A Mathematical Model for Lot-streaming Hybrid Flow Shop Scheduling Problem by Considering Learning Effect and Buffer Capacity

Roja Ruhbakhsh[1], EsmaeilMehdizadeh[*,2], Mohammad Amin Adibi[3]

[1]Department of Industrial Engineering, Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran; **E-mail addresses:** ruhbakhsh.ie.69@gmail.com;Mobile: +98(911)5961770

[2]Department of Industrial Engineering, Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran; **E-mail addresses:** Es.Mehdizadeh@iau.ac.ir;Mobile: +98(912)2812562

[3]Department of Industrial Engineering, Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran; **E-mail addresses:** ma.adibi@iau.ac.ir;Mobile: +98(912)*8862534*

Abstract

Lot streaming, is an effective technique to increase efficiency in a production system by splitting a job into several smaller parts to overlap operations between successive machines in a multi-stage production system. In this paper, a hybrid flow shop scheduling problem with lot streaming, learning effect and buffer's limitation is considered. A bi-objective mathematical model is presented for minimizing the total tardiness and the makespan. To validate the proposed model, the bi-objective model converted to single objective model by weighting method and an example is solved using GAMS software.Since the problem is NP-hard,NSGA-II and NRGA are used to solve the large scale of the problems and the first available machine (FAM) rule is used to assign the jobs to the machines from the first stage onwards in solution representation. Also, Taguchi method is used to tune the algorithms parameters. For evaluating the performance of the algorithms, the results obtained from GAMS compared with outputs of the GA algorithms. Also, 30 instance problems are randomly generated and six indicators are used to compare the algorithms together. After performing the experiments and comparingNSGA-II and NRGAalgorithms with each other based on each index, the appropriate algorithm is selected.

Keyword: Scheduling, Hybrid flow shop, Lot streaming, Buffer's capacity, Learning effect

[*] Corresponding author: Tel / Fax: +98 28 33670051.
E-mail address: Es.Mehdizadeh@iau.ac.ir(E. Mehdizadeh)

1. Introduction

Scheduling is the arrangement of available resources or the allocation of limited resources such as meetings, conferences, exams, people, jobs, machines, etc. to a set of activities in the form of a pattern over some time so that the constraints are satisfied and several specified goals are achieved. A hybrid flow shop (HFS) is a branch of flow shop scheduling problem, that is one of the most commonly used production scheduling issues and have a very important role in industrial production and production systems. Therefore, by HFS, many production systems can be displayed for example, in electronic[1], textiles [2], paper [3], concrete production [4], petrochemical and pharmaceutical industries [5], photography film production [6]. Also, there are many examples of this problem in non-production fields such as civil engineering [7], container shipping systems [8], architecture [9].

In hybrid flow shop studies mentioned above, each job is only allowed to move downstream when the operation is over. However, in today's competitive world, such do's and don'ts can have detrimental and negative effects on scheduling performance and cannot be found in many real manufacturing systems such as ceramic tile, electronics, and connectors industries where jobs consist of many identical subsets. By overlapping the required operation to process the production order can attain more productivity from a production system. This method is called “Lot streaming”. It means that in the operations a job becomes into several subsets of jobs that allow them to be scheduled separately in multi-stage production systems. The concept of “lot-streaming” as an approved method for reduction of makespan, required cycle time, required equipment for material transfer, and average work in process has emerged. So, in scheduling problems with lot streaming, three purposes should be considered; first, the number of job’s smaller sub-lots should be recognized. The second is, deciding on the allocation of machines for any sub-lot and finally the scheduling of allocation of any sub-lot to the machine. As an example, Figure 1 shows scheduling with and without the lot-streaming concept.

{Please insert Fig. 1. here}

The learning effect in scheduling configuration is one of the assumptions which we have considered in this article. In the case of HFS, any of the machines can process various jobs; so, learning increases in each machine because each machine may have a large amount of learning after any processing. Hence the processing time will be shorter. This concept is the

learning effect. According to [10], the position-based learning effect in a single machine environment is as follows:

$$p_{jr} = p_j r^{\alpha_i} \text{ j}, \quad \text{r} = 1, \ldots, \text{n} \tag{1}$$

He is considered $p_j$as the processing time of job j. If the job is placed in position r, then $p_{jr}$is the real-time of job j, and $\alpha \leq 0$ is the learning index where $\alpha = log_2 LR$ where LR is the learning rate. Therefore, if the location r is scheduled by worker i with the learning effect α_i, the real processing time of each job is denoted by $p_j r^{\alpha_i}$. We assumed that the processing time of the job depends on their position in the sequence arrived at each stage.

Finally, buffer capacity constraint has also been considered to make the problem more realistic. In a scheduling environment, when the job processing in one stage is completed, it should be going to the next stage and be assigned to its related machine at the next stage. But sometimes the next machine is being processed, so the job completed in the previous stage must stay on the same machine until the next machine is free, in which case it will block the relevant machine and prevent the next job from entering it. The existing intermediate buffer solves this problem partially. In this way, after each stage, a place with a limited capacity is considered, and the jobs after finishing their processing go there, until the machine in the next stage going to be free.

To the best of our knowledge, the hybrid flow shop scheduling problem with lot-streaming (HFS-LS), buffer capacity, and learning effect with two objectives have not yet been addressed. In the other hand, Gupta [11] showed that an environment with two stages can be NP-hard although there is only one machine in one of the stages. Wang et al. stated the hybrid flow shop batching with two-stage and lot streaming is NP-hard [12]. Also, Li et al. showed HFS problem and lot streaming with an efficient multi-objective algorithm for the variable sub-lots is belonging to the NP-hard class [13]. So, we can say that our problem is also NP-hard we will use meta-heuristic algorithms to solve our problem.

The next sections of our paper are as follows: In Section 2 a survey of the literature on the related field is provided. Section 3 defines the problem and the mathematical model.In Section 4, the algorithms to solve the multi-objective problems are proposed. Section 5 deals with the computational results of the proposed model and their review. At last, in Section 6, the conclusions and future studies are given.

2. Literature Review

2.1. Hybrid flow shop scheduling problem with lot-streaming

In recent years, the concept of lot streaming in hybrid flow shop scheduling has been taken into consideration.Cheng and Sarin [14], developed the lot streaming single product problem in hybrid flow shop. They minimized the makespan and the total completion times for all sub-lots and performed four heuristic methods based on mathematical programming. Nejati et al. [15], addressed the lot streaming problem in an assembly hybrid flow shop scheduling problem with two-stage and job shift constraints and used genetic and simulated annealing algorithms to solve their model.Zhang et al. [16], proposed an effective modified migrating birds optimization for HLFS, by minimizing the total flow time. They introduced the SWT rule for scheduling the jobs arriving at stages. Also, they presented some efficient modifications to raise the efficiency of the MBO algorithm, containing a composed the scouting phase, neighborhood search strategy, the dynamic solution acceptance criteria, and two competitive mechanisms. Lalitha et al. [17], formulated a MILP model for the HFS problem with lot streaming to create a program that minimizes makespan. Gong et al. [18], developed the Blocking lot-streaming flow shop scheduling problem with two objectives and used a multi-objective hybrid artificial bee colony algorithm for solving it. They also proposed several modifications to make the basic artificial bee colony algorithm adaptable to multi-objective optimization. Chen et al.[19], considered multi-objective hybrid programming for solving the HFS problem by minimized makespan and energy consumption. They also considered lot streaming in their research. They used NSGA2 and MOEES algorithms to optimize their model.Wang et al. [12], used a MILP model with two inequalities in the field of lot streaming and interchangeable sets with changeable sub-lots, incompatible product groups, and sequence-dependent setup times in a two-objective HFS for minimizing the total weighted completion time. They evaluated nine heuristics which C1J, C1W, and C1R (Stage 1's Capacity; J: Johnson's Rule; W: Weighted Shortest Processing Time; R: Random Key Method), presented optimal solutions for samples with 1 to 10 sizes. Qin et al. [20], investigated a multi-stage HFSP with lot sizing, sequence-dependent setup times, and calendar constraints. They proposed these assumptions in a real-world printed circuit board assembly shop.

2.2. Scheduling problem and Learning effect

The learning effect is well identified in several industrial environments which have been applied for many years ago.Biskup[10] considered learning effects on position for two objectives in single-machine with minimizing the due date and the makespan. Cheng and Wang [21], considered the learning effect in scheduling problem with a single machine. They minimized the maximum lateness by using a linear processing time. Also, Eren and Güner[22], addressed the learning effect in bi-criteria parallel machine scheduling problem to minimize the total tardiness and total completion time.Cheng et al. [23], adopted the learning effect in the scheduling area just in recent years.Gao et al. [24], considered a No-wait two-machine permutation flow shop scheduling problem with learning effect and common due date where the processing time of a job is given as a function of its position in the sequence and its amount of resource allocated to this job. Sun et al.[25] considered the flow shop scheduling problem of minimising the total weighted completion time in which the processing times of jobs are variable according to general position weighted learning effects.Xin et al. [26], studied on a permutation flow shop energy-efficient scheduling problem that considers multiple criteria aiming to find the optimal job processing sequence and conveyor speed that minimise both the makespan and total energy consumption.Bai et al. [27], addressed a flow shop scheduling problem to optimize maximum lateness, where learning effect is introduced for each task. They used effective branch and bound (B&B), a discrete artificial bee colony algorithm with hybrid neighbourhood search mechanism algorithms for small and medium-scale instances.

2.2.1. Hybrid flow shop with learning effect

Also, the learning effect in hybrid flow shop scheduling problem has been studied in recent years. For example; the position-based learning effect with sequence-dependent setup time by minimizing the total tardiness and makespan addressed by Pargar and Zandieh[28]. Mousavi et al. [29], addressed the *n* independent jobs in hybrid flow shop scheduling problem with minimizing the total tardiness and the makespan. Lei et al. [30], considered HFSP with total energy consumption, and teachers' teaching and its three sub-problems are optimized simultaneously to minimize the energy consumption and tardiness. Shahvari and Logendran[31], addressed machine-dependent and sequence-dependent family setup times in the HFS batch scheduling problem.

2.3. Buffer capacity and scheduling problem

Fu et al. [32], considered incompatible job families in FS scheduling problem with limited buffer. They developed a method with two-stage which embedded with a DE (Differential Evolution) algorithm. Their objective was to find out scheduling and batching for all jobs to minimize completion time. Zhao and et al. [33] considered permutation flow shop scheduling problem with limited buffers between successive machines.Zhang et al. [34], have considered limited buffer in FS problem with minimizing themakespan in two-stage which consisting of discrete processor, a batch processor, and incompatible job families. Gu et al. [35], tried to minimize the completion time of the two-stage flow shop scheduling problem with a limited buffer. They presented a Lagrangian relaxation that allows use for each problem which is based on the analysis approach. Zohali et al. [36], studied on lot scheduling problem of inflexible flow shops with buffer capacity aiming to minimize the setup and inventory holding of both semi-finished and final product costs. They solved their problem with two linear mathematical models.

2.3.1. Hybrid flow shop with buffer capacity

Yaurima et al. [37], presented a genetic algorithm for solving the HFS with unrelated parallel machines, sequence-dependent setup time, limited buffer, and machine availability constraints. They proposed a genetic algorithm for a problem without limited buffers. Hakimzade and Zandieh[38], studied a HFS environment with a limited buffer and sequence-dependent setup time with aiming to minimize total tardiness and completion time. Safari et al. [39], proposed a model for minimizing the makespan of a hybrid flow shop scheduling problem with intermediate buffers and resource constraints and used a hybrid algorithm based on genetic algorithm and variable neighborhood search. Yaurima et al. [40], considered the hybrid flow shop scheduling problem with minimizing the total completion time and energy consumption for solving. They used work in progress buffers, setup time, and unrelated machines. Jiang and Zhang[41], investigated an energy-oriented scheduling problem for HFS-LB with limited buffers. They minimized the non-processing energy objectives and total weighted tardiness. Lin et al. [42], addressed the re-entrant-HFS problem with stockers, and also, they proposed a hybrid harmony search and the genetic algorithm to solve their problem by minimizing mean flow time and the makespan. They considered limited inventory buffer capacity, job permutation, and transfer time between stages.

A review of the papers, our research shows that important assumptions such as learning effect and buffer capacity in the HFS scheduling problem whit lot streaming have not been

considered. For this purpose, at first, a mathematical model is presented. Then the Cplex solver of GAMS software is used to solve the small scale of the problems and since the problem is NP-hard,non-dominated sorting genetic algorithm(NSGA-II) and non-dominated ranking genetic algorithm (NRGA)algorithms are used to solve the large scale of the problems.

3. Problem definition and mathematical model

In this paper we considered multiple parallel machines in each stage in our problem, in which all machines are available at time zero. At one time only one job can process by each machine and every job turning into several sub-lots; so that they can be operated between successive stages and the sub-lots of different jobs cannot be intermingled. At one time, only one machine can process each sub-lot and they can't integrate, and at time zero the jobs are available. The processing time for each activity is a positive integer. There is an intermediate buffer between all stages. According to [41], an intermediate buffer of stages can be considered as a stage with machines with zero processing time, so these problems can be considered as a blocking problem that a buffer considered as a stage which has i machines that are parallel, and the job's processing times are zero. So, the completion time of the job in the buffer is equal to its leaving time from the previous stage. Finally, the completion time of each job in the buffer is equal to the time of leaving from the previous stage. In advanced manufacturing companies, due-date is a key index for managers to make decisions; because just-in-time sending from the factory to the customer has a great impact on cost savings and credit services, and the reputation of the company. Therefore, in this paper, we considered the total delay as one of the goals.

Sets:

M	Set of stages (indexed by k, and m where $m = \|M\|$)
J	Set of jobs (indexed by j, j', and n where $n = \|J\|$)
R	Set of position (indexed by r)
σ_k	The number of machines at stage k (indexed by i)
l_j	The number of sub-lots of job j (indexed by e)

Parameters:

p_{kj}	The processing time of job j in stage k
α	Learning ratio where $0 < \alpha \leq 1$

Q A very large positive number (e.g.$Q = \sum_{k\in M}\sum_{j\in J} p_{kj}$)

d_j Deadline of job j

r^{α} learning effect on rth position

Decision variables:

ST_{kjer} The starting time of eth sub-lot of job j at stage k in position r

CT_{kjer} The completion time of eth sub-lot of job j at stage k in position r

D_{kjer} The departure time of eth sub-lot of job j at stage k in position r

Cmax Makespan of schedule.

T_j The tardiness of job j

X_{kjir} Equal 1 if job j assigned to machine i at stage k in Position r; 0 otherwise.

$Y_{kjj'r}$ Equal 1 if job j precedes job j' to be assigned to the same machine at stage k in position r; 0 otherwise

Equations:

$$\min Z_1 = Cmax \tag{2}$$

$$\min Z_2 = \sum_{j=1}^{n} T_j \tag{3}$$

$$\sum_{i=1}^{\sigma_k}\sum_{r=1}^{n} X_{kjir} = 1 \tag{4}$$

$$ST_{kjer} \geq 0 \tag{5}$$

$$CT_{kjer} - ST_{kjer} = p_{kj} * r^{\alpha} \tag{6}$$

$$ST_{k+1jer} - CT_{kjer} \geq 0 \qquad \forall k = 1,\ldots,m-1 \tag{7}$$

$$ST_{kj,e+1,r} - CT_{kjer} \geq 0 \qquad e = \left[1,\ldots,l_j - 1\right] \tag{8}$$

$$Y_{kjj'r} + Y_{kj'jr} = 1 \tag{9}$$

$$ST_{kj1r} - CT_{kj'l_{j'}r} + Q\left(3 - Y_{kj'jr} - X_{kjir} - X_{kj'ir}\right) \geq 0 \tag{10}$$

$$CT_{kjer} = D_{k-1jer} + p_{kj} * r^{\alpha} \tag{11}$$

$$D_{kjer} + p_{kj} * r^{\alpha} \leq CT_{kjer} + Q\left(1 - Y_{kj'jr}\right) \tag{12}$$

$$Cmax \geq CT_{mjl_jr} \tag{13}$$

$$T_j \geq CT_{mjl_jr} - d_j \tag{14}$$

$$ST_{kjer},\ CT_{kjer},\ T_j,\ Cmax,\ D_{kjer} \geq 0 \tag{15}$$

$$X_{kjir} \in \{0,1\} \tag{16}$$

$$Y_{kjj'r} \in \{0,1\} \tag{17}$$

$$\forall j, j' \in J,\ k \in M,\ r \in R,\ i \in \{1,\ldots,\sigma_k\},\ e = [1,\ldots,l_j]$$

Equation (2) considers the minimization of the completion time of jobs. Equation (3) minimizes the total tardiness of jobs. Equation (4) ensures that each activity in all stages is processed by only one machine at any stage. Equation (5) ensures that the start time of each sub-lot must be a positive number. Equation (6) ensures that the difference between the start time and end time of each sub-lot at each stage in position r is equal to the processing time with the learning effect. Equation (7) ensures that the sub-lot of jobs is processed only if they were processed at the previous stage. Equation (8) ensures that each sub-lot is processed on the same stage if its previous sub-lot is processed. Equations (9) and (10) express the time capacity of the machine. According to these equations, the completion time of the last sub-lot of each job should be less than or equal to the start time of the first sub-lot of the next job in the same machine and at each stage. Equation (11) ensures that after the processing of each sub-lot of the job is completed, the next sub-lot of the same job begins immediately. Equation (12) ensures that two sub-lots are not processed in one position and on one machine at each stage. For equation (13) we have to calculate the completion time of jobs. Equation (14) calculated job tardiness. Equations (15), (16), and (17), show the range of problem decision variables.

4. Solution approach

Since our proposed model is NP-hard, we used the NRGA[43]and NSGA-II algorithms[44] for solving large scales of the problem.

4.1. Solution representation

In this paper, we use Vector chromosomes and considered the number of the stage is S, the number of jobs is J, and the number of machines in each stage is *M*.

For each stage, there is a permutation vector of dimensions $J + M - 1$. For example, suppose that we have 3 machines in the first stage and 8 jobs; the chromosome of this stage is a random permutation of dimensions 10 (8+3-1= 10). The first sequence created; 8, 2, 3, 9, 7, 6, 1, 10, 5, 4. Numbers greater than $J(j>8)$ are the separators between the jobs assigned to

each machine; 8, 2, 3, 9, 7, 6, 1, 10, 5, 4. Based on this way of displaying the answer, the allocation and arrangement of jobs on each machine in stage *s=1* is determined. For stage *s>1* the jobs are assigned to the machines in the next stage, according to the first available machine (FAM) rule. So, we have:

{Please insert Fig.2. here}

Then, according to Figure2, based on the number of each job sub-lots, the placement of the sub-lots and their assignment is determined.

4.2. Selection

In this paper, the roulette wheel method was used to select chromosomes. The relationship between the chance of selecting each chromosome and the value of the objective function is calculated according to the following formula:

$$P_i = e^{-\frac{SP*Z_i}{w}} \quad (18)$$

In the above formula, P_i represents the chance of selecting each i chromosome, and Z_i represents the value of the objective function for that chromosome, *SP* is the selection pressure and w is the worst value of the objective function in the present population. Finally, Pi values are normalized.

4.3. Crossover operator

In this paper, according to the structure of chromosomes, a two-point crossover has been used to create a crossover in each row of chromosomes. In a two-point crossover, two points from each row are randomly selected along both chromosomes and their middle parts replace with each other (Figure 3).

{Please insert Fig.3. here}

4.4. Mutation operators

The mutation operator includes three parts of swap, reversion, and insertion which are selected randomly to do mutation each time using these operators.

4.4.1 Swap operator

The two elements are randomly selected and swapped (Figure 4).

{Please insert Fig. 4. here}

4.4.2. Reversion

In this method, in addition to swapping the two elements, the location of their middle elements is also reversed (Figure 5).

{Please insert Fig. 5. here}

4.4.3. Insertion

In this method, the first selected element is deleted and transferred to the position after the second element (Figure 6).

{Please insert Fig. 6. here}

4.5. Impact of learning on scheduling jobs

To show the impact of learning effect on job scheduling in HFS systems we considered this factor in the following example. To calculate the due date, the following formula has been used:

$$d_j = \sum_{k=1}^{|M|} kp_j \times \left(\frac{\max\left(\sigma_{k\in m}^{k}\right)}{|M|} \right) \times (1 + \text{random} \times 3) \; \forall j \in n \tag{19}$$

The random means, random number of uniform distributions between (0,1). Here, to show the learning effect impact on the jobs assignment to machines and their sequence, three scenarios (a,b,c), once with and once without learning effect are evaluated. We showed p and q for weights of makespan and total tardiness, respectively. The job's due date is shown by dj. Figure 7 represents the optimal schedule of the three proposed scenarios and its aim is to figure out the learning effect impact and the objective ratios on the jobs optimum scheduling.

According to Figure 7 (a), the weight vectors are the same (*p=q=50%*), and the learning effect leads to reducing the total tardiness and makespan. In the below figure, we can see the impact of the learning effect in reducing the effective factors in optimizing and changing the optimal jobs schedule. In Figure 7 (b and c), we changed the weight vector to (0.25, 0.75), and again we can see that the learning rate reduced the makespan and total tardiness. The above example shows that the learning effect has a good performance in reducing processing

times through proper scheduling. We assumed that the job processing time and sub-lots are as Table 1:

{Please insert Table. 1. here}

So For above mentioned displaying way, we have Figure 7(a, b, c).

{Please insert Fig. 7 here}

4.6. Parameter tuning

In our paper, for adjusting the parameters of the algorithms, Taguchi method in MINITAB 17 software is used[45]. This method converts repeated data obtained from experiments into a marker of change, which is called signal-to-noise conversion. The purpose of the Taguchi method is to maximize this marker.

Based on these specified numbers, a sample problem based on Taguchi's proposed levels is run and the answers are recorded. The ratio of relative percentage deviation (RPD) was used, and its formula is:

$$RDP = \frac{\text{performance answer} - \text{best answer}}{best\ \ answer} \times 100 \tag{20}$$

It should be noted that the index used to set the parameter is the average distance from the ideal answer, which is used in many studies. Since the lower the RPD index is more desirable, so during a performance of the Taguchi method, the "smaller is better" option is used. Also, whereas higher the signal index to perturbation is more desirable; therefore, the maximum points of the graph for each parameter are considered and accept the corresponding level as the optimal level. The proposed NSGA-II and NRGA parameters for adjustment include N_{pop} (number of chromosomes), P_c (crossovers rate), P_m (mutation rate) and S_p(selection pressure).

Table 2 shows the levels of each parameter and the means of S/N ratios diagram for two algorithmsareshownFigure8 and9.Based on the above diagram, the optimal level value of the NSGA-II and NRGA algorithm parameters are as Table 3.

{Please insert Table 2 here}

{Please insert Fig 8 here}

{Please insert Fig 9 here}

{Please insert Table 3 here}

5. Computational results

In this section, at first, an example is developed to validate the proposed mathematical model. It is assumed that the processing time was the same on all machines. The bi-objective mathematical model converted to single objective mathematical model by weighting method and solved using the Cplex solver of GAMS.Table 4 shows the problem inputs, i.e., the number of activities with sub-lots related to each, processing time at each stage, and due date.

{Please insert Table 4. here}

The Gantt chart related to the problem after solving it by weighting method in GAMS is shown in Figure 10.

{Please insert Fig.10. here}

As mentioned, due to the NP-hard of the problem, we use the Genetic algorithm to solve the large-scale of the problems.To evaluate the efficiency of the proposed genetic algorithm,10 examples have solved by GAMS software and Genetic algorithmin single objective model. As can be seen in Table 5, the genetic algorithm has obtained acceptable answers. It is emphasized that in this section, the bi-objective model is converted into a single-objective model by weighting method, and then it is solved with the Cplex solver of GAMS software and genetic algorithm.

{Please insert Table 5. here}

To evaluate the algorithms and compare the performance of them, we randomly generated 30 sample problems and 6 criteria are presented: Number of Pareto Solutions (NOPS), Criteria for Maximum Diversity, Spacing criteria, Mean ideal distance (MID) and Multi-Objective Coefficient of Variation (MOCV). Then we execute the problems in different sizes

using NRGA and NSGA-II algorithms and calculate the evaluation indicators of the algorithms for them and compare the efficiency of the mentioned algorithms. In continuance, Table 6 shows the sample dimensions of random problems and the range of parameter are: The sub-lots for each job ~U (2,5), Processing time ~U (5,10) and Due date ~U (25,70)). Also, the computational results are given in this tables and Figures 11-17.

{Please insert Table 6 here}

{Please insert Fig. 11 here}

{Please insert Fig. 12 here}

{Please insert Fig. 13 here}

{Please insert Fig. 14 here}

{Please insert Fig. 15 here}

{Please insert Fig. 16 here}

{Please insert Fig. 17 here}

5.1. Statistical analysis of computational results

To examine the efficiency of the implemented algorithms, these two algorithms were analyzed based on their performance evaluation indicators to recognize the superiority of the algorithms.According to the obtained p-value in Table 7, there is a considerable difference between the diversity indexes of algorithms. Which according to Figure 18, the NRGA algorithm has better performance in this index.

{Please insert Table 7 here}

{Please insert Fig. 18 here}

Based on the p-value obtained from the variance analysis for the Spacing index in Table 8, it can be said that there is no important difference between our algorithms.

{Please insert Table 8 here}

According to the obtained p-value in Table 9, there is a considerable difference between the Nos index of algorithms, and as presented in Figure 19, the NRGA algorithm has better performance in this index.Based on the p-value obtained in Table 10, could say that for the MID index there is no significant difference between these algorithms.

{Please insert Table 9 here}

{Please insert Fig. 19 here}

{Please insert Table 10 here}

According to the obtained p-value in Table 11, there is a considerable difference between the DM index of algorithms, and as presented in Figure 20, the NRGA algorithm has better performance in this index.

{Please insert Table 11 here}

{Please insert Fig. 20 here}

According to the obtained p-value in Table 12, there is a considerable difference between the MOCV index of algorithms, and as presented in Figure 21, the NRGA algorithm has better performance in this index.

{Please insert Table 12 here}

{Please insert Fig. 21 here}

Based on the p-value obtained in Table 13, we could say that there is no significant difference between these algorithms for the CPU-time index.

{Please insert Table 13 here}

5.2. Results analysis

We used NRGA and NSGA-II algorithms to evaluate the proficiency of the proposed model. The HFS problem with lot-streaming, learning effect, and buffer capacity, has been raised for the first time. The main components of algorithms have been studied in the literature and the necessary mechanisms have been applied for this problem. The algorithms were evaluated in terms of performance measurement criteria by using analysis of variance. The p-value of each criterion is shown in the corresponding table. In cases where there is a significant difference, the box-plot diagram is used to display them.

Although smaller values are desirable for MOCV, CPU time, Spacing, and MID criteria, for NOS, DM, and Diversity criteria, larger values show the better performance of the algorithm. According to Table 7, at the 95% confidence level, there is a significant difference

for the Diversity Index, which according to Figure 11, it is clear that NRGA has a better performance in this index. According to Tables 8, 10, and 13, there is no significant difference between Spacing, MID, and CPU-time criteria between the two algorithms.

According to the results of Table 11, there is an important difference for the DM index at the level of 95% and as shown in Figure 20, NRGA is more desirable. Also, according to Table 12 and Figure 21, NRGA has better performance in MOCV criteria. Just according to Table 9 and Figure 19, NSGA-II has better performance than NRGA. Therefore,the following results can be obtained:For Diversity, DM, and MOCV criteria, the NRGA algorithm performs better than NSGA-II.The algorithms have similar performance for Spacing, MID, and CPU time criteria.For NOS criteria the NSGA-II algorithm performs better than NRGA.

From the theoretical point of view, it can be concluded that the proposed algorithms have appropriate efficiency for solving the proposed model of this paper, and in particular, the NRGA algorithm has better performance than NSGA-II in 3 criteria.From the practical point of view, the proposed model in this paper, we showed that using both the learning effect and lot streaming reduces the completion time and total tardiness. Because, as discussed in the literature, job splitting can improve the scheduling process; also, the impact of the learning effect for reducing the processing time, on the improvement of objective functions is inevitable.

6. Conclusion

In this study, we considered the hybrid flow shop scheduling problem with lot streaming by a bi-objective mathematical programming model which limited to buffer's capacity and learning effect to minimize makespan and total tardiness. Then, by weighting method, the multi objective model converted to single objective model and GAMS software was used to solve the small size problems to show the performance of the mathematical model. Inspired by previous studies, to optimize the model, NSGA-II and NRGA based on the Pareto solution were used to solve the large-scale problems. To increase the efficiency of the proposed algorithms, their input parameters have been adjusted by Taguchimethod at the most optimal level. To illustrate the performance of the proposed Meta heuristic algorithms at first, the obtained results of the algorithms compared with GAMS outputs in single mode. The obtained results show the performance of the GA algorithm. Finally, 30 instance problems are randomly generated and six indicators were used to compare the algorithms. After performing the experiments and comparing the algorithms with each other, the results show For Diversity, DM, and MOCV criteria, the NRGA algorithm performs better than NSGA-

II.The algorithms have similar performance for Spacing, MID, and CPU time criteria.For NOS criteria the NSGA-II algorithm performs better than NRGA.

For future study, this research can be expanded in several directions. First, develop a mathematical model by considering limitations such as interruption, human factors, machine failure, and the like. Second, other algorithms can be applied to solve the problem. Finally, this research can be considered in other multi stage fields such as job shop and open shop environment.

References

[1] Jin, Z. H., Ohno, K., Ito, T., et al. "Scheduling hybrid flowshops in printed circuit board assembly lines". *Production and Operations Management*, **11** (2) 216–230 (2002).

[2] Grabowski, J., Pempera, J. "Sequencing of jobs in some production system". *European Journal of Operational Research,* **125**, 535–550 (2000).

[3] Sherali, H. D., Sarin, S. C., Kodialam, M. S. "Models and algorithms for a two-stage production process". *Production Planning & Control*, **1** (1) 27–39 (1990).

[4] Guinet, A. G. P. "Textile production systems: a succession of non-identical parallel processor shops". *Journal of the Operational Research Society*, **42** (8) 655–671 (1991).

[5] Guinet, A., Solomon, M. "Scheduling hybrid flowshops to minimize maximum tardiness or maximum completion time". *International Journal of Production Research*, **34**, 1643–1654 (1996).

[6] Aghezzaf, E. H., Landeghem, H. V. "An integrated model for inventory and production planning in a two-stage hybrid production system". *International Journal of Production Research*, **40**, 4323-4333 (2002).

[7] Dror, M., Mullaserif, P.A. "Three stage generalized flowshop: scheduling civil engineering projects". *Journal of Global Optimization,* **9**, 321–344 (1996).

[8] Chen, L., Bostel, N., Dejax, P., et al. "A tabu search algorithm for the integrated scheduling problem of container handling systems in a maritime terminal". *European Journal of Operational Research,* **181**, 40–58 (2007).

[9] Allahverdi, A., Al-Anzi, F. S. "Scheduling multi-stage parallel-processor services to minimize average response time". *Journal of the Operational Research Society*, **57**, 101–110 (2006).

[10] Biskup, D. "Single-machine scheduling with learning considerations". *European Journal of Operational Research,* **115**, 173-178 (1999).

[11] Gupta, J.N. "Two-stage hybrid flowshop scheduling problem". *Journal of the Operational Research Society*, **39**, 359–364 (1988).

[12] Wang, S., Kurz, M., Mason, S. J., et al. "Two-stage hybrid flow shop batching and lot streaming with variable sub-lots and sequence-dependent setups". *International Journal of Production Research,* https://doi.org/10.1080/00207543.2019.1571251, (2019).

[13] Li, J. Q., Tao, B. X., Han, Y. Y., et al. "Efficient multi-objective algorithm for the lot-streaming hybrid flowshop with variable sub-lots". *Swarm Evolutionary Computation*, **52,** (2020).

[14] Cheng, M., Sarin, S. C. "Two-stage, Multiple-lot, Lot Streaming Problem for a 1+2 Hybrid Flow Shop". *International Federation of Automatic Control*, 448-453 (2013).

[15] Nejati, M., Mahdavi, I., Hassanzadeh, R., et al. "Lot streaming in a two-stage assembly hybrid flow shop scheduling problem with a work shift constraint". *Journal of Industrial and Production Engineering*, http://dx.doi.org/10.1080/21681015.2015.1126653, (2016).

[16] Zhang, B., Pan, Q. K., Gao, L., et al. "An Effective Modified Migrating Birds Optimization for Hybrid Flowshop Scheduling Problem with Lot Streaming". *Applied Soft Computing Journal,* http://dx.doi.org/10.1016/j.asoc.2016.12.021, (2016).

[17] Lalitha, J., Mohan, N., Pillai, V. "Lot streaming in [N-1](1)+N(m) hybrid flow shop". *Journal of Manufacturing* System, **44**, 12–21 (2017).

[18] Gong, D., Han, Y., Sun, J. "A Novel Hybrid Multi-Objective Artificial Bee Colony Algorithm for Blocking Lot-Streaming Flow Shop Scheduling Problems". *Knowledge-Based Systems*, 1-37 (2018).
[19] Chen, T. L., Cheng, C. Y., Chou, Y. H. "Multi-objective genetic algorithm for energy-efficient hybrid flow shop scheduling with lot streaming". *Annals of Operations Research*, https://doi.org/10.1007/s10479-018-2969-x, (2018).
[20] Qin, W., Zhuang, Z., Liu, Y., et al. "A two-stage ant colony algorithm for hybrid flow shop scheduling with lot sizing and calendar constraints in printed circuit board assembly". *Computers &Industrial Engineering*, doi: https://doi.org/10.1016/j.cie.2019.106115, (2019).
[21] Cheng, T. C. E., Wang, G. "Single machine scheduling with learning effect considerations". *Annals of Operations Research*, **98**, (1-4), 273-290 (2000).
[22] Eren, T., Güner, E. "A bicriteria parallel machine scheduling with a learning effect". *International Journal of Advanced Manufacturing Technology*, **40**, 1202-1205 (2009).
[23] Cheng, T. C. E., Kuo, W.H., Yang, D. L. "Scheduling with a position-weighted learning effect". *Optimization Letters, 293-306* (2014).
[24] Gao, F., Liu, M., Wang, J., Lu, Y. "No-wait two-machine permutation flow shop scheduling problem with learning effect, common due date and controllable job processing times". International Journal of Production Research, 2361-2369.https://doi.org/10.1080/00207543.2017.1371353, (2017).
[25] Sun, X., Geng, X., Liu, F. "Flow shop scheduling with general position weighted learning effects to minimize total weighted completion time". *Journal of the Operational Research Society*. 2674-2689, https://doi.org/10.1080/01605682.2020.1806746, (2020).
[26] Xin, X., Jiang, Q., Li, C., et al. "Permutation flow shop energy-efficient scheduling with a position-based learning effect". *International Journal of Production Research*, https://doi.org/10.1080/00207543.2021.2008041, (2021).
[27] Bai, D., Bai, X., Yang, J., et al. "Minimization of maximum lateness in a flow shop learning effect scheduling with release dates". *Computers & Industrial Engineering*, **158**, August 2021, 107309, (2021).
[28] Pargar, F., Zandieh, M. "Bi-criteria SDST hybrid flow shop scheduling with learning effect of setup times: water flow-like algorithm approach". *International Journal of Production Research*, 2609-2623, http://dx.doi.org/10.1080/00207543.2010.546380, (2012).
[29] Mousavi, S. M., Mahdavi, I., Rezaeian, J., et al. "An efficient bi-objective algorithm to solve re-entrant hybrid flow shop scheduling with learning effect and setup times". *Operational Research*, **16**, 1-36 (2016).
[30] Lei, D., Gao, L., Zheng, Y. "A Novel Teaching-Learning-Based Optimization Algorithm for Energy-Efficient Scheduling in Hybrid Flow Shop". *IEEE Transactions on Engineering Management*, **99**, 1-11 (2017).
[31] Shahvari, O., Logendran, R. "A comparison of two stage-based hybrid algorithms for a batch scheduling problem in hybrid flow shop with learning effect". *International Journal of Production Economics,* **195**, 227–248 (2018).
[32] Fu, Q., Sivakumar, A. I., Li, K. "Optimisation of flow-shop scheduling with batch processor and limited buffer". *International Journal of Production Research*, **50**, 8, 2267–2285 (2012).
[33] Zhao, F., Tanga, J., Wang, J., et al. "An improved particle swarm optimisation with a linearly decreasing disturbance term for flowshop scheduling with limited buffers". *International Journal of Computer Integrated Manufacturing,* **27**, 5, 488–499 (2014).
[34] Zhang, C., Shi, Z., Huang, Z., et al. "Flow shop scheduling with a batch processor and limited buffer". *International Journal of Production Research, DOI:* 10.1080/00207543.2016.1268730. (2016).
[35] Gu, H., Kononov, A., Memar, J., et al. "Efficient Lagrangian heuristics for the two-stage flow shop with job dependent buffer requirements". *Journal of Discrete Algorithms*, doi.org/10.1016/j.jda.2018.11.011. (2018).
[36] Zohali, H., Naderi, B., Mohammadi, M. "The economic lot scheduling problem in limited-buffer flexible flow shops: Mathematical models and a discrete fruit fly algorithm". *Applied Soft Computing Journal,* https://doi.org/10.1016/j.asoc.2019.03.054, (2019).
[37] Yaurima-Basaldua, V. H., Burtseva, L., Tchernykh, A. "Hybrid flowshop with unrelated machines, sequence-dependent setup time availability constraints and limited buffers". *Computers & Industrial Engineering,* **56**, 1452–1463 (2009).
[38] Hakimzadeh, A. S., Zandieh, M. "Bi-objective hybrid flow shop scheduling with sequence-dependent setup times and limited". *International Journal of Advanced Manufacturing Technology*, **58**, 309–325 (2012).
[39] Safari, G., Hafezalkotob, A., Khalilzadeh, M. "Hybrid genetic algorithm. A novel mathematical model for a hybrid flow shop scheduling problem under buffer and resource limitations-A case study". *Journal of Industrial and Systems Engineering*. **10**, 58- 77 (2018).

[40] Yaurima-Basaldua, V. H., Tchernykh, A., Villalobos, F. V., et al. "Hybrid Flow Shop with Unrelated Machines, Setup Time, and Work in Progress Buffers for Bi-Objective Optimization of Tortilla Manufacturing". *Algorithm*, **11**, 68, 1-23 (2018).
[41] Jiang, S. L., Zhang, L. "Energy-Oriented scheduling for hybrid flow shop with limited buffers through efficient multi-objective optimization". *IEEE,* **7**, 34477-34487 (2019).
[42] Lin, C. C., Liu, W. Y., Chen, Y. H. "Considering stockers in reentrant hybrid flow shop scheduling with limited buffer capacity". *Computers and Industrial Engineering*, **139**, 106-154 (2020).
[43] Al Jadaan, O., Rao, C. R., Rajamani, L. "Parametric study to enhance genetic algorithm performance using ranked based roulette wheel selection method". *Proceedings of the Genetic and Evolutionary Computation Conference*, 274-278 (2008).
[44] Deb, K. "Multi-objective Evolutionary Optimization: Past, Present and Future". *In Proceeding of the Fourth International Conference on Adaptive Computing in Design and Manufacture*, edited by I. C. Parmee, 225–236 (2000).
[45] Taguchi, G. "Introduction to quality engineering (White Plains, NY: Asian Productivity Organization, Unipub/Kraus International Publications), (1986).

Biographies

RojaRuhbakhshis currently a PhD student at the Department of Industrial engineering, Islamic Azad University, Qazvin Branch, Iran. She obtained her MSc degree from Islamic Azad University, Qazvin Branch in 2015 and her BSc degree from Mazandaran Institute of Technology in 2012 all in Industrial Engineering. Her research interests are in the areas of Project management, Production scheduling and meta-heuristic algorithms.

EsmaeilMehdizadeh is currently an Associate Professor at the Department of Industrial engineering, Islamic Azad University, Qazvin Branch, Iran. He received his PhD degree from Islamic Azad University, Science and Research Branch, Tehran in 2009, MSc degree from Islamic Azad University, South Tehran Branch in 1999, and BSc degree from Islamic Azad University, Qazvin Branch in 1996 all in Industrial Engineering. His research interests are in the areas of operation research, such as supply chain management, productionmanagement and inventory systems, production scheduling, fuzzy sets, soft computing and meta-heuristic algorithms. He has several papers in journals and conference proceedings. Also, he is a Managing Editor of International Journal of Optimization in Industrial Engineering.

MOHAMMAD AMIN ADIBI was born in 1984, Iran. He received the B.S. and M.S. degrees in Industrial Engineering from the Qazvin Islamic Azad University (QIAU), Qazvin, Iran, in 2006, and 2010 and the Ph.D. degree also in Industrial Engineering from Amirkabir University of Technology, Tehran, Iran, in 2014.Since 2015, he has been an Assistant Professor at the Industrial Engineering Department, QIAU. His research interests include

online optimization, optimization and data analysis interactions, dynamic scheduling, reinforcement learning, soft computing, and data analysis.

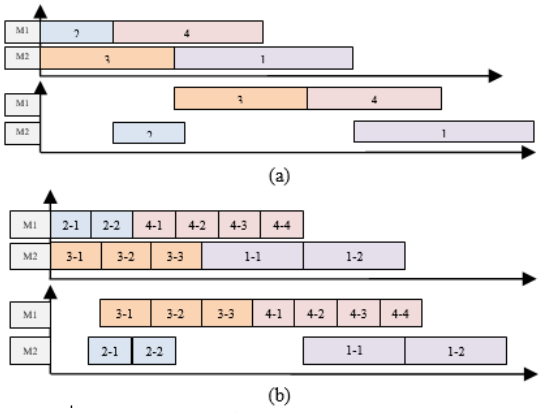


Fig. 1. Gantt chart for without (a) or with (b) lot-streaming

Assigned tasks to the machine1		Assigned tasks to the machine2		Assigned tasks to the machine3
8-2-3	9	7-6-1	10	5-4

Fig. 2. Example of job assigned to machines

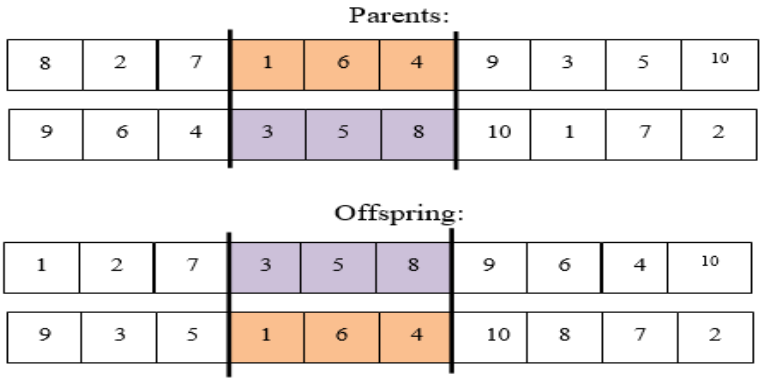


Fig. 3. Crossover operator

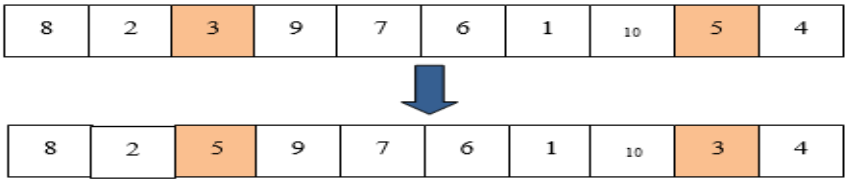


Fig. 4. Swap two elements

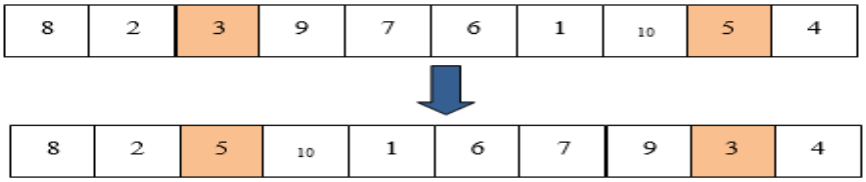


Fig. 5. Reverse two elements

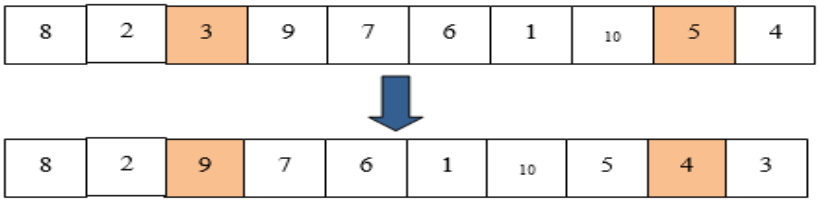
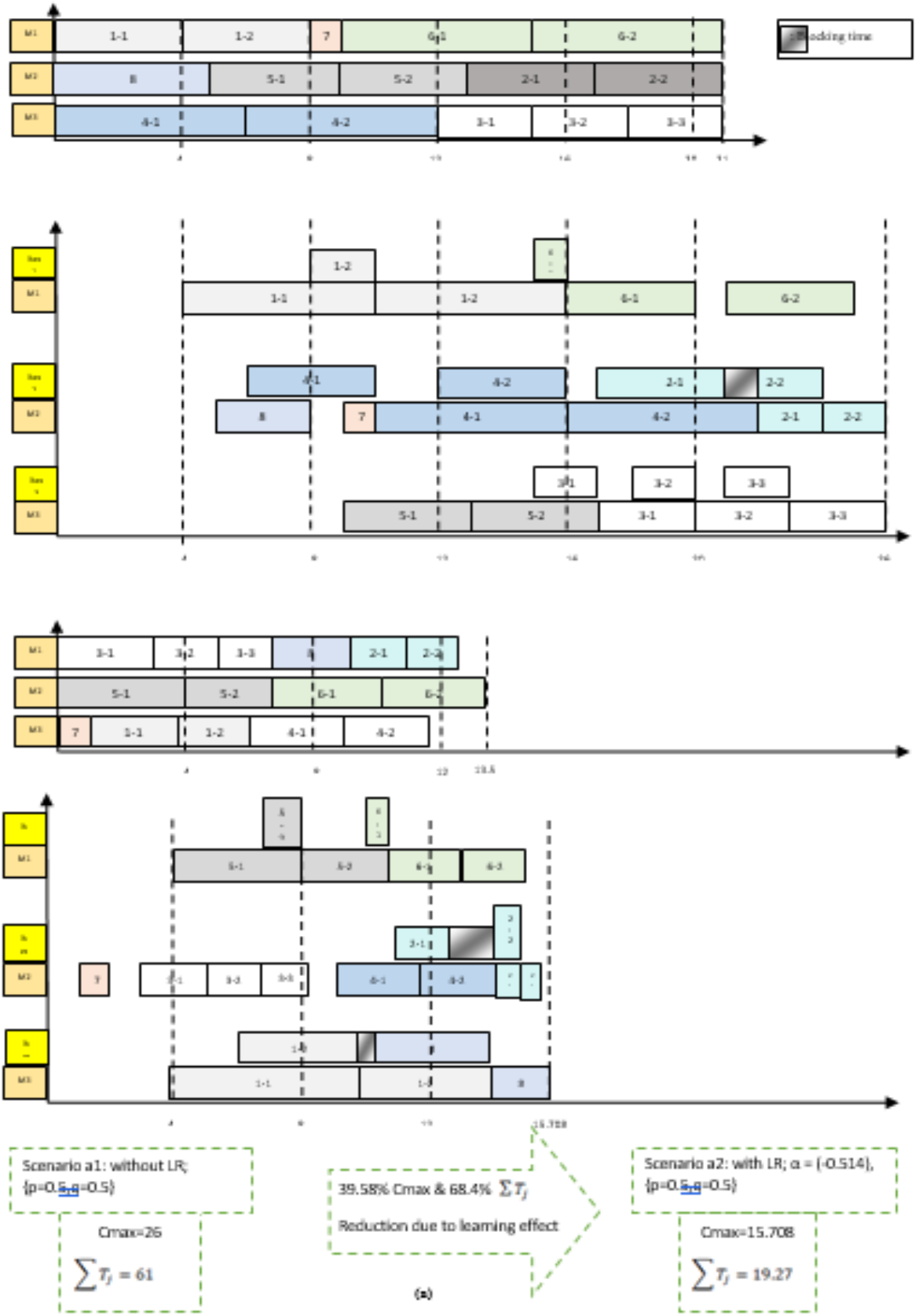
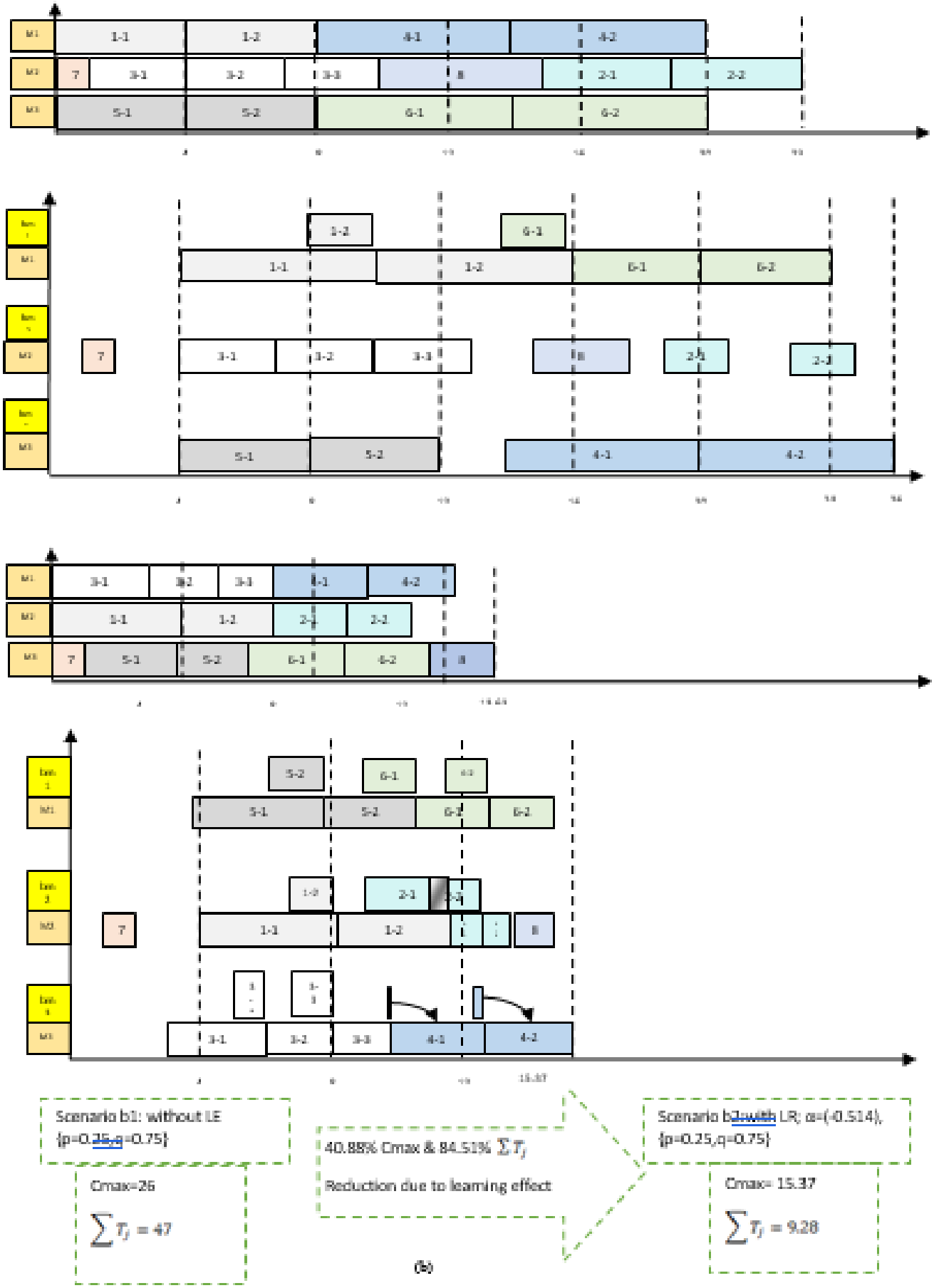


Fig. 6. Insertion of two elements

Table 1. The sub-lots and processing time of each job

Job (sub-lots)	S1	S2
1(2)	4	6
2(2)	4	2
3(3)	3	3
4(2)	6	6
5(2)	4	4
6(2)	6	4
7(1)	1	1
8(1)	5	3


Scenario a1: without LR;
{p=0.5, α=0.5}
Cmax=26
$\sum T_j = 61$
39.58% Cmax & 68.4% $\sum T_j$
Reduction due to learning effect
Scenario a2: with LR; α = (-0.514),
{p=0.5, α=0.5}
Cmax=15.708
$\sum T_j = 19.27$


Scenario b1: without LE
{p=0.25,q=0.75}
Cmax=26
$\sum T_j = 47$
40.88% Cmax & 84.51% $\sum T_j$
Reduction due to learning effect
Scenario b3:with LR; α=(-0.514),
{p=0.25,q=0.75}
Cmax= 15.37
$\sum T_j = 9.28$
(b)

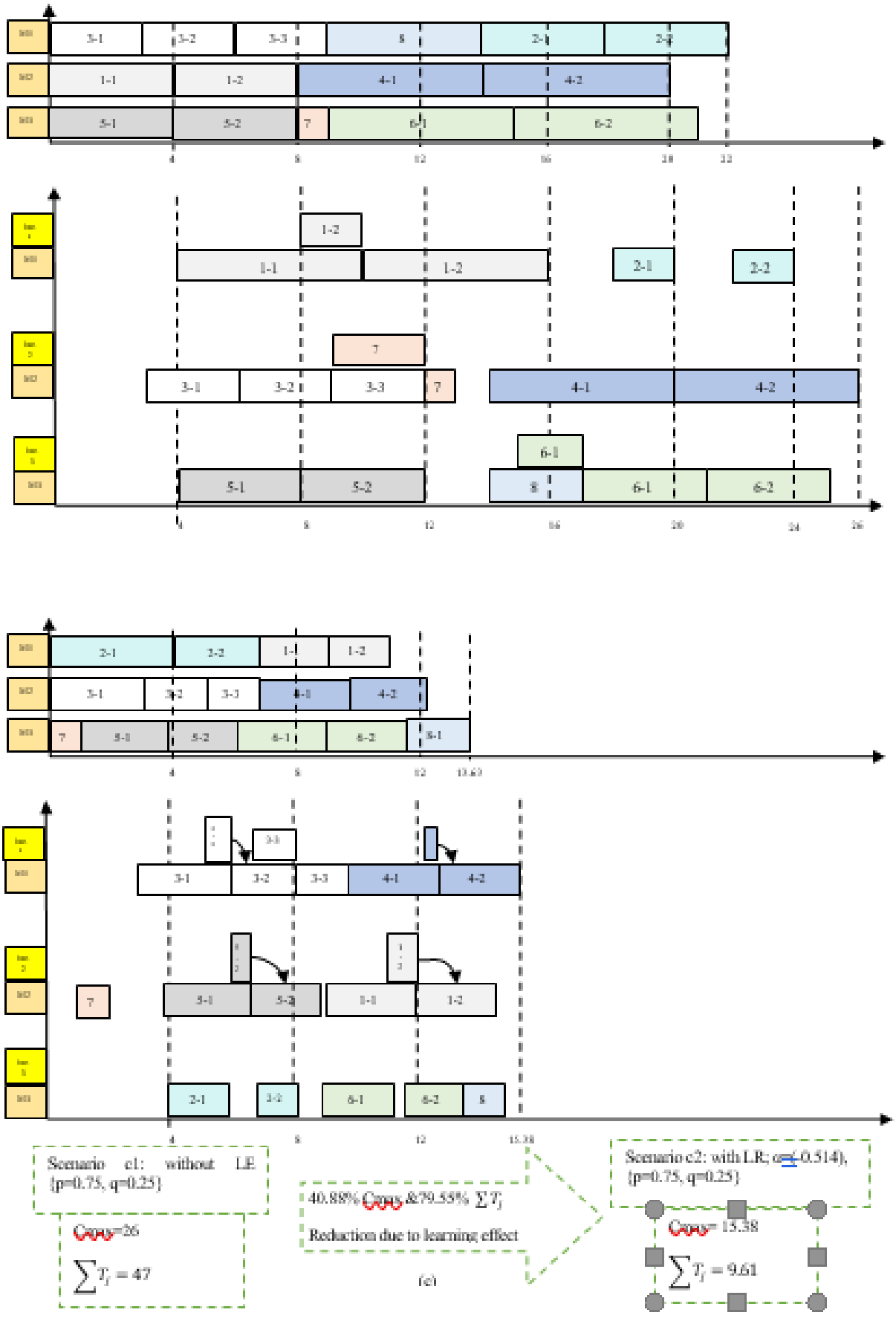


Fig. 7 The impact of the learning effect and objective coefficients on the (optimal) schedule of jobs

Table 2. The parameter level of NSGA-II and NRGA algorithm

Parameter	Layer		
	1	2	3
Npop	20	30	40
Pc	0.85	0.9	0.95
Pm	0.05	0.1	0.15
Sp	1	2	3

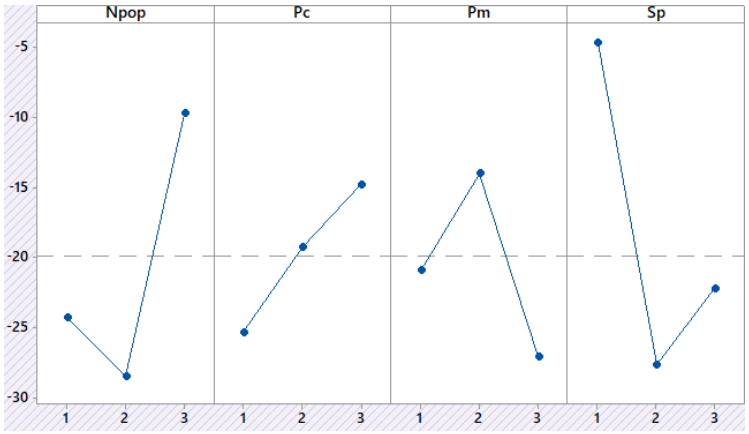


Fig. 8. The Means of S/N ratios diagram for the NSGA-II algorithm

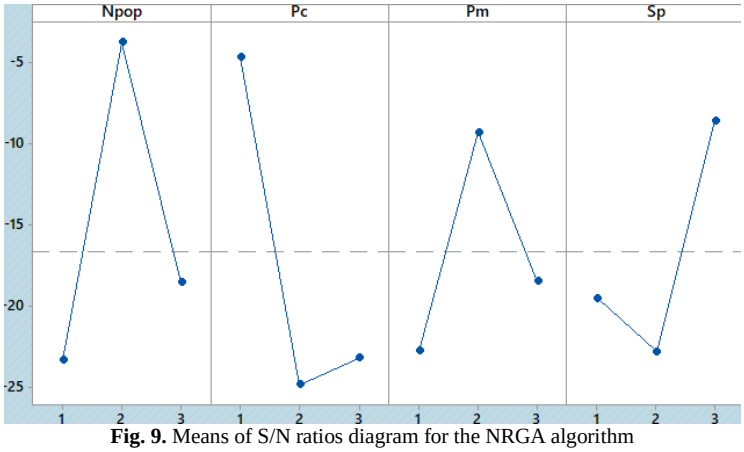


Fig. 9. Means of S/N ratios diagram for the NRGA algorithm

Table 3. Optimum value of the NSGA-II and the NRGAalgorithm parameters

Parameter	NSGA-II	NRGA
Npop	40	30
Pc	0.95	0.85
Pm	0.15	0.1
Sp	3	3

Table 4. Example parameters

job	Sublots	Processing time			Due date
		Stage 1	Stage 2	Stage 3	
1	2	7	5	7	30
2	2	8	8	2	25
3	3	2	8	4	12
4	2	8	3	8	17
5	1	6	8	7	15
6	2	2	8	8	35
7	3	3	5	6	18

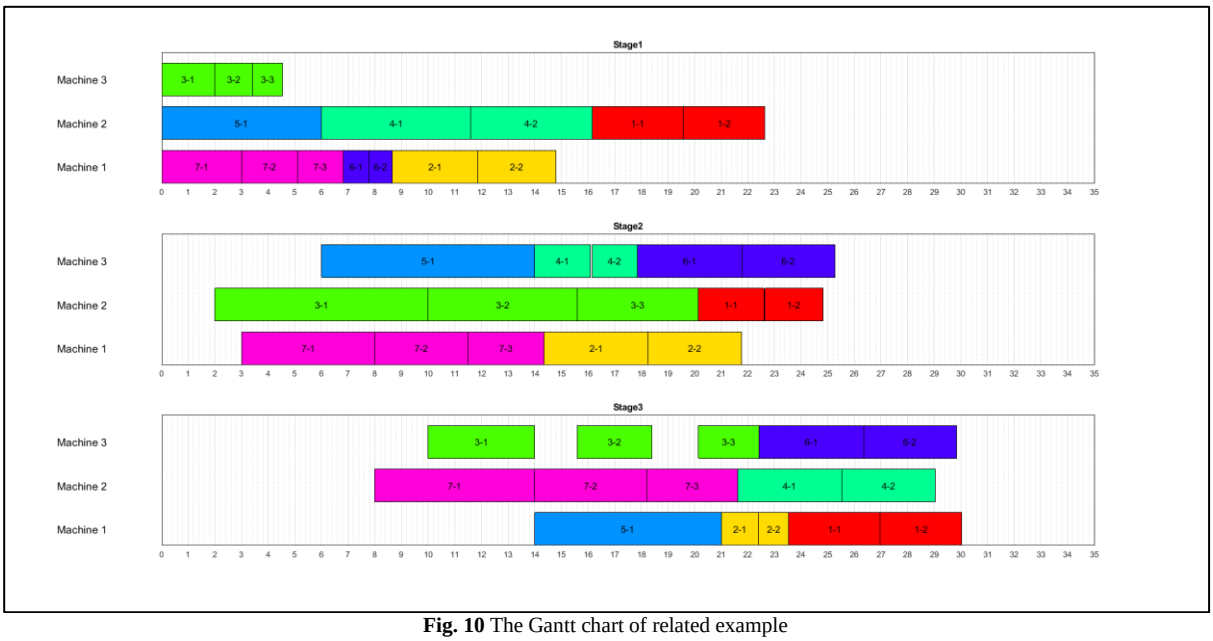


Fig. 10 The Gantt chart of related example

Table 5. Comparison between GAMS outputs and GA

N	**Structure**	**GAMS**			**GA**		
	$\|J\|/\|M\|/\|S\|$	Z_1	Z_2	CPU time (sec)	Z_1	Z_2	CPU time (sec)
1	$\|5\|/\|2\|/\|2\|$	14	4	3.19	14	4	20.65
2	$\|5\|/\|2\|/\|3\|$	17	5	4.05	17	5	23.645
3	$\|10\|/\|2\|/\|2\|$	18	7	81.21	18	7	26.317
4	$\|10\|/\|2\|/\|3\|$	23	9	86.65	23	9	30.212
5	$\|15\|/\|3\|/\|4\|$	30	13	265.65	32	15	35.318
6	$\|15\|/\|3\|/\|4\|$	35	15	243.12	37	17	39.804
7	$\|20\|/\|3\|/\|4\|$	56	18	631.73	60	21	46.332
8	$\|20\|/\|3\|/\|4\|$	66	26	642.49	69	29	53.051
9	$\|25\|/\|5\|/\|5\|$	93	33	1798.33	99	38	61.274
10	$\|25\|/\|5\|/\|5\|$	99	39	1938.62	108	27	70.711

Table 6. Computational results obtained problem samples by NSGA-II and NRGA algorithms

Test NO.	Problem number	The number of jobs	stage	The machinenumber in each stage	NSGA-II							NRGA						
					Diversity	Spacing	Nos	MID	DM	MOCV	CPU_time	Diversity	Spacing	Nos	MID	DM	MOCV	CPU_time
1	1	5	2	2	37.94	15.049	4	243.738	11.774	20.702	20.872	12.735	6.361	3	238.809	6.037	39.555	19.3361
2	2	5	3	2	150.052	4.356	9	152.999	35.209	4.345	20.68	139.632	9.513	6	180.828	27.55	6.564	18.8393
3	3	10	2	2	621.852	2.986	8	435.191	68.188	6.382	25.191	734.024	308.79	6	683.519	65.699	10.404	24.9274
4	4	10	3	2	501.779	5.045	8	429.533	61.334	7.003	29.398	715.325	257.845	6	634.449	63.233	10.034	27.4988
5	5	15	4	3	1715.572	117.441	9	1272.614	119.861	10.617	37.591	1176.19	435.125	8	1492.479	96.44	15.476	35.5994
6	6	15	4	3	631.354	176.696	10	1021.229	76.748	13.306	37.864	535.704	16.761	8	1331.775	59.905	22.231	36.685
7	7	20	4	3	785.616	261.599	7	1493.707	71.947	20.761	48.522	1033.631	179.374	8	1583.629	77.33	20.479	45.7194
8	8	20	4	3	1710.423	595.684	7	2632.062	106.224	24.778	111.629	4163.575	1143.413	8	2015.02	170.513	11.817	112.7414
9	9	25	5	5	3515.154	622.462	7	3868.239	150.945	25.627	199.267	6343.891	683.77	8	4323.132	209.724	20.613	180.205
10	10	25	5	5	5736.71	931.257	7	4367.714	171.052	25.534	172.057	5690.317	418.634	8	4870.445	201.576	24.162	156.2183
11	11	30	5	5	2769.472	39.464	8	5689.018	144.365	39.407	190.066	6311.748	1963.993	9	6359.395	234.263	27.146	173.7305
12	12	30	5	5	6874.746	33.985	8	6104.37	222.827	27.395	226.852	9589.341	1920.854	9	7410.957	260.042	28.499	209.5599
13	13	35	7	6	8391.917	60.294	10	9077.024	274.405	33.079	454.587	10338.51	230.684	9	11818.63	299.33	39.484	404.5746
14	14	35	7	6	9010.291	84.478	10	9505.241	286.451	33.183	422.326	9277.682	963.412	8	11532.97	260.942	44.197	387.2924
15	15	40	7	6	18534.94	3452.651	10	9849.839	377.952	26.061	602.592	15505.1	2854.829	8	11839.32	323.464	36.602	461.8561
16	16	40	7	6	17279.25	1097.252	10	12874.7	388.218	33.164	484.393	13474.99	793.684	10	17766.63	358.387	49.574	462.3092
17	17	45	7	6	24012.07	2376.468	9	18158.11	447.372	40.588	504.061	16292.35	3696.483	9	20813.06	363.24	57.298	476.9418
18	18	45	7	6	23884.85	2782.782	9	14325.53	401.172	35.709	542.494	20629.66	4461.13	7	19685.27	320.444	61.431	509.9744
19	19	50	10	8	32989.07	5557.542	7	26257.94	444.985	59.009	898.12	30114.56	3670.121	9	29772.16	486.676	61.175	924.2343
20	20	50	10	8	27922.66	6054.48	8	25849.68	452.511	57.125	892.49	33053.66	3542.657	9	30480.27	502.04	60.713	818.4159
21	21	55	10	8	5568.848	2208.034	4	41775.08	143.636	290.84	930.882	43354.96	4456.992	9	32404.62	547.967	59.136	908.8013
22	22	55	10	8	6156.592	1758.699	5	45383.26	167.468	270.997	904.15	41871.04	4045.067	9	32737.52	546.638	59.889	935.1524
23	23	60	10	8	16358.26	5349.714	5	45568.07	278.354	163.705	953.105	54151.43	3401.274	9	37015.64	611.685	60.514	1053.8692
24	24	60	10	8	12126.36	236.107	5	55663.32	244.522	227.642	930.913	46855.64	3633.349	9	40024.92	615.921	64.984	1150.0092
25	25	65	12	10	28692.01	11946.41	5	54349.57	374.738	145.034	1321.394	55499.57	5163.685	8	45162.26	621.384	72.68	1524.5642
26	26	65	12	10	10545.66	707.695	6	66254.3	238.544	277.745	1345.418	67265.13	5188.03	8	49647.06	667.478	74.38	1425.3859
27	27	70	12	10	23156.23	5036.398	6	69722.33	338.237	206.135	1451.531	75571.22	5639.474	8	48648.91	673.91	72.189	1510.2375
28	28	80	12	12	1820.76	369.513	5	72237.39	88.983	811.811	1580.876	75363.69	8369.203	8	58224.5	728.251	79.951	1585.6085

29	29	90	12	12	30784.28	10882.75	6	96988.4	418.914	231.523	1951.368	105796.9	11444.47	8	61739.25	802.284	76.954	1858.5288
30	30	100	12	12	29537.42	10705.29	6	122572.2	413.638	296.327	1991.304	123957.2	9766.726	9	89325.53	1000.101	89.317	2155.4818

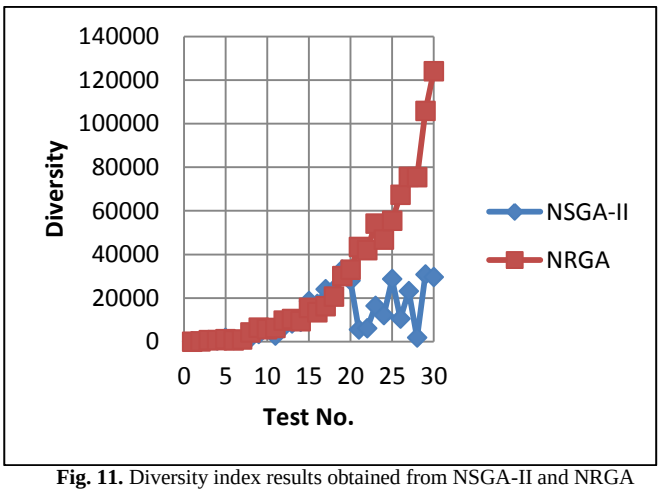


Fig. 11. Diversity index results obtained from NSGA-II and NRGA

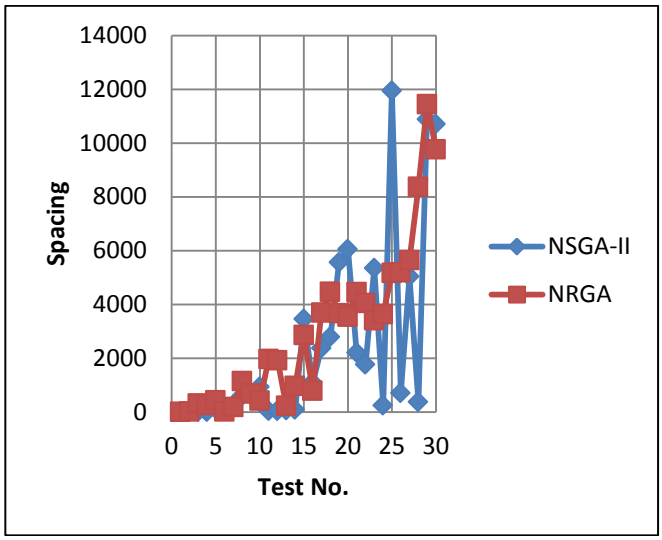


Fig. 12. Spacing index results obtained from NSGA-II and NRGA

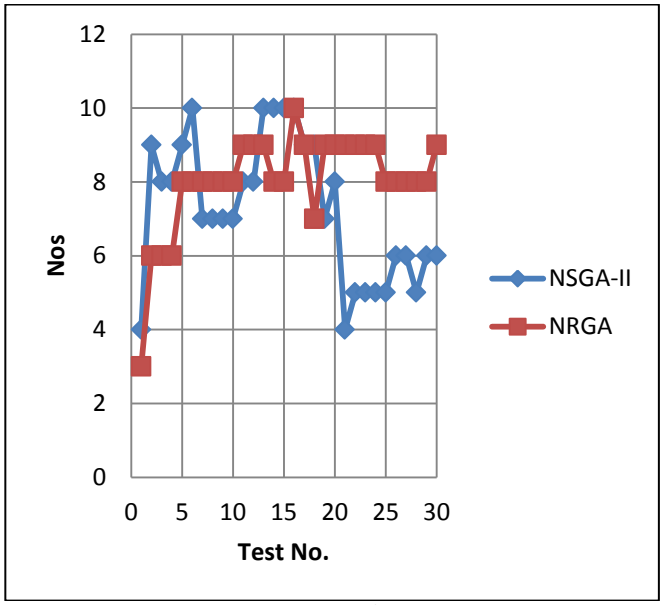


Fig. 13. Nos index results obtained from NSGA-II and NRGA

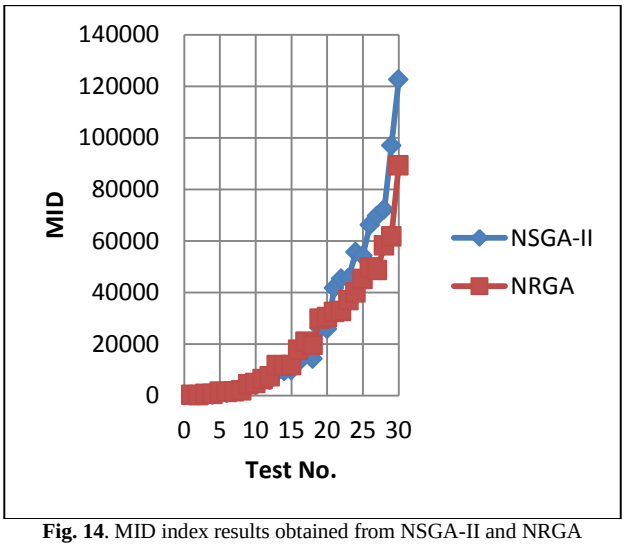


Fig. 14. MID index results obtained from NSGA-II and NRGA

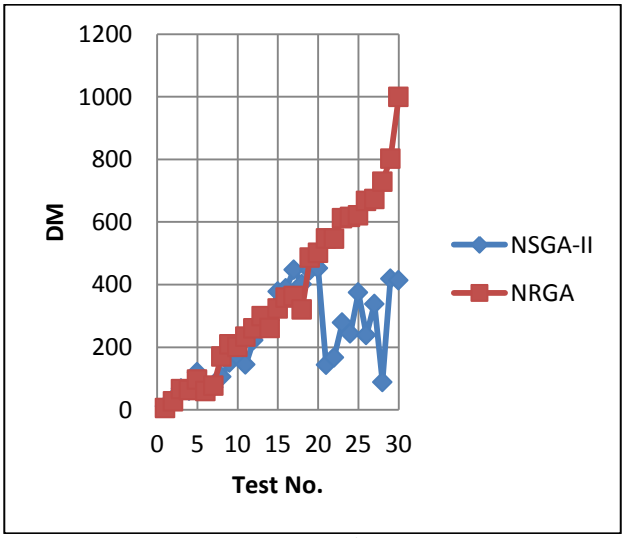


Fig. 15. DM index results obtained from NSGA-II and NRGA

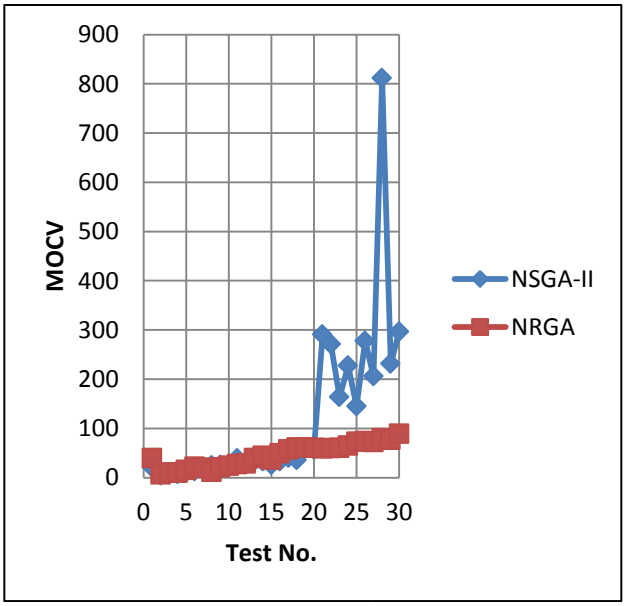


Fig. 16. MOCV index results obtained from NSGA-II and NRGA

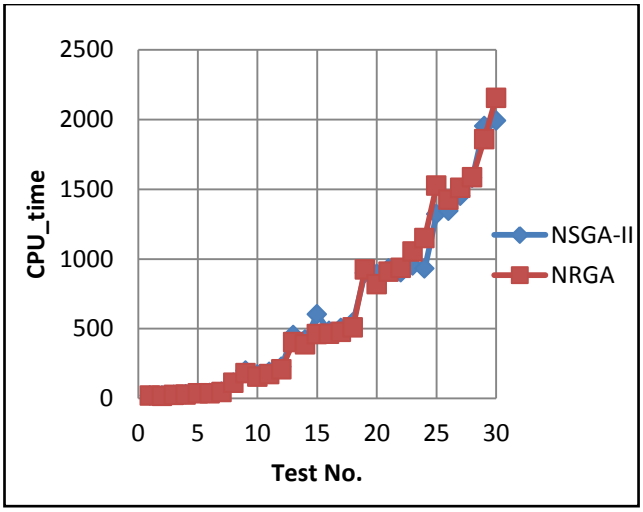


Fig. 17. CPU-time index results obtained from NSGA-II and NRGA

Table 7. Results of variance analysis for Diversity index

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Factor	1	4558768376	4558768376	7.36	0.009
Error	58	35903371899	619023653		
Total	59	40462140275			

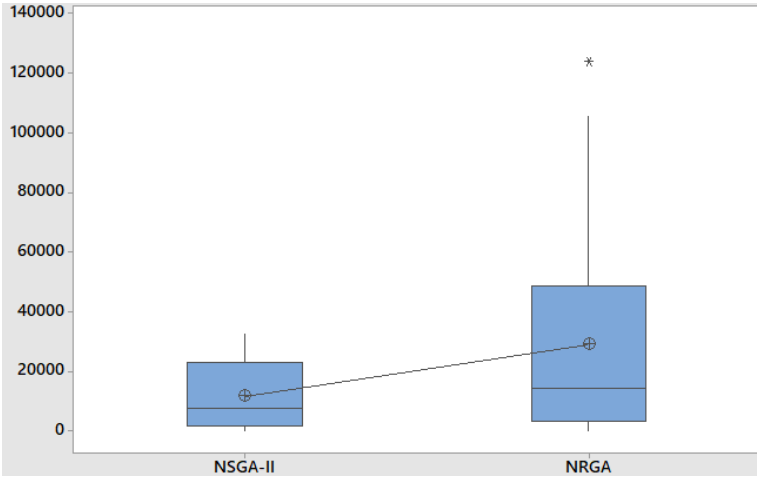


Fig. 18. Diversity diagram for NSGA-II and NRG algorithms

Table 8. Results of variance analysis for Spacing index

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Factor	1	3847189	3847189	0.36	0.549
Error	58	614218891	10589981		
Total	59	618066080			

Table 9. Results of variance analysis for Nos index

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Factor	1	8.817	8.817	3.22	0.078
Error	58	158.833	2.739		
Total	59	167.650			

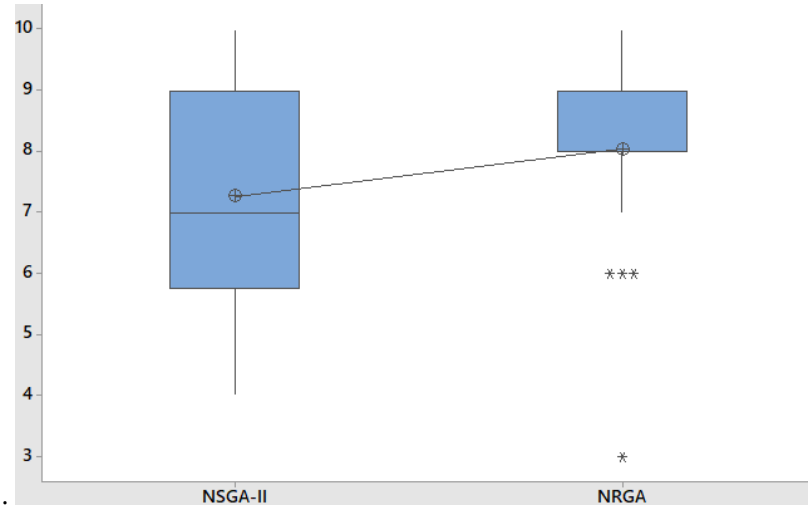


Fig. 19. Nos diagram for NSGA-II and NRG algorithms

Table 10. Results of variance analysis for MID index

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Factor	1	347327537	347327537	0.44	0.510
Error	58	45895491585	791301579		
Total	59	46242819122			

Table 11. Results of variation analysis for DM index

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Factor	1	291469	291469	6.50	0.013
Error	58	2600954	44844		
Total	59	2892423			

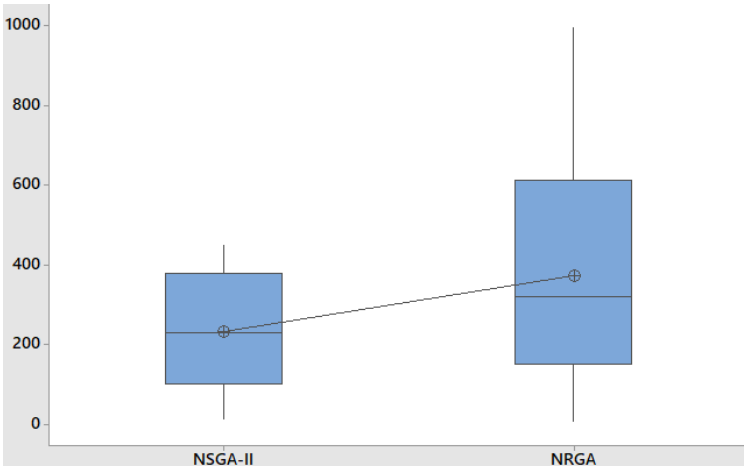


Fig. 20. DM diagram for NSGA-II and NRG algorithms

Table 12. Results of variation analysis for MOCV index

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Factor	1	74067	74067	5.28	0.025
Error	58	813888	14033		
Total	59	887955			

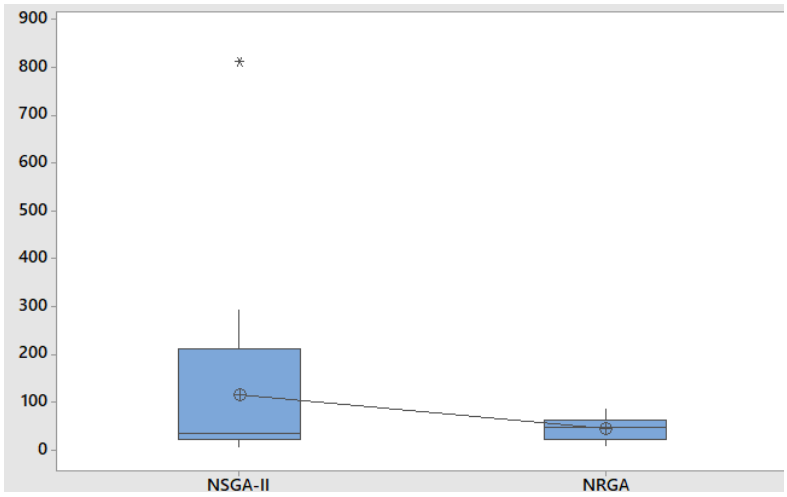


Fig. 21. MOCV diagram for NSGA-II and NRG algorithms

Table 13. Results of variation analysis for CPU-time index

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Factor	1	1626	1626	0.00	0.948
Error	58	21802258	375901		
Total	59	21803883			