	5	100	100	3	0	0	0	5	0
	−0.152	6	6	100	85.7	4	1	25	14.2	7	0
	−0.514	5	5	100	100	6	0	0	0	5	0

Table 7 The results of qualitative metrics for medium problems

Test problem	a^t	NSGA-DEA				MLPGA				No. NDS	No. subscriber solutions in NDS
		NPS	NDS	QM1 (%)	QM2 (%)	NPS	NDS	QM1 (%)	QM2 (%)		
TM1	0	1	1	100	100	8	0	0	0	1	0
	−0.152	21	19	90.4	76	10	6	60	24	25	0
	−0.514	11	11	100	100	7	0	0	0	11	0
TM2	0	5	5	100	100	4	0	0	0	5	0
	−0.152	16	15	93.7	83.3	8	3	37.5	16.7	18	0
	−0.514	11	11	100	100	11	0	0	0	11	0
TM3	0	4	3	75	60	3	2	66.36	40	5	0
	−0.152	8	8	100	100	4	0	0	0	8	0
	−0.514	14	14	100	100	20	0	0	0	14	0
TM4	0	5	3	60	42.8	5	4	80	57.2	7	0
	−0.152	6	6	100	75	8	2	25	25	8	0
	−0.514	9	9	100	100	6	0	0	0	9	0
TM5	0	2	2	100	66.6	6	1	16.6	33.4	3	0
	−0.152	7	7	100	70	5	3	60	30	10	0
	−0.514	10	10	100	100	9	0	0	0	10	0
TM6	0	2	2	100	100	6	0	0	0	2	0
	−0.152	6	5	83.3	83.3	5	1	20	16.7	6	0
	−0.514	13	13	100	100	8	0	0	0	13	0
TM7	0	3	3	100	100	2	0	0	0	3	0
	−0.152	3	3	100	60	5	2	40	40	5	0
	−0.514	22	19	86.3	95	11	1	9	5	20	0
TM8	0	5	0	0	0	3	3	100	100	3	0
	−0.152	2	2	100	100	7	0	0	0	2	0
	−0.514	12	12	100	100	14	0	0	0	12	0
TM9	0	5	5	100	100	6	0	0	0	5	0
	−0.152	3	3	100	100	4	0	0	0	3	0
	−0.514	26	18	69.2	81.8	8	4	50	18.2	22	0
TM10	0	4	4	100	100	3	0	0	0	4	0
	−0.152	12	10	83.3	71.4	8	4	50	28.6	14	0
	−0.514	20	11	55	55	12	9	75	45	20	0
TM11	0	2	2	100	100	1	0	0	0	2	0
	−0.152	5	5	100	100	7	0	0	0	5	0
	−0.514	7	7	100	100	9	0	0	0	7	0
TM12	0	14	10	71.4	76.9	3	3	100	23.1	13	0
	−0.152	5	5	100	100	8	0	0	0	5	0
	−0.514	9	9	100	100	10	0	0	0	9	0
TM13	0	8	8	100	100	3	0	0	0	8	0
	−0.152	6	2	33.3	66.6	3	1	33.3	33.4	3	0
	−0.514	20	20	100	100	9	0	0	0	20	0

Table 7 continued

Test problem	a^t	NSGA-DEA				MLPGA				No. NDS	No. subscriber solutions in NDS
		NPS	NDS	QM1 (%)	QM2 (%)	NPS	NDS	QM1 (%)	QM2 (%)		
TM14	0	6	3	50	42.8	4	4	100	57.2	7	0
	−0.152	6	6	100	100	5	0	0	0	6	0
	−0.514	10	10	100	83.3	6	2	33.3	16.7	12	0
TM15	0	3	2	66.6	50	2	2	100	50	4	0
	−0.152	7	7	100	100	10	0	0	0	7	0
	−0.514	15	15	100	100	8	0	0	0	15	0
TM16	0	5	0	0	0	1	1	100	100	1	0
	−0.152	12	12	100	100	5	0	0	0	12	0
	−0.514	16	15	93.7	83.3	14	3	21.4	16.7	18	0
TM17	0	2	2	100	100	6	0	0	0	2	0
	−0.152	6	6	100	100	7	0	0	0	6	0
	−0.514	5	5	100	100	6	0	0	0	5	0
TM18	0	1	1	100	50	5	1	20	50	2	0
	−0.152	6	5	83.3	83.3	5	1	20	16.7	6	0
	−0.514	8	8	100	100	9	0	0	0	8	0

Table 8 The results of qualitative metrics for large problems

Test problem	a^t	NSGA-DEA				MLPGA				No. NDS	No. subscriber solutions in NDS
		NPS	NDS	QM1 (%)	QM2 (%)	NPS	NDS	QM1 (%)	QM2 (%)		
TL1	0	5	5	100	83.3	4	1	25	16.7	6	0
	−0.152	3	3	100	100	6	0	0	0	3	0
	−0.514	12	12	100	100	8	0	0	0	12	0
TL2	0	2	2	100	100	6	0	0	0	2	0
	−0.152	12	11	91.6	91.6	4	1	25	8.4	12	0
	−0.514	11	11	100	100	8	0	0	0	11	0
TL3	0	1	1	100	100	2	0	0	0	1	0
	−0.152	11	11	100	100	12	0	0	0	11	0
	−0.514	23	23	100	100	8	0	0	0	23	0
TL4	0	8	8	100	100	13	0	0	0	8	0
	−0.152	12	12	100	100	9	0	0	0	12	0
	−0.514	12	12	100	100	8	0	0	0	12	0
TL5	0	6	6	100	66.6	4	3	75	33.7	9	0
	−0.152	4	4	100	100	6	0	0	0	4	0
	−0.514	8	8	100	100	5	0	0	0	8	0
TL6	0	3	3	100	100	7	0	0	0	3	0
	−0.152	8	8	100	100	6	0	0	0	8	0

Table 8 continued

Test problem	a^t	NSGA-DEA				MLPGA				No. NDS	No. subscriber solutions in NDS
		NPS	NDS	QM1 (%)	QM2 (%)	NPS	NDS	QM1 (%)	QM2 (%)		
	−0.514	7	7	100	100	7	0	0	0	7	0
TL7	0	2	2	100	33.3	5	4	80	66.7	6	0
	−0.152	5	4	80	50	6	4	66.6	50	8	0
	−0.514	7	7	100	100	7	0	0	0	7	0
TL8	0	2	2	100	50	3	2	66.6	50	4	0
	−0.152	4	4	100	100	6	0	0	0	4	0
	−0.514	3	3	100	100	8	0	0	0	3	0
TL9	0	3	3	100	100	6	0	0	0	3	0
	−0.152	3	3	100	100	4	0	0	0	3	0
	−0.514	6	6	100	100	8	0	0	0	6	0
TL10	0	1	1	100	100	2	2	100	0	3	0
	−0.152	6	6	100	100	1	0	100	0	6	0
	−0.514	11	11	100	100	4	0	0	0	11	0
TL11	0	1	1	100	50	4	1	25	50	2	0
	−0.152	6	6	100	100	4	0	0	0	6	0
	−0.514	8	8	100	100	10	0	0	0	8	0
TL12	0	3	3	100	75	2	1	50	25	4	0
	−0.152	6	6	100	100	4	0	0	0	6	0
	−0.514	10	10	100	100	8	0	0	0	10	0
TL13	0	8	4	50	57.1	3	3	100	42.9	7	0
	−0.152	14	12	85.7	75	4	4	100	25	16	0
	−0.514	11	0	0	0	5	5	100	100	5	0
TL14	0	6	2	33.3	40	3	3	100	60	5	0
	−0.152	6	6	100	100	4	0	0	0	6	0
	−0.514	6	6	100	100	9	0	0	0	6	0
TL15	0	5	5	100	100	5	0	0	0	5	0
	−0.152	8	8	100	100	2	0	0	0	8	0
	−0.514	4	4	100	50	5	4	80	50	8	0
TL16	0	5	5	100	100	7	0	0	0	5	0
	−0.152	7	7	100	100	1	0	0	0	7	0
	−0.514	12	12	100	100	6	0	0	0	12	0
TL17	0	5	5	100	100	4	0	0	0	5	0
	−0.152	3	3	100	100	4	0	0	0	3	0
	−0.514	3	3	100	100	14	0	0	0	3	0
TL18	0	4	4	100	100	2	0	0	0	4	0
	−0.152	5	5	100	100	4	0	0	0	5	0
	−0.514	19	4	21	44.4	10	5	50	55.6	9	0

Table 9 The results of quantitative metrics for small problems

Test problem	a^t	NSGA-DEA				MLPGA			
		MID	SNS	TM	FDH	MID	SNS	TM	FDH
TS1	0	266.1358	15.7851	280,089.0734	1.0000	262.0381	7.9851	238,993.7580	0.9814
	−0.152	200.5774	10.5846	263,064.9617	1.0000	198.4020	2.6784	134,753.0116	0.9847
	−0.514	136.6854	2.5192	39,601.3945	1.0000	136.6854	2.5192	39,601.3945	1.0000
TS2	0	996.2419	37.7760	1,079,315.2693	0.9959	976.7972	10.8638	603,755.9771	0.9987
	−0.152	536.3882	5.9054	238,822.7689	1.0000	545.3941	9.8325	363,421.9047	0.9928
	−0.514	174.2293	9.9858	139,838.1315	0.9999	171.6694	5.6385	99,189.2909	0.9931
TS3	0	1703.7915	81.2493	9,243,077.8740	1.0000	1668.3479	3.7716	1,425,356.0627	0.9909
	−0.152	775.9678	23.1398	890,037.9245	1.0000	814.2688	49.1956	1,806,641.8274	0.9863
	−0.514	180.0442	0.5929	81,420.5903	0.9998	180.4355	0.6663	89,678.7220	0.9992
TS4	0	1192.9237	29.6204	1,207,573.0103	1.0000	1204.7247	36.9428	1,464,132.5016	0.9993
	−0.152	796.6911	36.1181	896,214.0434	1.0000	792.5763	35.1313	920,480.0979	0.9992
	−0.514	353.3546	11.6493	545,652.2688	1.0000	349.6595	8.1830	392,452.9127	0.9956
TS5	0	3491.3172	16.4461	4,741,517.5274	0.9991	3529.1938	69.8638	23,136,655.0363	0.9950
	−0.152	1983.1037	37.4524	2,154,266.1937	1.0000	2014.2607	36.1048	2,330,360.4310	0.9919
	−0.514	551.3701	9.9057	1,292,612.1569	1.0000	497.6051	15.4674	455,960.0986	0.9877
TS6	0	6510.8733	50.7687	13,175,822.1306	0.9997	6576.6835	5.5316	10,274,405.9901	0.9977
	−0.152	3678.4719	54.0228	5,062,603.6779	1.0000	3661.4757	NaN	8,807,070.0815	1.0000
	−0.514	706.1997	36.8935	1,156,024.0473	0.9964	722.8708	23.2185	1,073,314.3954	0.9923
TS7	0	923.8373	23.7366	2,016,055.5940	1.0000	937.1823	42.6092	3,376,449.5858	0.9922
	−0.152	621.7102	4.8566	466,857.8801	1.0000	621.7102	4.8566	466,857.8801	1.0000
	−0.514	356.1755	12.6380	500,354.0587	1.0000	363.8846	25.5825	994,542.6729	0.9932
TS8	0	1845.5405	28.0110	1,747,170.0793	1.0000	1855.8154	9.5283	1,454,893.0753	0.9927
	−0.152	1265.4022	11.3353	1,114,492.2947	0.9999	1285.7012	45.8776	4,701,128.7730	0.9928

Table 9 continued

Test problem	a^t	NSGA-DEA				MLPGA			
		MID	SNS	TM	FDH	MID	SNS	TM	FDH
	−0.514	651.8521	9.0038	340,854.5644	0.9998	651.1789	6.7696	312,697.5178	0.9983
TS9	0	3053.6410	51.6369	3,849,044.4310	1.0000	3089.0207	40.7530	4,680,377.4146	0.9946
	−0.152	1928.2787	17.1870	2,212,127.8068	0.9970	1920.7798	2.1276	1,571,937.7429	0.9993
	−0.514	737.2608	18.3162	862,301.5116	0.9986	732.0789	20.5589	1,217,481.1803	0.9992
TS10	0	1207.0787	17.3503	1,386,139.0967	0.9966	1220.9594	58.6336	12,371,500.5944	0.9908
	−0.152	687.2662	28.5688	1,729,566.1803	1.0000	687.2662	28.5688	1,729,566.1803	1.0000
	−0.514	410.3258	18.2480	1,857,905.3329	0.9945	413.0042	23.8567	2,311,935.3677	0.9970
TS11	0	3869.8866	9.7302	5,495,578.4002	0.9890	3930.5684	29.6343	5,378,901.1819	1.0000
	−0.152	2265.3027	8.4527	2,468,660.5733	1.0000	2270.7780	5.2224	2,339,160.0437	0.9978
	−0.514	655.9750	22.6976	684,345.3717	0.9997	650.8290	10.8921	513,940.5780	0.9971
TS12	0	9096.2560	100.2629	19,122,926.6402	0.9959	9108.7740	NaN	49,436,164.5175	1.0000
	−0.152	5746.8400	NaN	20,296,674.1950	1.0000	5878.1027	42.9203	30,862,809.7456	0.9863
	−0.514	1950.1881	12.7491	3,064,753.1323	1.0000	1959.4328	8.5745	2,114,506.0679	0.9963
TS13	0	1174.6096	11.5978	802,462.6325	1.0000	1174.6096	11.5978	802,462.6325	1.0000
	−0.152	952.1678	11.5167	557,580.0582	1.0000	952.1678	11.5167	557,580.0582	1.0000
	−0.514	641.8675	20.3992	787,230.7975	0.9991	639.8793	19.3468	702,205.3178	1.0000
TS14	0	2168.3827	39.1509	2,167,953.3682	1.0000	2185.6491	28.5274	2,031,578.3979	0.9957
	−0.152	1480.7974	12.1935	1,986,127.2378	0.9981	1479.2622	1.9875	1,246,514.7476	0.9975
	−0.514	743.3034	17.0352	690,748.4060	0.9999	747.2396	18.1018	753,062.5172	1.0000
TS15	0	4184.2001	12.5220	4,916,128.8784	1.0000	4219.7614	41.2014	4,742,633.3489	0.9934
	−0.152	2920.2502	8.2622	2,521,176.0584	0.9955	2902.4176	2.3709	2,880,558.4840	1.0000
	−0.514	1561.1438	16.9614	4,442,056.9918	0.9967	1550.6114	5.1334	854,629.5377	0.9988
TS16	0	1931.9106	23.4720	2,111,314.5564	1.0000	1939.2625	18.7871	2,135,231.3126	0.9990

Table 9 continued

Test problem	a^t	NSGA-DEA				MLPGA			
		MID	SNS	TM	FDH	MID	SNS	TM	FDH
	−0.152	1482.5374	13.9019	4,061,409.4998	1.0000	1474.6962	4.1977	1,959,314.0136	1.0000
	−0.514	1160.9814	4.6923	941,625.2883	1.0000	1160.9814	4.6923	941,625.2883	1.0000
TS17	0	4110.7886	21.4025	5,994,394.3896	0.9967	4087.9844	31.3972	6,030,776.0139	1.0000
	−0.152	2404.1432	16.0722	3,013,827.2206	1.0000	2414.8421	23.7725	4,338,569.1175	0.9971
	−0.514	1287.8536	4.8490	1,894,035.7819	1.0000	1289.1874	5.0888	1,422,381.9741	0.9993
TS18	0	8584.8137	17.9243	17,939,001.5822	1.0000	8644.7265	30.3452	19,103,062.4079	0.9961
	−0.152	5338.3758	42.4800	11,495,680.1979	1.0000	5382.5848	62.1141	10,396,568.1530	0.9976
	−0.514	2200.7528	31.2874	7,278,470.4568	1.0000	2229.3678	9.4704	3,483,158.0455	0.9918

It denotes that there is only one solution in the non-dominated solution set if the value of SNS is NaN for a given algorithm

Table 10 The results of quantitative metrics for medium problems

Test problem	a^t	NSGA-DEA				MLPGA			
		MID	SNS	TM	FDH	MID	SNS	TM	FDH
TM1	0	10,852.9570	NaN	78,646,333.5501	1.0000	11,188.0601	158.8767	49,916,804.2884	0.9774
	−0.152	3706.5650	169.6497	22,900,224.5645	0.9997	3678.5616	42.8432	13,686,309.1806	0.9988
	−0.514	825.6750	16.3120	3,330,634.3635	1.0000	830.6438	6.2640	2,023,302.2304	0.9921
TM2	0	23,904.4579	160.4953	106,645,631.0502	1.0000	24,492.0274	131.8207	142,014,293.6225	0.9922
	−0.152	9831.0669	151.6905	39,673,982.0983	0.9999	9705.3621	51.8365	30,612,497.5266	0.9993
	−0.514	946.4366	7.6283	1,429,337.0139	1.0000	987.7061	35.5327	4,937,224.9667	0.9680
TM3	0	32,675.7527	405.8700	389,792,153.1448	0.9986	32,921.8946	125.2076	163,065,573.2219	0.9983
	−0.152	13,523.9723	153.9895	59,570,305.6549	1.0000	13,549.1050	38.8917	46,082,758.4218	0.9957
	−0.514	1069.7943	52.2725	4,846,927.4842	1.0000	1077.2657	43.2847	4,491,951.3539	0.9795
TM4	0	22,131.1918	195.4820	201,836,469.7577	0.9994	22,318.0924	104.2908	220,673,936.3627	0.9998
	−0.152	8165.5135	101.4185	56,700,893.9915	1.0000	8279.4952	136.1012	53,512,784.8253	0.9951
	−0.514	1057.7645	8.2335	902,243.7767	1.0000	1050.5268	7.1135	14,129,339.4988	0.9782
TM5	0	27,161.8840	23.6702	374,535,965.3289	1.0000	27,793.2696	105.6728	170,174,282.8310	0.9886
	−0.152	7418.8830	192.4921	49,756,491.0427	1.0000	7681.9189	194.6884	66,377,320.4187	0.9974
	−0.514	1041.4546	2.3036	2,821,109.2626	1.0000	1047.1847	6.4197	804,899,949.8936	0.9836
TM6	0	54,334.1787	58.9789	593,570,892.8920	1.0000	54,946.0568	148.2810	388,185,656.6241	0.9923
	−0.152	23,173.6421	54.05,115	127,902,591.9618	0.9998	23,225.2186	84.9662	124,261,781.4858	0.9979
	−0.514	1684.12,553	43.9665	7,810,152.7845	1.0000	1690.0935	9.4263	9,570,772.5037	0.9964
TM7	0	14,425.5281	57.5888	64,020,071.1590	1.0000	14,703.0147	73.4664	64,413,912.0751	0.9912
	−0.152	5446.2476	5.0800	16,745,512.4597	1.0000	5533.5935	60.2652	27,868,763.6082	0.9962
	−0.514	1514.3279	57.0408	6,188,006.2556	0.9999	1521.8205	61.3354	11,928,288.0049	0.9933
TM8	0	24,155.4467	278.7066	398,892,110.6032	0.9853	23,652.5158	29.2865	173,214,980.5707	1.0000
	−0.152	9706.6728	14.6909	35,072,650.9237	1.0000	10,227.0573	180.9320	110,214,930.0933	0.9718

Table 10 continued

Test problem	a^t	NSGA-DEA				MLPGA			
		MID	SNS	TM	FDH	MID	SNS	TM	FDH
	−0.514	1657.9916	20.5447	4,051,482.1561	1.0000	1746.0926	32.9163	6,353,707.6242	0.9751
TM9	0	32,744.0876	92.7674	181,884,079.6214	1.0000	33,425.9059	207.8011	231,440,062.5222	0.9905
	−0.152	12,298.1237	17.1767	85,039,357.8913	1.0000	12,648.1705	163.6660	69,144,290.6273	0.9825
	−0.514	970.2799	16.2362	4,522,901.7409	0.9960	969.8522	20.8881	7,401,083.4515	0.9878
TM10	0	22,494.3280	96.8662	217,401,720.2787	1.0000	22,885.9709	10.0926	287,184,805.4164	0.9919
	−0.152	8355.2100	161.1220	93,745,124.0054	0.9988	8449.6293	127.8910	60,100,415.2642	0.9972
	−0.514	1696.5551	51.8419	16,263,030.5112	0.9985	16,893.253	48.1513	18,203,360.5958	0.9983
TM11	0	35,708.0960	5.2559	1,311,524,815.2109	1.0000	35,705.1930	NaN	752,355,844.7309	0.9986
	−0.152	13,537.2200	323.5107	269,867,461.5750	1.0000	13,690.4361	75.9402	79,585,960.3005	0.9882
	−0.514	1862.2041	25.9661	17,116,831.7250	1.0000	1915.7762	18.0762	10,356,624.1891	0.9872
TM12	0	57,596.7927	349.1107	447,155,882.2897	0.9995	57,948.1315	48.7616	430,796,908.0620	1.0000
	−0.152	25,936.0533	48.2542	131,102,988.0405	1.0000	26,060.3519	263.5341	268,506,170.2668	0.9958
	−0.514	4562.0424	89.9129	25,777,361.1229	1.0000	4716.1458	92.9610	66,712,134.7305	0.9873
TM13	0	20,297.8754	127.6906	104,166,953.3016	1.0000	20,518.2108	308.3884	709,407,025.5701	0.9949
	−0.152	11,570.1525	90.7026	51,852,167.53417	0.9970	11,494.0387	45.9706	40,146,263.4540	0.9962
	−0.514	5250.5192	105.0104	16,977,196.5079	1.0000	5435.8667	46.8929	13,154,079.5353	0.9914
TM14	0	31,002.9656	356.2363	231,978,376.5327	0.9984	31,154.9466	164.4854	359,162,575.1897	1.0000
	−0.152	16,208.3429	56.4086	56,666,803.9866	1.0000	16,455.4258	85.4053	57,705,182.1644	0.9891
	−0.514	4545.1540	38.6288	9,554,546.7775	1.0000	4551.7246	71.6496	27,344,657.0400	0.9961
TM15	0	37,920.4567	160.6877	219,665,948.1403	0.9997	38,095.4929	69.68302	240,783,831.3610	1.0000
	−0.152	19,062.4102	251.8329	162,937,598.6855	1.0000	19,305.1784	110.8892	89,020,610.3701	0.9948
	−0.514	5133.5270	109.5157	31,805,239.9165	1.0000	5174.5465	144.1170	35,858,464.3149	0.9958

Table 10 continued

Test problem	a^t	NSGA-DEA				MLPGA			
		MID	SNS	TM	FDH	MID	SNS	TM	FDH
TM16	0	25,213.3938	103.0682	182,645,420.1173	0.9906	24,971.6102	NaN	384,745,494.6279	1.0000
	−0.152	10,513.8684	209.4380	81,592,376.8674	1.0000	10,606.8660	136.6562	71,407,541.5943	0.9952
	−0.514	3323.9551	57.4561	19,925,438.5566	0.9998	3319.7061	50.5494	19,769,036.7787	0.9983
TM17	0	440,949.0174	11.1398	1,658,509,476.6430	1.0000	44,575.6078	129.6776	332,358,275.0061	0.9923
	−0.152	20,113.9777	274.2202	170,732,663.9841	1.0000	20,109.1605	160.6352	102,587,026.4339	0.9928
	−0.514	6083.4712	65.9855	55,227,028.2389	1.0000	6254.1138	124.3830	47,965,621.3856	0.9879
TM18	0	64,771.5289	NaN	2,344,994,206.6256	1.0000	65,705.9379	131.0203	515,514,895.2597	0.9921
	−0.152	32,195.8799	158.3431	223,856,962.4509	0.9997	32,303.2893	48.8369	184,670,322.5775	0.9972
	−0.514	7379.6136	42.6015	25,556,902.0338	1.0000	7802.7448	41.7613	27,291,922.8540	1.0000

Table 11 The results of quantitative metrics for large problems

Test problem	a^t	NSGA-DEA				MLPGA			
		MID	SNS	TM	FDH	MID	SNS	TM	FDH
TL1	0	21,193.4830	728.2611	738,171,761.0785	1.0000	21,286.2134	283.4286	672,733,197.5959	0.9912
	−0.152	7730.7169	18.1497	53,931,190.1987	1.0000	8111.1134	217.2574	161,207,037.3999	0.9667
	−0.514	3218.1809	5.0228	15,361,852.0876	1.0000	3370.4712	65.8729	51,694,528.5372	0.9379
TL2	0	47,624.6427	411.8999	738,158,185.8804	1.0000	50,294.9923	295.7111	715,402,889.4719	0.9716
	−0.152	16,665.6391	586.2144	594,981,855.5267	0.9990	17,057.5800	269.5364	287,943,244.3858	0.9939
	−0.514	4034.5953	65.0076	83,401,491.8286	1.0000	4299.4175	78.3062	54,336,657.6962	0.9644
TL3	0	115,854.1278	NaN	7,564,701,904.3513	1.0000	116,439.1471	925.3113	3,905,027,971.5806	0.9973
	−0.152	38,469.2686	460.3596	465,781,226.4845	1.0000	40,184.6880	1322.2730	968,628,173.0634	0.9795
	−0.514	4630.5754	127.8674	63,095,299.8051	1.0000	4634.3906	35.0480	47,711,992.0466	0.9921
TL4	0	19,667.2896	691.6128	668,530,920.3565	1.0000	20,422.7240	1729.9436	1,407,852,262.5446	0.9785
	−0.152	7434.0031	348.5592	231,968,262.7321	1.0000	7488.8738	388.8035	257,980,076.2933	0.9859
	−0.514	4045.0138	32.1844	116,220,694.0179	1.0000	4162.2743	61.2742	121,931,935.6734	0.9515
TL5	0	73,858.1744	438.6632	1,517,635,816.9938	1.0000	76,080.7519	23,916.0513	1,605,238,163.2682	0.9976
	−0.152	26,212.4832	944.4669	1,062,408,539.6376	1.0000	26,469.0695	349.0000	602,458,805.4359	0.9876
	−0.514	5618.4867	30.2314	56,469,260.8676	1.0000	6289.2274	50.3864	115,898,837.2796	0.9184
TL6	0	128,253.9841	465.4005	2,472,805,969.7049	1.0000	132,457.6101	791.8036	2,817,728,536.9828	0.9821
	−0.152	41,227.9067	305.5547	886,772,429.1558	1.0000	43,152.4206	587.7288	3,918,287,339.1167	0.9840
	−0.514	5449.0684	35.8349	82,153,123.2123	1.0000	6561.4134	68.4980	81,543,656.8915	0.8670
TL7	0	129,090.9442	110.3268	2,447,743,595.0982	1.0000	132,747.9949	834.3018	2,995,302,611.9615	0.9981
	−0.152	45,374.1212	526.9391	983,709,817.5463	0.9995	467.3570	1055.6360	1,842,394,368.6622	0.9970
	−0.514	8898.1582	100.7803	99,910,111.7783	1.0000	9038.3986	56.9358	69,174,528.4365	0.9903
TL8	0	261,079.3250	440.3980	7,369,380,182.7820	1.0000	262,191.2855	843.7137	8,152,839,755.5554	0.9966
	−0.152	101,882.5732	1072.0853	14,865,820,129.0806	1.0000	105,116.6684	609.4816	2,633,816,550.3618	0.9806

Table 11 continued

Test problem	a^t	NSGA-DEA				MLPGA			
		MID	SNS	TM	FDH	MID	SNS	TM	FDH
	−0.514	10,149.9246	64.3114	235,678,302.7471	1.0000	11,056.5497	351.4159	253,771,767.8734	0.9498
TL9	0	372,505.7086	497.7722	20,585,628,017.2029	1.0000	376,587.3083	2790.7479	12,979,382,528.0728	0.9917
	−0.152	133,796.7686	577.1745	2,419,389,819.7609	1.0000	138,996.2219	353.0511	2,383,046,951.7183	0.9770
	−0.514	5153.9653	51.7053	157,914,014.2395	1.0000	5316.5029	84.0164	209,786,259.7288	0.9818
TL10	0	184,193.9965	NaN	19,278,390,497.0397	1.0000	182,106.7787	216.3518	4,803,168,263.2806	1.0000
	−0.152	58,171.1709	367.6011	1,036,888,837.0313	1.0000	59,743.6891	NaN	2,281,743,673.3140	0.9897
	−0.514	5875.0228	124.7028	253,130,612.8465	1.0000	5879.0700	60.9585	118,969,718.2830	0.9884
TL11	0	325,801.0814	NaN	57,982,285,988.0306	1.0000	328,078.9641	2992.9118	23,118,827,804.3225	0.9933
	−0.152	103,219.1104	494.0134	3,258,998,569.1436	1.0000	110,705.3485	758.2923	4,739,550,341.2628	0.9609
	−0.514	5954.3075	63.6033	119,008,611.3532	1.0000	6427.3528	33.6950	98,460,786.3832	0.9409
TL12	0	547,754.6373	1171.2806	23,640,250,543.1490	1.0000	551,985.2945	102.4558	43,172,076,826.0253	0.9983
	−0.152	200,884.6689	369.9210	6,386,469,460.4835	1.0000	204,748.7622	193.2211	8,534,489,511.3277	0.9872
	−0.514	6786.8562	9.9816	48,227,700.1171	1.0000	7043.2010	72.3630	187,679,872.1871	0.9692
TL13	0	146,639.0954	666.3464	3,226,791,585.1109	0.9988	147,695.3938	1133.0381	14,564,139,977.9281	1.0000
	−0.152	47,011.4024	729.8778	699,503,461.9329	0.9997	47,559.0436	252.4123	1,171,096,667.5069	1.0000
	−0.514	10,019.8500	143.3151	195,807,000.1758	0.9910	9881.7503	93.7911	355,648,701.5891	1.0000
TL14	0	264,741.4100	1692.0110	6,616,453,299.3779	0.9976	263,899.5142	556.8856	6,537,981,629.0217	1.0000
	−0.152	94,094.7765	616.0012	1,545,920,261.4242	1.0000	97,218.0899	297.6322	1,527,055,365.2051	0.9852
	−0.514	13,473.2951	70.3422	250,161,159.6256	1.0000	14,133.8948	117.6888	433,270,357.7239	0.9750
TL15	0	419,125.3515	2533.8600	28,007,092,309.2321	1.0000	419,197.2008	725.9572	20,505,159,428.8226	0.9938
	−0.152	167,707.3942	1290.7738	3,202,539,284.3409	1.0000	170,732.1053	1628.8325	27,054,455,719.8942	0.9909
	−0.514	13,919.9728	119.4342	523,993,506.9798	1.0000	14,486.8120	87.9223	283,705,754.2554	0.9975

Table 11 continued

Test problem	a^t	NSGA-DEA				MLPGA			
		MID	SNS	TM	FDH	MID	SNS	TM	FDH
TL16	0	218,275.6727	923.0126	9,172,409,231.1840	1.0000	221,307.3452	1050.2234	11,101,389,215.0555	0.9937
	−0.152	82,316.2717	464.1686	1,873,684,418.1754	1.0000	83,267.3761	NaN	4,188,199,445.8709	0.9927
	−0.514	17,390.0328	239.0716	309,173,436.9460	1.0000	18,889.5274	48.1118	184,586,371.2006	0.9625
TL17	0	377,407.0855	1704.7996	13,760,997,269.5119	1.0000	380,653.0473	763.1042	12,234,348,046.3133	0.9951
	−0.152	141,741.2480	591.0366	5,915,751,251.5991	1.0000	147,067.3358	592.3749	3,558,097,286.2295	0.9791
	−0.514	17,350.1185	107.4640	509,862,970.3050	1.0000	18,515.6845	135.5849	257,410,011.6212	0.9634
TL18	0	583,749.4538	641.9073	29,327,938,371.1075	1.0000	588,474.4627	304.0512	59,688,909,076.5570	0.9946
	−0.152	224,680.8709	5426.5238	5,204,687,202.8626	1.0000	232,745.2126	1712.2465	24,839,264,503.8831	0.9796
	−0.514	19,658.3179	208.9766	554,370,874.9112	0.9966	195,828.1025	152.2231	1,355,483,410.6485	1.0000

Table 4 shows the data which is transformed into S/N value. The rate of S/N is shown in Fig. 2. As it can be seen in Table 4, factor C (probability of crossover) is prominent in the execution process of determining NSGA-DEA. In addition, the influence of four factors on minimizing makespan and the total tardiness in NSGA-DEA is, in the order of: probability of crossover, number of initial population, number of generation, and probability of mutation. As it can be seen in Fig. 2, the optimal factors are: A (3), B (3), C (2), and D (1).

3.4 The validation of proposed algorithm

To demonstrate the validation of proposed algorithm, the experiments were conducted on special small problems. It is impossible to find the Pareto-optimal solutions using the enumeration algorithm because of extreme complexity of the problems. The full enumeration algorithm is used to find the Pareto-optimal solutions for only several special small problems. The total of possible states of problems with $n = 5$, 7 and 10 is the 120, 5040 and 3,628,800 solutions to explore the full enumeration algorithm, respectively. Details special small size problems and the results are shown in Table 5. In there, the first column indicates the abbreviation codes of each test problem, the second and third columns describe the details problems (number of jobs × number of stages × number of re-entrants, and number of machines per stage), the fourth describes learning indices, the fifth and sixth columns describe number of Pareto-optimal solutions and the number of non-dominated solutions obtained from algorithm respectively, the seventh describes the number of non-dominated solutions obtained from algorithm in Pareto-optimal solutions, and the last column describes the average CPU time (second unit). Based on the results of given in Table 5, the following observations can be made.

Due to the fifth and seventh columns, the proposed algorithm is able to find all solutions in Pareto-optimal in 62.5 % cases (the Pareto-optimal solutions exactly). In other cases, more solutions were found, except for one or two. This result indicates that the proposed algorithm has a very high reliability (excellent performance) to solve the problems. Due to the sixth and seventh columns, the all solutions obtained of proposed algorithm are efficient in 83.3 % cases, because they exist at Pareto-optimal. Due to the last column, the proposed algorithm is able to solve the problems in the length of the interval from 26.4266 to 41 s. The full enumeration algorithm has spent the interval from 0.0780 to 2713.5281 s. This result indicates that the proposed algorithm has a shorter range in solving problem.

3.5 Numerical result

The performance of the proposed NSGA-DEA is compared with a MLPGA algorithm proposed by Cho et al. (2011). It is noticeable that all of algorithms are implemented in MATLAB 2009a, and run on a PC with 2.30 GHz Intel Core and 4 GB of RAM memory. To show the efficiency and effectiveness of the proposed algorithm in comparison with a MLPGA, computational experiments were done on various test problems (i.e. small, medium and large). For each algorithm, we run each test problem ten times and four qualitative metrics and four quantitative metrics are computed for each of them. The comparisons are performed on the basis of the sets of non-dominant solutions obtained by each algorithm.

Tables 6, 7 and 8 represent the values of the four qualitative metrics for various problems. The value of NPS, NDS, QM1 and QM2 are shown in these tables. Now, the first line of Table 6 is explained. The results shown in NPS column only evaluate the number of non-dominated solutions found by each algorithm, not their quality. However, the quality of solutions can be measured by NDS, QM1 and QM2, as summarized in Tables. The values of NPS of NSGA-DEA and MLPGA are equal to 9 and 8 respectively. Consequently, the NSGA-DEA front has more non-dominated solutions than MLPGA front, corresponding the value of NPS. The values of NDS of NSGA-DEA and MLPGA are equal to 9 and 4 respectively. It shows number of the net non-dominated solutions (NDS) of each algorithm. The larger value of NDS, the better of solution quality we have. The values of QM1 of NSGA-DEA and MLPGA are equal to 100 and 50 % respectively. The value of QM1 of NSGA-DEA is calculated by divided NDS = 9 with the NPS = 9. It means that all solutions obtained of the NSGA-DEA are efficient. Consequently, there are none solutions in NSGA-DEA front that are dominated by at least one solution from MLPGA front. The value of QM1 of MLPGA is calculated by divided NDS = 4 with the NPS = 8. It means that four solutions of the MLPGA are efficient. Consequently, four solutions on the MLPGA front are dominated by at least one solution on the NSGA-DEA front. The values of QM2 of NSGA-DEA and MLPGA are equal to 100 and 44.4 % respectively. The value of QM2 of NSGA-DEA is calculated by divided "NDS of algorithm = 9" with the "NDS = 9". It means that the all solutions in NSGA-DEA front have been included in net non-dominated solutions. The value of QM2 of MLPGA is calculated by divided "NDS of algorithm = 4" with the "NDS = 9". It means that the four solutions in MLPGA front have been included in net non-dominated solutions. It is notable that, net non-dominated solutions may be both algorithms. For example, 4 solutions of net non-dominated solutions (NDS) are both algorithms.

As it can be observed, four metrics (NDS, NPS, QM1 and QM2) have better values for NSGA-DEA in comparing with other algorithm in more cases. The averages and summarized results of these experiments are shown in Table 12. As it can be seen in average row in Table 12, the number of non-dominated solutions of MLPGA is less than that of NSGA-DEA, 79 % of Pareto members of NSGA-DEA are efficient. Also, 87.1 % of the net Pareto set members made by members of the Pareto set which belongs to NSGA-DEA, while the solutions of MLPGA only cover 8.6 % of the members of the Pareto set. As shown in Table 12, the proposed algorithm is more effective than the MLPGA algorithm in terms NDS, NPS, QM1 and QM2 for small, medium and large-sized problems.

Tables 9, 10 and 11 present the values of four quantitative metrics on all test problems for each algorithm. The averages and summarized results of these experiments are shown in Table 12. As it can be seen in average row in Table 12, algorithm NSGA-DEA can obtain better performances on metrics MID and TM than other algorithm. The values of MID and TM of NSGA-DEA are less than that of other algorithm in 75.3 and 49.4 % cases respectively. It indicates that the obtained solutions of NSGA-DEA converge towards the ideal point (0, 0). It is closer to the

Table 12 The summarized results of qualitative and quantitative metrics (%)

Test problem	Algorithm	NPS	NDS	QM1	QM2	MID	SNS	TM	FDH
Small problem	NSGA-DEA	63.0	75.9	64.8	87.0	53.7	50.0	42.6	64.8
	MLPGA	14.8	11.1	20.4	11.1	35.2	38.9	46.3	20.4
	Equal	22.2	13.0	14.8	1.9	11.1	11.1	11.1	14.8
Medium problem	NSGA-DEA	59.3	88.9	85.2	88.9	79.6	55.6	50.0	90.7
	MLPGA	37.0	7.4	13.0	7.4	20.4	44.4	50.0	9.3
	Equal	3.7	3.7	1.8	3.7	0.0	0.0	0.0	0.0
Large problem	NSGA-DEA	51.9	83.3	87.0	85.2	92.6	48.1	55.6	88.9
	MLPGA	42.6	9.3	9.3	7.4	7.4	51.9	44.4	9.3
	Equal	5.5	7.4	3.7	7.4	0.0	0.0	0.0	1.8
Average	NSGA-DEA	58.1	82.7	79.0	87.1	75.3	51.2	49.4	81.5
	MLPGA	31.5	9.3	14.2	8.6	21.0	45.1	46.9	13.0
	Equal	10.4	8.0	6.8	4.3	23.7	3.7	3.7	5.5

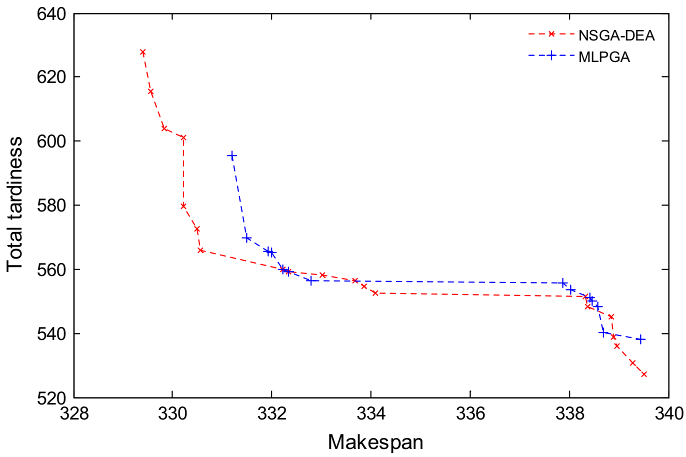

Fig. 3 *Plot* of the obtained non-dominated solutions for NSGA-DEA and MLPGA in a small problem

ideal point. Consequently, the Pareto set of NSGA-DEA algorithm is better than the other one. The values of SNS of NSGA-DEA are also more than that of other algorithm in 51.2 % cases. It indicates that the non-dominated solutions of NSGA-DEA are compacted in greater space than that of other algorithm. Similarly, the values of FDH of NSGA-DEA are also more than that of other algorithm in 81.5 % cases. It indicates that the non-dominated solutions of NSGA-DEA dominate some members of the Pareto set which belongs to MLPGA. These results show that the proposed algorithm works effectively in all size of problems. Therefore, the average values of all the metrics in Table 12 show that the NSGA-DEA is able to obtain more diversified and competitive Pareto sets than the MLPGA.

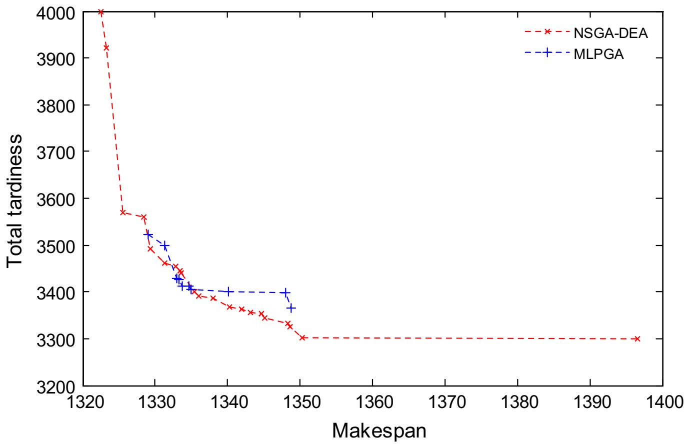

Fig. 4 *Plot* of the obtained non-dominated solutions for NSGA-DEA and MLPGA in a medium problem

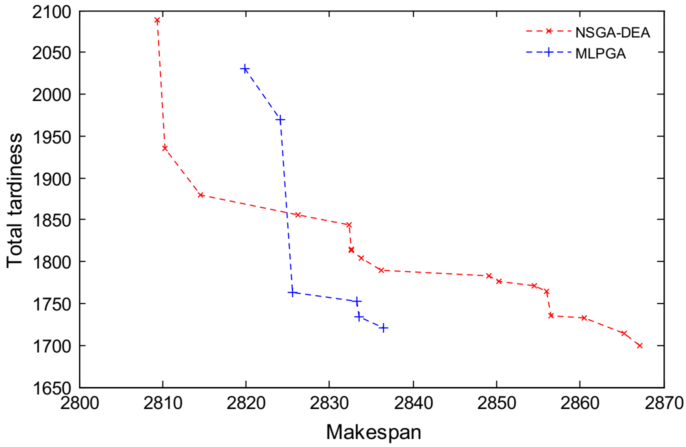

Fig. 5 *Plot* of the obtained non-dominated solutions for NSGA-DEA and MLPGA in a large problem

Graphical representation is provided to demonstrate output results of the NSGA-DEA and MLPGA. Figures 3, 4 and 5 represent the non-dominated solutions of a single run by proposed algorithm and MLPGA for problems of small, medium and large size respectively. It is obvious that the solutions obtained in Pareto-front by the NSGA-DEA algorithm are more desirable. Also, we plot the initial set and the final set of solutions obtained by each algorithm for an instance in Fig. 6. The initial set is the same for all algorithms. This figure shows that NSGA-DEA generates more efficient solutions. Therefore, these figures illustrate and confirm the conclusion derived from the numerical results based on the performance criteria.

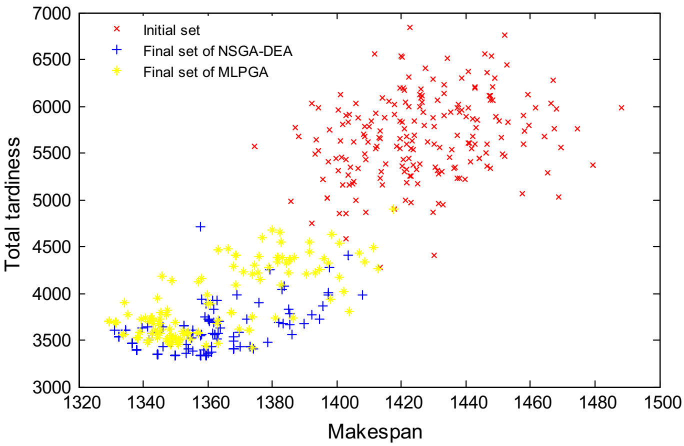

Fig. 6 Initial and final sets for NSGA-DEA and MLPGA in a medium problem

4 Conclusion and further researches

This paper considers the problem of scheduling *n* independent jobs in a hybrid flow shop with the objectives of minimizing both the makespan and the total tardiness. There are considerable numbers of practical assumptions in real world scheduling settings. To address the realistic assumptions of the proposed problem, three additional traits were added to the scheduling problem. These include re-entrant lines, setup times and position-dependent learning effects. A genetic algorithm is proposed for solving this bi-objective optimization problem. The performance of the proposed algorithm is compared with a genetic algorithm proposed in the literature on a set of test problems. Several computational tests are used to evaluate the effectiveness and efficiency of the proposed algorithm in finding good quality schedules. Computational results show that the proposed algorithm provides better results than genetic algorithm in the literature by qualitative and quantitative criteria. For future study, the scheduling with other system characteristics, which have not been included in this paper, such as release date, limited intermediate buffers, machine availability constraints, and unrelated parallel machines at each stage can be a practical extension, although the problem would be very difficult to solve.

References

Allahverdi A, Gupta JND, Aldowaisan T (1999) A review of scheduling research involving setup considerations. OMEGA Int J Manag Sci 27(2):219–239

Allahverdi A, Ng CT, Cheng TCE, Kovalyov MY (2008) A survey of scheduling problems with setup times or costs. Eur J Oper Res 187(3):985–1032

Andrés C, Albarracín JM, Tormo G, Vicens E, García-Sabater JP (2005) Group technology in a hybrid flowshop environment: a case study. Eur J Oper Res 167:272–281

Arthanary TS, Ramaswamy KG (1971) An extension of two machine sequencing problems. Oper Res 8(1):10–22
Attar SF, Mohammadi M, Tavakkoli-Moghaddam R, Yaghoubi S (2014) Solving a new multi-objective hybrid flexible flowshop problem with limited waiting times and machine-sequence-dependent set-up time constraints. Int J Comput Integr Manuf 27(5):450–469
Bagchi TP (1999) Multi-objective scheduling by genetic algorithms. Kluwer, Boston
Behnamian J, Zandieh M (2013) Earliness and tardiness minimizing on a realistic hybrid flowshop scheduling with learning effect by advanced metaheuristic. Arab J Sci Eng 38:1229–1242
Behnamian J, Fatemi Ghomi SMT, Zandieh M (2009) A multi-phase covering Pareto-optimal front method to multi-objective scheduling in a realistic hybrid flowshop using a hybrid metaheuristic. Expert Syst Appl 36:11057–11069
Biskup DA (1999) Single machine scheduling with learning considerations. Eur J Oper Res 115(1):173–178
Biskup DA (2008) A state-of-the-art review on scheduling with learning effects. Eur J Oper Res 188:315–329
Chang PC, Hsieh JC, Wang YW (2003) Genetic algorithms applied in BOPP film scheduling problems: minimizing total absolute deviation and setup times. Appl Soft Comput 3:139–148
Cho H-M, Bae S-J, Kim J, Jeong I-J (2011) Bi-objective scheduling for reentrant hybrid flow shop using Pareto genetic algorithm. Comput Ind Eng 61:529–541
Corne DW, Knowles JD, Oates MJ (2000) The Pareto envelope based selection algorithm for multi-objective optimization. In: Proceedings of the parallel problem solving from nature VI conference, Springer Lecture Notes in Computer Science, vol 1917/2000, pp 839–848
Corne DW, Jerram NR, Knowles JD, Oates MJ (2001) PESA-II: region-based selection in evolutionary multiobjective optimization. In: Proceedings of the genetic and evolutionary computation conference, (GECCO-2001). Morgan Kaufmann, San Francisco, CA, pp 283–290
Davoudpour H, Ashrafi M (2009) Solving multi-objective SDST flexible flow shop using GRASP algorithm. Int J Adv Manuf Technol 44(7–8):737–747
Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multi-objective genetic algorithm: NSGA II. IEEE Trans Evolut Comput 6(2):182–197
Dugardin F, Yalaoui F, Amodeo L (2010) New multi-objective method to solve reentrant hybrid flow shop scheduling problem. Eur J Oper Res 203:22–31
Fadaei M, Zandieh M (2013) Scheduling a bi-objective hybrid flow shop with sequence-dependent family setup times using metaheuristics. Arab J Sci Eng 38:2233–2244
Fonseca CM, Fleming PJ (1993) Genetic algorithms for multi-objective optimization: formulation, discussion and generalization. In: Proceedings of the fifth international conference on genetic algorithms, July 1993. Morgan Kaufmann, San Mateo, pp 416–423
Gupta JND (1988) Two-stage, hybrid flowshop scheduling problem. J Oper Res Soc 39(4):359–364
Hakimzadeh Abyaneh S, Zandieh M (2012) Bi-objective hybrid flow shop scheduling with sequence-dependent setup times and limited buffers. Int J Adv Manuf Technol 58:309–325
Holland JH (1975) Adaptation in natural and artificial systems. University of Michigan, Ann Arbor
Horn J, Nafpliotis N, Goldberg DE (1994) A niched Pareto genetic algorithm for multi-objective optimization. In: Proceedings of the 1st IEEE-ICEC conference, computational computation, 27–29 June 1994, Orlando, FL, vol 1, pp 82–87
Hyun CJ, Kim Y, Kim YK (1998) A genetic algorithm for multiple objective sequencing problems in mixed model assembly lines. Comput Oper Res 25(7–8):675–690
Ishibuchi H, Murata T (1998) A multi-objective genetic local search algorithm and its application to flowshop scheduling. IEEE Trans Syst Manuf Cybern 28(3):392–403
Jolai F, Asefi H, Rabiee M, Ramezani P (2013) Bi-objective simulated annealing approaches for no-wait two-stage flexible flow shop scheduling problem. Sci Iran E 20(3):861–872
Jungwattanakit J, Reodecha M, Chaovalitwongse P, Werner F (2008) Algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. Int J Adv Manuf Technol 37(3–4):354–370
Jungwattanakit J, Reodecha M, Chaovalitwongse P, Werner F (2009) A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. Comput Oper Res 36(2):358–378
Karimi N, Zandieh M, Karamooz HR (2010) Bi-objective group scheduling in hybrid flexible flowshop: a multi-phase approach. Expert Syst Appl 37(6):4024–4032

Knowles JD, Corne DW (1999) The Pareto archived evolution strategy: a new baseline algorithm for multiobjective optimization. In: Proceedings of the 1999 congress on evolutionary computation, 6–9 July, IEEE Service Center, Washington, DC, pp 98–105
Lin D, Lee CKM (2011) A review of the research methodology for the re-entrant scheduling problem. Int J Prod Res 49(8):2221–2242
Mousavi SM, Zandieh M, Amiri M (2011a) An efficient bi-objective heuristic for scheduling of hybrid flow shops. Int J Adv Manuf Technol 54(1–4):287–307
Mousavi SM, Zandieh M, Amiri M (2011b) Comparisons of bi-objective genetic algorithms for the hybrid flowshop scheduling with sequence-dependent setup times. Int J Prod Res 50(10):2570–2591
Mousavi SM, Mousakhani M, Zandieh M (2012a) Bi-objective hybrid flow shop scheduling: a new local search. Int J Adv Manuf Technol 64(5–8):933–950
Mousavi SM, Zandieh M, Yazdani M (2012b) A simulated annealing/local search to minimize the makespan and total tardiness on a hybrid flowshop. Int J Adv Manuf Technol 64(1–4):369–388
Naderi B, Zandieh M, Balagh AKG, Roshanaei V (2009a) An improved simulated annealing for hybrid flowshops with sequence-dependent setup and transportation times to minimize total completion time and total tardiness. Expert Syst Appl 36(6):9625–9633
Naderi B, Zandieh M, Roshanaei V (2009b) Scheduling hybrid flowshops with sequence-dependent setup times to minimize makespan and maximum tardiness. Int J Adv Manuf Technol 41:1186–1198
Pargar F, Zandieh M (2012) Bi-criteria SDST hybrid flow shop scheduling with learning effect of setup times: water flow-like algorithm approach. Int J Prod Res 50(10):2609–2623
Pasupathy T, Rajendran C, Suresh RK (2007) A multi-objective genetic algorithm for scheduling in flowshops to minimize the makespan and total flow time of jobs. Int J Adv Manuf Technol 27(7–8):804–815
Pinedo ML (1995) Scheduling theory, algorithms, and systems. Prentice-Hall, Englewood Cliffs
Pinedo ML (2008) Theory, algorithms, and systems, 3rd edn. Springer Science + Business Media, New York
Rashidi E, Jahandar M, Zandieh M (2010) An improved hybrid multi-objective parallel genetic algorithm for hybrid flow shop scheduling with unrelated parallel machines. Int J Adv Manuf Technol 49:1129–1139
Ruiz R, Vázquez-Rodríguez JA (2010) The hybrid flow shop scheduling problem. Eur J Oper Res 205:1–18
Ruiz-Torres AJ, Lopez FJ (2004) Using the FDH formulation of DEA to evaluate a multi-criteria problem in parallel machine scheduling. Comput Ind Eng 47:107–121
Schaffer JD (1985) Multi-objective optimization with vector evaluated genetic algorithms. In: Grefenstette J (ed) Proceedings of the international conference on genetic algorithms and their applications. Lawrence Erlbaum, Hillsdale, pp 93–100
Srinivas N, Deb K (1995) Multi-objective function optimization using non-dominated sorting genetic algorithms. IEEE Trans Evol Comput 2(3):221–248
Wang S, Liu M (2014) Two-stage hybrid flow shop scheduling with preventive maintenance using multi-objective tabu search method. Int J Prod Res 52(5):1495–1508
Wang MY, Sethi SP, Van de Velde SL (1997) Minimizing makespan in a class of reentrant shops. Oper Res 45(5):702–712
Ying KC, Lin SW, Wan SY (2014) Bi-objective reentrant hybrid flowshop scheduling: an iterated Pareto greedy algorithm. Int J Prod Res. doi:10.1080/00207543.2014.910627
Zitzler E, Thiele L (1999) Multi-objective evolutionary algorithm: a comparative case study and the strength Pareto approach. IEEE Trans Evol Comput 3(4):251–257
Zitzler E, Laumanns M, Thiele L (2001) SPEA2: improving the strength Pareto evolutionary algorithm. Report 103. Computer Engineering and Networks Laboratory (TIK), Zurich