

## Research Article

# A New Approach Based on the Learning Effect for Sequence-Dependent Parallel Machine Scheduling Problem under Uncertainty

Maryam Ebrahimi <sup>1</sup>, Parviz Dinari,<sup>2</sup> Mohamad Samaei,<sup>3</sup> Rouhollah Sohrabi <sup>4</sup>,  
and Soheil Sherafatianfini<sup>5</sup>

<sup>1</sup>Department of Information Technology Management, Electronic Branch, Islamic Azad University, Tehran, Iran

<sup>2</sup>Department of Industrial Engineering, Kharazmi University, Karaj, Iran

<sup>3</sup>Department of Industrial Engineering, Yazd University, Yazd, Iran

<sup>4</sup>Faculty of Economics and Social Sciences, Bu-Ali Sina University, Hamedan, Iran

<sup>5</sup>Department of Industrial Engineering, Tarbiat Modares University, Tehran, Iran

Correspondence should be addressed to Maryam Ebrahimi; ebrahimim@modares.ac.ir

Received 7 June 2022; Revised 28 June 2022; Accepted 18 August 2022; Published 10 September 2022

Academic Editor: Anibal Coronel

Copyright © 2022 Maryam Ebrahimi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Production system design has lots of restrictions and complex assumptions that cause difficulty in decision making. One of the most important of them is the complexity of the relationship between man and machine. In this regard, operator learning is recognized as an effective element in completing tasks in the production system. In this research, a mathematical model for scheduling the parallel machines in terms of job degradation and operator learning is presented. As one of the most important assumptions, the sequence-dependent setup time is of concern. In other words, jobs are processed sequentially, and there is a sequence-dependent setup time. Moreover, the processing time and delivery due date are considered uncertain, and a fuzzy conversion method is used to deal with this uncertainty. The proposed mathematical model is a multiobjective one and tries to minimize speed and completion time. In order to optimize this mathematical model, the genetic algorithm (GA) and variable neighborhood search (VNS) algorithms have been used. A new hybrid algorithm has also been developed for this problem. The results show that the hybrid algorithm can provide more substantial results than classical algorithms. Moreover, it is revealed that a large percentage of Pareto solutions in the proposed algorithm have a generation time of more than 80% of the algorithm's execution time.

## 1. Introduction

The parallel machine scheduling problem is very important because this category of issues is considered not only in the production environment but also in information systems, i.e., computation of distribution and parallel processing of information. In real problems, the main problem is the uncertainty of the parameters affecting the problem. Parameters such as processing times and delivery times are generally indefinite, and fuzzy numbers can be used to indicate this uncertainty. However, in the workplace, other

factors affect the condition of the problem and its basic parameters [1]. Too often, raw materials waiting in line for processing and production deteriorate over time, which can even result in the complete loss of that raw material. Such an effect is called the “decay effect.” On the production line, on the other hand, workers with production equipment perform operations that increase their ability and skill over time and with time can speed up the production process. Such an effect is called a “learning effect.” But another parameter that affects the time of completion of jobs in a production system is the setup time of the device depending on the sequence [2].

This type of setup time can depend on different conditions, one of which is the setup time depending on the sequence. It is also clear that in a workshop environment, devices can be heterogeneous, that is, not of the same type (for example, with different brands) and that such a difference between devices leads to differences in their processing speeds [3].

Nowadays, due to the competitive market conditions, organizations must seek to increase the efficiency and optimization of their production operations in order to survive. For this reason, they should consider different factors in order to optimize their schedule and respond to customer requests in a timely manner. This situation encourages companies to move from separate planning processes to integrated planning. They can respond quickly to change and deliver higher quality products and lower production costs using integrated scheduling systems [4, 5].

The scheduling of parallel machine problems in a definite environment without considering the effects of degradation and learning has been considered by many researchers, including the articles of Unlu and Mason [1], Bozorgirad and Logendran [2], Tirkolaee et al. [3], and Chobar et al. [4]. Considering the effect of deterioration, several articles have been published. Ji and Cheng [6] and Liu et al. [7] investigated the scheduling of parallel machines with the aim of minimizing the sum of the completion times and in the case where the degradation follows the linear function. Mazdeh et al. [8] examined the scheduling of parallel machines with the aim of minimizing the sum of delays as well as the cost of machine degradation. Ruiz-Torres et al. [9] investigated this problem in a situation where the rate of decline is a function of the sequence of jobs and solved it with the help of SA. Moreover, Goli et al. [10] and Zhang and Luo [11] in their research examined the problem of scheduling the parallel machines in conditions where the effect of deterioration is seen in the works, taking into account conditions such as maintenance and rejection. Recently, Ding et al. [12] solved the problem of parallel machines by considering the effect of degradation using the exit chain algorithm (ECA).

Considering the effect of learning alone on scheduling problems, one can refer to the articles of Lee et al. [13] by examining the problem of uniform parallel machine scheduling; Goli et al. [14] considered both the effect of learning and the delivery time of goods; Przybylski [15] considered the effect of integral learning; Expósito-Izquierdo et al. [16] noted the addition of sequence-dependent setup times. Taking into account both the learning effect and the effect of degradation in the deterministic environment, Toksarı and Güner [17] investigated the problem of identical parallel machine scheduling with the aim of minimizing ET and nonlinear degradation coefficients and common delivery times.

Although much research has been conducted in the field of parallel machine scheduling problems in the definite environment, only a few limited articles have been published so far in the context of parameter uncertainty. Without the effects of learning and deterioration, Al-Khamis and

M'Hallah [18] have modeled and solved the basic problem of job scheduling on parallel machines in two steps. Peng and Liu [19] examined the basic parallel machine scheduling problem in terms of fuzzy processing times. They developed three methods for modeling potential problems: EVP, CCP, and DCP in a fuzzy environment by defining a new module called credibility. Finally, by presenting a hybrid intelligent algorithm (hybrid genetic algorithm and simulation annealing), they examined the results of the models.

Considering the effect of learning and deterioration without considering the time-dependent setup times, two articles have recently been published in a fuzzy environment. Rostami et al. [20] examined this problem for the first time. By presenting important features for this problem, they proposed a branch and boundary algorithm to solve the problems. Aric and Toksarı [21] also investigated and solved the same problem by considering the effects of deterioration and learning in a fuzzy environment with the help of a local search algorithm.

To the best of our knowledge, no research has been carried out on parallel machine scheduling so far, considering the learning effect and the effect of work degradation in a situation where processing times and delivery times are fuzzy and setup times depend on the sequence in question. Accordingly, the main contribution of this research can be summarized as presenting a developed mathematical model and a hybrid meta-heuristic algorithm to find the optimal scheduling of parallel machines, considering job degradation and operator learning. In this regard, NSGA-II and VNS algorithms are combined as a new hybrid algorithm. In other words, the contribution can be presented as follows:

- (i) Optimizing the sequence-dependent parallel machine scheduling under uncertainty
- (ii) Considering the operator learning effect in parallel machine scheduling
- (iii) Considering the role of work degradation in parallel machine scheduling
- (iv) Proposing a new hybrid algorithm based on NSGA-II and VNS

In Section 2, while defining the problem, first, a multiobjective mixed-integer nonlinear programming (MOMINLP) model is presented. Next, the proposed model is then finalized with the help of validity criteria as well as chance constraint planning (CCP). Section 3 includes the development of hybrid meta-heuristic algorithms to solve the problem in question on a large scale. Sections 4 and 5 show the computational results and discussion, respectively, and finally, Section 5 contains the conclusions and future studies.

## 2. Proposed Mathematical Model

*2.1. Problem Definition.* There are  $N$  jobs that are all available in zero time, and failure is not allowed. In the production workshop, there are  $M$  machines that have

TABLE 1: MOMINLP decision parameters and variables.

Indices	$i$	Index of jobs	$1, 2, \dots, N$
	$k, r$	Index of priorities	$1, 2, \dots, N$
	$j$	Index of machines	$1, 2, \dots, M$
Parameters	$N$	Total number of jobs	
	$\beta_j$	Learning effect coefficient of machine $j$	
	$\alpha_i$	Work-related deterioration effect factors for job $i$	
	$b_j$	Speed rate of machine $j$ setup time	
	$\bar{p}_i$	Nominal processing time of job $i$	
	$d_i$	Delivery due date of job $i$	
Variables	$V_j$	Speed rate of machine $j$	
	$\tilde{p}_{rj}$	Actual processing time of job $j$ in priority $r$ on machine $j$	
	$S_{rj}$	Setup time of priority $r$ on machine $j$	
	$\tilde{C}_{rj}$	Completion time of job $j$ on priority $r$	
	$\tilde{C}_{\max}$	Last job completion time (makespan)	
	$\tilde{T}_{rj}$	Delay of job in priority $r$ on machine $j$	
	$\tilde{E}_{rj}$	Earliness of job in priority $r$ on machine $j$	
	$x_{irj}$	Equal to 1 if job $i$ is given priority $r$ on machine $j$	

different processing speeds ( $V_j$ ). Jobs must be processed on one of these machines. Workers working with each machine have different skills in gaining skills, which is indicated by a learning factor  $\beta_j$ , which is a negative number. Jobs also decompose at different rates over time, denoted by a factor of  $\alpha_i$ . To begin processing everything on each machine, we need to spend setup time, which depends on the sequence of the past (see research by [22–25]). Processing times and delivery times are also fuzzy triangular numbers. The goal is to minimize the sum of early and late (ET) and make span at the same time in accordance with the standard introduced by Graham et al. [26].

In this regard, LE,  $D$ , and ST show the effect of training, deterioration, and setup time depending on the previous sequence, respectively. Since the  $Pm||C_{\max}$  problem has been proven to be NP-hard by Liao and Sheen [27], therefore, by adding more difficult conditions to the problem, this problem remains NP-hard.

## 2.2. Fuzzy Nonlinear Multiobjective Programming Model.

In this section, a fuzzy multiobjective mixed-integer programming model is presented. First, it is assumed that the sequence on each machine is divided into  $N$  hypothetical priorities, which are denoted by  $r$ . Then, the process of assigning jobs to these priorities is carried out by the model. The model searches for the solution space to finally find the optimal solution. Table 1 summarizes the parameters and decision variables of the proposed model.

According to the parameters and decision variables defined in Table 1, the proposed model is defined as follows: MOMIP model:

$$\text{Min } \tilde{f}_1 = \sum_{r=1}^N \sum_{j=1}^M \tilde{T}_{rj} + \sum_{r=1}^N \sum_{j=1}^M \tilde{E}_{rj}, \quad (1)$$

$$\text{Min } \tilde{f}_2 = \tilde{C}_{\max}. \quad (2)$$

Subject to

$$\tilde{p}_{rj} = \frac{1}{V_j} \left[ \sum_{i=1}^N [x_{irj} \times ([\bar{p}_i + \alpha_i \times (r-1)] \times r^{\beta_j})] \right] \forall r, j, \quad (3)$$

$$\begin{cases} s_{rj} = b_j \sum_{k=1}^{r-1} \tilde{p}_{kj}, & \forall r, j, \\ s_{1j} = 0, & \forall j, \end{cases} \quad (4)$$

$$\begin{cases} \tilde{C}_{rj} = \tilde{C}_{r-1,j} + s_{rj} + \tilde{p}_{rj}, & \forall r, j, \\ \tilde{C}_{0j} = 0, & \forall j, \end{cases} \quad (5)$$

$$\tilde{C}_{\max} \geq \tilde{C}_{n,j} \quad \forall j, \quad (6)$$

$$\begin{cases} \tilde{T}_{rj} \geq \left( \tilde{C}_{r,j} \times \sum_{i=1}^N x_{irj} \right) - \sum_{i=1}^N (\tilde{d}_i \times x_{irj}), \\ \tilde{E}_{rj} \geq \sum_{i=1}^N (\tilde{d}_i \times x_{irj}) - \left( \tilde{C}_{r,j} \times \sum_{i=1}^N x_{irj} \right), \end{cases} \quad (7)$$

$$\sum_{r=1}^N \sum_{j=1}^M \tilde{x}_{irj} = 1 \quad \forall i, \quad (8)$$

$$\begin{cases} \sum_{i=1}^N x_{i,r,j} \leq 1, & \forall i, \\ x_{I,r+1,j} \leq \sum_{i=1}^N \tilde{x}_{irj}, & \forall i, r (r \neq N), (I = 1, \dots, N), (I \neq i), \\ \tilde{p}_{rj}, \tilde{C}_{rj}, \tilde{T}_{rj}, \tilde{E}_{rj}, \tilde{C}_{\max} \geq 0, \end{cases} \quad (9)$$

$$x_{irj} \in \{0, 1\}. \quad (10)$$

The objective function (1) minimizes the sum of all late and earliness and tardiness. The second objective function in (2) minimizes the maximum time to complete jobs. Constraint (3) is designed to calculate the actual amount of processing time of jobs in each priority on each machine. From this relationship, it is clear that by increasing the

priority number, the amount of the deterioration effect on each job increases, and the processing time increases. Moreover, as the value of  $r$  increases, the learning effect will play a role and reduce the processing time of the priority. Constraint (4) for calculating the setup times depends on the previous sequence on each machine. It is clear that with an increase in  $r$ , the amount of setup time also increases on each machine. Constraint (5) for calculating the completion time of each priority of each machine is equal to the completion time of the previous priority on the same machine plus the actual setup and processing times of that priority. Constraint (6) is to convert the Max-Min objective function to the simplified Min objective and calculate the makespan. Constraint (7) shows the value of tardiness or earliness for each job.

Constraint (8) implies that each job must be assigned to only one priority of a machine. Constraint (9) is designed to calculate the absence of more than one job in each priority of each machine as well as the absence of idle time in the machines. Finally, Constraint (10) defines the decision variables.

**2.3. Chance Constraint Programming Based on Credibility Theory.** Chance-constraint programming (CCP) is a process for uncertain parameters within a problem while guaranteeing a specific performance. Uncertain parameters in a mathematical model lead to questions regarding reliability and risk, creating difficulties in determining the most likely result. In stochastic optimization, expected or nominal values are used for these uncertainties. At the same time, there is some risk in making these stochastic optimization decisions. However, CCP allows for certain unexpected events that violate specific constraints as long as the overall constraint satisfaction is maintained with a given level of probability. In addition, in CCP, a system's performance can be optimized with uncertain constraints using the chance-constraint optimization method by accounting for these constraints and ensuring they satisfy some well-defined reliability values [28, 29].

In this section, the model presented above becomes a definite model with the tool of measuring credit and getting help from CCP. In the following model, the concept of fuzzy chance constraint programming [19] is used. The difference with probability is that instead of the odds function, the validity value function of an uncertain event is used. A noteworthy point in the following model is that in calculating ET, care must be taken that for jobs with delays in the value of  $(\tilde{C}_{r,j} \times \sum_{i=1}^N x_{irj}) - \sum_{i=1}^N (\tilde{d}_i \times x_{irj})$  and for jobs with acceleration, the value  $\sum_{i=1}^N (\tilde{d}_i \times x_{irj}) - (\tilde{C}_{r,j} \times \sum_{i=1}^N x_{irj})$ , so the concept of a distance between two numbers is used here to establish such conditions.

In the following model, two new variables  $\bar{f}_1$  and  $\bar{f}_2$  are defined to model the problem using CCP, which are the upper bound of the objective functions 1 and 2, respectively. Also, instead of the two variables  $\tilde{E}_{r,j}$  and  $\tilde{T}_{r,j}$ , a new variable called  $\tilde{E}\tilde{T}_{r,j}$  is used, which is a combination of the two. The two parameters  $\lambda_1$  and  $\lambda_2$  are also the level of reliability of the first and second objective functions.

Credibility-based CCP model:

$$\text{Min}(\bar{f}_1, \bar{f}_2). \quad (11)$$

Subject to

$$Cr \left\{ \sum_{r=1}^N \sum_{j=1}^M \tilde{E}\tilde{T}_{r,j} \leq \bar{f}_1 \right\} \geq \lambda_1, \quad (12)$$

$$Cr \{ \tilde{C}_{n,j} \leq \bar{f}_2 \forall j \} \geq \lambda_2, \quad (13)$$

$$\tilde{E}\tilde{T}_{r,j} = d \left( \left( \tilde{C}_{r,j} \times \sum_{i=1}^N x_{irj} \right), \sum_{i=1}^N (\tilde{d}_i \times x_{irj}) \right) \forall r, j. \quad (14)$$

(3)–(5), (8), and (9) constraints

$$\tilde{p}_{r,j}, \tilde{C}_{r,j}, \tilde{C}_{\max} \geq 0 \quad x_{irj} \in \{0, 1\}. \quad (15)$$

In this model, the objective function in (11) sequentially minimizes the two objective functions under consideration, Constraints (12) and (13). To define the odds limit in fuzzy problems, with the help of the credit validation tool, the validity value of the two objective functions is considered higher than a predetermined confidence level and seeks to find the minimum values of  $\bar{f}_1$ . This is true in these relationships. In Constraint (14), the expression on the right represents the distance between the two fuzzy numbers, namely  $\tilde{d}_i$  and  $\tilde{C}_{r,j}$ , which ultimately creates a triangular fuzzy number. It should be noted that in the above model, the confidence levels are assumed to be greater than 0.5, i.e.,  $\lambda_i > 0.5$ . The  $\varepsilon$ -constraint method is then used to single-purpose the proposed model by considering Objective 2 as the main objective function of the problem and adding Objective 1 to the constraints.

### 3. Developed Meta-Heuristic Algorithms

In this section, first, the steps of the NSGA-II algorithm developed for the problem under this paper are described, and next, the structure of the hybrid algorithm with the neighborhood variable search algorithm (VNS) is presented. The implementation steps of the NSGA-II algorithm can be described as follows:

**3.1. Solution Representation.** The type of display of solutions in this paper is considered as two-row and  $n$ -column matrices, where  $n$  represents the total number of available jobs. The first line of the matrix indicates the machine number on which each job (column number) is processed, and the second row indicates the priority of processing each job on that machine. For example, Figure 1 shows the actual problem space and the chromosome display space for 5 jobs and 2 parallel machines.

**3.2. Fitness Function.** In order to determine the value of each of the generated chromosomes, it is necessary to create a special fit function based on the Pareto rank and the swarm distance. Once the value of  $f_i^k$  has been calculated for each

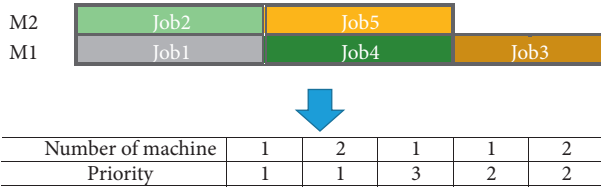


FIGURE 1: Illustration of solution representation.

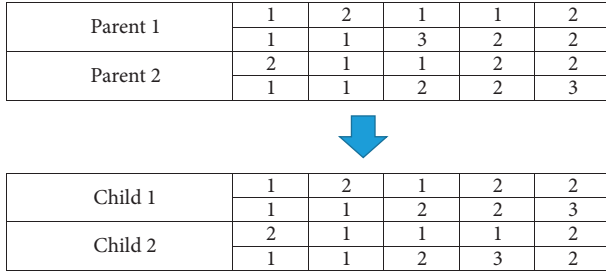


FIGURE 2: An example of the crossover operator.

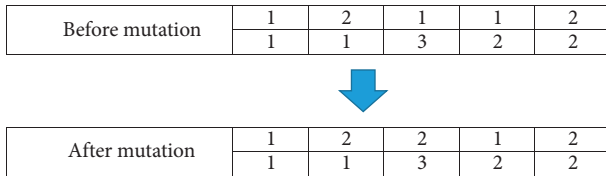


FIGURE 3: An example of the mutation operator.

solution  $i$  (by the mathematical relations of the objective functions presented in Section 3.3), then the congestion distance of each of the solutions can also be calculated.

**3.3. Crossover Operator.** In order to perform the intersection operation, a single-point intersection operator has been used. An example of a crossover operator is presented in Figure 2.

**3.4. Mutation Operator.** To perform mutation on the chromosomes, the number of columns  $P_{mut} * n$  is first randomly selected (although this value is rounded to the nearest larger integer). Then, the value of the components of the selected columns is randomly based on a uniform relation, adopting an integer between 1 and the maximum value of each row. In this algorithm,  $P_{mut} = 0.03$ . An example of the mutation operator is presented in Figure 3.

**3.5. Feasibility of Chromosomes.** In this structure, the constraints related to the sequence of the number of priorities in each machine may be violated after jumping and crossing operations. To solve this problem using the repair strategy, unjustified solutions are corrected after each execution of the intersection and jump operators. The following steps are taken to correct the solutions:

Step 1. For machine  $j$ , the amount of priority assigned to the machine's jobs is sorted in ascending order

Step 2. The rank of each job in the ordered layout of step 1 is considered the new priority value

Step 3. The above operation is performed on all machines to repair all the solutions

**3.6. Stop Condition.** One of the well-known conditions in the literature is the condition of achieving the maximum number of repetitions, which is also considered in this article.

**3.7. Hybrid VNS Algorithm.** In this section, we will present the features of the VNS algorithm in order to combine it with the NSGA-II algorithm. This compound can be classified as a low-level and computational heuristic (LCH). In this combination, after generating the solutions obtained from the intersection operators, some of the chromosomes will be affected by the VNS algorithm with two neighboring structures (NSS) in order to find better solutions to extract a better solution (close local optimization). In order to evaluate the results obtained from the VNS algorithm, an innovative approach based on the concept of one-way resonance has been used. In this algorithm, two neighborhood structures are proposed as follows:

The first structure: random selection of two machines, a random selection of a job on each machine, and movement in working with each other

The second structure: random selection of two machines, the random selection of a job on each machine, shifting in working with each other, random reselection of a job on each machine, and shifting in working with each other

Moreover, to stop the number between 50 and 100 repetitions, it has been proposed as two options, which have been selected as a better option in the parameter setting section. On the other hand, due to the hybrid VNS and NSGA-II algorithm, it is necessary to define other parameters, which is the start algorithm (VNS). In this study, the VNS algorithm will be called for one-third of the total NSGA-II iterations that expire. Step length of step means that after starting the VNS algorithm, the VNS algorithm will be run every three times. The number of chromosomes ( $VNS_{number}$ ), i.e., each time a VNS is performed, a percentage of the total number of offspring created will be used for local improvement. This parameter contains a numeric value of 15% by setting the parameter.

## 4. Computational Results

In order to validate and achieve the efficiency of the mathematical model and algorithm presented in the previous section, in this section, problems are randomly generated and solved by these methods to discuss the efficiency of the proposed methods. As the data are the most important aspect of implementing a mathematical model, it is necessary to use the validated data from the literature. Accordingly, to create random test problems, the method presented

TABLE 2: Comparison of MINLP and NSGA-II results.

$M$	$n$	MINLP					NSGA-II				
		ONVG	CPU time (sec.)	Avg. running time (sec.)	G D	C (MINLP, NSGA)	ONVG	CPU time (sec.)	Avg. running time (sec.)	G D	C (NSGA, MINLP)
2	5	13	437	33.61	4.8	1	8	5.06	0.632	5.4	1
	7	18	1038	129.75	2.1	5	8	5.54	0.692	4.5	0
3	9	13	2074	159.53	0	13	10	12.68	1.268	33.1	0
	7	13	4001	307.76	9.3	3	11	7.43	0.675	14.2	2
	9	15	6985	465.66	5.9	6	10	14.24	1.424	12.7	3

TABLE 3: Comparison of the quality of simple and hybrid mode results.

Metrics	Basic NSGA-II		Hybrid NSGA-II and VNS	
	Mean	SD	Mean	SD
Running time (sec)	112.85	53.19	634.76	22.74
ONVG metric	25.67	1.15	27.67	2.31
Spacing metric	0.00072	0.00006	0.00056	0.00008
Max (ET)-min (ET)	709.60	17.74	799.03	50.16
Max (C max)-min (C max)	96.54	6.73	109.54	3.49
Mean of ET	1099.58	9.73	1133.36	20.12
Mean of cmax	164.10	1.34	157.39	2.56

in the research of Rostami et al. [20] has been used. Moreover,  $b_j = U(0.1, 0.3)$  is selected. In this paper, the confidence levels of the first and second objective functions are considered 0.95 and 0.9, respectively.

The mathematical programming model has been modeled and solved by Gomez software. In the NSGA-II algorithm, the number of iterations is assumed to be 200. The parameters of mutation rate, crossover rate, and population size after setting the parameter are  $N_p = 20$ ,  $P_c = 0.9$ , and  $P_m = 0.15$ , respectively. The NSGA-II algorithm is encoded by MATLAB software.

Here, two criteria are used to evaluate the solutions. The first criterion is the error rate of the Parthian points obtained by the algorithm  $k$  from the optimal Pareto front obtained by the other algorithm ( $GD_k$ ). It should be noted that in these relationships, only  $d_i$  is related to points that are defeated by other points. In other words,  $d_i$  exists when the value of  $\min_j \{ (f_1^i(x) - f_2^j(x)) + (f_2^i(x) - f_1^j(x)) \}$  is positive.  $PF_{\text{known}}$  is the number of generated Pareto solutions. In equations (16) and (17), index  $i$  corresponds to points created by the algorithm  $k$ , and the  $j$  index corresponds to points created by another algorithm.

$$GD_k = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{|PF_{\text{known}}|}, \quad (16)$$

$$d_i = \max\{0, \min_j \{ (f_1^i(x) - f_2^j(x)) + (f_2^i(x) - f_1^j(x)) \}\}. \quad (17)$$

The second criterion is the number of coverages. If we show the Pareto front obtained from the mathematical model with  $X$  and the Pareto front obtained with NSGA-II

with  $Y$ , then the criterion  $C(X, Y)$  indicates the number of points at which the  $Y$  front is defeated by the  $X$  front.

Table 2 shows the results for comparing the mathematical model and the NSGA-II for small-scale problems. In this table, ONVG shows the number of members of the optimal Parthian front created.  $C$  represents the number of coverage points by each algorithm, and  $GD$  is calculated for each algorithm from equation (16). The results in Table 2 show the very high speed of NSGA-II compared to MINLP in the production of the Pareto front. But on the other hand, in this category of problems, according to the two criteria of  $GD$  and the number of coverages, the mathematical programming model provides a relatively better solution. The noteworthy point in this table is that for problems with three machines, some Pareto points of the meta-heuristic algorithm overcome some Pareto points of the mathematical model (given the value of  $C$ ), which indicates that the nonlinear model in this category of problems has failed to create a global Parthian front.

With the exception of the small size category, the rest of the problems can only be achieved by a good Parthian front with NSGA-II. Therefore, in order to evaluate the solutions obtained by the meta-innovative method, it is necessary to compare them with the solutions that are very close to Parthian optimal solutions. In this paper, a suitable solution is obtained with the help of the NSGA-II algorithm and with a large population size, which is called the best-known solution (BKS). To obtain BKS, the number of repetitions equals 1000, and the population size equals 100. Moreover, the number of Pareto solutions is another important factor to compare the methods. In this regard, ONVG is defined as the number of Pareto solutions. This is shown in Table 2.

To evaluate large test problems, the criteria of the population number, solution times, number of Pareto points created by the algorithm, spacing criterion, slaves and mean of objective functions,  $GD$  criterion, and finally, the average of generated response errors were used. The mean error rate measure actually calculates the  $GD$  ratio relative to the mean point distance of the two objective functions from the origin of the coordinates. This criterion is used because  $GD$  values are proportional to the values of the objective functions. Therefore, if the value of the objective functions is a large number, then the value of  $GD$  also increases in proportion to them, which by comparison cannot be appropriate on its own. The error value is calculated as follows:

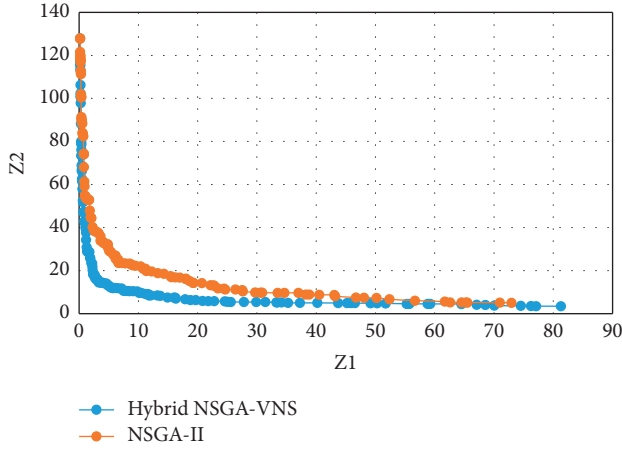


FIGURE 4: Pareto optimal solutions for different algorithms.

$$\text{Error} = \frac{GD}{\sqrt{(\text{mean } ET)^2 + (\text{mean } C \text{ max})^2}} \quad (18)$$

The spacing criterion is also calculated by using equations (19) and (20), which indicates the order of the solutions on the Pareto boundary. However, before calculating the criterion, the values of the objective functions are linearly normalized to take values between zero and one.

$$\text{spacing} = \sqrt{\frac{1}{|PF_{\text{known}}|} \sum_{i=1}^{|PF_{\text{known}}|} (\bar{d} - \bar{d}_i)^2}, \quad (19)$$

$$\bar{d}_i = \min\left(|f_1^i(x) - f_1^j(x)| + |f_2^i(x) - f_2^j(x)|\right). \quad (20)$$

In this analysis, all the criteria for the BKS obtained for it are calculated. For this purpose, by considering the VNS algorithm and executing the algorithm for 16 jobs and 4 machines, Table 3 compares the results with each other. As it turns out, according to the claim, the scatter value of Pareto responses in the one-way resonance approach is more remarkable than the average.

On the other hand, in order to analyze the rate of convergence, Figure 4 is proposed. In this Figure, the vertical axis represents the time of birth of each of the Parthian final solutions, and the horizontal axis represents the final Parthian solution number. As a result, the more the chart tends to be 100%, the later it means that the final Pareto solutions converge and stabilize later. As can be seen in Figure 2, a large percentage of Pareto responses in each of the approaches have a birth time of more than 80% of the execution time of the algorithm, which means that there is no early convergence in the proposed model.

## 5. Discussion

In recent years, a large number of multiobjective evolution algorithms have been introduced. We can mention the second version of the multiobjective genetic algorithm with uncoordinated ranking among these algorithms. Their

drawback is their computational complexity which is of the degree  $mN$  where  $m$  is the number of targets and  $N$  is the size of the population. Evolution will increase dramatically.

The results presented in this research, and especially the results in Figure 4, show that the development of the NSGA-II algorithm and the presentation of a hybrid algorithm based on it can increase the quality of Pareto solutions. This algorithm also has a great variety of solutions to help decision makers choose the most appropriate one according to their opinions.

## 6. Conclusion

In this paper, the scheduling of heterogeneous parallel machines is investigated in a fuzzy environment in which the effect of decay and learning effects and setup sequences depend on past sequences that simultaneously affect processing times. First, introductory definitions and a review of the literature on the subject were given. Then, considering the processing times and delivery times of works as fuzzy triangular numbers, a hybrid mathematical model of non-linear integers was presented.

In this problem, the two objective functions, namely, the sum of the accelerations and the sum of the delays and the maximum completion time, are minimized simultaneously. Then, with the help of the validity measurement criterion, whose features are very close to the probable concepts, and with the help of the odds constraint programming method, the multiobjective fuzzy model becomes a definite model that can be coded. It has the solution software. Since the proposed mathematical model does not have the ability to solve large-scale problems at the time of logical solution, the NSGA-II meta-heuristic method was used to solve large-scale problems. Moreover, in order to increase the power of the NSGA-II algorithm, the hybrid algorithm is presented in the form of a multiobjective genetic algorithm based on the concept of one-way resonance.

In order to evaluate the methods presented in this article, first, problems were generated completely randomly in small sizes and large sizes. For small problems that both methods can solve logically, Pareto sets were created by two methods and compared with each other. The results showed that in small-scale problems, the mathematical model provided relatively better solutions than the NSGA-II, although in some problems, the mathematical model performed worse than the NSGA-II. By solving large-scale problems by comparing the results between the simple and hybrid NSGA-II modes, the results showed that the idea of a one-way intensification approach plays a significant role in obtaining the Pareto front solutions.

This research has various management insights, the most important of which can be mentioned in creating a holistic view of the characteristics of a production system. Considering the effect of learning on the processing speed of jobs is one of the issues that exist in almost all production systems, and its study can lead to providing appropriate solutions to improve the performance of the production system. On the other hand, one of the most important limitations of this research is the lack of access to data from



multiple companies for a comprehensive evaluation of the proposed method. To develop this research, it is suggested that the developed mathematical model be optimized with new algorithms such as the multiobjective gray wolf optimizer.

## Data Availability

Data are available within the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] Y. Unlu and S. J. Mason, "Evaluation of mixed integer programming formulations for non-preemptive parallel machine scheduling problems," *Computers & Industrial Engineering*, vol. 58, no. 4, pp. 785–800, 2010.
- [2] M. A. Bozorgirad and R. Logendran, "Sequence-dependent group scheduling problem on unrelated-parallel machines," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9021–9030, 2012.
- [3] E. B. Tirkolaee, A. Goli, P. Ghasemi, and F. Goodarzi, "Designing a sustainable closed-loop supply chain network of face masks during the COVID-19 pandemic: Pareto-based algorithms," *Journal of Cleaner Production*, vol. 333, Article ID 130056, 2022.
- [4] A. P. Chobar, M. A. Adibi, and A. Kazemi, "Multi-objective hub-spoke network design of perishable tourism products using combination machine learning and meta-heuristic algorithms," *Environment, Development and Sustainability*, vol. 5, pp. 1–28, 2022.
- [5] A. P. Chobar, M. A. Adibi, and A. Kazemi, "A novel multi-objective model for hub location problem considering dynamic demand and environmental issues," *Journal of industrial engineering and management studies*, vol. 8, no. 1, pp. 1–31, 2021.
- [6] M. Ji and T. E. Cheng, "Parallel-machine scheduling with simple linear deterioration to minimize total completion time," *European Journal of Operational Research*, vol. 188, no. 2, pp. 342–347, 2008.
- [7] M. Liu, F. Zheng, S. Wang, and Y. Xu, "Approximation algorithms for parallel machine scheduling with linear deterioration," *Theoretical Computer Science*, vol. 497, pp. 108–111, 2013.
- [8] M. M. Mazdeh, F. Zaerpour, A. Zareei, and A. Hajinezhad, "Parallel machines scheduling to minimize job tardiness and machine deteriorating cost with deteriorating jobs," *Applied Mathematical Modelling*, vol. 34, no. 6, pp. 1498–1510, 2010.
- [9] A. J. Ruiz-Torres, G. Paletta, and E. Pérez, "Parallel machine scheduling to minimize the makespan with sequence dependent deteriorating effects," *Computers & Operations Research*, vol. 40, no. 8, pp. 2051–2061, 2013.
- [10] A. Goli, H. Khademi-Zare, R. Tavakkoli-Moghaddam, A. Sadeghieh, M. Sasanian, and R. Malekalipour Kordestanizadeh, "An integrated approach based on artificial intelligence and novel meta-heuristic algorithms to predict demand for dairy products: a case study," *Network: Computation in Neural Systems*, vol. 32, no. 1, pp. 1–35, 2021.
- [11] M. Zhang and C. Luo, "Parallel-machine scheduling with deteriorating jobs, rejection and a fixed non-availability interval," *Applied Mathematics and Computation*, vol. 224, pp. 405–411, 2013.
- [12] J. Ding, L. Shen, Z. Lü, and B. Peng, "Parallel machine scheduling with completion-time-based criteria and sequence-dependent deterioration," *Computers & Operations Research*, vol. 103, pp. 35–45, 2019.
- [13] W. C. Lee, M. C. Chuang, and W. C. Yeh, "Uniform parallel-machine scheduling to minimize makespan with position-based learning curves," *Computers & Industrial Engineering*, vol. 63, no. 4, pp. 813–818, 2012.
- [14] A. Goli, H. Khademi Zare, R. Tavakkoli-Moghaddam, and A. Sadeghieh, "Hybrid artificial intelligence and robust optimization for a multi-objective product portfolio problem Case study: the dairy products industry," *Computers & Industrial Engineering*, vol. 137, Article ID 106090, 2019.
- [15] B. Przybylski, "A new model of parallel-machine scheduling with integral-based learning effect," *Computers & Industrial Engineering*, vol. 121, pp. 189–194, 2018.
- [16] C. Exposito-Izquierdo, F. Angel-Bello, B. Melián-Batista, A. Alvarez, and S. Báez, "A metaheuristic algorithm and simulation to study the effect of learning or tiredness on sequence-dependent setup times in a parallel machine scheduling problem," *Expert Systems with Applications*, vol. 117, pp. 62–74, 2019.
- [17] M. D. Toksarı and E. Güner, "Parallel machine earliness/tardiness scheduling problem under the effects of position based learning and linear/nonlinear deterioration," *Computers & Operations Research*, vol. 36, no. 8, pp. 2394–2417, 2009.
- [18] T. Al-Khamis and R. M'Hallah, "A two-stage stochastic programming model for the parallel machine scheduling problem with machine capacity," *Computers & Operations Research*, vol. 38, no. 12, pp. 1747–1759, 2011.
- [19] J. Peng and B. Liu, "Parallel machine scheduling models with fuzzy processing times," *Information Sciences*, vol. 166, no. 1–4, pp. 49–66, 2004.
- [20] M. Rostami, A. E. Pilerood, and M. M. Mazdeh, "Multi-objective parallel machine scheduling problem with job deterioration and learning effect under fuzzy environment," *Computers & Industrial Engineering*, vol. 85, pp. 206–215, 2015.
- [21] O. A. Arık and M. D. Toksarı, "Multi-objective fuzzy parallel machine scheduling problems under fuzzy job deterioration and learning effects," *International Journal of Production Research*, vol. 56, no. 7, pp. 2488–2505, 2018.
- [22] M. Alinaghian and A. Goli, "Location, allocation and routing of temporary health centers in rural areas in crisis, solved by improved harmony search algorithm," *International Journal of Computational Intelligence Systems*, vol. 10, no. 1, pp. 894–913, 2017.
- [23] A. Goli and B. Malmir, "A covering tour approach for disaster relief locating and routing with fuzzy demand," *International Journal of Intelligent Transportation Systems Research*, vol. 18, no. 1, pp. 140–152, 2020.
- [24] E. Asgharizadeh, M. Kadivar, M. Noroozi, V. Mottaghi, H. Mohammadi, and A. P. Chobar, "The intelligent traffic management system for emergency medical service station location and allocation of ambulances," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 2340856, 2022.
- [25] P. Ghasemi, H. A. Khalili, A. P. Chobar, S. Safavi, and F. M. Hejri, "A new multiechelon mathematical modeling for pre-and postdisaster blood supply chain: robust optimization



- approach,” *Discrete Dynamics in Nature and Society*, vol. 2022, Article ID 2976929, 10 pages, 2022.
- [26] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. R. Kan, “Optimization and approximation in deterministic sequencing and scheduling: a survey,” *Annals of discrete mathematics*, vol. 5, pp. 287–326, 1979.
- [27] L. W. Liao and G. J. Sheen, “Parallel machine scheduling with machine availability and eligibility constraints,” *European Journal of Operational Research*, vol. 184, no. 2, pp. 458–467, 2008.
- [28] M. Rahmaty, A. Daneshvar, F. Salahi, M. Ebrahimi, and A. P. Chobar, “Customer churn modeling via the grey wolf optimizer and ensemble neural networks,” *Discrete Dynamics in Nature and Society*, vol. 2022, pp. 1–12, Article ID 9390768, 2022.
- [29] A. Geletu, M. Klöppel, H. Zhang, and P. Li, “Advances and applications of chance-constrained approaches to systems optimisation under uncertainty,” *International Journal of Systems Science*, vol. 44, no. 7, pp. 1209–1232, 2013.