Hindawi
Discrete Dynamics in Nature and Society
Volume 2022, Article ID 8962052, 10 pages
https://doi.org/10.1155/2022/8962052

Research Article

Modeling and Solving the Flow-Shop Scheduling Problem with Sequence-Dependent Setup Times by Firefly Algorithm (Case Study: Automotive Industry)

Mustafa Mohammadi,[1] Seyed Ahmad Shayannia ,[1] Mohamadreza Lotfi,[2] and Javad Rezaeian Zaidi[3]

[1]*Department of Industrial Management, Firoozkooh Branch, Islamic Azad University, Firoozkooh, Iran*
[2]*Department of Industrial Engineering, Islamic Azad University, Firoozkooh Branch, Firoozkooh, Iran*
[3]*Department of Industrial Engineering, Mazandaran University of Science and Technology, Babol, Iran*

Correspondence should be addressed to Seyed Ahmad Shayannia; sheibat@yahoo.com

Received 22 May 2022; Revised 30 July 2022; Accepted 2 August 2022; Published 24 August 2022

Academic Editor: Abdellatif Ben Makhlouf

Progress in today's modern industry requires a lot of knowledge, one of which is scheduling. Flow-shop scheduling is one of the most widely used optimization problems. In this research, considering the importance of simultaneously order in different stages of production in the automotive industry, and also in order to make the problem more practical, we have investigated the problem of scheduling the flow-shop, taking into account the lead time and the costs of each order. Due to the fact that in most research studies, the lead time and costs of an order have been ignored because they made it difficult to find the initial solution to the problem. In this research, using the firefly meta-heuristic method, a suitable solution is provided to overcome this problem. Therefore, considered objective function is to minimize the total completion time. Absolute relative error (ARE) has been used to validate the model in a deterministic and meta-heuristic mode. According to the ARE result, the difference in the results between the two algorithms is negligible. Then, the sequence results are determined according to the desired algorithm for 5 tasks considered for automobile parts. The results show that the completion time of job 1 is 1397.85; job 2 is 771.44; job 3 is 608.65; job 4 is 1163.87; and job 5 is 479.45.

1. Introduction

Planning is the making of decisions for the future, and production planning means determining the production strategy for allocating production lines to meet orders. One of the most prominent cases in preparing the manufacturing schedule for production lines in determining the accumulated size and sequence of orders and how to allocate resources over time [1]. We always use the term schedule in our everyday conversations. However, we may not always have a proper definition in mind. Although schedules generally seem tangible and simple, creating them is complex without a deep understanding of scheduling. Scheduling problems in the industry have a similar structure. They include a set of activities and resources available to perform those activities. Also, some decisions are known as planning decisions in the industry. The planning process determines the resources needed to produce and the activities required for scheduling. In the scheduling process, we need to determine the type and amount of each resource, and as a result, we can determine the possible time of completion of jobs [2]. Scheduling is allocating resources limited to activities over time to optimize one or more objective functions. Resources include manpower, machinery, materials, and auxiliary equipment. Machinery operations, movements, transfers, loads, etc. are also examples of activities. Activities can have the earliest start time, the latest end time, and the delivery time. The purpose of scheduling is cases such as the minimum completion time for a set of orders, minimum delay, the maximum number of activities or orders completed in a given time, minimum intermediate inventory, and interaction in the use of resources. According to the intended objectives and with regard to the existing constraints such as production capacity, resource capacity, resource inventory, budget constraints, and time constraints, the problem of scheduling or allocating resources to activities over time is done [3, 4].

Based on the abovementioned discussion, scheduling is allocating resources over time to perform a set of jobs. This definition has two different meanings. First, scheduling is a kind of decision-making process during which the schedule is set. Second, scheduling is a theoretical topic that encompasses a set of principles, models, methods, and logical outcomes that provide us with in-depth insights into the practice of scheduling. Therefore, considering that, it is assumed that there is a production strategy for ordering in this research. In this strategy, production is done based on the orders received. Unlike the production strategy for the warehouse where the production is done first, and then, the goods are stored in the warehouse to find a suitable customer, no products will be produced without an order from the customer. Many orders are received in the first period, some of which are accepted based on production capacity, and others are rejected. In this case, production needs to be scheduled in such a way as to minimize all tangible costs such as delay costs, adjustment costs, and maintenance costs, which are among the most critical criteria in terms of production scheduling and planning in many industries. In such a situation, a multicriteria scheduling model is usually needed.

As mentioned above, in this study, jobs in usual flow-shop problems are replaced by orders. Scheduling several orders where each order contains a certain number of similar jobs and a specific customer with a definite delivery time and a specific delay cost will be the main focus of the research. Maintenance and delay costs will be very important in the problem under study. In such problems, scheduling decisions have two levels. Scheduling of orders is at a higher level so that which order and how and in what order should be processed. The next level is scheduling of jobs within each order in a way that in what order the jobs need to be processed. Here, only the first level is examined, and the order of processing the jobs within each order is not the subject of this research. For this, the main contributions of this article are as follows:

(i) Determined a schedule for sequence orders in the flow-shop problem.

(ii) Proposed an integer linear programming model for flow-shop scheduling problems, assuming sequence-dependent setups.

The rest of the article is organized as it is clear. The second section provides the historical background of past studies to identify the research gap. In the third section, the proposed framework and analysis method are introduced. The fourth section provides the research results of solving the proposed model. In the fifth section, managerial insights is presented. Finally, a general conclusion and suggestions for future research are provided in the sixth section.

2. Literature Review

In this section, we introduce studies that have been conducted in the past. Wang and Cheng [5] investigated the permutation flow-shop problem assuming the existence of only two machines and considering the capacity constraint for only the first machine. The setups in their study depended on the sequence. Schaller et al. [6] investigated the scheduling of families' jobs within each family in the flow-shop environment. In their study, the setup times between families are assumed to depend on sequences. The objective is to minimize the total completion time when jobs within each family are being processed. Ruiz et al. [7] introduced two genetic algorithms for the permutation flow-shop problem and showed that the algorithms introduced by them work better than other algorithms, especially Mercado and Brad [8]. They used the modified NEH method to find the initial solution to the flow-shop problem. The name of the heuristic method is NEHT-RMB, which can be used as an effective method to find a set of appropriate answers to the flow-shop problem. In the genetic algorithms introduced, the answers were used as the pool of answers in the algorithm. Also, Ruiz and Stutzle [9] in their paper introduced two simple local search methods based on an interactive greedy algorithm. Their algorithm is two-stage: the destruction stage, in which some jobs are removed from the initial answer, and the construction stage, in which the deleted jobs are assigned to the initial answer using the NEH heuristic method. They showed that their algorithm works better than the algorithm of Ruiz et al. [7]. The introduced algorithm against a wide range of case studies was compared with a set of algorithms introduced in the literature, which shows the superiority of the introduced algorithm. Of course, the greedy algorithm alone is not very powerful without using local search. Ekşioğlu et al. [10] have provided an article in flow-shop flow problems, and in their article, they have used the modified tabu search method to solve the problem. However, with a general review, their method does not have the simplicity and accuracy of the greedy algorithm. Also, the studied space in the greedy algorithm is more comprehensive than the algorithm developed by them. In their recent article, Allahverdi et al. [11] provide a comprehensive overview of the studies conducted on the problems of flow-shop and permutation flow-shop. In recent studies, Balaji and Porselvi [12] consider the problem of scheduling batches of parts in a multicell flexible manufacturing system with batch setup time. The goal is to find the best sequence of batches and thus minimize the time interval. For this purpose, two mathematical models have been created: the batch availability model and the job availability model. Since the problem is NP-hard, a particle swarm optimization algorithm and a simulated annealing algorithm are proposed to solve the problem. The experimental results show that the simulated annealing method offers a better solution than particle swarm optimization. Celik and Dal [13] have developed a simulated annealing-based meta-heuristic method for cluster-based job scheduling in which sequential and parallel modes of the workflow are implemented in C++ software according to the model in this method. The effectiveness of the proposed approach is shown through 12 well-known criteria from the Brown dataset. In their study, Hashemi et al. [14] proposed a new approach based on particle swarm optimization for cellular problems with alternative routing to minimize intercellular motions. The applied approach offers a very near-optimal solution with only one change in component configuration compared to the best-known solution with the exact method. Finally, the computational results of the applied approach suggest a new adjusted optimization scenario for cellular manufacturing systems with alternative routes. This approach shows that particle swarm optimization with the number of particles as close as possible to the number of parts, inertia weights proportional to the maximum, and a minimum number of alternative routing and learning factors with a normal value of 2 can achieve an optimal value in less time. Brum et al. [15] address the problem of nonpermutation flow-shop scheduling, a more general type of flow-shop problem in which machines can have different sequences of jobs. The goal of this study is to minimize the total completion time. For this purpose, a model for generating meta-heuristic algorithms is proposed, and an automated algorithm configuration is used to obtain efficient methods. Algorithms start by building a high-quality permutation solution, which is then improved in the second stage, creating nonpermutation solutions by changing the order of jobs on some machines. Tamssaouet et al. [16] have proposed a scheduling framework for solving the complex multiobjective job-shop scheduling problem resulting from production. To produce practical and meaningful industrial schedules, this study extends the proposed batch approach by considering periods of unavailability and minimum time delays and simultaneously optimizing various industry-related criteria. To this end, a new criterion for the satisfaction of higher-level decision-making production goals has also been proposed. Abolghasemian et al. [17] presented a delay scheduling based on discrete-event simulation for construction projects. For this purpose, a combined approach of discrete-event simulation and computational modeling was applied; then, we compare the results. Measurements show that the systems fragmented by repeated and short repetitions while referring to early are in optimal performance. Rashidi Komijan et al. [18] presented a new bus routing. For this purpose, a multiobjective mixed-integer model is proposed to handle the associated problem. The minimization of transportation cost as well as traveling time is the main objective. The proposed model is applied in a real case study including 4 schools in Tehran. The results indicate the efficiency of the proposed model in comparison with the existing system. Khanchehzarrin et al. [19] presented a new mixed-integer nonlinear programming model for the time-dependent vehicle routing problem with time windows and intelligent travel times. The aim is to minimize fixed and variable costs, with the assumption that the travel time between any two nodes depends on traffic conditions and is considered to be a function of vehicle departure time. Depending on working hours, the route between any two nodes has a unique traffic parameter. For this purpose, considered each working day to be divided into several equal and large intervals, termed as a time interval of traffic. For this purpose, a Tabu search optimization algorithm is devised for solving large problems. Also, after linearization, a number of random instances are generated and solved by the CPLEX solver of GAMS to assess the effectiveness of our proposed algorithm. The results indicate that the initial travel time is estimated appropriately and updated properly in accordance with the repeating traffic conditions. Rezaei et al. [20] presented a vehicle routing problem in relief supply chain under crisis condition considering blood types. For this purpose, a bi-objective mixed-integer linear programming (MILP) model was developed for relief supply under crisis condition. The mentioned model has two objectives: maximizing the amount of blood collected by bloodmobiles and minimizing the arrival time of the blood receiver buses and a helicopter to a crisis-stricken city after the collected blood is used up. The model is coded by CPLEX software, and the results obtained from solving the model indicate that, without considering a helicopter, the demand is not supplied within the critical period after crisis. Momenitabar et al. [21] for the first time considered the impacts of the backup suppliers and lateral transshipment/resupply simultaneously on designing a sustainable closed-loop supply chain network (SCLSCN) to decrease the shortage that may occur during the transmission of produced goods in the network. In this manner, the fuzzy multiobjective mixed-integer linear programming model is proposed to design an efficient SCLSCN resiliently. Moreover, the concept of circular economy has been studied in this article to reduce environmental effects. This study aims at optimizing total and environmental costs, including energy consumption and pollution emissions, while increasing job opportunities. Pourghader chobar et al. [22] designed a multiobjective hub-spoke network of perishable tourism products. In order to consider the perishable factor of the products, some collection centers are considered for the products, which are perished. Accordingly, the combination of Hub-Spoke network and supply chain is assessed here. Moreover, this combination is to use transportation discounts in the supply chain network. The desired combination is done in such a way that the distributors are considered a set of hubs.

2.1. Research Gap. According to previous studies, most of the papers examined in this review have studied sequence-dependent setups. Therefore, studies in which time and cost are sequence-dependent are not considered. The most important gap in this study is to consider similar jobs within each order, in which the time and cost of setup between two jobs from one order are ignored. For this purpose, to fill the research gap, adding such an assumption makes it very difficult to find an initial answer to the problem based on the existing heuristic methods. To overcome such a problem, a

Table 1: Categorized literature review.

References		Solution approach						Goal
		Meta-heuristic						
Author	Number	PSO	AGIG	NSGA-II	Exact	Simulation	Computational	
Hashemi et al	[14]	*						Minimize intercellular motion
Brum et al	[15]		*					Nonpermutation flow-shop
Tamssaoret et al	[16]				*			Flow-shop scheduling
Abolghasemian et al	[17]					*	*	Construction delay scheduling
Rashidi komijan et al	[18]				*			Minimizing transportation cost
Khanchehzarrin et al	[19]				*			Minimize fix and variable cost
Rezaei et al.	[20]				*			Minimize amount of blood collected
Momenitabar et al.	[21]				*			Optimize total and environmental costs
Pourghader chobar et al.	[22]			*				Designed a multiobjective hub-spoke network of perishable tourism products

specific delivery time and a certain delay cost are assumed for each order (based on the customer's opinion). Table 1 shows previous studies categorized based on solution approach and main goals.

Therefore, the main contributions of this study are as follows:

(i) Determined a schedule for prioritizing orders in the job-shop problem.

(ii) Determining the optimal time for the determined sequences.

(iii) Proposed an integer linear programming model for flow-shop scheduling problems, assuming sequence-dependent setups.

(iv) Using a novel meta-heuristic solution approach to optimize the problem for executing concurrent work on each order using a specific lead time and cost impact factor for each order.

3. Proposed Method

This section introduces an integer linear programming model for flow-shop scheduling problems, assuming sequence-dependent setups taken from research sources [23]. For this purpose, details of the proposed mathematical modeling such as symbols, problem statement, and solution approach are stated.

3.1. Symbols

3.1.1. Sets. n: Number of jobs

m: Number of machines

i: Subscript of job $i\,(i = 1, \ldots, n)$

j: Subscript of machine $j\,(j = 1, \ldots, n)$

3.1.2. Variables

P_{ij}: Processing time, indicating the time required to process job i on machine j.

S_{ikj}: Setup time from a job i to job k on machine j ($i = 0$ refers to the initial setup time of the scheduled job).

C_{ij}: Completion time of job i on machine j.

X_{ik}: If job i is processed before job k, it is equal to one; otherwise, it is equal to zero.

3.1.3. Objective Function. The objective is to minimize the total completion time or the completion time of the last job on the last machine, which is considered Minimize makespan = C_{nm}.

3.1.4. Set of Constraints. Constraint (1) guarantees that all jobs are scheduled and the completion time of job i on machine 1 is at least as large as the processing time of that job on the machine:

$$p_{i1} \geq c_{i1};\, i = 1, \ldots, n. \tag{1}$$

Processing job i cannot start on machine j unless it is finished on machine $j-1$. Constraint (2) guarantees that the completion time of job i on machine j must be at least as large as the processing time on machine j, one greater than the completion time on machine $j-1$:

$$p_{ij} + c_i[j-1] \geq c_{ij}, i = 1, 2, \ldots, n;\, j = 2, 3, \ldots, m. \tag{2}$$

Constraints (3) and (4) guarantee that there is only one constraint for each sequence of jobs. This case shows the relationship between precedence and latency between jobs:

$$c_{ij} - c_{kj} + Mx_{ik} \geq s_{kij} + p_{ij}, \tag{3}$$

$$c_{ij} - c_{kj} + M[1 - x_{ik}] \geq s_{kij} + p_{ij}, \tag{4}$$

where $k > i \geq 1$ and $i = 1, 2, \ldots, n;\, k = 1, 2, \ldots, n;\, j = 1, 2, \ldots, m$. Also, M is a very large number. Constraints (5) and (6) guarantee that only one job can follow another job in each schedule:

$$\sum_{i=1}^{n} x_{ik} = 1; k = 1, 2, \ldots, n \text{ for } i \neq k, \quad (5)$$

$$\sum_{k=1}^{n} x_{ik} = 1; k = 1, 2, \ldots, n \; for \; i \neq k. \quad (6)$$

3.1.5. Model Assumption. The main model assumption is as follows:

(i) There is a certain number of jobs that can be assigned to a station.
(ii) Each operation is performed on its own machine.
(iii) The processing time of each job is determined.
(iv) The prerequisites for each work have been determined. Therefore, a task is executable when its prerequisite is completely completed.

3.2. Problem Statement. Azintaneh Factory, which is studied in this research, was established in 1994 and operated in manufacturing light- and heavy-vehicle brake system components. This factory has received special attention by employing specialized and experienced employees required by a part of the country's automotive industry in the company's management, along with increasing manufacturing capacity and product quality. This company succeeded and became one of the important suppliers of Iran-Khodro, Saipa, and Renault Pars Khodro. It also operates under the license of Lockheed A.P. of the United Kingdom and Bosch of Germany. This factory produces five types of booster products as follows.

(1) 8-inch ABS booster (Bardo) for use in Peykan Pickup.
(2) 9-inch ABS booster—for use in Peugeot 405.
(3) RANA booster—for use in Rana.
(4) 10-inch ABS booster—for use in Samand.
(5) 10-inch booster (LX)—for use in Samand LX.

We have five jobs, four types of machines, and 58 activities. We have measured the time of each of these activities, which we examine below. The stages of each of these products are shown in Tables 2–6 .

3.3. Solution Approach: A Firefly Optimization Algorithm. The firefly algorithm is one of the most powerful asset optimization algorithms, which is highly regarded for its convergence to the global optimal solution. The firefly algorithm, like other meta-heuristic algorithms, includes stages. This algorithm consists of seven stages: first, selecting the parameters; second, the initial random solution (Rand of size k) and writing the main loop of the algorithm, which is different according to the type of algorithms; third, moving around the firefly towards brighter fireflies; fourth, merging; fifth, choosing the best; sixth, if the stopping condition was set, stop the algorithm; otherwise, return to the second stage and re-run the stages, and finally the output of the algorithm.

Table 2: 8-inch ABS booster line.

Name of 8-inch ABS booster line station	Time (minute : seconds)
First station	27 : 27
Second station	22 : 28
Third station	10 : 06
Fourth station	11 : 05
Fifth station	38 : 29
Sixth station	34 : 17
Seventh station	19 : 94
Eighth station	25 : 66
Ninth station	36 : 09
Tenth station	36 : 04
Eleventh station	23 : 75
Twelfth station	18 : 63
Thirteenth station	32 : 63
Fourteenth station	29 : 01
Fifteenth station	30 : 78
Sixteenth station	42 : 23
Seventeenth station	17 : 71

Table 3: 9-Inch ABS booster line.

Name of 9-inch ABS booster line station	Time (minute : seconds)
First station	20 : 55
Second station	14 : 17
Third station	21 : 91
Fourth station	41 : 20
Fifth station	78 : 48
Sixth station	18 : 19
Seventh station	3 : 31
Eighth station	39 : 68
Ninth station	14 : 14
Tenth station	12 : 14

Table 4: 10-inch ABS booster line.

Name of 10-inch ABS booster line station	Time (minute : seconds)
First station	31 : 64
Second station	26 : 31
Third station	39 : 37
Fourth station	35 : 11
Fifth station	25 : 19
Sixth station	20 : 96
Seventh station	27 : 01
Eighth station	35 : 94
Ninth station	45 : 84

3.3.1. Setting Required Parameters. To determine and adjust the parameters of the firefly algorithm, we first design a number of scenarios using a design of experiment. In order to design experiments in the firefly algorithm, the Taguchi method has been used. To use this method, first, 3 different levels (low-level code 1, medium-level code 2, and high-level code 3) are defined for their parameters. And then, the predefined tests in this algorithm are executed for all possible combinations. The recommended values for the parameters of this algorithm are according to Table 7.

Table 5: RANA booster line.

Name of RANA booster line station	Time (minute : seconds)
First station	39 : 69
Second station	25 : 17
Third station	28 : 64
Fourth station	23 : 99
Fifth station	15 : 45
Sixth station	32 : 35
Seventh station	11 : 99
Eighth station	21 : 04
Ninth station	22 : 22
Tenth station	24 : 39
Eleventh station	39 : 69
Twelfth station	13 : 32
Thirteenth station	14 : 06

Table 6: LX booster line.

Name of LX booster line station	Time (minute : seconds)
First station	25 : 12
Second station	23 : 74
Third station	18 : 64
Fourth station	15 : 99
Fifth station	26 : 71
Sixth station	18 : 76
Seventh station	17 : 86
Eighth station	21 : 51
Ninth station	30 : 12

Table 7: Parameters and levels for the firefly algorithm.

Parameters	Value		
	Level 1	Level 2	Level 3
Gamma	0.025	0.05	0.1
Beta	0.05	0.1	0.2
Alpha	0.2	0.3	0.5
Alpha_damp	0.25	0.75	0.99

Then, with Taguchi's L9 design, we create different experiments and implement the firefly algorithm for each one. The execution results are presented in Table 8. In Table 8, all possible states are shown for different levels considered for the firefly algorithm factors. For example, in the first experiment, all the factors have been included in the experiment for their lowest level. In the second experiment, the gamma factor with the lowest level value and other factors with their respective average level value are present. In the same way, other possible states are completed based on the permutation rule in statistics. By running each design and calculating the value of the firefly algorithm, the desired response level is estimated using this algorithm.

By calculating the signal-to-noise ratio for each of the parameters and evaluating the calculated values, we determine the best level for each of the parameters. The lower the value of the signal-to-noise ratio for each parameter level, the more that parameter level is selected. Table 9 shows the calculated signal-to-noise ratio.

Table 8: Results of design of experiment response.

Run	Algorithm parameters				Firefly output
	Gamma	Beta	Alpha	Alpha_damp	
1	1	1	1	1	0.534
2	1	2	2	2	0.612
3	1	3	3	3	0.537
4	2	1	2	3	0.491
5	2	2	3	1	0.576
6	2	3	1	2	0.637
7	3	1	3	2	0.599
8	3	2	1	3	0.973
9	3	3	2	1	0.642

Table 9: S/N results.

Parameters	S/N		
	Level 1	Level 2	Level 3
Gamma	4.9	5.1	0.2
Beta	5.2	0.3	4.3
Alpha	0.3	4.8	4.9
Alpha-damp	4.3	4	3.7

Therefore, the best value of each parameter according to the S/N results is selected as follows:

Gamma = 0.1: light absorption coefficient,

Beta0 = 0.1: Absorption coefficient base value,

Alpha = 0.2: mutation coefficient,

Alpha_damp = 0.99: Alpha decreases by one percent in each iteration. We want it to converge towards a good solution.

4. Computational Results

The developed model has been implemented in a personal system with CPU Intel Core i5 and RAM 4 GB specifications. To solve exact model GAMS 24.1.2 software and firefly metaheuristic model MATLAB are used. For this, first, the problem data includes the times of each activity on the machines; we have five types of jobs and a total of 58 activities, and each activity has prerequisites. According to Table 10, the prerequisites of each activity are specified. Then, after entering the data, we explain the parameters as follows. After running the model based on the items set above, it calculates the value of the objective function, which is the maximum value of time for the machines, in such a way as to minimize the machine that performs its job the latest. For this purpose, the function $z = \max[(\text{mach}.t)]$ calculates the maximum time of machines taking into account Max it = 10.

According to the results, the completion time of the busiest machine is 275.83 seconds. After the evolutionary process that has been done, it reaches 271.27 seconds the second time, which is 4.56 seconds better than the previous one, and thus improves. Table 11 shows the other computational results of the objective function of the problem based on the implementation of the firefly algorithm.

Table 10: Prerequisites for each activity.

Station	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Job 1	—	—	—	3	—	5	6	7	8	—	—	11	—	—	14	—	16
Job 2	—	1	—	—	—	5	6	—	—	9	—	—	—	—	—	—	—
Job 3	—	—	—	3	4	5	—	7	8	—	—	—	—	—	—	—	—
Job 4	—	1	2	—	4	5	—	7	8	9	—	—	12	—	—	—	—
Job 5	—	1	2	3	—	—	6	7	8	—	—	—	—	—	—	—	—

Table 11: Results of objective functions in the ten iterations.

Iteration 1: Best fit = 275.83
Iteration 2: Best fit = 271.27
Iteration 3: Best fit = 266.83
Iteration 4: Best fit = 261.79
Iteration 5: Best fit = 259.83
Iteration 6: Best fit = 259.83
Iteration 7: Best fit = 259.83
Iteration 8: Best fit = 254.83
Iteration 9: Best fit = 254.83
Iteration 10: Best fit = 254.83

Then, by plotting the objective function according to Figure 1, it is clear that the decreasing trend from the first to the tenth iteration is evident. Therefore, the objective function has a downward trend.

In Tables 12–16, the priority of the execution of each activity is shown according to their operations. In addition, the implementation of each operation using each machine is specified. Also, the duration of the execution of each activity in the optimal sequence mode has also been calculated. In Table 12, the sequence of job 1 activities is shown. The sequence of job 1 activities is 1; 2; 5; 6; 7; 11; 12; 13; 14; 15; 9; 10; 16; 17; 4; 3; 8. Based on the optimal sequence, the completion time of Job 1 is equal to 1397.85 minutes.

In Table 13, the sequence of job 2 activities is shown. The sequence of job 2 activities is 8; 4; 9; 10; 7; 1; 2; 3; 5 and 6. Based on the optimal sequence, the completion time of Job 2 is equal to 771.44 minutes.

In Table 14, the sequence of job 3 activities is shown. The sequence of job 3 activities is 5; 6; 1; 2; 3; 4; 7; 8; and 9. Based on the optimal sequence, the completion time of Job 3 is equal to 608.65 minutes.

In Table 15, the sequence of job 4 activities is shown. The sequence of job 4 activities is 12; 13; 2; 3; 9; 10; 11; 4; 5; 6; 7; 8 and 1. Based on the optimal sequence, the completion time of Job 4 is equal to 1163.87 minutes.

In Table 16, the sequence of job 5 activities is shown. The sequence of job 5 activities is 5; 6; 1; 2; 3; 4; 7; 8; and 9. Based on the optimal sequence, the completion time of Job 5 is equal to 479.45 minutes.

4.1. Validation. In order to evaluate the effectiveness of the algorithm, 7 numerical samples with different aspects are considered. Then, according to the response of the CPLEX exact method and the firefly algorithm, absolute relative error (ARE) is calculated that recommended Abolghasemian et al. [24]. ARE is calculated as shown in Equation (7) based on the comparison of genetic output and branches and borders. Table 17 compares the results obtained from the branch and bound algorithm.

$$\frac{|\text{Cplex output} - \text{firefly output}|}{\text{Gfirefly output}}. \quad (7)$$

The ARE results shown in Table 17 are calculated to be less than 0.1 for each sample therefore, the error difference between the results of the two algorithms is negligible.

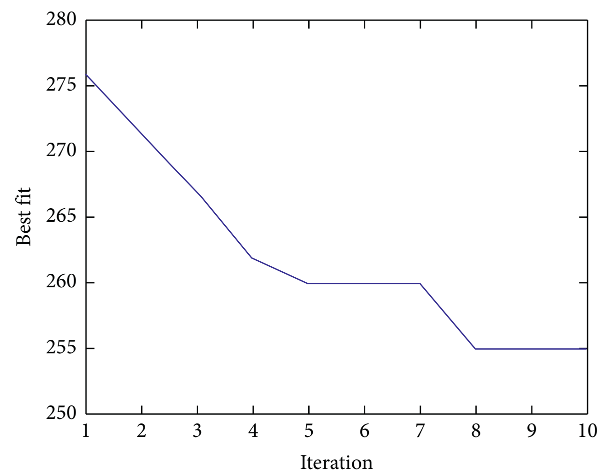

Figure 1: The trend of changes in the value of the objective function in each iteration.

Table 12: Sequence of job 1 operations.

Job	Operation	Machine	Start time	Final time	Duration
1	1	1	0	91.27	91.27
1	2	2	91.27	186.55	95.28
1	5	5	0	129.29	129.29
1	6	6	146.15	190.32	44.17
1	7	7	190.32	280.26	89.94
1	11	11	0	36.75	39.75
1	12	12	39.32	78.95	39.63
1	13	13	78.95	148.58	69.63
1	14	14	148.58	199.59	51.01
1	15	15	199.59	260.37	60.78
1	9	9	0	133.9	133.9
1	10	10	305.35	360.39	55.04
1	16	16	260.37	347.6	57.23
1	17	17	347.6	451.31	103.71
1	4	4	213.79	248.29	226.5
1	3	3	186.55	261.61	75.06
1	8	8	311.1	346.76	35.66

Table 13: Sequence of job 2 operations.

Job	Operation	Machine	Start time	Final time	Duration
2	8	8	0	115.68	115.68
2	4	4	0	57.2	57.2
2	9	23	115.68	174.82	59.14
2	10	10	174.82	254.96	80.14
2	7	22	0	19.31	19.31
2	1	1	160.03	186.58	26.55
2	2	19	186.58	297.05	110.47
2	3	20	297.05	415.96	118.91
2	5	21	57.2	213.65	156.45
2	6	6	333.6	361.19	27.59

Table 14: Sequence of job 3 operations.

Job	Operation	Machine	Start time	Final time	Duration
3	5	26	0	104.19	104.19
3	6	6	104.19	146.15	41.96
3	1	1	122.39	160.03	37.64
3	2	24	160.03	272.24	112.21
3	3	25	0	101.75	101.75
3	4	4	248.29	287.4	39.11
3	7	27	146.15	271.16	12.01
3	8	8	271.16	311.1	39.94
3	9	28	311.1	430.94	119.84

Table 15: Sequence of job 4 operations.

Job	Operation	Machine	Start time	Final time	Duration
4	12	12	0	39.32	39.32
4	13	35	39.32	153.92	193.24
4	2	29	0	94.17	94.17
4	3	30	94.17	144.81	50.64
4	9	33	0	108.22	108.22
4	10	10	254.96	305.35	50.39
4	11	34	305.35	432.04	126.69
4	4	4	144.81	183.8	38.99
4	5	31	183.8	287.25	103.45
4	6	6	287.25	333.6	46.35
4	7	32	0	90.99	90.99
4	8	8	346.76	387.8	41.04
4	1	1	186.58	240.27	53.69

5. Managerial Insight

The issue of scheduling operations with dependent start-up times in a n-work model and n-machine (n/n) has always been of interest to researchers. This issue, despite the developed methods, the complexity of the problems, and the time-consuming nature of their solution, continues to be of interest to researchers and especially executives. In this article, based on the application of research theorems in operations and the properties of the solution space of mathematical planning models, the integer of a new algorithm can be developed that can easily present the n/n operation sequence if it has dependent start-up times. Quick access to answers and problem-solving is one of the salient features of this method, while all previous methods have very long solution times.

Therefore, in this research, by overcoming the difficulty of the problem after considering the assumption of simultaneous execution of operation in each order, taking into account the lead time and the cost of each order, we were able to provide a prioritization of the execution of operations in the production of various parts in the automotive industry. Therefore, the most important managerial applications of the present research are

(i) Determining the optimal sequence of parts based on their completion time.

(ii) Calculating the execution time of each operation in each part and also the total time of making the final part.

Table 16: Sequence of job 4 operations.

Job	Operation	Machine	Start time	Final time	Duration
5	5	38	0	38.71	38.71
5	6	6	38.71	100.47	61.76
5	1	1	91.27	122.39	31.12
5	2	36	0	52.74	52.74
5	3	37	52.74	118.38	65.64
5	4	4	183.8	213.79	29.99
5	7	39	100.47	196.33	95.86
5	8	8	196.33	224.84	28.51
5	9	40	224.84	299.96	75.12

Table 17: Comparing the obtained results from branch and bound algorithm.

The number of problem	Machines	Operations	Response		
			Firefly time (seconds)	CPLEX time (second)	ARE
1	36	2	203	194	0.04
2	48	11	230	250	0.08
3	10	3	436	450	0.03
4	22	13	434	442	0.01
5	4	14	1053	1100	0.04
6	6	4	1899	2100	0.09
7	28	5	4049	4100	0.01

6. Conclusion and Future Work

This article develops the flow-shop problem from its classic mode to the flow-shop problem with setup and sequence-dependent times. The objective function considered in this research is to minimize the completion time of the last job. Usually, in scheduling problems, the processing time of each job operation is assumed to be given and fixed. However, this point is not emphasized in the relevant literature. Therefore, studies in which time and cost are sequence-dependent are not considered. For this purpose, in the present study, similar jobs are within each order in which the time and cost of setup between two jobs from one order are ignored. Therefore, considered objective function is to minimize the total completion time or the completion time of the last job on the last machine. Also, the main contribution of this paper is determining a schedule for prioritizing orders in the flow-shop problem. For this purpose, an integer linear programming model for flow-shop scheduling problems is proposed, assuming sequence-dependent setups. Therefore, using a novel meta-heuristic as solution approach optimized the problem. For this purpose, using the firefly algorithm, the optimal order for each production unit that you want to do is determined. The algorithm considers the initial 100 vectors and generates 100 new vectors and, after merging all the answers, selects the best solution from the 100 solutions obtained. By sorting the answers based on quality, which is due to the completion time of the last machine, each one that has less time is placed at the beginning of the sequence. Therefore, the algorithm gives the vector of the optimal solution that the vector of the solution in our problem is the order of operations. Finally, the sequence results are determined according to the desired algorithm for 5 tasks considered for automobile parts. The results show that the completion time of job 1 is 1397.85; job 2 is 771.44; job 3 is 608.65; job 4 is 1163.87, and job 5 is 479.45. In order to show the validity of the obtained results, absolute relative error (ARE) has been used. For this purpose, the results between the mathematical model and meta-heuristic have been compared with each other, and a slight difference between the results has been observed, which can be ignored. For further research, it is suggested that the results be implemented with other optimization algorithms such as genetics and simulated annealing, which is also mentioned in the literature. The computational results are compared with the results of this research. The complexity of the problem in case of increasing the dimensions of the answer space is the most important limitation of this research. In contrast, the most important advantage of using the high-end response method is problem-solving. Given that there are indefinite parameters in the real world, it is recommended to provide a model based on indefinite parameters for model development.

Data Availability

Article data are available upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] C. H. Arnoldo and D. Candea, *Production and Inventory Management*, 2th, New York, 1984.

[2] K. R. Baker, *Elements of Sequencing and Scheduling*, 3th. University of Toronto bookstores, Canada, 1996.

[3] D. Philip, "Scheduling reentrant flexible job shops with sequence dependent setup times," MS Thesis, Montana State University, Bozeman MT 59717 USA, 2005.

[4] R. Mirataollahi olya, S. A. Shayannia, and M. Mehdi movahedi, "Designing a multiobjective human resource scheduling model using the tabu search algorithm," *Discrete Dynamics in Nature and Society*, vol. 2022, Article ID 5223535, 1 pages, 2022.

[5] X. Wang and T. E. Cheng, "An approximation scheme for two-machine flowshop scheduling with setup times and an availability constraint," *Computers & Operations Research*, vol. 34, no. 10, pp. 2894–2901, 2007.

[6] J. E. Schaller, J. N. Gupta, and A. J. Vakharia, "Scheduling a flowline manufacturing cell with sequence dependent family setup times," *European Journal of Operational Research*, vol. 125, no. 2, pp. 324–339, 2000.

[7] R. Ruiz, C. Maroto, and J. Alcaraz, "Solving the flowshop scheduling problem with sequence dependent setup times using advanced metaheuristics," *European Journal of Operational Research*, vol. 165, no. 1, pp. 34–54, 2005.

[8] R. Z. Ríos-Mercado and J. F. Bard, "An enhanced TSP-based heuristic for makespan minimization in a flowshop with setup times," *Journal of Heuristics*, vol. 5, no. 1, pp. 53–70, 1999b.

[9] R. Ruiz and T. Stutzle, "An Iterated Greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives," *European Journal of Operational Research*, vol. 187, no. 3, pp. 1143–1159, 2008.

[10] B. Ekşioğlu, S. D. J. P. Ekşioğlu, and P. Jain, "A tabu search algorithm for the flowshop scheduling problem with changing neighborhoods," *Computers & Industrial Engineering*, vol. 54, pp. 1–11, 2008.

[11] A. Allahverdi, C. Ng, T. Cheng, and M. Y. Kovalyov, "A survey of scheduling problems with setup times or costs," *European Journal of Operational Research*, vol. 187, no. 3, pp. 985–1032, 2008.

[12] A. N. Balaji, S. Porselvi, and N. Jawahar, "Particle swarm optimisation algorithm and multi-start simulated annealing algorithm for scheduling batches of parts in multi-cell flexible manufacturing system," *International Journal of Services and Operations Management*, vol. 32, no. 1, p. 83, 2019.

[13] E. Celik and D. Dal, "A novel simulated annealing-based optimization approach for cluster-based task scheduling," *Cluster Computing*, vol. 24, no. 4, pp. 2927–2956, 2021.

[14] A. Hashemi, H. Gholami, U. Venkatadri, A. A. Salameh, M. Jafari, and G. Abdul-Nour, "A novel approach to solve cell formation problems with alternative routing using particle swarm optimisation," *Transformations in Business and Economics*, vol. 21, no. 1, 2022.

[15] A. Brum, R. Ruiz, and M. Ritt, "Automatic generation of iterated greedy algorithms for the non-permutation flow shop scheduling problem with total completion time minimization," *Computers & Industrial Engineering*, vol. 163, Article ID 107843, 2022.

[16] K. Tamssaouet, S. Dauzère-Pérès, S. Knopp, A. Bitar, and C. Yugma, "Multi-objective optimization for complex flexible job-shop scheduling problems," *European Journal of Operational Research*, vol. 296, no. 1, pp. 87–100, 2022.

[17] M. Abolghasemian, A. P. Chobar, M. AliBakhshi, A. Fakhr, and S. Moradi, "Delay scheduling based on discrete-event simulation for construction projects," *Iranian Journal of Operations Research*, vol. 12, no. 1, pp. 49–63, 2021.

[18] A. Rashidi Komijan, P. Ghasemi, K. Khalili-Damghani, and F. HashemiYazdi, "A new school bus routing problem considering gender separation, special students and mix loading: a genetic algorithm approach," *Journal of Optimization in Industrial Engineering*, vol. 14, no. 2, pp. 23–39, 2021.

[19] S. Khanchehzarrin, M. Shahmizad, I. Mahdavi, N. Mahdavi-Amiri, and P. Ghasemi, "A model for the time dependent vehicle routing problem with time windows under traffic

conditions with intelligent travel times," *RAIRO - Operations Research*, vol. 55, no. 4, pp. 2203–2222, 2021.

[20] M. Rezaei Kallaj, M. Abolghasemian, S. Moradi Pirbalouti, M. Sabk Ara, and A. Pourghader Chobar, "Vehicle routing problem in relief supply under a crisis condition considering blood types," *Mathematical Problems in Engineering*, vol. 2021, Article ID 7217182, pp. 1–10, 2021.

[21] M. Momenitabar, Z. Dehdari Ebrahimi, M. Arani, J. Mattson, and P. Ghasemi, "Designing a Sustainable Closed-Loop Supply Chain Network Considering Lateral Resupply and Backup Suppliers Using Fuzzy Inference System," *Environment, Development and Sustainability*, pp. 1–34, 2022.

[22] 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*, pp. 1–28, 2022.

[23] S. R. Das and C. Canel, "An algorithm for scheduling batches of parts in a multi-cell flexible manufacturing system," *International Journal of Production Economics*, vol. 97, no. 3, pp. 247–262, 2005.

[24] M. Abolghasemian, A. G. Kanai, and M. Daneshmandmehr, "A two-phase simulation-based optimization of hauling system in open-pit mine," *Iranian Journal of Management Studies*, vol. 13, no. 4, pp. 705–732, 2020.