Journal of Advances in Computer Research
Quarterly pISSN: 2345-606x eISSN: 2345-6078
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 9, No. 2, May 2018), Pages: 1-20
www.jacr.iausari.ac.ir

Evaluation of Bi-objective Scheduling Problems by FDH, Distance and Triangle Methods

S. Mostafa Mousavi✉

Department of Technical and Engineering, Faculty of Industrial Engineering, Islamic Azad University, Nowshahr Branch, Mazandaran, Iran

mousavi@iauns.ac.ir

Received: 2017/04/03; Accepted: 2017/06/05

Abstract

In this paper, two methods named distance and triangle methods are extended to evaluate the quality of approximation of the Pareto set from solving bi-objective problems. In order to use evaluation methods, a bi-objective problem is needed to define. It is considered the problem of scheduling jobs in a hybrid flow shop environment with sequence-dependent setup times and the objectives of minimizing both the makespan and the total tardiness. The bi-objective genetic algorithm in literature is applied to solve this problem belongs to NP-hard class. In the structure of algorithm, 3 and 4 alternatives for dispatching rules and neighborhood search structure have been introduced respectively. Therefore, twelve algorithms are derived from a combination of dispatching rules and neighborhood search structures. After the execution of algorithms, efficient sets are compared through several evaluation methods. Computational results show that the FIFO rule is the best alternative for the dispatching rule in order to find the job sequence for the second to end stages.

Keywords: *Data Envelopment Analysis, Distance Method, Triangle Method, Bi-Objective Problem*

1. Introduction

The main topic of this paper is to evaluate the performance of approximation of the Pareto set obtained of bi-objective scheduling problems by several evaluation methods. In order to use evaluation methods, a bi-objective scheduling problem is needed to define. Therefore, a bi-objective hybrid flow shops (HFS) is considered for research. It should be noted that any other scheduling problem could be considered.

Production scheduling plays a key role in the manufacturing systems of enterprises for maintaining a competitive position in fast-changing markets. One of the most applied and recognized scheduling problem is HFSs which have various applications in real world industries. The HFS scheduling problem involves the sequence of jobs in a flow shop in which, at any stage, there might be two or more identical machines [1]. Also, the setup time of a job is required when a switch between two different jobs occurs. After completing processing of one job and before beginning processing of the next job, some sort of setup must be performed. The length of time required to do the setup depends on both the prior and the current job to be processed; that is, the setup times are sequence-dependent.

In the n-job g-stage HFS scheduling problem, there are m_i identical machines at stage i, where i = 1, 2, 3,…,g and a set of n simultaneously available jobs to be processed sequentially on g stages of a production facility. To begin, a job sequence is determined according to a particular sequencing rule or random way. Then, jobs are assigned to the machines at stage 1 using the determined job sequence. The jobs arrive at stage 1 where the corresponding operations are performed and the jobs are delivered to stage 2 for the completion of succeeding operations. The sequence of jobs for the other stages, i.e., from stage two to stage g, determine according to dispatching rules. Consequently, the first discussion in this paper is that among dispatching rules which one will be more proper. Hence, several dispatching rules should test to determine the sequence of jobs from two to end stage. In Table 1 some abbreviations have been presented that express specific title, and in Table 2 several dispatching rules have been shown. The rules 1 to 4 are single-attribute rules, and rules 5 to 8 are multi-attribute rules [2].

Table 1. Abbreviations of specific title

Specific title	Abbreviation
Arrival time of each job	AT*
Due date of a job	DD
Processing time of each job	PT
Current time of each job	CT
Remaining processing time of each job	RPT
Processing time modified of each job	$\tilde{p}$

*It is assumed in this paper that AT is zero for all jobs.

Table 2. Candidate dispatching rules

Name	Description	Criteria
FIFO	First In First Out	First in first out
SPT	Shortest Processing Time	Shortest processing time first
EDD	Earliest Due Date	Earliest due date first
SRPT	Shortest Remaining Processing Time	Shortest remaining processing time first
AT-RPT	CT – AT + RPT	Largest first
CR	Critical Ratio (CR) = (DD – CT)/RPT	Smallest first
SL	Slack Time (SL) = DD –CT – RPT	Smallest first
MDD	Modified Due Date=max{CT + $\tilde{p}$, DD}	Largest first

In order to determine remaining process time, in addition to processing time, there is setup time which related to the sequence of jobs. Since the jobs sequence of stage 2 to end is not clear, the formula obtained by Kurz and Askin [3] are used as follows:

n = Number of jobs to be scheduled

g = Number of serial stages

s_{ij}^{t} = setup time between job j and job i at stage t while job j is scheduled immediately after job i

s_{0j}^{t} = setup time job j at stage t when job j is assigned to a machine at the first position

p_i^t = Processing time for job i at stage t

$\tilde{p}_i^t$ = Modified processing times for job i in stage t (this time represents the minimum time at a stage t that must elapse before job i could be completed.)

RPT_i^k = The sum of modified processing times for job i of stage k to stage g (k=2,…,g)

$$\tilde{p}_i^t = p_i^t + \min_{j\in\{0,1,..,n\},\, j\neq i} s_{ji}^t \quad \text{for} \quad t = 2, 3, \ldots, g \quad i = 1, 2, \ldots, n \tag{1}$$

$$RPT_i^k = \sum_{t=k}^{g} (p_i^t + \min_{j \in \{0,1,..,n\}, j \neq i} s_{ji}^t) = \sum_{t=k}^{g} \tilde{p}_i^t \quad \text{for } k = 2, 3, \ldots, g \quad i = 1, 2, \ldots, n \qquad (2)$$

The objective of scheduling is to assign jobs to the machines at the corresponding stages and determine the processing sequences on the machines so that one or some selected criteria are optimized. Scheduling problems have often a multi-objective nature. Therefore, the analysis of the performance of a schedule often involves more than one aspect and requires a multi-objective treatment. The scheduling objectives in such industries may vary. The maximum completion time (makespan or $C_{\max}$) criterion has been chosen as one of the objectives in the present article. Nowadays, many companies are concerned with meeting customers' demand in terms of due dates and, therefore, scheduling problems with due date-related measures have more practical meaning than before. Consequently, another criterion namely total tardiness ($\overline{T}$) is considered simultaneously. The simultaneous consideration of these objectives is the bi-objective optimization problem under study. According to the simultaneous approach, alternatively named Pareto approach, both criteria are considered as primary objectives and the target is finding efficient solutions. The bi-objective HFS problem with sequence-dependent setup times (SDST) has been investigated in several studies [4-9]. Also, Barzegar and Motameni [10] considered the flexible job shop scheduling problem that is one of the most general and difficult of all traditional scheduling problems. Tavakkoli-Moghaddam et al. [11] provided a summary of some basic definitions in order to better understand the multi-objective problems (MOP).

The remainder of the paper is organized as follows: The bi-objective genetic algorithm and the details of each step are explained in Section 2. In Section 3; methods are introduced which have been used to evaluate the quality of efficient solutions. The computational results and numerical comparisons are reported in Section 4. Finally, Section 5 consist conclusions and future work.

2. Bi-Objective Genetic algorithm

The aim of this paper is to compare non-dominated solutions obtained with several evaluation methods. Therefore, the algorithm in literature named bi-objective genetic algorithm (GA) [5] is applied to achieve non-dominated solutions for the scheduling problem. GA-based optimization techniques were first used by Holland [12] and have been applied widely to solve both single-objective and MOPs. For example, Barzegar et al. [13] proposed a genetic algorithm for the open shop scheduling with Makespan and total completion time.

In this section, the bi-objective GA is presented for solving the *n*-job *g*-stage HFS scheduling problem. In bi-objective GA, probability function of the sum of weighted is applied to combine objectives. Also, neighborhood operator (step 10) is used to generate a part of solutions of the next generation. Consequently, the second discussion in this paper is that among neighborhood search structures (NSS) which one will be more proper for neighborhood operator. Hence, several NSSs should examine so that the best is selected. In Table 3 several NSSs have been shown. A candidate solution is presented by its configuration vector $X=(x_1,\ldots,x_n)$ and denote by π_i a subsequence of X of arbitrary length, by (x_i) the subsequence consisting of a single configuration x_i, and by ''**.**'' the concatenation operator.

Table 3. Candidate neighborhood search structures

Name	Description
Swap move	$SWP(X,i,j)=SWP(\pi_1.(x_i).\pi_2.(x_j).\pi_3,i,j)=\pi_1.(x_j).\pi_2.(x_i).\pi_3.$
Shift move (backward)	$BSH(\pi_1.(x_i).\pi_2.(x_j).\pi_3,i,j)=\pi_1.(x_j).(x_i).\pi_2.\pi_3.$
Shift move (forward)	$FSH(\pi_1.(x_i).\pi_2.(x_j).\pi_3,i,j)=\pi_1.\pi_2.(x_j).(x_i).\pi_3.$
Neighborhood swapping	$\eta(X)=\{X':X'=SWP(X,i,j),\ i=1,...,n-1,\quad j=i+1,...,n\}$
Random insertion scheme	$RIS(X,i,j)=RIS(\pi_1.(x_i)\pi_2(x_j).\pi_3,i,j)=\pi_1.\pi_2(x_i)(x_j).\pi_3.$

The steps of bi-objective GA are shown for solving the n-job g-stage m_i-identical parallel machines in each stage SDST HFS scheduling problem below.

Step 1: Encoding

Integer coding is used in this research. For example a solution consists of 5-bit as 3-1-4-2-5 denotes that job 3 is process first, and then job 1, job 4, job 2, job 5 are processed successively. For doing the order of jobs, from the second stage to next, the dispatching rules have been used.

Step 2: Initialization:

Parameters setting: Set the number of initial population (np), Number of generation (ng), Probability of crossover (P_c), Probability of mutation (P_m), Probability of reproduction (P_r), Probability of NSS (P_n).

Generate initial population: Initial solutions are randomly generated and these initial solutions form the first population.

Step 3: Record efficient Solutions

Calculate the objective values of chromosomes in the population and record the efficient solutions.

Step 4: Calculate Objective Value

The total objective function is constituted of the linear combination of objective functions. And the certain weights are assigned. For a solution x, the objective function in the study is represented as follows:

$$f(x)=\lambda_1\times f_1(x)+\lambda_2\times f_2(x) \text{ where } f_1(x)=\text{makespan};\quad f_2(x)=\text{total tardiness}; \lambda_1+\lambda_2=1 \qquad (3)$$

Step 5: Evaluate Fitness

In this paper, minimization of objectives is considered. Thus, the fitness value of chromosome is given by:

$$fitness\,(x)=\frac{1}{\lambda_1\times f_1(x)+\lambda_2\times f_2(x)+1} \quad \text{that } \lambda_1+\lambda_2=1,\quad x=1,2,...,np \qquad (4)$$

In the denominator, the value one is added in order to prevent a division by zero when the weight λ_1 and the total tardiness jobs ($f_2(x)$) are equal to zero.

Step 6: Selection Scheme

Roulette wheel, elitist and purely random selections are employed to reproduce the next generation. For the chromosome x with fitness $fitness(x)$, its selection probability $prob(x)$ is calculated as follows:

$$prob(x) = \frac{fitness(x)}{\sum_{x=1}^{np} fitness(x)} \quad (5)$$

Step 7: Reproduction

Select $np \times P_r$ solutions from current population. The elitist selection is applied in this step.

Step 8: Crossover operator

Select $np \times P_c$ pairs of parents from current population (based on roulette wheel selection), and perform crossover on the parents. One-Point Crossover (1PX), Order Crossover (OX) and the Position-Based Crossover (PBX) are applied in this paper.

Step 9: Mutation operator

Select $np \times P_m$ chromosome from current population (based on purely random selection), and mutate the individual bits. Shift move and swap move are applied in this paper.

Step 10: Neighborhood operator

A neighborhood relation is defined on the search space to generate $np \times P_n$ solutions. For selection of the chromosome, current population is combined with solutions achieved from operators (crossover and mutation). Then, the NSS performs on solution with maximum probability ($prob(x)$).

$$prob(x) = \frac{fitness(x)}{\sum_{x=1}^{np(1+Pc+Pm)} fitness(x)} \quad (6)$$

Step 11: Replacement

The new population generated by the previous steps, and updates the old population.

Step 12: Update efficient Solutions

Search the non-dominated (efficient) solutions in the new population and update the old non-dominated solutions with new ones.

Step 13: Stopping Rule

If the number of generations equals to the pre-specified number (ng) then stop, otherwise go to step 4.

3. Performance Measures

In this paper, three methods are used to evaluate the effectiveness of the algorithms in finding good quality schedules. From three to, we propose two methods named distance and triangle methods to evaluate the quality of approximation of the Pareto set obtained from solving bi-objective problems. The third method named FDH approach is available in the literature and only the stages of method are described briefly.

In single-objective optimization, quality is defined by means of the objective function: the smaller (or larger) the value, the better the solution. If two solutions in the presence of multiple optimization criteria are compared, the concept of Pareto dominance can be used, although the possibility of two solutions being incomparable, i.e., neither dominates the other, complicates the situation. However, it gets even more complicated when we compare two sets of solutions because some solutions in either set may be dominated by solutions in the other set, while others may be incomparable. Accordingly, it is not clear what quality means with respect to approximations of the Pareto-optimal set: closeness to the optimal solutions in objective space, coverage of a wide range of diverse solutions, or other properties? It is difficult to define appropriate quality measures for approximations of the Pareto-optimal set. Here, the various MOP metrics are reviewed that have been extracted from published works.

Several studies can be found in the literature that address the problem of comparing approximations of the Pareto-optimal set in a quantitative manner. The conventional way of comparing non-dominated set of solutions is through visual comparison in the objective space. This method is simple and straightforward. The criterion is to have solutions close to the true Pareto front and must be well distributed over the Pareto frontier. Following, various performance metrics were proposed. Some of the infamous metrics proposed were diversity [14], attainment surface [15], attainment surface sampling [16], generational distance, spacing, error ratio, maximum Pareto front error, overall non-dominated vector generation and ratio [17], size of the dominated space, coverage of two sets and coverage difference of two sets [18], etc. Detailed summaries of the metrics were discussed in references [19-22]. Some methods are based on binary quality measures, which assign numbers to pairs of approximation sets, e.g., Zitzler and Thiele [23] and Hansen and Jaszkiewicz [24]. Hansen and Jaszkiewicz [24] discussed several measures and proposed a general framework to compare and evaluate an approximation (so-called A). A cardinal measure defines the proportion of the number of efficient solutions in the approximate set to the number solutions in Pareto-optimal (so-called R). Note that this proportion can be zero and yet A can be a good approximation. A distance measure aims at establishing an average (maximum) distance between the points of A and R. However, the proposed metrics only emphasized on the distance between competing non-dominated sets or distance between competing non-dominated sets and a reference set.

Van Veldhuizen and Lamont [25] proposed an additional metric that measures the spread of the points in A. graphical plots have been used to compare the outcomes of multi-objective evolutionary algorithms as Van Veldhuizen and Lamont [25] point out. A conceptually different method is the attainment function approach [26], which consists of estimating the probability of attaining arbitrary goals in objective space from multiple approximation sets. Knowles et al. [27] compared the information provided by different assessment techniques on two database management applications. Knowles [28] and Knowles and Corne [21] discussed and contrasted several commonly used quality measures in light of Hansen and Jaszkiewicz's approach, as well as according to other criteria such as, e.g., sensitivity to scaling. They showed that about one third of the investigated quality measures are not compliant with any of the "outperformance" relations introduced by Hansen and Jaszkiewicz [24]. In this research, three methods are used to evaluate the quality of solutions. The methods applied in this paper are presented as follows:

3.1 FDH approach

In order to evaluate and compare non-dominated solutions, a particular case of data envelopment analysis (DEA) named free disposal hull (FDH) formulation is used. DEA is generally considered to be an approach to for evaluation of the performance of a set of decision-making units (DMU) by the calculation of efficiency and related measures. DEA is a non-parametric approach that uses linear programming to measure the relationship of produced goods and services (outputs) to assigned resources (inputs). Therefore, it is proper method to assess relative efficiency among similar entities called DMUs. In our problem, each scheduling solution is a DMU. From the definition of efficiency we will have DEA: weighted sum of outputs divided by weighted sum of inputs.

DEA assigns a score of 1 to a unit only when comparisons with other relevant units do not provide evidence of inefficiency in the use of any input or output. DEA assigns an efficiency score less than one to (relatively) inefficient units. A score less than one, means that a linear combination of other units from the sample could produce the same vector of outputs using a smaller vector of inputs. Now, the efficient DMUs generate an efficient frontier that 'envelops' all DMUs in different ways depending on the assumption made about returns to scale.

The following is a simple rule of thumb to classify the DEA attributes in a model as inputs or outputs. When it is necessary to decrease the value of an attribute, it is an input; in this case we hold everything else constant. The opposite is used for outputs: when it is necessary to increase the value of an attribute, it is an output, and everything else is constant. In other words, inputs are the makespan and the total tardiness because if they are smaller, it will be better. For illustrate application of FDH, see Ruiz-Torres and Lopez [29]. Also, Shirvani [30] applied the method DEA FDH model on multi objective scheduling. In this section of paper, only the stages of method are described briefly.

a) The scheduling solutions (or DMUs) obtained with each algorithm are combined in one single data set to generate an 'FDH problem set. Remember that each algorithm only 'provides' non-dominated solutions.
b) In this stage, each DMU comparing it 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, the degree of efficiency for DMU G is calculated, which is inefficient.

R_y	Set of DMUs provided by approximation method *y*
$\lvert R_y \rvert$	Number of DMUs in set R_y
$C^s_{\max}$	Makespan corresponding to DMU S
$\overline{T}^s$	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

Now the degree of efficiency of any DMU (efficient or inefficient) is defined as follows.

$$E^G = \begin{cases} 1 & \text{if G is efficient} \\ \text{Max}_{S \in Q^G} \left\{ w_1 \left(\frac{C^S_{\max}}{C^G_{\max}} \right) + w_2 \left(\frac{\overline{T}^S}{\overline{T}^G} \right) \right\} & \text{if G is inefficient} \end{cases} \quad (7)$$

The 'heuristic's efficiency' (A_y) is calculated as follows:

$$A_y = \frac{\sum_{G \in R_y} E^G}{|R_y|} \qquad (8)$$

3.2 Distance method

The decision maker is required to select a solution from a finite set by making compromises. A suitable solution should provide acceptable performance for all objectives. The solutions should represent a trade-off between the various objectives. In Figure 1, acceptable trade-offs solutions have been shown for a bi-objective problem. If there is an efficient solution along only an axis (e.g gray circles in Figure 1), it won't be appropriate. Since these solutions are just proper for one individual objective. Therefore, algorithm will be proper when it present more number of solutions with the trade-offs between the various objectives. Now, an approach should propose to determine algorithm with cited characteristic.

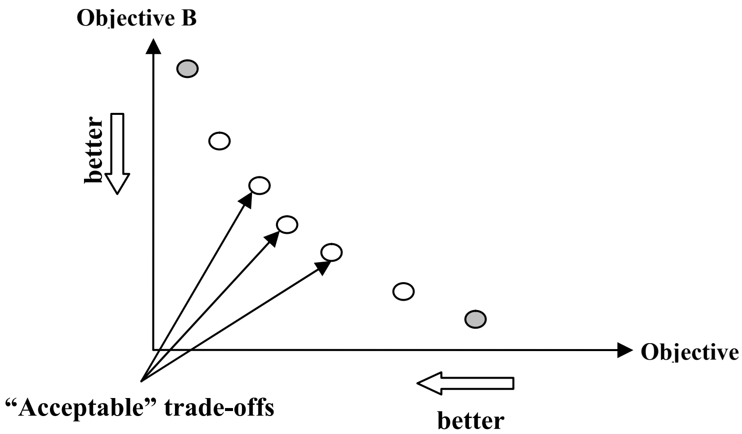

Figure 1. The presentation of suitable solutions

Distance method is suggested to evaluate the set of non-dominated solutions. It needs to indicate some concepts in order to better understand measure. First, it is assumed that the origin is (0, 0). Second, the performance of algorithm will be proper, if efficient solutions converge to side an origin or ideal solution (see Figure 2). The Euclidean distance (ed_i) of efficient solutions will be calculated in relation to the origin by Eq. 9. The average of these distances ($\overline{D}$), is radius of an arc in the search space. Now, each solution will be transferred to a point on the arc in the along which connects to the origin (see Figure 3). Lower values ($\overline{D}$) represent better sets.

$$ed_i = \sqrt{{C^{i}_{\max}}^{2} + \overline{T}^{i2}} \quad i = 1,2,\ldots,a \qquad (9)$$

$$\overline{D} = \frac{\sum_{i=1}^{a} ed_i}{n} \qquad (10)$$

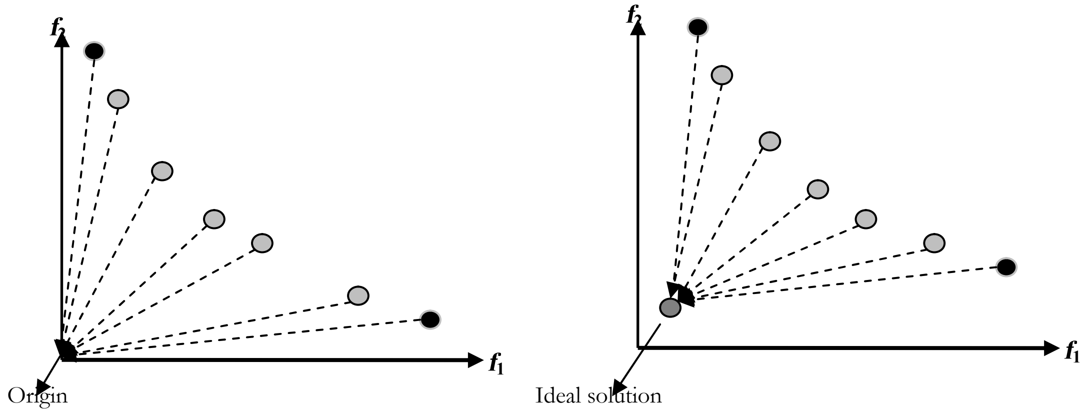

Figure 2. Converge to side an origin or ideal solution

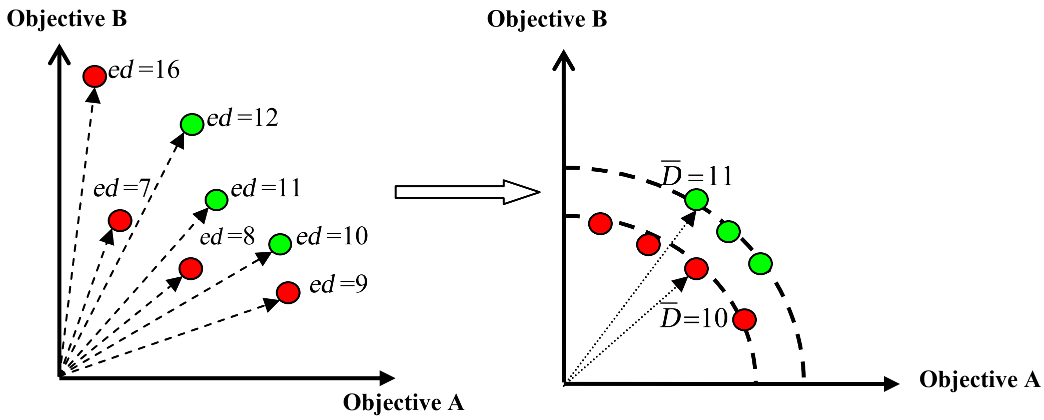

Figure 3. Distance method

Where *a* is the number of non-dominated solutions which obtained and C^{i}_{max} and $\overline{T}^{i}$ are, respectively, the makespan and total tardiness values of the solution *i*. Euclidean distance for solutions along only an axis is the greater. Consequently $\overline{D}$ for efficient solutions will be greater. Note that the distance method consists first of normalization of the criteria evaluation vectors (between 0 and 1) in order to convert the performance measures into non-dimensional ones.

3.3 Triangle Method

In this section, a simple but efficient method based on the area of coverage is proposed to compare non-dominated solution sets. The original concept of the proposed method is similar to the famous metric introduced by Zitzler [18] that is the area of coverage named "Hyper-area (H)". This metric, defines the area of coverage that an approximation of the Pareto-optimal set has with respect to the objective space. This would equate to the summation of all the areas of rectangles, bounded by the origin and ($f_1(x)$, $f_2(x)$), for a two-objective problem. Where $f_1(x)$ and $f_2(x)$ are the objective values. In this metric, shared regions are calculated several times. In order to solving this problem, this method is proposed as follows.

Suppose the objective values were obtained at objective A and B. These data are plotted in Figure 4. This figure clearly suggests that there is a relation between objective A and B, in the sense that the higher the objective B, the lower tends to be the objective A. In this figure, we have plotted a line of relationship that describes the statistical relation between objective A and B. Thence, we can use regression line estimation as $y = a\,x + b$. Here, y is the dependent or response variable, and x as the independent, explanatory or predictor variable.

In this paper, one time each objective as the response variable and another one as the predictor variable are invested to consider analytic outcomes of the two cases $C_{\max} = b_1 + a_1 \times \overline{T}$ and $\overline{T} = b_2 + a_2 \times C_{\max}$. Where $C_{\max}$ is the makespan, $\overline{T}$ is the total tardiness. a_1, a_2, b_1 and b_2 are calculated as follows:

$$a_1 = \overline{C}_{\max} - b_1 \times \overline{\overline{T}} \tag{11}$$

$$b_1 = \frac{\sum_{i=1}^{n} (\overline{T}^i - \overline{\overline{T}})(C^i_{\max} - \overline{C}_{\max})}{\sum_{i=1}^{n} (\overline{T}^i - \overline{\overline{T}})} \tag{12}$$

$$a_2 = \overline{\overline{T}} - b_2 \times \overline{C}_{\max} \tag{13}$$

$$b_2 = \frac{\sum_{i=1}^{n} (\overline{T}^i - \overline{\overline{T}})(C^i_{\max} - \overline{C}_{\max})}{\sum_{i=1}^{n} (C^i_{\max} - \overline{C}_{\max})} \tag{14}$$

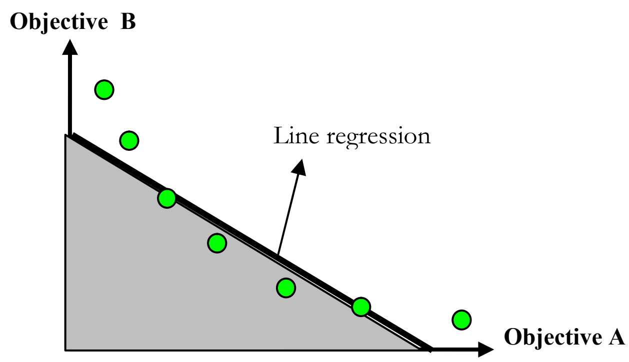

Figure 4. Linear regression

Where n is the number of non-dominated solutions which obtained. $C^i_{\max}$ and $\overline{T}^i$ are, respectively, the makespan and total tardiness values of the solution i in the reference set n. $\overline{C}_{\max}$ and $\overline{\overline{T}}$ are average makespan and total tardiness in the reference set, respectively.

After regression line estimation, the meeting point of this line with the axes (x=0 and y=0) are determined. Then the area of the triangle is calculated (gray region in Figure 4). Lower values represent better sets.

In order to estimate regression line, at least three statistical data are needed. We consider the passing line which forms a 45 angle with that axis, when there is only one statistical data (see Figure 5a). Also, we apply the exact line instead of estimating regression line, when there is two statistical data (see Figure 5b).

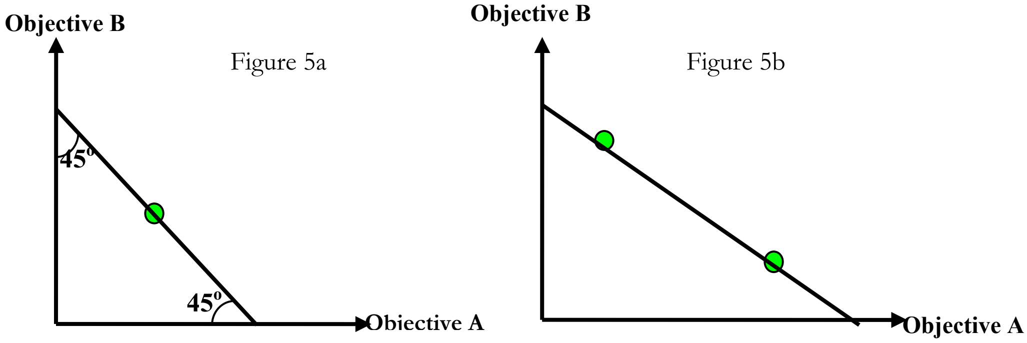

Figure 5. Particular cases in triangle method

4. Experimental results

4.1 Generation of test problem

The problem data can be characterized by five factors: range of processing times, range of setup times, number of stages, number of jobs, range in number of machines per stage and range of due date. Processing times are distributed uniformly over two ranges with a mean of 60: [50–70] and [20–100]. The setup times are uniformly distributed from 12 to 24 which are 20% to 40% of the mean of the processing time.

Problems involve three sections: small size problem (namely S1, S2, S3, S4, and S5), medium size problem (namely M1, M2, M3, M4 and M5) and large size problem (namely L1, L2, L3, L4 and L5). We used problems with 15 jobs × 5 stages (for small), 25 jobs × 10 stages (for medium), and 40 jobs × 20 stages (for large). Numbers of machines are distributed uniformly over two ranges [1-4] and [1-10]. Also, the largest number of machines in a stage must be less than the number of jobs. Due dates can be generated from a composite uniform distribution based on R and τ; with probability τ the due date is uniformly distributed over the interval $[\bar{d} - R\bar{d}, \bar{d}]$ and with probability (1-τ) over the interval $[\bar{d},\ \bar{d} + (C_{\max} - \bar{d})R]$, where τ and R are two parameters called the tardiness factor ($\tau = 1 - \bar{d} / C_{\max}$) and the due date range ($R = (d_{\max} - d_{\min}) / C_{\max}$), respectively. It should be noted that $d_{\max}$, $d_{\min}$ and $\bar{d}$ are maximum, minimum and average due date, respectively.

Values of τ close to 1 indicate that the due dates are tight, and values close to 0 indicate that the due dates are loose. A high value of R indicates a wide range of due dates, whereas a low value indicates a narrow range of due dates [31]. The values of τ are taken as 0.25, 0.50 and 0.75 and the values of R are taken as 0.25, 0.50, and 0.75. For each problem structure, data based on five different R and τ combinations are used: (0.2, 0.2), (0.2, 0.8), (0.5, 0.5), (0.8, 0.2), (0.8, 0.8). This means that small size problems namely S1, S2, S3, S4, and S5 differ only in due date's parameter value and too, for medium and large size problems.

4.2 Numerical result

Instances are tested with $\lambda_1 \in \{0.25, 0.5, \text{and } 0.75\}$ in the objective function of the weighted sum. Also, six replications for each problem size have been performed since there are some random conditions when applying the algorithm. As previously mentioned, 3 rules of table 2 and 4 NSS of table 3 are considered in the form of combination. Therefore, 12 combined algorithms are obtained which have been presented in Table 4 with their names and abbreviation codes.

The purpose of this paper is to evaluate the quality of efficient sets of algorithms combined output with three evaluation methods. The outcome of the assessment methods is to determine the best option for the dispatching rules and structures of neighborhood search.

Table 4. Candidate algorithms

No	Abbreviation	Neighborhood search structure	Dispatching rule
1	GA1	Inversion move	FIFO
2	GA2	Random insertion scheme	FIFO
3	GA3	Neighborhood swapping	FIFO
4	GA4	Swapped sequence	FIFO
5	GA5	Inversion move	AT-RPT
6	GA6	Random insertion scheme	AT-RPT
7	GA7	Neighborhood swapping	AT-RPT
8	GA8	Swapped sequence	AT-RPT
9	GA9	Inversion move	SL
10	GA10	Random insertion scheme	SL
11	GA11	Neighborhood swapping	SL
12	GA12	Swapped sequence	SL

4.2.1 Results based on FDH method

In FDH approach, weights W_l=0.25, 0.5 and 0.75 are used to calculate the degree of efficiency. Based on the results of efficiency given in Tables 5 to 7, the following observations can be made.

For S1 problem in Table 6 (section I), the algorithm of GA6 with the effectiveness of 0.9945 is better than others. Therefore, the best dispatching rule and NSS are "AT-RPT" and "random insertion scheme" by considering Table 4. In Table 6 (section I), values of W_l and $\boldsymbol{\lambda_1}$ are equal to 0.25 and 0.75, respectively. Without changing W_l, the $\boldsymbol{\lambda_1}$ is changed to 0.25. For S1 problem in Table 5 (section I), the algorithm of GA2 with the effectiveness of 1 is better than others. It has been found that the results differ in different search directions.

Therefore, we suggest that the searching process in algorithm starts along the direction of the first selected objective, and the search process progresses such that the weight for the first objective function decreases gradually and the weight for the second objective function increases gradually. This means that different weights vectors are used simultaneously to define different directions of search.

For M3 problem in Table 5 (section I), GA3 with the effectiveness of 0.9983 and in Table 5 (section II), the algorithm of GA2 with the effectiveness of 0.9978 is chosen as the best. A review of results in tables shows that, the results differ with different W_ls. Therefore, changing the value of W_l had significant effect in the overall relative performance of the algorithms.

Table 5. Heuristics efficiency when λ_1 = 0.25 I) W_1=0.25 II) W_1=0.75

Problem		Meta-heuristic											
		GA1	GA2	GA3	GA4	GA5	GA6	GA7	GA8	GA9	GA10	GA11	GA12
S1	I	0.9348	1.0000	0.9925	0.9854	0.8867	0.9932	0.9634	0.9743	0.9078	0.9225	0.8973	0.9561
	II	0.9700	1.0000	0.9955	0.9931	0.9540	0.9977	0.9847	0.9903	0.9586	0.9659	0.9609	0.9783
S2	I	0.9392	0.9744	0.9766	0.9823	0.9327	0.9442	0.8671	0.9298	0.9755	0.7308	0.9973	0.7302
	II	0.9760	0.9897	0.9906	0.9928	0.9719	0.9809	0.9526	0.9730	0.9873	0.9090	0.9978	0.9081
S3	I	0.9255	0.9965	0.9966	0.9971	0.9412	0.9140	0.9825	0.9901	0.9741	0.6953	0.9911	0.9955
	II	0.9659	0.9983	0.9976	0.9989	0.9767	0.9659	0.9891	0.9914	0.9886	0.8984	0.9946	0.9971
S4	I	0.9905	0.9997	0.9989	0.9988	0.9938	0.9985	0.9964	0.9957	0.9889	0.9926	0.9971	0.9496
	II	0.9939	0.9996	0.9993	0.9991	0.9961	0.9985	0.9968	0.9958	0.9888	0.9957	0.9975	0.9969
S5	I	0.9926	1.0000	0.9960	0.9995	0.9872	1.0000	0.9955	0.9971	0.9935	0.8398	0.9975	0.9911
	II	0.9953	1.0000	0.9957	0.9991	0.9930	1.0000	0.9969	0.9976	0.9971	0.9457	0.9953	0.9847
M1	I	0.9150	1.0000	0.9809	0.9258	0.8998	0.9815	0.9442	0.9790	0.9918	0.9858	0.7613	0.7615
	II	0.9686	1.0000	0.9768	0.9682	0.9532	0.9857	0.9737	0.9899	0.9958	0.9938	0.9187	0.9178
M2	I	0.9332	0.9979	0.9942	0.9945	0.9719	0.9851	0.9865	0.9907	0.9952	0.7119	0.9997	0.9975
	II	0.9742	0.9978	0.9946	0.9977	0.9718	0.9849	0.9934	0.9920	0.9967	0.8927	0.9999	0.9986
M3	I	0.9834	0.9941	0.9983	0.9914	0.9727	0.9922	0.9939	0.9957	0.9959	0.8694	0.8048	0.9955
	II	0.9903	0.9978	0.9969	0.9934	0.9725	0.9926	0.9955	0.9962	0.9957	0.9553	0.9305	0.9973
M4	I	0.9903	0.9991	0.9970	0.9949	0.9957	0.9955	0.9950	0.9926	0.9938	0.9909	0.9954	0.9920
	II	0.9936	0.9992	0.9987	0.9940	0.9965	0.9913	0.9957	0.9943	0.9971	0.9908	0.9978	0.9946
M5	I	0.9222	0.9939	0.9838	0.9928	0.9754	0.9900	0.9886	0.9774	0.6654	0.9989	0.9994	0.9997
	II	0.9686	0.9940	0.9913	0.9975	0.9787	0.9871	0.9920	0.9850	0.8879	0.9991	0.9995	0.9999
L1	I	0.9995	0.9998	1.0000	0.9991	0.7503	0.2479	0.9635	0.6348	0.9828	0.3843	0.3811	0.9953
	II	0.9986	0.9994	1.0000	0.9972	0.9145	0.7436	0.9760	0.8752	0.9897	0.7826	0.7907	0.9966
L2	I	0.9789	0.9973	0.9914	1.0000	0.9953	0.9977	0.9967	0.9972	0.9972	0.9983	0.9976	0.7932
	II	0.9892	0.9972	0.9961	1.0000	0.9961	0.9985	0.9968	0.9982	0.9973	0.9987	0.9973	0.9289
L3	I	0.9853	0.9978	0.9952	1.0000	0.9954	0.9979	0.9975	0.9968	0.9942	0.9971	0.9248	0.9247
	II	0.9929	0.9981	0.9973	1.0000	0.9934	0.9967	0.9973	0.9964	0.9940	0.9951	0.9698	0.9687
L4	I	0.9978	0.9998	0.9973	0.9996	0.9971	0.9975	0.9981	0.9985	0.9967	0.9975	0.9968	0.9867
	II	0.9979	0.9999	0.9981	0.9997	0.9969	0.9977	0.9979	0.9984	0.9965	0.9983	0.9972	0.9944
L5	I	0.9404	0.9992	0.9969	0.9989	0.9678	0.9864	0.9882	0.9830	0.9980	0.8765	0.9678	0.9670
	II	0.9776	0.9997	0.9976	0.9993	0.9874	0.9919	0.9946	0.9905	0.9985	0.9559	0.9981	0.9880

Table 6. Heuristics efficiency when λ1 = 0.75 I) W1=0.25 II) W1=0.75

Problem		Meta-heuristic											
		GA1	GA2	GA3	GA4	GA5	GA6	GA7	GA8	GA9	GA10	GA11	GA12
S1	I	0.9718	0.9787	0.9810	0.9762	0.9238	0.9945	0.9849	0.9939	0.9407	0.9670	0.9343	0.9514
	II	0.9813	0.9929	0.9909	0.9920	0.9697	0.9980	0.9903	0.9958	0.9727	0.9856	0.9737	0.9789
S2	I	0.9022	0.9883	0.9635	0.8927	0.8637	0.9628	0.9742	0.8999	0.9909	0.8222	0.9977	0.9946
	II	0.9635	0.9958	0.9856	0.9559	0.9510	0.9845	0.9891	0.9630	0.9912	0.9407	0.9968	0.9979
S3	I	0.9452	1.0000	0.9757	0.9993	0.9500	0.9324	0.9780	0.9797	0.9818	0.5833	0.9949	0.6256
	II	0.9749	1.0000	0.9895	0.9997	0.9785	0.9725	0.9852	0.9889	0.9888	0.8587	0.9955	0.8709
S4	I	0.9982	0.9987	0.9971	0.9970	0.9952	1.0000	0.9964	0.9981	0.9854	0.9927	0.9918	0.9912
	II	0.9965	0.9993	0.9979	0.9959	0.9961	1.0000	0.9981	0.9978	0.9898	0.9945	0.9960	0.9968
S5	I	0.9962	1.0000	0.9977	0.9969	0.9837	0.9926	0.9974	0.9928	0.9940	0.8628	0.9974	0.9952
	II	0.9927	1.0000	0.9975	0.9976	0.9897	0.9953	0.9969	0.9944	0.9943	0.9509	0.9976	0.9937
M1	I	0.9493	1.0000	0.9432	0.9866	0.9604	0.9787	0.9805	0.9861	0.9838	0.9911	0.9900	0.7904
	II	0.9773	1.0000	0.9734	0.9950	0.9763	0.9775	0.9868	0.9919	0.9934	0.9921	0.9952	0.9246
M2	I	0.9577	1.0000	0.9878	0.9896	0.9331	0.9829	0.9776	0.9903	0.9939	0.9976	0.6525	0.9968
	II	0.9792	1.0000	0.9937	0.9945	0.9653	0.9804	0.9856	0.9905	0.9954	0.9982	0.8806	0.9976
M3	I	0.9720	0.9993	0.9966	0.9963	0.9752	0.9913	0.9938	0.9944	0.9972	0.9316	0.9957	0.8681
	II	0.9851	0.9996	0.9984	0.9945	0.9870	0.9894	0.9944	0.9930	0.9966	0.9766	0.9975	0.9526
M4	I	0.9959	1.0000	0.9956	0.9958	0.9938	0.9921	0.9958	0.9941	0.9938	0.9881	0.9929	0.9936
	II	0.9958	1.0000	0.9926	0.9945	0.9930	0.9933	0.9956	0.9961	0.9960	0.9937	0.9929	0.9947
M5	I	0.9142	0.9983	0.9861	0.9953	0.9634	0.9890	0.9803	0.9947	0.9931	0.5594	0.9989	0.7731
	II	0.9634	0.9985	0.9881	0.9953	0.9804	0.9912	0.9852	0.9956	0.9954	0.8515	0.9994	0.3243
L1	I	0.5468	1.0000	0.5263	0.6656	0.7530	0.9255	0.9461	0.7538	0.9870	0.8388	0.9960	0.6827
	II	0.8451	1.0000	0.8414	0.8854	0.9704	0.9684	0.9736	0.8502	0.9917	0.9441	0.9973	0.8902
L2	I	0.9688	1.0000	0.9961	1.0000	0.9779	0.9832	0.9967	0.9907	0.9948	0.8843	0.8824	0.9132
	II	0.9875	1.0000	0.9930	1.0000	0.9880	0.9932	0.9961	0.9958	0.9921	0.9598	0.9596	0.9697
L3	I	0.9956	0.9990	0.9958	0.9991	0.9946	0.9958	0.9952	0.9984	0.9921	0.9536	0.9229	0.9988
	II	0.9961	0.9986	0.9978	0.9978	0.9957	0.9972	0.9972	0.9978	0.9901	0.9834	0.9737	0.9995
L4	I	0.9966	0.9993	0.9962	0.9996	0.9944	0.9974	0.9989	0.9975	0.9960	0.9959	0.9971	0.9872
	II	0.9973	0.9992	0.9971	0.9997	0.9966	0.9958	0.9980	0.9971	0.9969	0.9971	0.9983	0.9983
L5	I	0.9673	1.0000	0.9982	0.9973	0.9950	0.9943	0.9968	0.9966	0.9971	0.7837	0.9985	0.8544
	II	0.9824	1.0000	0.9975	0.9985	0.9953	0.9939	0.9966	0.9974	0.9971	0.9182	0.9985	0.9448

Table 7. Heuristics efficiency when λ1 = 0.5 and W1=0.5

Problem	Meta-heuristic											
	GA1	GA2	GA3	GA4	GA5	GA6	GA7	GA8	GA9	GA10	GA11	GA12
S1	0.9462	1.0000	0.9966	0.9897	0.9730	0.9706	0.9881	0.9886	0.9178	0.9906	0.9662	0.9580
S2	0.9030	0.9623	0.9757	0.9963	0.8090	0.9739	0.9744	0.9402	0.9844	0.8853	0.9917	0.9974
S3	0.9494	1.0000	0.9931	0.9832	0.9505	0.9753	0.9862	0.9889	0.9821	0.7716	0.9906	0.9923
S4	0.9942	1.0000	0.9969	0.9990	0.9944	0.9967	0.9962	0.9976	0.9906	0.9939	0.9963	0.9965
S5	0.9935	0.9973	0.9966	0.9988	0.9950	1.0000	0.9981	0.9985	0.9970	0.8984	0.9955	0.9972
M1	0.9178	0.9773	0.9437	1.0000	0.9768	0.9733	0.9833	0.9762	0.9963	0.7963	0.9999	0.8615
M2	0.9596	1.0000	0.9888	0.9987	0.9679	0.9824	0.9869	0.9940	0.9950	0.8232	0.9981	0.9955
M3	0.9783	0.9971	0.9925	1.0000	0.9745	0.9945	0.9957	0.9880	0.9934	0.8630	0.9984	0.9311
M4	0.9915	0.9957	0.9940	1.0000	0.9917	0.9924	0.9934	0.9949	0.9970	0.9944	0.9927	0.9811
M5	0.9843	0.9938	0.9910	0.9662	0.9783	0.9768	0.9912	0.9848	0.9981	0.9988	0.9977	0.7090
L1	0.6646	0.9997	0.9980	1.0000	0.7328	0.6367	0.9079	0.7202	0.9888	0.6602	0.9948	0.9976
L2	0.9736	1.0000	0.9783	0.9898	0.9809	0.9940	0.9953	0.9968	0.9949	0.9543	0.9974	0.9324
L3	0.9934	0.9990	0.9990	0.9991	0.9981	0.9935	0.9942	0.9948	0.9961	0.9974	0.9950	0.9421
L4	0.9987	0.9948	0.9976	1.0000	0.9940	0.9979	0.9962	0.9977	0.9967	0.9896	0.9971	0.9985
L5	0.9806	0.9971	0.9981	0.9975	0.9711	0.9893	0.9957	0.9943	0.9986	0.8821	0.9980	0.9989

4.2.2 Results based on distance method

In distance method, the influence of λ_1 is similarity with expressed for above method. Therefore, algorithm performance tables are only presented for different directions of search (see Table 8 to 10).

Table 8. Result of $\overline{D}$ when λ1=0.25, S and M (×103), L (×104)

Problem	Meta-heuristic											
	GA1	GA2	GA3	GA4	GA5	GA6	GA7	GA8	GA9	GA10	GA11	GA12
S1	1.5484	1.4349	1.4452	1.4342	1.5235	1.4316	1.4554	1.4407	1.5916	1.5164	1.5471	1.5466
S2	1.4761	1.4257	1.4242	1.4198	1.4860	1.6149	1.7193	1.6115	3.8442	3.6678	3.6978	3.6696
S3	1.8180	1.6466	1.6508	1.6380	2.0204	1.7692	1.7812	1.8014	3.4374	3.2591	3.2903	3.2994
S4	7.0716	6.8342	6.8412	6.8710	6.9976	6.8629	6.9223	6.8719	7.3318	7.1603	7.2034	7.1317
S5	6.3587	6.1851	6.2223	6.1681	6.4368	6.1786	6.2237	6.2029	8.1581	8.0187	8.0801	8.0681
M1	3.1284	2.9402	3.0863	3.0735	3.6550	3.1327	3.2437	3.1054	4.9373	4.8937	4.8259	4.8244
M2	10.2695	8.6283	8.7491	8.6736	11.3271	9.1453	9.7451	9.4229	18.4031	18.3140	18.4787	18.4707
M3	11.2317	10.7195	10.7159	10.6787	11.7134	10.8530	10.8667	10.7835	16.8432	16.5960	16.5799	16.6954
M4	21.6920	21.0241	21.0546	21.2118	21.7135	21.3512	21.6940	21.4493	22.3765	22.0881	22.2745	2.2929
M5	7.0645	6.0060	6.2319	6.0431	7.0942	6.5416	6.6692	6.7243	14.3870	14.4041	14.4180	4.4090
L1	0.3210	0.2961	0.2948	0.2959	0.3011	0.2998	0.3013	0.3006	1.0448	1.0157	0.3179	1.0208
L2	2.3582	2.2610	2.2852	2.2461	2.3759	2.3348	2.3267	2.3339	3.4008	3.3813	3.3909	3.3772
L3	3.3164	3.2125	3.2460	3.2059	3.2915	3.2317	3.2488	3.2316	3.7624	3.7377	3.7292	3.7277
L4	5.3894	5.3368	5.3750	5.3531	5.4118	5.3835	5.3989	5.3775	5.5684	5.5252	5.5441	5.5104
L5	2.4966	2.2708	2.2890	2.2789	2.4870	2.3516	2.3501	2.4803	3.6190	3.5985	3.6115	3.6085

Table 9. Result of $\overline{D}$ when λ1=0.5, S and M (×103), L (×104)

Problem	Meta-heuristic											
	GA1	GA2	GA3	GA4	GA5	GA6	GA7	GA8	GA9	GA10	GA11	GA12
S1	1.5001	1.4262	1.4425	1.4359	1.4821	1.4412	1.4513	1.4428	1.6624	1.4995	1.5252	1.5557
S2	1.4702	1.5322	1.6442	1.4115	1.4890	1.5909	1.6604	1.5212	3.8474	3.6648	3.7166	3.7120
S3	1.7856	1.6396	1.6454	1.6642	1.9074	1.7831	1.7773	1.7913	3.4129	3.2651	3.4047	3.3022
S4	7.1328	6.8201	6.9150	6.8248	7.0216	6.8877	6.8850	6.8423	7.2448	7.1499	7.1798	7.1626
S5	6.3700	6.2239	6.2573	6.2476	6.3672	6.1630	6.2222	6.2469	8.1264	8.0414	8.0891	8.0696
M1	3.3506	2.9920	3.0555	2.8849	3.4878	3.1121	3.2296	3.2542	4.9204	4.8513	4.8586	4.8621
M2	10.2056	8.5868	8.7279	8.6225	11.6472	9.2681	9.7784	9.2976	18.5895	18.3140	18.3600	18.4816
M3	11.4359	10.6988	10.9222	10.6473	11.7645	10.8059	10.7837	10.9983	16.7784	16.6503	16.6582	16.6271
M4	21.5349	21.3044	21.2832	21.1772	21.9217	21.3407	21.4950	21.4364	22.5053	21.9568	22.4270	22.4266
M5	6.3708	6.0380	6.1561	6.0177	7.2332	6.5893	6.7459	6.6380	14.4381	14.3811	14.4241	14.3591
L1	0.3624	0.2948	0.2959	0.2946	0.3662	0.3001	0.3013	0.3005	1.0482	1.0190	1.0294	1.0209
L2	2.4157	2.2284	2.3702	2.2666	2.4825	2.3033	2.3172	2.3073	3.4025	3.3793	3.3809	3.3766
L3	3.2852	3.2211	3.2260	3.2199	3.2864	3.2462	3.2805	3.2364	3.7614	3.2614	3.7467	3.7281
L4	5.3971	5.3422	5.3680	5.3167	5.4288	5.3730	5.3816	5.3759	5.5671	5.5177	5.5471	5.5326
L5	2.3678	2.2677	2.2775	2.2761	2.5628	2.3524	2.3523	2.3099	3.6150	3.5976	3.6081	3.5994

Table 10. Result of $\overline{D}$ when λ1=0.75, S and M (×103), L (×104)

Problem	Meta-heuristic											
	GA1	GA2	GA3	GA4	GA5	GA6	GA7	GA8	GA9	GA10	GA11	GA12
S1	1.5062	1.4376	1.4505	1.4331	1.5221	1.4314	1.4529	1.4358	1.6369	1.5389	1.5515	1.5849
S2	1.5857	1.4023	1.5731	1.4161	1.5224	1.4167	1.4085	1.4189	3.8481	3.7319	3.6981	3.7499
S3	1.8173	1.6385	1.6727	1.6476	1.9912	1.7989	1.7728	1.7897	3.4486	3.3107	3.3335	3.2819
S4	6.9470	6.9039	6.9445	6.8694	7.0434	6.8094	6.8949	6.8984	7.3526	7.1286	7.1639	7.1451
S5	6.3387	6.1433	6.2581	6.2724	6.4562	6.2269	6.2413	6.1924	8.1252	8.0375	8.0617	8.0681
M1	3.2536	2.9551	3.1137	2.9534	3.6212	3.1193	3.3424	3.1372	4.9297	4.8409	4.9258	4.7901
M2	9.2413	8.4563	8.6564	8.5952	11.2997	9.4014	9.5472	8.8432	18.6081	18.4158	18.3559	18.4829
M3	11.2619	10.7095	10.6611	10.6761	11.8990	10.9843	11.3449	10.8185	16.7289	16.6560	16.7339	16.6149
M4	21.5369	21.2005	21.2268	21.2200	21.9122	21.4732	21.6209	21.6530	22.6486	22.3998	22.4510	22.0730
M5	7.2040	5.9957	6.1904	6.0796	7.4153	6.3982	6.8477	6.4604	14.5152	14.3811	14.4000	28.7597
L1	0.3418	0.2934	0.2951	0.2951	0.3591	0.3002	0.3005	0.2995	1.0469	1.0250	1.0300	1.0228
L2	2.3961	2.2516	2.2796	2.2418	2.4771	2.3290	2.3373	2.3333	3.4038	3.3736	3.3819	3.3730
L3	3.2807	3.2201	3.2425	3.2154	3.3088	3.2339	3.2772	3.2316	3.7635	3.7294	3.7216	3.7286
L4	5.3867	5.3390	5.3752	5.3467	5.4310	5.3761	5.3750	5.3848	5.5663	5.5225	5.5416	5.4970
L5	2.4100	2.2802	2.2828	2.2856	2.4219	2.3509	2.3547	2.3670	3.6215	3.5969	3.6098	3.6047

4.2.3 Results based on triangle method

Results of triangle method are presented in Table 11 to 13. In triangle method, analytic outcomes are done based on the two cases $C_{\max} = b_1 + a_1 \times \overline{T}$ and $\overline{T} = b_2 + a_2 \times C_{\max}$. Tables 11 to 13, respectively, show that in 53%, 80% and 66% two linear patterns have had the same results. The decision maker has the right choice when the results aren't the same. It is known that the tardiness depends on the process and completion time of jobs. By the fact, line estimation in the form of $\overline{T} = b_2 + a_2 \times C_{\max}$ should be used.

Table 11. Result of triangle when λ1=0.25, I) $C_{\max} = b_1 + a_1 \times \overline{T}$, II) $\overline{T} = b_2 + a_2 \times C_{\max}$, S and M (×106), L (×108)

Problem		Meta-heuristic											
		GA1	GA2	GA3	GA4	GA5	GA6	GA7	GA8	GA9	GA10	GA11	GA12
S1	I	34.3345	24.8506	4.0102	6.1507	23.2216	124.331	15.012	6.6150	4.2463	4.0711	40.351	67.7647
	II	18.3989	23.1281	3.2503	5.6180	9.4538	124.3311	12.8177	5.1429	3.9163	4.0711	40.351	6.8723
S2	I	17.1297	15.2436	1.9435	1.7416	20.6367	143.6958	59.9096	223.883	14.8007	17.4184	13.587	23.0198
	II	13.9096	11.0041	1.8979	1.6850	16.4622	33.7595	50.0540	66.2389	13.1051	17.4184	13.4965	23.0198
S3	I	30.4351	9.1565	11.2595	13.4935	31.7563	11.9798	4.4946	4.6226	30.6384	9.3870	27.6668	16.6211
	II	16.6102	5.0785	7.9028	13.4935	26.9915	11.9798	4.3809	3.8020	16.3428	9.3870	27.6433	12.3959
S4	I	22.3838	23.8998	18.2868	18.7536	19.3257	53.5477	18.6252	22.4129	40.6121	22.6280	25.0598	19.4335
	II	20.6840	37.7436	18.2868	18.4434	19.6793	47.5251	21.9794	18.7095	39.7748	22.1865	24.6981	19.4335
S5	I	33.7666	18.2428	23.6487	17.4270	48.0613	27.2580	20.7725	17.6560	36.9820	42.9544	36.4627	4.3222
	II	35.3386	18.2428	24.6936	19.3344	36.1957	27.2580	20.0002	17.5418	37.0580	42.9544	42.5473	4.7381
M1	I	20.9346	9.9825	39.6370	39.2034	172.5909	53.6374	130.3940	20.2107	37.2017	21.3886	22.1013	23.3010
	II	15.9767	9.9495	22.8313	19.5108	132.1541	51.3511	130.3940	20.2107	34.1480	21.6142	22.1013	23.3010
M2	I	183.223	48.755	83.838	74.622	193.165	50.101	127.602	65.377	101.365	213.769	217.382	217.215
	II	161.539	46.572	75.985	61.439	155.254	47.899	74.671	63.629	101.365	213.769	217.382	17.215
M3	I	84.784	118.237	122.217	56.130	83.528	145.693	54.942	72.875	85.808	91.951	179.494	96.486
	II	78.143	118.237	122.217	53.459	77.484	124.216	57.210	62.049	85.807	97.454	179.494	90.519
M4	I	122.006	111.212	104.726	108.724	109.825	281.011	122.887	107.508	123.199	300.517	139.293	50.765
	II	126.370	196.448	111.936	105.915	110.762	281.011	116.922	108.204	123.196	300.517	139.239	44.905
M5	I	216.598	27.451	61.608	60.978	34.165	44.002	90.644	124.061	138.844	139.161	139.412	139.245
	II	185.7937	28.307	53.815	43.268	34.165	35.717	56.674	87.668	138.844	139.161	139.412	139.245
L1	I	2.7262	0.0438	0.0435	0.0438	0.0668	0.0349	0.5538	0. 2223	1.1088	0.8067	0.5133	0.8777
	II	2.7262	0.0438	0.0435	0.0438	0.0668	0.0349	0.3922	0.1625	1.1088	0.8067	0.5133	0.8777
L2	I	3.7969	1.6936	1.3469	1.4588	2.7834	1.7243	1.8599	1.8185	6.8062	2.0209	2.2168	2.0172
	II	2.7724	1.7486	1.4077	1.3626	2.6537	1.6474	1.7347	1.6378	6.8062	2.0204	2.2685	2.0411
L3	I	2.1952	1.9740	1.9354	6.0778	30.9881	2.8212	4.1650	11.3337	2.8888	3.5000	2.2313	2.3752
	II	2.0074	1.9141	1.9733	6.0778	30.9881	2.8212	4.1650	11.3337	3.3896	5.7173	2.2313	2.3752
L4	I	3.5065	3.3108	3.2803	3.5692	3.2413	5.3023	3.3598	3.6439	3.5481	16.8966	3.5528	6.8009
	II	4.0353	3.6435	3.2008	3.2272	3.2413	4.4215	3.5250	3.5957	3.5481	16.8966	3.5528	6.8009
L5	I	195.4807	1.3384	1.3641	1.3424	3.7328	3.3149	1.4049	36.5765	13.4269	6.2813	2.1829	2.3056
	II	195.4807	1.3383	1.3482	1.3955	3.2567	3.1294	1.4049	14.4525	13.4269	6.2813	2.1829	2.3680

Table 12. Result of triangle when λ1=0.5, I) $C_{\max} = b_1 + a_1 \times \overline{T}$, II) $\overline{T} = b_2 + a_2 \times C_{\max}$, S and M (×106), L (×108)

Problem		Meta-heuristic											
		GA1	GA2	GA3	GA4	GA5	GA6	GA7	GA8	GA9	GA10	GA11	GA12
S1	I	11.2261	2.4358	11.4108	3.4001	4.1501	6.5138	8.0268	8.7289	20.5750	1.9456	7.7497	6.2168
	II	9.8581	1.8762	7.5372	2.8721	2.9664	6.2649	7.5238	8.4732	19.3193	1.9456	7.6397	5.0190
S2	I	17.7476	57.4084	43.5852	1.8152	19.3016	89.6963	79.2268	60.3110	16.8987	12.7563	10.7447	24.0780
	II	16.7275	28.6731	38.7049	1.5248	15.7077	34.8272	37.1497	42.0240	13.9321	12.7563	9.9635	23.9595
S3	I	24.6108	9.1448	2.5470	15.5198	22.0679	5.4953	4.7708	6.1740	11.0391	8.3241	41.4033	3.9992
	II	11.5976	9.0856	2.5043	9.2335	9.7908	4.3352	4.2065	5.7806	9.9683	8.2087	25.6649	11.4735
S4	I	24.6450	19.9054	32.2056	19.7267	25.1582	30.0806	19.6002	25.4360	54.9865	20.2595	25.1585	39.3989
	II	22.0372	19.9054	28.1753	21.4462	23.1974	25.7933	18.8301	25.4360	47.7167	19.3185	20.3862	39.3989
S5	I	17.6882	16.5854	16.7382	19.7158	73.9983	16.6360	17.4384	21.4995	22.4994	25.1356	23.9748	43.5403
	II	17.5095	16.5978	16.7547	16.6691	49.1973	16.6360	17.8604	21.1091	22.4994	26.7722	31.6089	43.5403
M1	I	68.00	11.90	3618.00	7.500	12.90	25.20	26.10	99.30	28.40	24.30	22.40	28.00
	II	58.90	10.00	3618.00	7.500	12.90	23.70	18.40	48.70	28.40	24.30	22.40	22.30
M2	I	207.231	122.004	162.636	42.329	320.025	149.186	110.266	55.505	133.251	213.769	214.784	217.445
	II	189.967	117.044	135.947	41.982	236.248	149.186	95.362	55.505	133.251	213.769	214.784	217.445
M3	I	174.664	158.053	60.874	73.319	97.056	54.594	62.345	157.582	92.464	99.440	85.330	87.837
	II	136.648	116.704	56.690	73.319	87.347	53.791	60.440	130.454	90.855	99.440	85.330	87.837
M4	I	114.909	150.165	107.705	106.459	110.943	146.661	150.069	111.620	173.145	297.241	186.639	15.938
	II	122.306	142.992	105.652	104.302	111.408	210.810	134.187	111.620	171.009	297.241	181.008	127.570
M5	I	30.776	79.884	66.066	27.905	160.877	112.722	59.204	45.716	139.779	138.745	139.445	138.345
	II	29.932	68.039	31.868	31.392	65.546	61.502	40.374	34.544	139.779	138.745	139.445	138.345
L1	I	23.1778	0.0434	0.0438	0.0434	73.0273	1.0041	0.2338	0.3820	0.6077	0.8110	0.7873	0.7730
	II	22.9844	0.0434	0.0438	0.0434	73.0273	0.5190	0.1679	0.2897	0.6077	0.8110	0.7873	0.7008
L2	I	4.6148	1.3357	27.4425	2.8670	5.3912	3.3336	1.8379	1.6626	2.0467	2.4307	2.033	2.0231
	II	3.9370	1.3357	9.0710	2.8670	4.8498	3.3336	1.7140	2.3785	2.0467	2.3404	2.033	2.0255
L3	I	2.2492	1.8883	6.1502	2.7526	2.2104	2.3236	4.5334	23.4701	3.3831	2.7860	2.260	3.9046
	II	2.1777	1.8955	6.1502	2.3878	2.2104	2.3236	4.4568	23.4701	3.9214	2.8304	2.2568	2.6820
L4	I	3.1902	3.3946	3.2038	3.2628	3.2533	3.3860	5.9681	3.3762	3.6615	3.9599	3.5069	24.5548
	II	3.3947	3.4919	3.1752	3.2628	3.4847	3.2639	6.1861	3.3762	3.6690	3.9599	3.3372	24.5548
L5	I	4.1356	6.0593	1.7395	1.3350	13.1032	1.5629	1.4288	16.0642	7.6230	2.2956	2.639	2.3176
	II	2.9749	6.0593	1.4071	1.3635	11.6525	1.4724	1.4218	16.0642	7.6230	2.2956	2.641	2.3176

Table 13. Result of triangle when λ1=0.75, I) $C_{\max} = b_1 + a_1 \times \overline{T}$ **, II)** $\overline{T} = b_2 + a_2 \times C_{\max}$ **, S and M (×106), L (×108)**

Problem		Meta-heuristic											
		GA1	GA2	GA3	GA4	GA5	GA6	GA7	GA8	GA9	GA10	GA11	GA12
S1	I	28.1215	27.8012	14.0158	12.5383	10.8589	9.0632	6.8268	3.4268	15.7287	8.2213	4.1753	24.8591
	II	19.8578	17.3457	10.8923	12.3187	9.3458	8.4603	6.6933	2.6670	14.6639	4.5155	4.1728	18.2424
S2	I	37.8293	3.9809	111.3289	7.5109	28.0244	0.8655	3.5355	9.7648	18.6734	27.4428	42.0897	33.4368
	II	26.9218	2.6988	40.7874	7.1063	22.5887	0.7981	2.4108	5.2304	12.0718	26.7106	42.0897	31.8025
S3	I	32.5399	6.0221	12.8787	7.1546	45.9461	4.7553	6.2272	3.9467	13.2812	12.5626	12.6055	8.7094
	II	20.5970	5.4124	12.3500	6.9529	43.6931	3.9896	6.2272	3.6603	12.1620	9.5436	10.3681	8.9187
S4	I	32.3877	20.6502	19.0605	22.0096	23.2015	32.2308	30.6137	23.1244	30.4221	20.5084	19.7187	42.7933
	II	29.1840	18.4438	18.8675	18.7619	21.0655	32.2308	41.5346	22.3163	23.6771	19.7413	19.3291	34.6317
S5	I	17.7459	17.5693	17.2261	16.8594	53.1758	19.3115	16.7024	17.7322	44.1809	22.3439	43.4545	36.1800
	II	17.4672	17.5693	16.6881	16.7790	28.3864	16.8802	16.6488	18.2353	44.1809	22.3439	43.4545	40.6493
M1	I	46.0093	81.8489	34.8124	32.9481	157.3490	32.5347	61.5807	22.5772	22.9449	21.7624	40.6311	56.5940
	II	30.3810	57.7063	15.6197	32.9276	112.0160	20.7410	59.0328	15.6301	22.9449	21.4895	26.0861	56.5940
M2	I	99.030	103.229	48.730	43.070	195.293	118.934	221.627	60.529	220.332	216.008	130.677	217.507
	II	90.243	97.602	47.376	42.834	158.466	78.898	155.575	60.529	220.332	216.008	130.677	217.507
M3	I	81.10	106.60	169.90	52.20	196.50	392.10	2458.30	65.60	87.10	107.60	118.80	89.20
	II	80.10	90.10	139.60	54.50	139.00	303.80	2458.30	55.20	88.30	124.30	107.30	89.20
M4	I	1912.80	198.10	112.80	126.00	123.40	124.20	137.10	342.20	180.90	308.20	202.60	299.90
	II	1912.80	194.60	114.40	127.60	114.00	120.70	129.00	232.30	174.50	308.20	200.40	299.90
M5	I	566.310	34.855	30.060	48.598	237.670	30.695	75.151	59.571	76.699	138.745	139.078	768.987
	II	382.418	32.607	29.062	31.900	157.771	30.695	66.106	59.571	76.699	138.745	139.078	768.987
L1	I	15.0790	0.0431	0.2283	0.2593	6.4214	0.2082	0.0312	0.1639	0.8317	0.5912	0.9607	1.9027
	II	6.2472	0.0431	0.0885	0.2593	4.1929	0.2082	0.0312	0.1627	0.8204	0.5912	0.9607	1.9027
L2	I	4.3405	1.9177	2.6096	3.1689	11.9004	2.2507	1.9988	3.4126	3.5671	2.4445	2.2610	2.0200
	II	3.9155	1.9177	2.8930	3.1689	11.5031	2.2507	1.7046	3.4126	3.7275	2.0260	2.0193	2.0200
L3	I	2.1628	2.8059	8.6757	1.8921	4.1009	5.1754	5.1250	2.0459	2.4176	2.5856	2.2311	2.2478
	II	2.5509	2.6198	5.8538	2.0060	2.5703	5.1116	5.1250	2.0459	2.4475	2.4357	2.2311	2.2253
L4	I	3.2574	3.8328	3.3135	3.5274	3.2505	3.2213	4.4844	18.4634	17.1411	3.4538	16.9921	16.7267
	II	3.3612	3.9154	3.2063	3.3950	3.3944	3.3851	4.4717	19.8809	17.1411	3.4538	16.9921	16.7267
L5	I	3.2479	1.9552	1.3728	1.7471	2.9367	1.5466	1.4397	1.4671	2.3063	4.3840	2.2269	2.3653
	II	2.1880	1.5892	1.3785	1.6099	2.8435	1.4782	1.4043	1.4097	2.3887	4.3840	2.3023	2.2013

4.2.4 Final results

As can be seen from tables, no combination case of 12 algorithms was able to outperform all other cases on all problems. Therefore, Tables 14 and 15 are provided to classify and obtain final results for determining a suitable combination case.

Table 14 is provided to determine a suitable dispatching rule. The acceptable percentage of each rule has been obtained in this table. As can be seen from Table 14, the first in, first out (FIFO) rule is better in order to find the job sequence for the second to end stages in all evaluation methods.

Table 15 is provided to determine an appropriate NSS. In this table, the acceptable percentage of each NSS has been obtained. As can be seen from Table 15, the structure of "swap moves" is the worst operator in comparison with others NSSs in all evaluation methods. And the "random insertion scheme" is better than others in FDH and distance methods. Structure of "RIPS" is the best operator of triangle method with average 34.5%.

Table 14. The comparison of dispatching rules as percentage (T=Table, number %)

Rule	FDH method					Distance method			Triangle method					
	T5 (I)	T5 (II)	T6 (I)	T6 (II)	T7	T8	T9	T10	T11 (I)	T11 (II)	T12 (I)	T12 (II)	T13 (I)	T13 (II)
FIFO	80	80	74	74	74	94	94	87	73	86	54	80	93	53
AT_RPT	6	6	13	13	6	6	6	13	27	14	40	20	13	47
SL	20	20	13	20	20	0	0	0	0	6	6	0	0	0

Table 15. The comparison of neighborhood search structures as percentage (T=Table, number %)

Rule	FDH method					Distance method			Triangle method					
	T5(I)	T5(II)	T6(I)	T6(II)	T7	T8	T9	T10	T11(I)	T11(II)	T12(I)	T12(II)	T13(I)	T13(II)
Swap moves	0	0	0	0	0	0	0	0	6	6	0	0	6	6
Random insertion scheme	54	60	74	74	47	54	47	74	33	40	27	40	40	13
Neighborhood swapping	26	20	13	6	0	6	0	6	47	13	40	20	20	47
RIPS	26	26	20	20	53	40	53	20	14	47	33	40	40	33

4.2.5 Short discussion

In the literature of multi-objective optimization problems, several methods have been presented to evaluate the quality of the obtained non-dominated front and assess the performance of multi-objective optimizers. Each approach has its own advantages and drawbacks. It is unclear up to now what their advantages and disadvantages are. There is no common agreement on which measure(s) should be used.

Suppose that we have an algorithm and would compare our algorithm with other efficient algorithm. The question arises that among evaluation methods which one is more proper? If an evaluation method is selected and comparison between algorithms is carried out, the question arises whether selected algorithm would again choose with another evaluation method? To answer such a question, we have to know the advantages and drawbacks of each evaluation method. To state some of the advantages and drawbacks of methods used in this paper, we give examples.

In FDH approach, each solution is evaluated by comparing it to the non-dominated solutions of other algorithms on a one-to-one basis. Therefore, the advantage of this method is that all solutions of algorithms are influential in performance measure of each algorithm (see Figure 6a).

To state drawbacks, if an algorithm has several solutions just along an individual axis (in one corner of the solution space), and these solutions aren't covered by solutions of other algorithms, this algorithm has high efficiency or equal to 1 (i.e., set X in Figure 6b). While by considering quality, these solutions aren't suitable at all.

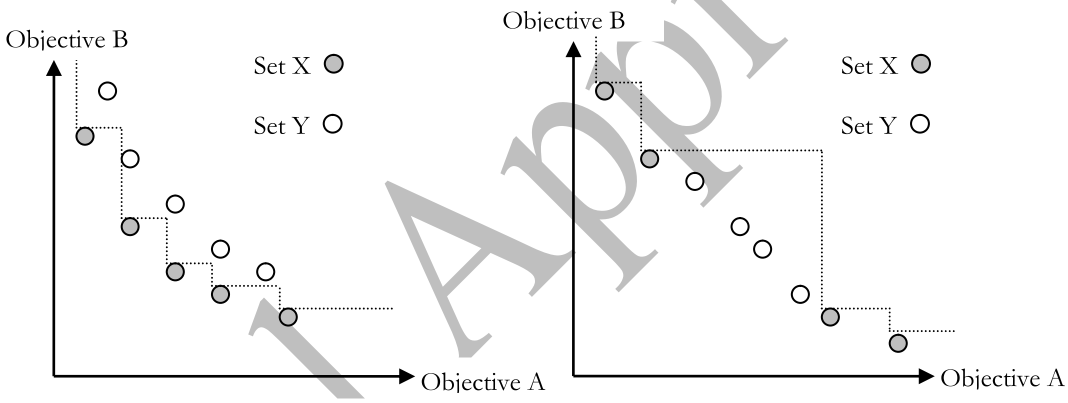

Figure 6a
Set X dominates set Y. Set X has the effectiveness of 1 and set Y has the effectiveness of lower 1. Hence, set X is better than set Y.

Figure 6b
Set X and set Y have the effectiveness of 1. But considering quality, set X isn't suitable.

Figure 6. Particular case in FDH approach

To state the advantages in triangle method (covers FDH drawbacks), if there is a solution just along an individual axis and too far to origin (see Figure 7a and Figure 7b), the area will be greater (i.e., cause bad result for algorithm). To state the drawbacks, in this performance measure, no attention is paid to solutions of other algorithms. This means that the performance of algorithms is calculated independent of other algorithms (in opposite of FDH advantage).

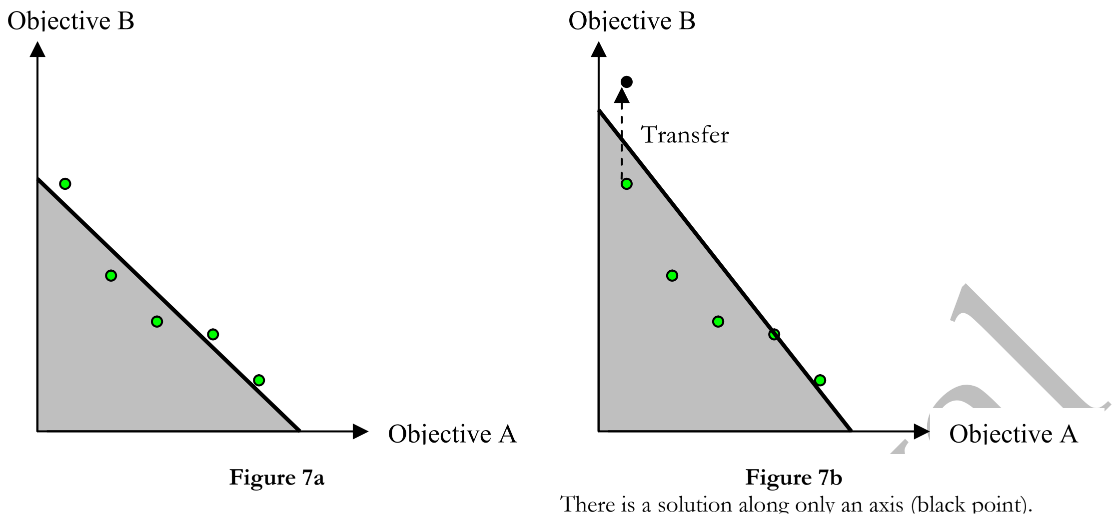

Figure 7. Example of particular case in triangle method

Finally, we can express that the results of such methods is not necessarily identical and results are obtained according to the concepts used in formulas.

5. Conclusions and future work

In this paper, three techniques are presented to analyze the non-dominated set of solutions, which in turn can assist in the comparison process. Then, evaluation methods are used on the certain problem. The problem under study is scheduling jobs in a hybrid flow shop environment with sequence-dependent setup times and the objectives of minimizing both the makespan and the total tardiness. A meta-heuristic procedure based on the genetic algorithm is applied to solve these problems. Two parts of the structure of algorithm are considered to improve the output results including: dispatching rule, and neighborhood search structure. In the structure of algorithm, 3 and 4 alternatives for dispatching rules and neighborhood search structure have been introduced respectively. Therefore, twelve algorithms are derived from a combination of dispatching rules and neighborhood search structures. Then efficient sets of algorithms combined output are compared through several evaluation methods. The FIFO rule is better in order to find the job sequence for the second to end stages in each of evaluation methods. The SL rule is worst in distance and triangle methods. The structure of "swap moves" is the worst operator in comparison with others for each of methods. The "random insertion scheme" is better in FDH and distance methods. The "RIPS" is the best operator by triangle method. Future research under consideration by the authors includes the development of other evaluation method for multi-criteria sets.

References

[1] M. Zandieh, S.M.T. Fatemi Ghomi, and S.M. Moattar Husseini, "An immune algorithm approach to hybrid flowshops scheduling with sequence-dependent setup times", Applied Mathematics and Computation, 180 (1), 111–127 (2006).

[2] T. Yang, Y. Kuo, and C. Cho, "A genetic algorithms simulation approach for the multi-attribute combinatorial dispatching decision problem", European Journal of Operational Research, 176 (3), 1859–1873 (2007).

[3] M.E. Kurz, and R.G. Askin, "Scheduling flexible flow lines with sequence-dependent setup times", European Journal of Operational Research, 159 (1), 66–82 (2003).

[4] S.M. Mousavi, M. Zandieh, and M. Amiri, "An efficient bi-objective heuristic for scheduling of hybrid flow shops", International Journal of Advanced Manufacturing Technology, 54 (1-4), 287-307 (2011).

[5] S.M. Mousavi, M. Zandieh, and M. Amiri, "Comparisons of bi-objective genetic algorithms for the hybrid flowshop scheduling with sequence-dependent setup times", International Journal of Production Research, 50 (10), 2570-2591 (2011).

[6] S.M. Mousavi, M. Mousakhani, and M. Zandieh, "Bi-objective hybrid flow shop scheduling: a new local search", International Journal of Advanced Manufacturing Technology, 64 (5-8), 933-950 (2012).

[7] S.M. Mousavi, M. Zandieh, and M. Yazdani, "A simulated annealing/local search to minimize the makespan and total tardiness on a hybrid flowshop", International Journal of Advanced Manufacturing Technology, 64 (1-4), 369-388 (2012).

[8] S. M. Mousavi, and M. Zandieh, "An efficient hybrid algorithm for a bi-objectives hybrid flow shop scheduling", Intelligent Automation and Soft Computing, 1-8, (2016).

[9] S. M. Mousavi, I. Mahdavi, J. Rezaeian, and M. Zandieh, "An efficient bi-objective algorithm to solve re-entrant hybrid flow shop scheduling with learning effect and setup times", Operational Research: An International Journal, 1-36, (2016).

[10] B., Barzegar, and H. Motameni, "Optimality of the flexible job shop scheduling system based on Gravitational Search Algorithm", Journal of Advances in Computer Research, 2(4), 1-12 (2011).

[11] R. Tavakkoli-Moghaddam, A. R. Rahimi-Vahed, and A. H. Mirzaei, "Solving a multi-objective no-wait flow shop scheduling problem with an immune algorithm", International Journal of Advanced Manufacturing Technology, 36 (9), 969-981 (2006).

[12] J.H. Holland, "Adaptation in natural and artificial systems", University of Michigan, Ann Arbor, Michigan, (1975).

[13] B. Barzegar, H. Motameni, A. Khosrozadeh ghomi, and A. Divsalar, "A Hybrid Genetic Algorithm for the Open Shop Scheduling with Makespan and Total Completion Time", Journal of Advances in Computer Research, 3(4), 17-24 (2011).

[14] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II", Kan GAL report, Indian Institute of Technology, Kanpur, India, (2000).

[15] C.M. Fonseca, and P.J. Fleming, "On the Performance Assessment and Comparison of Stochastic Multi-objective Optimizers", Parallel Problem Solving from Nature - PPSN IV, Berlin, Germany, pp. 584-593, Springer-Verlag Lecture Notes on Computer Science no. 1141 (1996).

[16] J.D. Knowles, and D.W. Corne, "Approximating the non-dominated front using the Pareto Archived Evolution Strategy", Evolutionary Computation, 8(2), 149-172 (2000).

[17] D.A. Van Veldhuizen, "Multi-objective Evolutionary Algorithms: Classifications, Analyses, and New Innovations", PhD Thesis, Dept. of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, (1999).

[18] E. Zitzler, "Evolutionary Algorithms for Multi-objective Optimization: Methods and Applications", PhD Thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, (1999).

[19] K.H. Ang, and Y. Li, "An Overview of Benchmarking Techniques for Multi-Objective Evolutionary Algorithms", Proceedings of the 6th Online World Conference on Soft Computing in Industrial Applications (WSC6), pp. 337-348, (2001).

[20] K. Deb, "Multi-Objective Optimization using Evolutionary Algorithms", John Wiley & Sons, Chichester, (2001).

[21] J.D. Knowles, and D.W. Corne, "On Metrics for Comparing Non-Dominated Sets", Proceedings of the Congress on Evolutionary Computation (CEC'02), pp. 711-716, Hawaii, U.S.A, (2002).

[22] R. Sarker, and C.A.C. Coello, "Assessment Methodologies for Multi-objective Evolutionary Algorithms", Kluwer Academic Publishers, New York, pp. 177-195, (2002).

[23] E. Zitzler, and L. Thiele, "Multi-objective optimization using evolutionary algorithms: A comparative case study", In Parallel Problem Solving from Nature (PPSN V), A. E. Eiben et al., Eds. Berlin, Germany: Springer, pp. 292–301, (1998).

[24] M.P. Hansen, and A. Jaszkiewicz, "Evaluating the Quality of Approximations to the Nondominated Set", Technical Report, Institute of Mathematical Modelling, Technical University of Denmark, (1998).

[25] D.A. Van Veldhuizen, and G.B. Lamont, "On Measuring Multiobjective Evolutionary Algorithm Performance", in Congress on Evolutionary Computation, vol. 1. Piscataway, New Jersey, pp. 204–211, (2000).

[26] V.G. Fonseca, C.M. Fonseca, and A.O. Hall, "Inferential performance assessment of stochastic optimizers and the attainment function," In Evolutionary Multi-Criterion Optimization (EMO2001), E. Zitzler et al., Eds. Berlin, Germany: Springer, pp. 213–225, (2001).

[27] J. D. Knowles, D.W. Corne, and M.J. Oates, "On the assessment of multi-objective approaches to the adaptive distributed database management problem", In Parallel Problem Solving from Nature (PPSN VI), M. Schoenauer et al., Eds. Berlin, Germany: Springer, pp. 869–878, (2000).

[28] J.D. Knowles, "Local-search and hybrid evolutionary algorithms for Pareto optimization", Ph.D. dissertation, Dept. of Comput. Sci., Univ. of Reading, Reading, U.K (2002).

[29] A.J. Ruiz-Torres, and F.J. Lopez, "Using the FDH formulation of DEA to evaluate a multi-criteria problem in parallel machine scheduling", Computers & Industrial Engineering, 47 (2-3), 107–121 (2004).

[30] M.H. Shirvani, "Evaluating of Feasible Solutions on Parallel Scheduling Tasks with DEA Decision Maker", Journal of Advances in Computer Research, 6(2), 109-115 (2015).

[31] T. Eren, and E. Güner, "The tricriteria flowshop scheduling problem", International Journal of Advanced Manufacturing Technology, 36 (11), 1210-1220 (2007).