Reducing noise pollution by flexible job-shop scheduling with worker flexibility: Multi-subpopulation evolutionary algorithm

M. Hajibabaei, J. Behnamian*

Department of Industrial Engineering, Faculty of Engineering, Bu-Ali Sina University, Hamedan, Iran
*Corresponding author, E-mail: Behnamian@basu.ac.ir; Tel.: (+9881) 38292505; Fax: (+9881) 38292631

Abstract: Flexible job-shop scheduling is one of the most critical production management topics. In this paper, it is also assumed job interruption due to the machine breakdown is allowed, and the processing time depends on the speed of the machines and requires both human and machine resources to process the jobs. Although, as the speed of the machine increases, the time of job' completion reduces, an increase in speed results in an increase in noise pollution in the production environment, and with the aim of applying a cleaner production that is a preventative approach, it has been tried to reduce noise pollution by minimizing the increase in speed. After modeling the problem using the mixed-integer programming and solving it using the ε-constraint method, since the problem is NP-hard, a multi-subpopulation evolutionary algorithm is proposed to solve it. The results showed that considering the mean ideal distance criterion, the ε-constraint method has a better performance than the proposed algorithm but considering other criteria the proposed algorithm has is better. Also, the proposed algorithm was compared with the NSGAII in large-size instances and the computational results showed that the proposed algorithm performs better than the NSGAII in most cases.

Keywords: Flexible job-shop scheduling, Sustainability, Machine breakdown, Worker flexibility, Sub-population meta-heuristic

1. Introduction

The industry of each country is one of the major sources of employment and income for that country and is essential for the production of goods and services [1]. Hence, industrial development is a necessity for economic growth. At the same time, the industry sector is a major consumer of resources and materials, and industrial activities impose environmental pollution. In the past, productive activities have been much related to the profit of the production unit, and environmental issues have not been considered vastly. Given the importance of the environmental issue, it is important to use a preventive approach to reduce the environmental impacts of manufacturing plants. Therefore, the cleaner production concept was born in 1997. In this approach, the focus is on preventing contamination in the production unit [2].

Cleaner production involves the continued application of a comprehensive environmental strategy for the process of products and services in order to increase overall efficiency and reduce harmful effects on humans and the environment. It is a strategy to make the changes needed in existing technology and industry to build a sustainable development society. The concept of cleaner production has been further developed with an incentive to conserve the environment [3]. From an environmental point of view, minimizing the amount of noise generated by the production unit is one of the criteria for cleaner production [4]. Noise is the kind of unpleasant sound that interrupts a person's efficiency in some situations. Noise, like air pollution, has been introduced as a new threat to humans. It has also been proved that a certain level of noise can be harmful to human hearing [5]. The sound level is measured in units of decibels (dB), and the human ear is capable of hearing 1 to 140 decibels (dB).

Scheduling is a decision process used in many service and manufacturing industries. The scheduling problem deals with allocating resources to activities over time, and its purpose is to optimize one or more goals. Goals can also take many forms. One of the key success factors in any manufacturing organization is to determine the scheduling and sequence of jobs on production scheduling problems that play an important and effective role in the performance of the manufacturing unit. The job-shop scheduling problem is one of the most important topics in

production management, which is a branch of production scheduling and is one of the most complex hybrid optimization topics [6] .
In this paper, it is assumed there are *m* machines and *n* jobs that each job follows its predetermined path in the flexible job-shop environment. A flexible job-shop problem is a generalization of the job-shop problem and the problem of parallel machines. This problem has several stages, each containing several parallel machines. In this workshop, every operation of any job can be performed on a machine from a set of machines available for processing [7]. Therefore, in the case of the flexible job-shop problem, in addition to sequencing jobs on machines, jobs are assigned to machines. The problem addressed in this paper is an extension of [8] in which in a flexible job-shop environment with sequence-dependent setup time, the makespan is minimized. In the case under study, many assumptions have been made to make the situation more realistic, some of which are: (*i*) the interruption of job is allowed, (*ii*) the machines are not constantly available, (*iii*) the speeds of the machines are also different, and (*iv*) the noise pollution are taken into account in this study. In this research, it is assumed, as the speed of the machines increases, the time to complete jobs is reduced, but the noise pollution in the production environment is increased. After modeling the problem as mixed inter programming, for simultaneous minimizing of the total completion times and the sum of the speed increments, and due to the NP-hardness of this problem, a sub-population algorithm is proposed.
This paper is presented in six sections. In this section, the introduction was discussed. Section 2 reviews the literature on the subject. Section 3 deals with problem definition and its modeling. Section 4 introduces the multi-objective meta-heuristic method, which is used. Section 5 presents computational results, and Section 6 devotes to the conclusion and future research.

2. Literature Review

This section provides an overview of research on the issues of job-shop scheduling and flexible job-shop scheduling. The papers studied are categorized into cleaner production scheduling and environmental impacts.
Cleaner production. de Oliveira Neto et al. [9] introduced a cleaner approach to minimizing the environmental impacts that are imposed on the environment by production in industrial plants. They also compared the features of the end-of-line control approach and cleaner production. Ultimately, they concluded that cleaner production would be effective in the production units. Rajaram et al. [10] presented a multi-objective model including minimizing environmental impacts and maximizing economic aspects of cleaner production and used a two-objective genetic algorithm to solve it. Mokhtari et al. [4] presented a multi-objective model including minimization of production costs, transportation, and environmental impacts, including CO_2 emissions, waste generation, noise generation, and occupational injury in transportation scheduling at manufacturing plants. Then, they were solved the presented model using ideal programming.
Cleaner production and environmental impacts. Zarrouk et al. [11] minimized the makespan in the multi-process job-shop scheduling by considering maintenance in cleaner production and solved the problem using ant colony algorithms and particle swarm optimization. Amjad et al. [12] proposed a mathematical model in the case of flexible job-shop scheduling with stochastic processing time. They then used the multi-objective genetic algorithm to minimize completion time and energy consumption. They also compared the proposed algorithm with hybrid particle swarm optimization algorithms and hybrid simulated annealing optimization algorithm. Dai et al. [13] proposed a multi-objective nonlinear programming model for energy efficiency in flexible job-shop environments considering transportation constraints. In the proposed model, the goals of minimizing energy consumption and maximizing completion time were considered. An improved genetic algorithm was used to solve the problem, and to solve the proposed algorithm, and then it was solved in several problem instances. Maine et al. (2019) proposed a

multi-objective non-linear programming model for energy efficiency in flexible job-shop scheduling problem with consideration of transportation constraints. To minimize the energy consumption and makespan, they used an improved genetic algorithm to solve the problem. Abedi et al. [14] considered the job-shop scheduling problem with machines' breakdown. They proposed a multi-objective memetic algorithm to minimize the sum of the weighted delay and the total energy consumption.

Scheduling with makespan. Li & Gao [15] proposed a mathematical model and a meta-heuristic tabu search algorithm for the problem of job-shop scheduling with parallel machines. The purpose of this study was to minimize the makespan. The proposed algorithm was also compared with the combined genetic and ant colony algorithm. Kundakcı and Kulak [16] proposed a mathematical model and genetic algorithm to minimize the makespan in the dynamic job-shop scheduling problem and solved it in several problems to investigate the performance of the proposed algorithm.

AitZai et al. [17] proposed a mixed-integer model for the job-shop scheduling problem to minimize the makespan. Then, a detailed branch and bound method using valid equations were presented and solved in different numerical examples to investigate the efficiency of the proposed method. Wu et al. [18] proposed an ant colony optimization algorithm for a flexible job-shop scheduling problem with the aim of minimizing the makespan. They also solved the proposed algorithm in several problems to evaluate the performance of the proposed algorithm.

To minimize the makespan, Wang et al. [19] proposed an ant colony optimization algorithm for the flexible job-shop scheduling problem. Jamrus et al. [20] presented a discrete particle swarm optimization algorithm for scheduling a flexible job-shop production with parallel machines whose aim was to minimize the makespan and to use innovative methods to evaluate the performance of the proposed algorithm. In the flexible job-shop scheduling problem with flexible human resources, Gong et al. [21] proposed an integer non-linear programming model and a memetic algorithm to minimize the makespan, maximum machine workload, and total machine workload. Shen et al. [8] solved the problem of flexible job-shop scheduling by considering sequence-dependent setup times with the objective of minimizing the makespan using the discrete graph method and the tabu search algorithm. Peng et al. [22] presented a mathematical model and genetic algorithm to minimize the makespan for the flexible job-shop scheduling problem with dual human-machine resources. Tamssaouet et al. [23] presented tabu search and simulated annealing to minimize the makespan in job-shop scheduling problem with limited machine access. They also used a graph theory-based method to solve small-size instances. Yazdani et al. [7] employed flexible job-shop scheduling with consideration of human and machine resources to minimize the makespan. They also proposed a mathematical model and a hybrid algorithm of neighborhood search and simulated annealing. Also, the results of the presented hybrid algorithm were compared with the results of other algorithms. Gong et al. [24] minimized the makespan and the maximum total delay time in the job-shop scheduling problem by a memetic algorithm. Zhang et al. [25] proposed an improved genetic algorithm for the flexible job-shop scheduling problem, to minimize the makespan, total start-up time, and total shipping time. To minimize the makespan in the flexible job-shop scheduling problem, Ding and Gu [26] proposed a mathematical model and an improved particle swarm optimization algorithm. Yang et al. [27] proposed a multi-objective non-linear mixed-integer model and NSGAII for a flexible job-shop scheduling problem with the objectives of the makespan and energy consumption.

Scheduling with ET objective function. Heydari & Aazami [28] presented the maximum tardiness and makespan for a job-shop scheduling problem with sequence-dependent setup times. The ε-constraint method solved the presented problem. Gao et al. [29] proposed a harmonic search algorithm for a flexible job-shop scheduling problem with multiple objectives. The objectives are the weighted combination of two minimization criteria, namely, the maximum of the completion time and the mean of earliness and tardiness. They also solved the problem in several instances to evaluate the performance of the proposed algorithm. Ebrahimi et al. [30] presented a mixed linear programming model and a cluster-based algorithm for the flexible job-shop problem that aims to minimize the makespan and tardiness time. Yazdani et al. [31] proposed a mixed-integer programming model and an approximate optimization method based on imperialist competitive algorithm and neighborhood search to minimize the tardiness and earliness in the job-shop scheduling problem. Yu and Lee [32] proposed a mixed-integer programming model and two other methods to minimize the tardiness of job-shop with group processing. They also solved the problem in small size using the branch and bound method. Mendoza et al. [33] proposed a linear programming model to minimize the tardiness in the job-shop problem and solved it in numerical examples to investigate the efficiency of the proposed model. Sadaghiani et al. [34] developed a multi-objective mixed-integer programming model for the flexible job-shop problem that minimized the makespan, total workload, and maximum workload of machines. They also proposed an algorithm based on Pareto and NSGAII methods for large-scale problem-solving. Then, in order to evaluate the efficiency and effectiveness of the proposed algorithm, they compared it with other algorithms. Dalfard et al. [35] proposed a mixed-integer nonlinear programming model for the problem of flexible job-shop scheduling considering maintenance constraints. To solve the large-scale problem, they presented a combination of a genetic algorithm and an innovative algorithm and compared it with another algorithm to evaluate the performance of the proposed method. Huang et al. [36] proposed a modified particle swarm optimization algorithm for a flexible multi-objective job-shop scheduling problem. The objectives of this study were to minimize the makespan, minimize the machine load, and minimize the maximum machine load. Table (1) shows the difference between the present study and previous researches. According to the studied papers, flexible job-shop scheduling, with the objective of minimizing the total completion time and the sum of the speed increments (to reduce the noise pollution) in the manufacturing environment, machine breakdown and interruption of the job have not been considered simultaneously. Therefore, in this paper, these assumptions are studied simultaneously in a cleaner production environment with the aim of approaching the conditions of the problem under study to the real world.

******** Insert Table 1 here ********

3. Problem definition

In the case of flexible job-shop scheduling, each job i consists of operation op_i, which o_{ij} represents the set of operations performed for each job. This operation is performed on a set of machines that can perform the above operations on the machines and each job is assigned a machine. In this problem, each job requires both human and machine resources for processing, and each operation is processed by a set of ms_{ijm}, which includes machines capable of performing operation o_{ij} to produce m part. The interruption is due to the machine breakdown permitted and the duration of the interruption is predetermined. The speed of the machines varies, and each machine generates less noise while working at a lower speed. Due to the importance of reducing the time to complete the parts by increasing the speed of the machines, this goal can be achieved. But as the speed of the machines increases, the amount of noise generated in the industrial unit increases. On the other hand, it endangers the physical and mental health of workers, and the physical and mental problems of workers reduce the efficiency of production. Therefore, to remedy this problem, the rate of acceleration in production is determined by the amount of noise that people have the ability to hear. The amount of noise generated in the production unit is measured by dB. The maximum noise that workers can handle is 140 dB. Therefore, the speed of the machines can be increased so that the noise level does not exceed the maximum value.

After each job is completed, a corresponding part is produced. The purpose of this paper is to minimize the sum of completion times and reduce the rate of acceleration to reduce noise production and less injury to workers. Following Aurich et al. [37], the sound intensity is calculated using Equation (1).

$$94 - 10\,Log T = Lpa \tag{1}$$

where T is the time of exposure to sound in hours and Lpa is the sound pressure level allowed for exposure time, dB.

The following assumptions are presented in this study:

Jobs are available at zero time.

Machines are available at zero time.

Interruption is allowed.

Once the job is being processed, the job continues as it has been interrupted after the repairs are completed.

Workers are available at zero time and cannot leave the machine during the processing of an operation.

A machine can only perform one operation at a time.

Every process needs both machine and worker resources.

All parameters are definite.

As the speed of the machines increases, the amount of noise created increases.

3.1. Mathematical model

In this section, a nonlinear mixed-integer programming model is formulated to formulate a flexible job-shop scheduling problem with job interruption. Before presenting the proposed model, indexes, parameters, and decision variables are introduced.

Indexes:

i, i' :	Job index	$i = 1, \dots, n \;, i' = 1, \dots, n'$
j, j' :	Operation index	$j = 1, \dots, e \;, j' = 1, \dots, e'$
k :	Machine index	$k = 1, \dots, K$
l :	Worker index	$l = 1, \dots, L$
m, m' :	Part index	$m = 1, \dots, M \;, m' = 1, \dots, M'$

Parameters:

p'_{ijml} :	Standard processing time of operation o_{ij} for part *m* by worker *l*
H :	A large positive number
v_k :	The minimum speed of machine *k*
tt_k :	Repair time of machine *k*
mt_k :	Number of times that machine *k* needs repair
ff_k :	The maximum speed of machine *k*
fI_k :	The minimum speed of machine *k*

Decision variables:

a_{ijmk} :	A binary variable that takes a value of 1 if the operation o_{ij} for part *m* is performed by the machine *k*
ap_{lk} :	The binary variable that takes the value of 1 if the worker l is assigned to the machine *k*

pm_{ijmk} : A binary variable that takes a value of 1 if operation o_{ij} for part m be interrupted by machine k

$ba_{ijmi'j'm'}$: A binary variable that takes a value of 1 if the operation o_{ij} for part m is performed before operation $o_{i'j'}$ for part m'

t'_{ijm} : Start time of operation o_{ij} for part m

cc_{ijm} : Completion time o_{ij} for part m

c'_m : Completion time of part m

pt_{ijmkl} : processing time of operation o_{ij} on machine k by worker l

bb_{ijmk} : Increase speed of machine k for processing o_{ij} operation for part m

The proposed mathematical model with respect to the symbols defined is as follows.

$$\min \sum_{i=1}^{n}\sum_{j=1}^{e}\sum_{m=1}^{M}\sum_{k\in ms_{ijm}} bb_{ijmk}$$

$$min \sum_{m=1}^{M} c'_m$$

$$s.t.: \sum_{k\in ms_{ijm}} a_{ijmk} = 1 \qquad \forall i, j, m$$

$$t'_{ijm} \geq t'_{i(j-1)m} + \sum_{k\in ms_{ijm}}\sum_{l=1}^{L} a_{i(j-1)mk} . pt_{i(j-1)mkl} + \sum_{k\in ms_{ijm}} pm_{i(j-1)mk} .tt_k \qquad \forall j, i, m$$

$$t'_{ijm} \geq t'_{i'j'm'} + \sum_{l=1}^{L} a_{i'j'm'k} . pt_{i'j'm'kl} - \mathrm{H}\left(2 - a_{ijmk} - a_{i'j'm'k} + ba_{ijmi'j'm'}\right) + pm_{i'j'm'k} .tt_k$$
$$\forall\ j\ , j', i, i', m, m \neq m', k \in ms_{ijm}$$

$$t'_{i'j'm'} \geq t'_{ijm} + \sum_{l=1}^{L} a_{ijmk} . pt_{ijmkl} - \mathrm{H}\left(3 - a_{ijmk} - a_{i'j'm'k} - ba_{ijmi'j'm'}\right) + pm_{ijmk} .tt_k$$
$$\forall\ j, j', i, i', m, m \neq m'\ , k \in ms_{ijm}$$

$$cc_{ijm} \geq t'_{ijm} + \sum_{k\in ms_{ijm}}\sum_{l=1}^{L} a_{ijmk} . pt_{ijmkl} + \sum_{k\in ms_{ijm}} pm_{ijmk} .tt_k \qquad \forall j, i, m$$

$$c'_m = cc_{ijm} \qquad \forall m\ , i = m\ , j = e$$

$$pm_{ijmk} \leq a_{ijmk} \qquad \forall j, i, m, k \in ms_{ijm}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{e}\sum_{m=1}^{M} pm_{ijmk} \leq mt_k \qquad \forall\ k \in ms_{ijm}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{e}\sum_{m=1}^{M} pm_{ijmk} \geq 1 \qquad \forall\ k \in ms_{ijm}$$

$$pt_{ijmkl} = \frac{ap_{lk} . p'_{ijml}}{bb_{ijmk} + v_k} \qquad \forall\, j, i, m\ , , l\ \ k \in ms_{ijm}$$

$$\sum_{l=1}^{L} ap_{lk} = 1 \qquad \forall\, k$$

$$\sum_{k=1}^{K} ap_{lk} = 1 \qquad \forall\ l$$

$$\log\left(pt_{ijmkl} \right) \le 1.4 - 6\left(\frac{v_k - fI_k}{ff_k - fI_k} \right)$$

$$bb_{ijmk} \le ff_k - v_k \qquad \forall\ j\ , i, m\ , k \in ms_{ijm}$$

$$\frac{ap_{lk} . p'_{ijml}}{ff_k} \le pt_{ijmkl} \le \frac{ap_{lk} . p'_{ijml}}{v_k} \qquad \forall\ \ i, j, m\ , l\ \ \ k \in ms_{ijm}$$

$$a_{ijmk} \in \left(0,1\right) \quad , \quad pm_{ijmk} \in \left(0,1\right) \qquad \forall\ \ i, j, m\ , k \in ms_{ijm}$$

$$ba_{ijmi'j'm'} \in \left(0,1\right) \qquad \forall\ \ i, j, m, i', j', m'$$

$$t'_{ijm} \ge 0\ \ ,\ \ cc_{ijm} \ge 0 \quad c'_m \ge 0 \qquad \forall i, j, m$$

$$pt_{ijmkl} \ge 0 \qquad \forall\ \ i, j, m, l, k \in ms_{ijm}$$

$$ap_{lk} \in \left(0,1\right) \qquad \forall l, k$$

Equation (1) as the first objective function minimizes the sum of the speed increments. Equation (2) as the second objective function minimizes the total completion time. Equations (3) to (17) indicate the constraints. Equation (3) guarantees that each operation for each part is assigned to one and only one of its eligible machines. Equation (4) guarantees preconditioning relationships between successive operations of the same job for a part. Equations (5) and (6) indicate the time relationship between two operations of two different parts if two operations are performed by a machine. Equation (7) indicates when each operation of each part will be completed. Equation (8) indicates the time of completion of each part. Equation (9) indicates that the processing of each operation of each part on each machine is interrupted when that operation is assigned to that machine. Constraints (10) and (11) indicate the minimum and the maximum number of interruptions of each machine. Equation (12) indicates the processing time of each operation of each part with respect to the speed of each machine. Equation (13) indicates that a worker is assigned to each machine. Equation (14) indicates that each worker is assigned a machine. Equation (15) shows the maximum amount of sound intensity in the workstation. Equation (16) indicates the maximum speed value of each machine. Equation (17) indicates the upper and lower limits of processing time. Equations (18) - (22) indicate the status of the variables.

3.2. *ε*-constraint method

The *ε*-constraint method (known as the *ε*-constraint method) is one of the multi-objective problem-solving methods. In this method, except for one objective function, the rest of the objective functions will be appeared as a constrained upper bound in the minimization problem. In multi-objective problems, the Pareto layer is created by applying parametric changes to the right-hand side of this constraint. In this regard, it is assumed that there is a mathematical model *k* objective function as follows:

$$\min z_k = f_k(x_1, x_2, \ldots, x_n) \qquad k = 1, \ldots, p \tag{23}$$
s.t:
$$g_j(x_1, x_2, \ldots, x_n) \le b_j \qquad j = 1, \ldots, m$$

To solve this model, one of the objective functions is minimized by the ε-constraint method as a single-objective problem, while the other objective functions are added to the constraints with upper bounds. The following model shows this method, schematically.

$$\min z_1 = f_1(x_1, x_2, \ldots, x_n) \tag{24}$$
s.t:
$$f_2(X) \ge \varepsilon_2$$
$$f_3(X) \ge \varepsilon_3$$
$$\vdots$$
$$f_p(X) \ge \varepsilon_p$$
$$g_j(x_1, x_2, \ldots, x_n) \le b_j \qquad j = 1, \ldots, m$$

In this study, as the main objective function, the rate of increase in speed is minimized, and the other one is considered as a constraint.

4. Subpopulation genetic algorithm

The main goal of multi-objective decision-making is to achieve the set of Pareto solutions. In this study, the multi-objective method of the multi-objective genetic algorithm was used. In multi-objective optimization, failure to achieve solutions with appropriate diversity indicates an immature process in the evolutionary algorithm. In this algorithm, in order to generate diverse solutions, by dividing the initial population into several sub-populations, the weighted method is used [38].

******** Insert Figure 1 here ********

In this study, the subpopulation genetic algorithm (SPGA) is used. In this algorithm, the initial population is subdivided into several subpopulations to generate effective responses, and each subpopulation is weighted. For each subpopulation, the genetic algorithm is fully implemented, and Pareto solutions are generated [39]. Figure (1) shows the Pareto algorithm solutions.

4.1. Initial solution representation

The random key method is used for initial solution representation. In a flexible job-shop environment, there are a number of parallel machines at each stage capable of performing the same operation. Each job has a predetermined path for processing. For example, Figure (2) shows the processing path of each job. This is a matrix of three rows and four columns whose number is equal to the number of stages and the number of columns to the number of jobs. S1, S2, and S3 represent the first, second, and third stages, respectively. As this figure shows, the first job is first processed in the second stage, then in the first stage, and does not need to be processed in the third stage. The asterisk shows that one job does not need to be processed at that stage. The second job is processed first in the first stage, then in the third stage, and finally in the second stage. The third job is processed first in the second stage, then in the first stage and finally in the third stage. The fourth job is processed first in the first stage, then in the second stage, and finally in the third stage. The fifth job is first processed in the third stage, then in the first stage, and finally in the second stage, and finally, the sixth job, first in the third stage, then in the second stage, and finally in the first stage. The first, second, and third stages consist of two, two, and three machines, respectively. The solution is a matrix with n rows and h columns. n is the number of stages and h is the number of jobs. According to the matrix, for each row where the first operation of each job is processed at that stage, a random number (1, m + 1) is generated at random. m is the number of machines in each stage. For example, the representation of the solution for Figure (2) is shown in Figure (3). As Figure (3) shows, for the second and fourth columns in the first row, two random numbers are generated in the interval (3 1). The number 1.25 indicates that the first operation of the second job at this stage is performed by the first machine. The number 2.75 indicates that the first operation of the fourth job is performed by the second machine at this stage. In the second row, the numbers 1.05 and 1.15 indicate that the first operation of the first and third jobs is processed in the second stage by the first machine. Because the decimal part of the number in the second row and the third column is smaller than the number in the second row and the first column, so first the first operation of the third job and then the first operation of the first job is done on the first machine. Finally, in the third row, the first operation of the sixth job on the first machine and the first operation of the fifth job on the second machine are done.

******** Insert Figure 2 here ********

******** Insert Figure 3 here ********

4.2. Crossover operator

In this paper, as shown in Figure (4), a single-point method is used to create a crossover. This is where a point is randomly assigned along the two chromosomes selected as parents, and the chromosomes are split from that point into two portions. They are replaced with one another and result in the production of children.

******** Insert Figure 4 here ********

4.3. Mutation operator

Mutation operators are random-shift operators in which one or more cells of a specific chromosome are taken into account, and values in those chromosomes change. In this study, as shown in Figure (5), two cells are randomly selected, and the values within those cells are exchanged.

******** Insert Figure 5 here ********

4.4. Selection operator

After the new population is created using the crossover and mutation operators, it merges with the new population, and the best of the original population is selected.

5. Computational results

To investigate the efficiency of the developed model, using 75 test problems, the performance of the proposed algorithm is evaluated by the ε-constraint method in small-size instances and by the

NSGAII in Ahmadi et al. [40] in large-size instances. The algorithms are coded by MATLAB and implemented on a Win 7 (64Bit) with 16GB RAM.

5.1. Parameter setting

Following Cuiyu et al. [41] and Ehtesham Rasi [42], in the present study, an experimental method was used to determine the parameter. In this regard, after solving several test problems in different sizes and different rates, the crossover and mutation rates were set to 0.8 and 0.2, respectively, and the number of replicates was set to 50. Weights were also assigned to each subpopulation, from zero to one with a distance of "0.1".

5.2. Evaluation metric

There are many criteria to evaluate the performance of multi-objective algorithms, each with its own advantages and disadvantages. Some of these criteria only take into account the number of Pareto solutions, whereas, in the problem-solving environment, the quality of the solutions is usually the most important. However, given that there is no single solution as an optimal solution in multi-objective space, another criterion is called the diversity of solutions is considered in this paper.
In this study, we used three performance measurements that cover both the quality of the solution and the variety of solutions. These criteria are:
Mean ideal distance (MID)
The rate of achievement to two objectives simultaneously (RAS)
The spread of non-dominance solution (SNS)

Equation (38) is used to calculate the first criterion. In this respect, n is the number of vectors in the Pareto layer, and c_i is the Euclidean distance between each member of the set of coordinates, obtained from the equation $\sqrt{f_{1i}^{\;2} + f_{2i}^{\;2} + \ldots + f_{ki}^{\;2}}$. In this respect, f_{ki} is the k value of the objective function in the Pareto i vector solution vector. Obviously, the lower the value, the greater the utility of that set.

$$MID = \frac{\sum_{i=1}^{n} c_i}{n} \tag{25}$$

In relation to the second criterion, if the solution is along one axis because it is only fit for one objective and does not perform well for the other, it is less desirable, but where we have achieved an acceptable balance between objectives. Equation (26) represents the second criterion where $F_i = \min\{f_{1i}, f_{2i}\}$.

$$RAS = \frac{\sum_{i=1}^{n}\left[\left(\frac{f_{1i} - F_i}{F_i}\right) + \left(\frac{f_{2i} - F_i}{F_i}\right)\right]}{n} \tag{26}$$

Finally, Equation (27) is used to calculate the last criterion.

$$SNS = \sqrt{\frac{\sum_{i=1}^{n}(MID - c_i)^2}{n-1}} \tag{27}$$

Note that the lower values of MID and RAS criteria are better and the higher value of SNS is better.

5.3. Computational results in small-size instances

Table (2) shows the results obtained by comparing the SPGA and the ε-constraint method in the small-size instances. Then, these results are analyzed considering controlled factors, algorithms and used evaluation metrics in Figure (6). This figure shows that in all cases, the ε-constraint method with respect to the MID has better performance than the SPGA. Also, Figure (6) shows that considering the RAS and SNS, the SPGA has a better performance.

******** Insert Table 2 here ********

******** Insert Figure 6 here ********

5.4. Computational results in large-size instances

Table (3) shows the results of 45 randomly generated large-size instances in which the SPGA and NSGAII are compared. As the table shows, considering the two criteria of MID and RAS, there is a relative superiority with the proposed algorithm, but in the SNS criterion, which is the criterion for the diversity of the generated solutions, it is the SPGA that has absolute superiority.

******** Insert Table 3 here ********

The results of Table (3) are analyzed according to controlled factors, algorithms and used evaluation metrics in Figure (7). Figure (7a) shows that in all cases, the NSGAII, with respect to the MID, performs better than the proposed algorithm. Considering the RAS, Figure (7b) shows that the performance of the proposed algorithm and NSGAII are similar in small-size instances but in large test problems, e.g., in 15 and 50, the SPGA performs better than the NSGAII.

******** Insert Figure 7 here ********

Figure (7c) shows that according to SNS, the proposed algorithm, in most cases, performs better than the NSGAII. To further examine the results, statistical analysis has been performed, the results of which are presented in Tables (4-6).

******** Insert Table 4 here ********

******** Insert Table 5 here ********

******** Insert Table 6 here ********

The results of the Tukey test in small size show that in all criteria, the difference between the ε-constraint method and the proposed algorithm and the difference between the ε-constraint method and NSGAII are significant. But the difference between the NSGAII and SPGA is not significant. Figure (8) shows the mean and error bar of the algorithms according to the three criteria.

******** Insert Figure 8 here ********

Furthermore, as the results of the Tukey test in large size shown in Tables (7-9), there is no significant difference between SPGA and NSGAII in MID, SNS and RAS criteria. Figure 9 shows the mean and error bar of the proposed algorithm compared to NSGAII in three criteria of MID, RAS and SNS.

******** Insert Table 7 here ********

******** Insert Table 8 here ********

******** Insert Table 9 here ********

******** Insert Figure 9 here ********

6. Conclusion and future research

The main purpose of this research is to determine the sequence of operations on machines, reducing the amount of completion time and noise generated by increasing speed. In this problem, jobs are manufactured in a flexible job-shop environment where job interruption is permitted. The purpose of this study is to minimize the sum of part completion times and the sum of the speed increases using the ε-constraint method. To compare the algorithm, three criteria were used. Due to the bi-objective problem and its NP-hardness, the subpopulation genetic algorithm (SPGA) was used to solve it. The proposed algorithm was compared with the ε-constraint method and the NSGAII in small and large-size instances, respectively. In the small-size instances, the ε-constraint method and the proposed algorithm were compared. The results showed that in the mean ideal distance (MID) criterion the ε-constraint method was better compared to the proposed algorithm, but considering the rate of achievement to two objectives simultaneously (RAS) and spread of non-dominance solution (SNS) criteria the proposed algorithm had a better performance than the ε-constraint method. The proposed algorithm was also compared with the NSGAII in large-size instances. The results showed that considering the MID criterion, the NSGAII was somewhat better than the SPGA, in the RAS criterion, the proposed algorithm was better in more instances (especially in larger size), but in the SAS criterion, complete superiority was with the SPGA. Considering MID, SNS and RAS criteria, the results of the Tukey test in small dimensions showed that the difference between the ε-constraint method with the SPGA and NSGAII were significant. But the difference between SPGA and NSGAII was not significant. Furthermore, the results of the Tukey test in large-size instances showed that there is no significant difference between the SPGA and NSGAII.
Given that our aim in this paper was to present an applied model that is close to reality, further assumptions can be made to get closer to the real issues. Therefore, the following are suggestions for further research:
Considering uncertainty in the problem parameters,
Simultaneous studying of energy issues, speeding up and noise pollution,
Using the exact method to solve the problem, and
Developing novel algorithms such as math-heuristic and hyper-heuristic algorithms.

Conflicts of interest: none

References

[1] Behnamian, J., Fatemi Ghomi, S.M.T., Jolai, F., et al. "Realistic two-stage flowshop batch scheduling problems with transportation capacity and time" *Applied Mathematical Modelling*, 36, pp. 723–735 (2012).
[2] Hillary, R. *Environmental management systems and cleaner production*: Wiley Toronto (1997).
[3] Scarazzato, T., Panossian, Z., Tenório, J., et al. "A review of cleaner production in electroplating industries using electrodialysis" *Journal of Cleaner Production, 168*, pp. 1590-1602 (2017).
[4] Mokhtari, H., & Hasani, A. "A multi-objective model for cleaner production-transportation planning in manufacturing plants via fuzzy goal programming" *Journal of Manufacturing Systems, 44*, pp. 230-242 (2017).

[5] Jariwala, H. J., Syed, H. S., Pandya, M. J., et al. "Noise Pollution & Human Health: A Review" *Noise and Air Pollutions: Challenges and Opportunities. Ahmedabad: LD College of Engineering* (2017).
[6] Zhang, J., Ding, G., Zou, Y., et al. "Review of job shop scheduling research and its new perspectives under Industry 4.0" *Journal of Intelligent Manufacturing, 30*(4), pp. 1809-1830 (2019).
[7] Yazdani, M., Zandieh, M., Tavakkoli-Moghaddam, R. "Evolutionary algorithms for multi-objective dual-resource constrained flexible job-shop scheduling problem" *OPSEARCH, 56*(3), pp. 983-1006 (2019).
[8] Shen, L., Dauzère-Pérès, S., Neufeld, J. S. "Solving the flexible job shop scheduling problem with sequence-dependent setup times" *European Journal of Operational Research, 265*(2), pp. 503-516 (2018).
[9] de Oliveira Neto, G. C., Santana, J. C. C., Godinho Filho, M., et al. "Assessment of the environmental impact and economic benefits of the adoption of cleaner production in a Brazilian metal finishing industry" *Environmental technology*, pp. 1-15 (2018).
[10] Rajaram, R., Jawahar, N., Ponnambalam, S., et al. "Multi-Objective Optimization of Economic and Environmental Aspects of a Three-Echelon Supply Chain" In *Industry 4.0 and Hyper-Customized Smart Manufacturing Supply Chains* (pp. 127-158): IGI Global (2019).
[11] Zarrouk, R., Bennour, I. E., Jemai, A. "A two-level particle swarm optimization algorithm for the flexible job shop scheduling problem" *Swarm Intelligence, 13*(2), pp. 145-168 (2019).
[12] Amjad, M. K., Butt, S. I., Kousar, R., et al. "Recent research trends in genetic algorithm based flexible job shop scheduling problems" *Mathematical Problems in Engineering,* (2018).
[13] Dai, M., Tang, D., Giret, A., et al. "Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints" *Robotics and Computer-Integrated Manufacturing, 59*, pp. 143-157 (2019).
[14] Abedi, M., Chiong, R., Noman, N., et al. "A multi-population, multi-objective memetic algorithm for energy-efficient job-shop scheduling with deteriorating machines" *Expert Systems with Applications*, 157, pp. 1-33 (2020).
[15] Li, X., Gao, L. "An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem" *International Journal of Production Economics, 174,* pp. 93-110 (2016).
[16] Kundakcı, N., Kulak, O. "Hybrid genetic algorithms for minimizing makespan in dynamic job shop scheduling problem" *Computers & Industrial Engineering, 96,* pp. 31-51 (2016).
[17] AitZai, A., Benmedjdoub, B., Boudhar, M. "Branch-and-bound and PSO algorithms for no-wait job shop scheduling" *Journal of Intelligent Manufacturing, 27*(3), pp. 679-688 (2016).
[18] Wu, J., Wu, G., Wang, J. "Flexible job-shop scheduling problem based on hybrid ACO algorithm" *International Journal of Simulation Modelling, 16*(3), pp. 497-505 (2017).
[19] Wang, L., Cai, J., Li, M., et al. "Flexible job-shop scheduling problem using an improved ant colony optimization" *Scientific Programming*, 2017, pp. 1-11, (2017).
[20] Jamrus, T., Chien, C.-F., Gen, M., et al. "Hybrid particle swarm optimization combined with genetic operators for flexible job-shop scheduling under uncertain processing time for semiconductor manufacturing" *IEEE Transactions on Semiconductor Manufacturing, 31*(1), pp. 32-41 (2017).
[21] Gong, X., Deng, Q., Gong, G., et al. "A memetic algorithm for multi-objective flexible job-shop problem with worker flexibility" I*nternational Journal of Production Researc*h, 7(56), pp. 2506-2522 (2017).
[22] Peng, C., Fang, Y., Lou, P., et al. "Analysis of double resource flexible job-shop scheduling problem based on genetic algorithm*" 15th International Conference on Networking*, Sensing and Control, pp. 1-6 (2018).
[23] Tamssaouet, K., Dauzère-Pérès, S., Yugma, C." Metaheuristics for the Job-Shop Scheduling Problem with Machine Availability Constraints" *Computers & Industrial Engineering*, 125, pp. 1-16 (2018).
[24] Gong, G., Deng, Q., Chiong, R., et al. "An effective memetic algorithm for multi-objective job-shop scheduling*" Knowledge-Based Systems*, 182, pp. 1-14 (2019).
[25] Zhang, G., Hu, Y., Sun, J., et al. "An improved genetic algorithm for the flexible job shop scheduling problem with multiple time constraints" *Swarm and Evolutionary Computation*, 54, pp. 1-15 (2020).
[26] Ding, H., Gu, X. "Improved particle swarm optimization algorithm based novel encoding and decoding schemes for flexible job shop scheduling problem" *Computers and Operations Research*, 121, pp. 1-15 (2020).
[27] Yang, D., Zhou, X., Yang, Zh., et al. "Multi-objective optimization Model for Flexible Job Shop Scheduling Problem Considering Transportation Constraints: A Comparative Study" *IEEE Congress on Evolutionary Computation (CEC)*, pp. 1-18 (2020).
[28] Heydari, M., Aazami, A. "Minimizing the maximum tardiness and makespan criteria in a job shop scheduling problem with sequence dependent setup times" *Journal of Industrial and Systems Engineering, 11*(2), pp. 134-150 (2018).
[29] Gao, K.-Z., Suganthan, P. N., Pan, Q.-K., et al. "Discrete harmony search algorithm for flexible job shop scheduling problem with multiple objectives" *Journal of Intelligent Manufacturing, 27*(2), pp. 363-374 (2016).

[30] Ebrahimi, A., Jeon, H. W., Lee, S., et al. "Minimizing total energy cost and tardiness penalty for a scheduling-layout problem in a flexible job shop system: A comparison of four metaheuristic algorithms" *Computers & Industrial Engineering, 141*, pp. 106295 (2020).
[31] Yazdani, M., Aleti, A., Khalili, S. M., et al. "Optimizing the sum of maximum earliness and tardiness of the job shop scheduling problem" *Computers & Industrial Engineering, 107*, pp. 12-24 (2017).
[32] Yu, J.-M., Lee, D.-H. "Solution algorithms to minimise the total family tardiness for job shop scheduling with job families" *European Journal of Industrial Engineering, 12*(1), pp. 1-23 (2018).
[33] Mendoza, A.M., Acosta, R.H., Reyes, J.C. "Production scheduling for a job shop using a mathematical model" *Methodology, 11*(12), pp. 13 (2018).
[34] Sadaghiani, J., Boroujerdi, S., Mirhabibi, M., et al. "A Pareto archive floating search procedure for solving multi-objective flexible job shop scheduling problem" *Decision Science Letters, 3*(2), pp. 157-168 (2014).
[35] Dalfard, V. M., Mohammadi, G. "Two meta-heuristic algorithms for solving multi-objective flexible job-shop scheduling with parallel machine and maintenance constraints" *Computers & Mathematics with Applications, 64*(6), pp. 2111-2117 (2012).
[36] Huang, S., Tian, N., Wang, Y., et al. "Multi-objective flexible job-shop scheduling problem using modified discrete particle swarm optimization" *SpringerPlus, 5*(1), pp. 1432 (2016).
[38] Behnamian, J., Zandieh, M., Fatemi Ghomi, S.M.T. "A multi-phase covering Pareto-optimal front method to multi-objective parallel machine scheduling" *International Journal of Production Research*, 48(17), pp. 4949–4976 (2010).
[37] Aurich, J. C., Yang, X., Schröder, S., et al. "Noise investigation in manufacturing systems: An acoustic simulation and virtual reality enhanced method" *CIRP Journal of Manufacturing Science and Technology, 5*(4), pp. 337-347 (2012).
[39] Behnamian, J., Fatemi Ghomi, S.M.T., Jolai, F. et al. "Minimizing makespan on a three-machine flowshop batch scheduling problem with transportation using genetic algorithm" *Applied soft computing*, 12, pp. pp. 768-777 (2012).
[40] Ahmadi, E., Zandieh, M., Farrokh, M., et al. "A multi objective optimization approach for flexible job shop scheduling problem under random machine breakdown by evolutionary algorithms" *Computers & Operations Research, 73*, pp. 56-66 (2016).
[41] Cuiyu, W., Yang, L., Xinyu, L. "Solving flexible job shop scheduling problem by a multi-swarm collaborative genetic algorithm" *Journal of Systems Engineering and Electronics*, 32, 2, pp. 261-271 (2021).
[42] Ehtesham Rasi, R." Optimization of the multi-objective flexible job shop scheduling model by applying NSGAII and NRGA algorithms" *Journal of Industrial Engineering and Management Studies*, 8, 1, pp. 45-71 (2021).

List of Figures

List of Tables

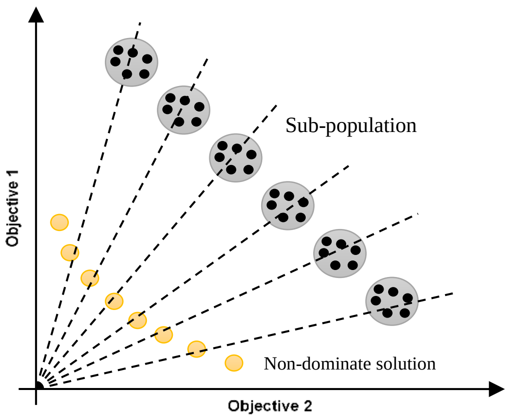


Figure 1: Pareto solutions

	J1	J2	J3	J4	J5	J6
S1	2	1	2	1	2	3
S2	1	3	1	2	3	2
S3	*	2	3	3	1	1

Figure 2: Jobs processing route

	J1	J2	J3	J4	J5	J6
S1		1.25		2.75		
S2	1.15		1.05			
S3					2.5	1.5

Figure 3: Solution representation

Parent 1

	J1	J2	J3	J4	J5	J6
S1		1.25		2.75		
S2	1.15		1.05			
S3					2.5	1.5

Parent 2

	J1	J2	J3	J4	J5	J6
S1		2.21				1.28
S2	2.95			1.75		
S3			1.25		2.3	

Child 1

	J1	J2	J3	J4	J5	J6
S1		1.25				1.28
S2	1.15			1.75		
S3			1.25		2.3	

Child 2

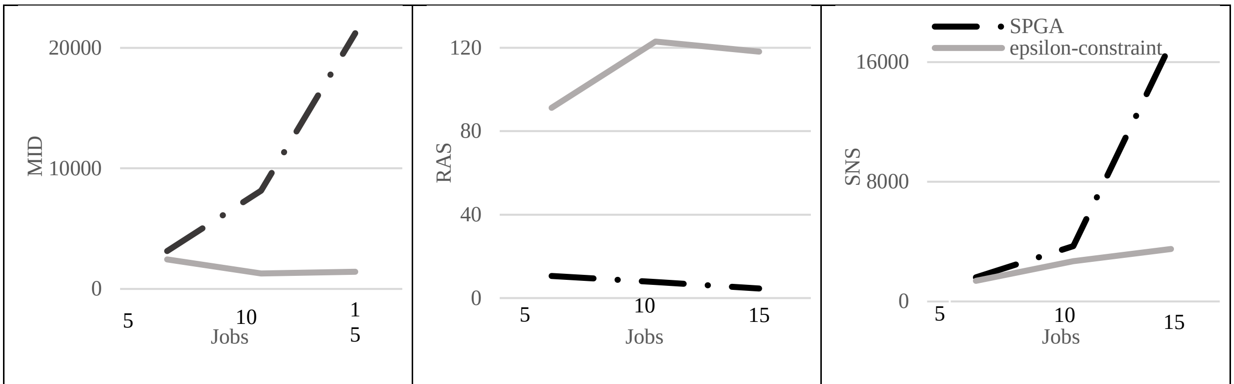
	J1	J2	J3	J4	J5	J6
S1		2.21		2.75		
S2	2.95		1.05			
S3					2.5	1.5

Figure 4: Crossover operator

	J1	J2	J3	J4	J5	J6
S1		1.25		2.75		
S2	1.15		1.05			
S3					2.5	1.5

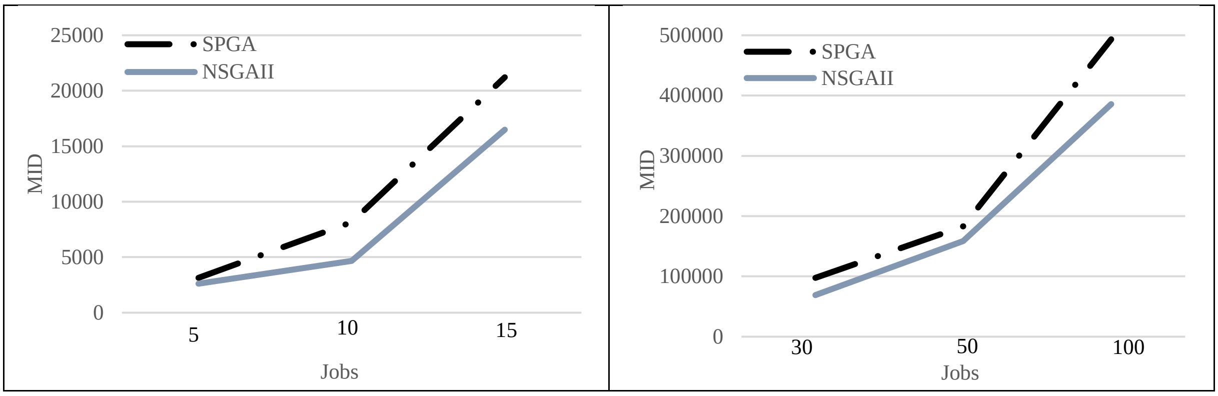
	J1	J2	J3	J4	J5	J6
S1				2.75	1.25	
S2	1.15		1.05			
S3		2.5				1.5

Figure 5: Mutation operator

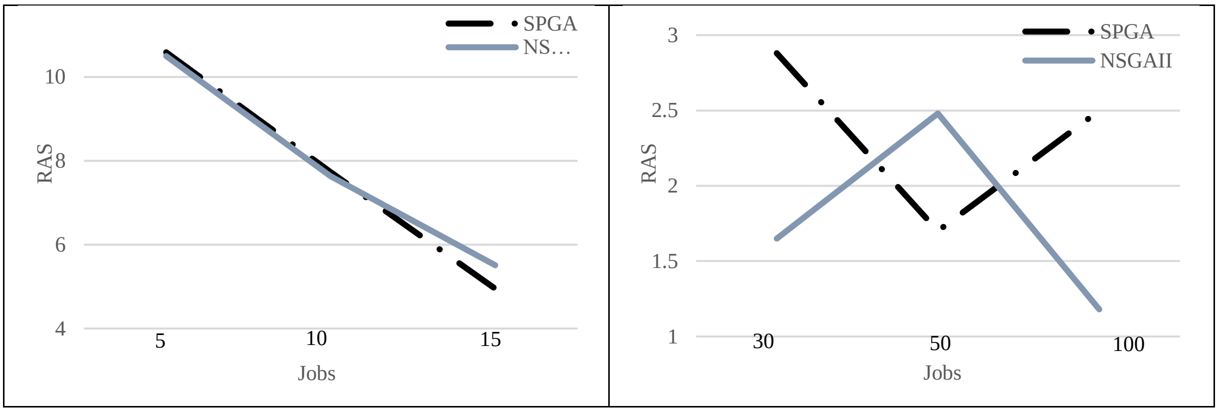


considering the MID criterion | considering the RAS criterion | considering the SNS criterion

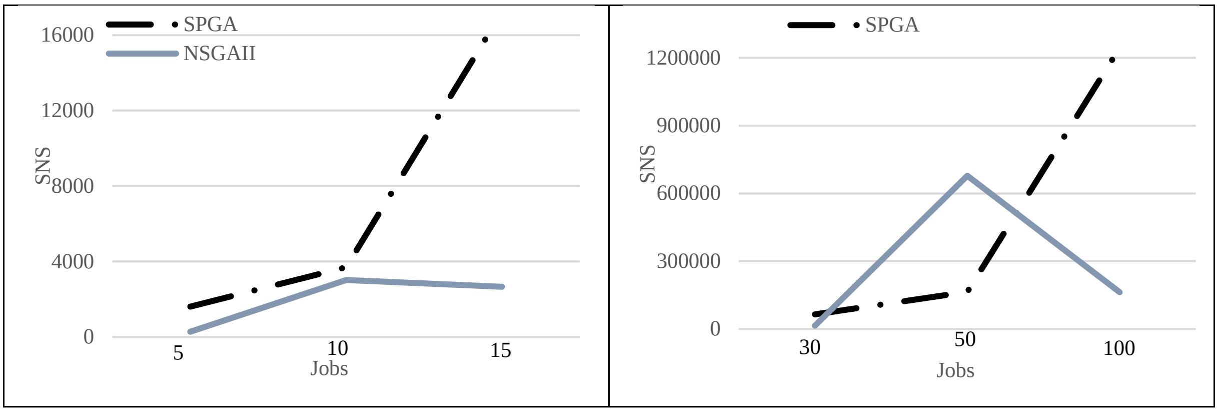
Figure 6: Comparisons of the ε-constraint method and SPGA



(*a*): Comparisons of SPGA and NSGAII considering the MID criterion



(*b*): Comparisons of SPGA and NSGAII considering the RAS criterion



(***c***)**:** Comparisons of SPGA and NSGAII considering the SNS criterion

Figure 7: Comparisons of the SPGA and NSGAII

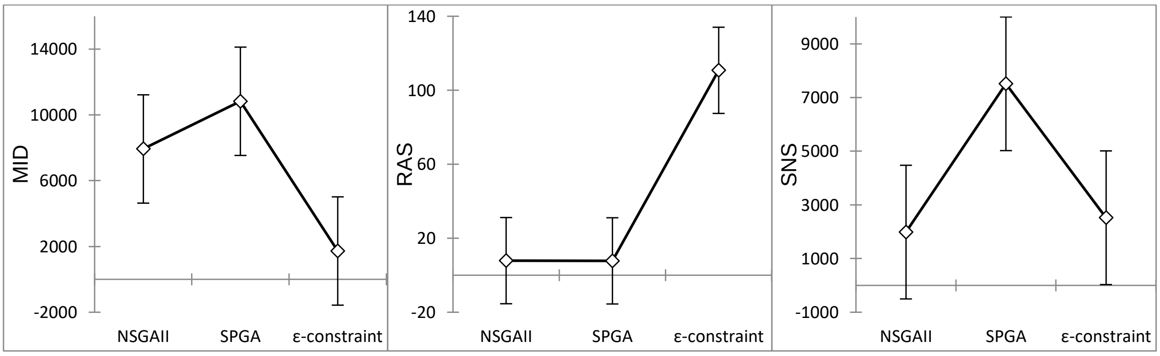


Figure 8: Comparison of algorithms in small-size instances

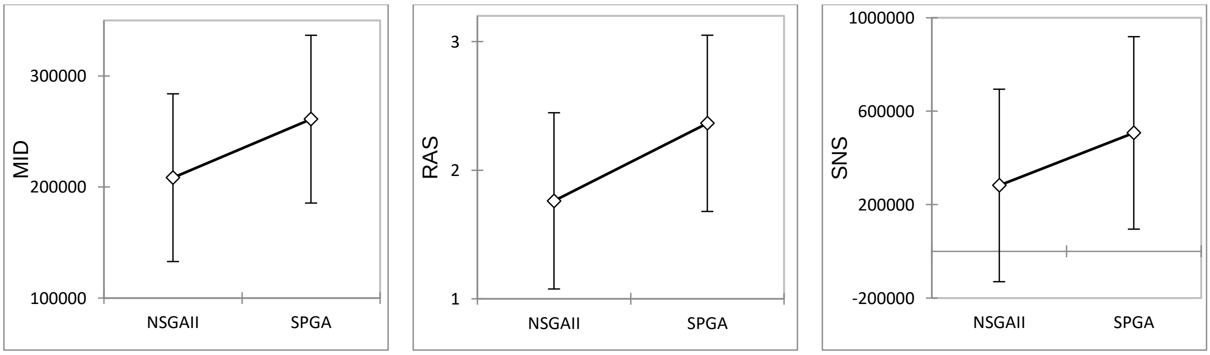


Figure 9: Comparison of algorithms in large-size instances

Table 1. A summary of the reviewed research related to the present study

Authors/Year	Objective function		Solving method		Environment		Resource		Impact Environment		
	Single	Multiple	Heuristic/ Meta-Heuristic	Exact	Job-shop	Flexible jobshop	Machine	Worker	Noise Pollution	Energy	CO_2
Rajaram et al. (2019)		*	*			*					*
Mokhtari et al. (2017)		*		*					*		*
Zarrouk et al. (2019)	*		*		*		*				
Amjad et al. (2018)		*	*			*	*			*	
Lu & Jiang (2019)	*		*		*		*			*	*
Dai et al. (2019)		*	*			*	*			*	
Maine et al. (2019)		*	*			*	*			*	
Abedi et al. (2020)		*	*		*		*			*	
Li & Gao (2016)	*		*		*						
Kundakcı and Kulak (2016)	*		*		*						
AitZai et al. (2016)	*			*	*		*				
Li and Gao(2016)	*		*		*		*				
Wu et al. (2017)		*	*			*	*				
Jamrus et al.(2017)	*		*			*	*				
Shen et al.(2018)	*		*			*	*				
Yazdani et al. (2019)	*		*			*	*				
Wang et al. (2017)	*		*			*	*				
Gong et al. (2017)	*		*			*	*	*			
Peng et al. (2018)	*		*			*	*	*			
Tamsuet et al. (2018)	*			*	*		*				
Gong et al. (2019)	*		*		*		*				
Zhang et al. (2020)	*		*			*	*				
Ding and Gu (2020)	*		*				*				
Heydari and Aazami(2018)	*			*	*		*				
Gao et al. (2016)							*				
Ebrahimi et al. (2020)		*	*			*	*				
Yazdani et al. (2017)	*		*		*		*				
Yu and Lee (2018)	*			*	*		*				
Mendoza et al. (2018)	*				*		*				
Sadaghiani et al. (2014)		*	*			*	*				
Dalfard et al. (2012)	*		*			*	*				
Huang et al. (2016)		*	*			*	*				
Present Study		*	*	*		*	*	*	*		

Table 2: Computational results in small-size instances

Jobs * Machines	Test problem	MID			RAS			SNS		
		SPGA	NSGAII	ε-constraint	SPGA	NSGAII	ε-constraint	SPGA	NSGAII	ε-constraint
5*2	1	325.95	258.86	1373.98	6.78	12.06	26.24	151.76	97.84	745.89
	2	690.88	541.01	1337.83	7.89	6.5	265.22	391.69	64.26	734.68
	3	1225.97	543.58	1409.66	6.74	6.6	29.7	588.52	104.02	758.07
	4	2292.74	2102.6	3542.05	8.94	15.8	197.72	3385.35	273.62	2400.44
	5	2566.18	3203.34	1337.63	15.07	9.11	29.68	1011.12	183.16	734.85
	6	4362.97	2074.17	1383.58	10.97	7.13	31.05	2763.83	95.14	759.1
	7	4002.73	4165.25	1381.53	12.63	9.5	31.6	1400.21	960.92	758.17
	8	4945.06	3016.22	2393.12	12.03	13.08	54.62	2387.02	589.99	1316.87
	9	4535.18	7207.08	5219.76	13.04	13.23	124.41	1741.84	346.16	2661.48
	10	6356.76	3050.38	5104.26	11.87	12.03	121.9	2266.5	55.34	2809.45
	average	3140.42	2616.24	2448.34	10.59	10.50	91.21	1608.78	277.04	1367.9
10*3	1	823.86	1623.21	1073.85	2.19	9.68	12.65	1174.78	233.88	1425.86
	2	1623.05	1124.86	896.26	3.32	1.33	48.6	1530.86	20.009	2663.26
	3	2290.03	3446.04	1162.48	7.6	4.5	19.21	1167.12	557.82	2429.51
	4	4400.77	4086.18	3576.15	7.18	18.51	124.41	2868.09	116.44	3261.93
	5	6547.86	3058.54	1001.55	14.36	7.6	121.94	2887.35	59.54	2557.48
	6	12646.17	3383.85	1073.73	8.71	1.85	617.66	2084.88	2696.08	2742.83
	7	11712.04	10666.91	1125.04	14.66	8.6	73.41	4911.86	843.84	3083.23
	8	13188.52	9026.74	909.76	7.61	6.08	63.47	6390.96	5373.81	2686.26
	9	13164.01	2204.94	900.65	6.48	11.43	64.43	7160.48	20142.66	2667.24
	10	15021.33	8157.23	1110.92	5.28	6.8	84.07	6743.61	159.83	3394.38
	average	8141.76	4677.85	1283.03	7.73	7.63	122.98	3691.99	3020.39	2691.19
15*5	1	1622.12	1147.91	1362.53	3.25	6.46	73.39	1113.59	550.76	3480.71
	2	4554.44	833.76	1476.49	2.33	1.66	102.91	2103.99	60.83	3771.95
	3	7400.54	6957.08	1850.35	3.33	2.006	124.38	4510.55	1474.95	1438.76
	4	23363.1	23762.84	1580.97	4.8	15.52	124.36	22575.88	2025.05	4038.92
	5	13977.3	25635.97	1604.86	5.68	3.8	131.61	7721.09	2900.81	4089.85
	6	20042.44	12492.78	1200.97	5.1	3.87	101.59	13575.66	547.36	3068.02
	7	20393.02	37305.88	1636.05	4.69	4.05	144.02	13905.24	15171.51	4179.69
	8	36027.13	9353.77	1449.86	3.7	4.07	128.28	33326.05	326.67	3703.95
	9	34167.02	10790.16	978.71	11.6	5.58	133.14	27943.94	322.3	4062.97
	10	50613.65	36638.75	1225.96	5.07	8.1	117.39	45645.93	3231.65	3223.81
	average	21216.07	16491.89	1436.67	4.95	5.51	118.1	17242.19	2661.18	3505.86

Table 3: Computational results in large-size instances

Jobs × Machines	Test problem	MID		RAS		SNS	
		SPGA	NSGAII	SPGA	NSGAII	SPGA	NSGAII
30*6	1	4333.4	10882.19	5.26	0.95	2056.009	1324.86
	2	11140.13	21603.34	2.43	0.26	4279.85	29429.1
	3	19706.49	4253.75	1.44	0.13	9565.36	1299.23
	4	57237.5	31186.01	2.68	2.68	59965.7	10371.53
	5	50910.97	48861.49	2.5	1.74	48678.88	1438.21
	6	55376.03	13275.72	1.63	0.42	18848.76	4023.26
	7	161510.32	83474.2	1.9	2.8	229358.84	43068.9
	8	89796.53	19760.74	1.76	1.98	31405.93	1264.13
	9	93314.52	132008.69	1.9	1.65	78044.66	3665.79
	10	115893.54	53154.76	1.78	1.53	57909.53	2580.25
	11	152219.39	115472.41	2.06	2.13	84060.74	3037.21
	12	90149.19	104973.8	2.36	2.002	51720.005	43900.36
	13	142337.31	100313.17	8.38	2.1	70191.82	21966.84
	14	165544.91	212943.99	2.33	2.33	150712.73	48300.09
	15	160852.4	82669.69	4.84	2.16	72131.51	920.33
	Average	**97570.65**	**68988. 93**	**2.88**	**1.65**	**64595.35**	**14439.33**
50*8	1	17016.04	18761.93	4.06	0.72	13290.66	5106.2
	2	34093.11	18003.38	1.91	0.38	35405.37	5573.79
	3	83768.32	29610.83	0.68	1.09	74624.37	1940.49
	4	57816.35	67078.68	0.91	2.76	118350.48	6918.54
	5	120212.6	144254.11	0.83	0.19	55732.36	38984.47
	6	165645.83	48475.95	1.79	0.4	211904.25	7018.59
	7	113019.47	206681.24	0.34	0.4	79943.18	192453.89
	8	247863.49	245373.2	2.75	11.99	236020.96	10475.82
	9	197662.8	217937.07	0.64	1.03	172556.3	50304.9
	10	161097.17	263347.01	1.07	14.14	106821.87	76759.94
	11	378178.71	255297.34	0.78	0.95	395352.9	7951.76
	12	196968.12	63812.42	0.76	0.27	324687.64	3075.95
	13	262847.87	262713.33	4.43	0.78	133696.29	9676124.57
	14	260656.68	71091.81	0.85	1.63	149771.05	51080.59
	15	450472.35	467839.27	3.7	0.56	347488.4	37549.01
	Average	**183154.59**	**158685.17**	**1.7**	**2.48**	**163709.73**	**678087.9**
100*6	1	44161.72	15138.73	8.6	1.62	34252.01	385788.81
	2	111760.16	73492.12	2.87	1.69	223480.12	5782.45
	3	158320.44	21425.56	1.82	1.05	232807.93	1817.11
	4	212199.58	201660.16	1.15	0.70	383493.13	6372.27
	5	317530.45	67732.17	2.81	0.89	511114.24	5179.58
	6	277298.86	299425.15	1.81	0.1	140435.72	11025.48
	7	542362.06	852384.2	0.33	1.49	798158.55	1606868.18
	8	715527.13	424537.99	1.72	0.03	1404039.67	101922.45
	9	302113.11	554287.27	1.2	0.11	188018.9	105556.1
	10	403661.83	436321.61	1.15	0.51	201272.42	24861.86
	11	472083.42	293809.84	2.03	0.26	207686.83	20958.25
	12	523411.69	760693.33	0.87	0.17	259353.18	88374.04
	13	1422182.04	687892.45	3.35	0.43	2939556.25	11009.31
	14	1195072.67	723506.95	4.38	0.36	2343493.81	15537.02
	15	702018.34	374524.96	3.42	8.29	8796135.17	41385.07
	Average	**493313.56**	**385788.83**	**2.5**	**1.18**	**1244219.86**	**162162.53**

Table 4: Tukey analysis with a confidence interval of 95% considering the MID criterion in small-size instances

Contrast	Difference	Standardized difference	Critical value	Pr > Diff	Significant
SPGA vs. ε-constraint	9106.743	3.886	2.384	**0.001**	Yes
SPGA vs. NSGAII	2900.764	1.238	2.384	0.434	No
NSGAII vs. ε-constraint	6205.978	2.648	2.384	**0.026**	Yes

Table 5: Tukey analysis with a confidence interval of 95% considering the RAS criterion in small-size instances

Contrast	Difference	Standardized difference	Critical value	Pr > Diff	Significant
ε-constraint vs. SPGA	103.005	6.213	2.384	**<0.0001**	Yes
ε-constraint vs. NSGAII	102.884	6.205	2.384	**<0.0001**	Yes
NSGAII vs. SPGA	0.121	0.007	2.384	1.000	No

Table 6: Tukey analysis with a confidence interval of 95% considering the SNS criterion in small-size instances

Contrast	Difference	Standardized difference	Critical value	Pr > Diff	Significant
SPGA vs. NSGAII	5528.117	3.119	2.384	**0.007**	Yes
SPGA vs. ε-constraint	4992.671	2.817	2.384	**0.016**	Yes
ε-constraint vs. NSGAII	535.445	0.302	2.384	0.951	No

Table 7: Tukey analysis with a confidence interval of 95% considering the MID criterion in large-size instances

Contrast	Difference	Standardized difference	Critical value	Pr > Diff	Significant
SPGA vs. NSGAII	52672.299	0.980	1.987	0.330	No

Table 8: Tukey analysis with a confidence interval of 95% considering the RAS criterion in large-size instances

Contrast	Difference	Standardized difference	Critical value	Pr > Diff	Significant
SPGA vs. NSGAII	0.603	1.236	1.987	0.220	No

Table 9: Tukey analysis with a confidence interval of 95% considering the SNS criterion in large-size instances

Contrast	Difference	Standardized difference	Critical value	Pr > Diff	Significant
SPGA vs. NSGAII	224990.980	0.767	1.987	0.445	No

Biographies

Mrs M. Hajibabaei received her MSc degree in Industrial Engineering from Bu-Ali Sina University in 2021 under supervision of Dr. Javad Behnamian. Her research activities are mainly in the areas of supply chain management, and production planning.

J. Behnamian obtained his Ph.D. in Industrial Engineering from Amirkabir University of Technology, Tehran, Iran. Currently, he is an Associate Professor at Industrial Engineering Department, Bu-Ali Sina University, Hamedan, Iran. His main research areas are: production scheduling, distributed systems, multicriteria decision making, and soft computing.