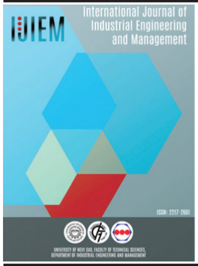
ISSN 2683-345X

journal homepage: http://ijiemjournal.uns.ac.rs/




International Journal of Industrial Engineering and Management

Volume 14 / No 2 / June 2023 / 122 - 137

Original research article

# A novel mathematical model for Integration of Production Planning and Maintenance Scheduling

I. Rastgar[a], J. Rezaeian[a,*], I. Mahdavi[a], P. Fattahi[b]

[a] *Department of Industrial Engineering, Mazandaran University of Science and Technology, Babol, Iran;*

[b] *Department of Industrial Engineering, Alzahra University, Tehran, Iran*

## ABSTRACT

The integration of production and scheduling of a shop floor, which has recently received attention both in practice and theory, is critical to achieving the optimal operational performance of production planning. The main novelty of this paper is the integration of the strategic decision with the tactical-operational decision to propose a new integrated production and scheduling mathematical model. In small instances, the model is solved by an epsilon-constraint method. To address the high complexity of large-scale problems, this study innovates new hybrid optimization algorithms. In this regard, MOPSO, NSGA-II, MOHS, and IMOHS are considered the optimization tools to solve the proposed model. To confirm their efficiency, an extensive comparison with individual algorithms is carried out by different multi-objective optimization metrics. Finally, some sensitivity analyses are performed to discuss some practical implications. The results show the advantages and effectiveness of the IMOHS in reporting the optimal Pareto for the proposed model compared to the other three algorithms.

## ARTICLE INFO

Article history:

Received October 9, 2022
Revised April 14, 2023
Accepted April 26, 2023
Published online May 10, 2023

Keywords:
Production planning and scheduling;
Hybrid flow-shop;
Imperfect maintenance;
IMOHS

*Corresponding author:
Javad Rezaeian
j.rezaiean@ustmb.ac.ir

## 1. Introduction

A more challenging issue is the development to handle an integrated production planning and scheduling problem. Production planning and scheduling are two central, related, and interconnected levels of decision-making in production systems. Due to different deadlines and goals, planning and scheduling are often handled sequentially separately, which leads to infeasible or suboptimal solutions.

In the literature, various types of integration of decision levels in a production environment have been studied. This article focuses on the coordination or integration of production and scheduling (see Fig. 1). Many companies manage these two operations independently with little or no integration which can result in missed opportunities for cost savings and improved customer service through optimized production and scheduling operations [1]. The integration of production and scheduling of a shop floor has recently received attention in both practice and theory, and it is critical to achieve optimal operational performance in production planning [2]. The commonly used planning and scheduling decision-making strategy generally follows a hierarchical approach, in which the planning problem is solved first to define the production targets, and the scheduling problem is solved next to meet these targets. The generated

Published by the University of Novi Sad, Faculty of Technical Sciences, Novi Sad, Serbia.

http://doi.org/10.24867/IJIEM-2023-2-328

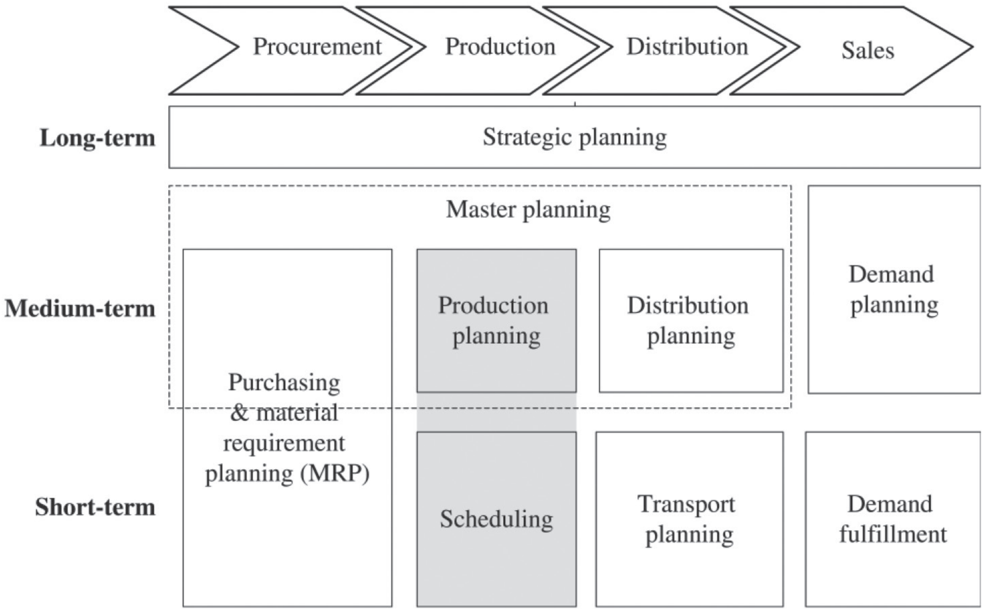



**Figure 1.** The supply chain matrix [3]

planning decisions might cause infeasible scheduling sub-problems. At the planning level, the effects of changeovers and daily inventories are neglected, which tends to produce optimistic estimates that cannot be realized at the scheduling level (i.e., a solution determined at the planning level does not necessarily lead to feasible schedules). Furthermore, the optimality of the planning solution cannot be ensured because the planning level problem might not provide an accurate estimation of the production cost, which should be calculated from detailed tasks determined by the scheduling problem.

Therefore, the main two motivations of this study are the following:

- Developing efficient methods for the integration of production and scheduling problems, and
- Obtaining good solutions to production planning that considers tactical and operational planning, aligned with the corporate objectives of the company.

From the production scheduling point of view, most literature assumed that machines are always available during the planning time horizon. However, in real manufacturing systems, this assumption is unreasonable. In recent decades, the relationship between production and maintenance has been identified as a conflict in management decisions, which has been examined in several recent studies [4]. However, if the production plan does not consider the expected maintenance period, interruptions resulting from maintenance interference or machine breakdown may lead to unsatisfied demand. This article considers an integrated multi-objective model with system availability for maintenance. The multi-objective model enables the decision-maker to obtain a compromise solution meeting at best two criteria: one related to maintenance activities, and the other to the production plan.

The production also involves various types of resources, e.g., plastics production requiring injection machines and molds. Each of these resources requires a specific set of maintenance activities. Failure to harmonize the maintenance schedules for these resources can lead to significant disruptions in planned production due to the non-availability of resources. Herein, this article considers mold and machine maintenance as critical resources in production [5].

In this article, an integrated production planning and scheduling model is developed for a hybrid flow shop environment, taking maintenance constraints into consideration. The contributions of this article to the literature are as follows:

(1) Strategic-Tactical-Operational Integration: The article proposes a new integrated production and scheduling mathematical model.
(2) Maintenance Policies: two maintenance policies, imperfect and perfect maintenance activity are considered in the model
(3) Metaheuristic Algorithm: The article presents a new metaheuristic algorithm to solve the problem and compares it with other multi-objective metaheuristic algorithms

Dauzere-Peres and Lasserre [6, 7] were among the pioneers in proposing an integrated planning and scheduling model that considers complex scheduling problems and the exact capacity of the shop floor. They propose a solution procedure that iterates between planning and a scheduling module. In the scheduling module, an optimal schedule is determined for fixed lot sizes, while, in the planning module, the best lot sizes are calculated for a fixed sequence of operations. This procedure allows a feasible production plan to be determined. Zhou et al [8] concurrently considered planning and scheduling decisions. They presented a MILP formulation to integrate resource allocation and production planning in multiproduct batch plants. Yan et al [9] addressed the multi-period production planning and scheduling problem with setup time and mixed batches. They presented a non-linear mixed-integer programming model, and applied an alternant iterative GA to solve it. Chu et al [10] linked the planning and scheduling problems via service level constraints. To solve the integrated problem, a hybrid method is developed, which iterates between a mixed-integer linear programming solver for the planning problem and an agent-based reactive scheduling method.

Han et al [11] proposed a mathematical model of bi-level integration production planning and scheduling problem in the manufacturing system. A hybrid solution method was presented that consists of PSO to solve the production planning problem level and a new heuristic method to solve the scheduling problem under fuzzy manufacturing conditions. In the integrated problem of production planning and scheduling, Bhosale and Pawar [12] developed the problem of operations allocation at the scheduling level with the assumption of waiting time in their model. They used the proposed GA to solve the problem. An integrated model of cell production, group scheduling, production, and PM in a dynamic cell production system was put forward by Alimian et al. [13]. The maintenance problem in this model determines the time of perfect non-periodic maintenance in the form of corrective maintenance operations. To solve the model, the Benders decomposition method was used in GAMS. Han et al. [11] developed an integrated two-level mathematical model of the problem of production planning and scheduling in the production system. They utilized the PSO optimization method to solve the problem at the production planning level and the new heuristic method to solve the problem at the scheduling level under fuzzy production conditions. Based on the literature, it can be concluded that the integrated production planning and scheduling problem with a maintenance strategy in hybrid flow-shop environments is a crucial topic in academia and industry. To the best of our knowledge, no research has yet focused on this area.

The remainder of the article is organized as follows: The mathematical formulation of the problem is presented in Section 2. In Section 3, the proposed algorithm, data generation, and parameter tuning are described in more detail. The experimental study conducted is explained in detail in Section 4. Finally, Section 5 concludes the paper.

## 2. Problem formulation

### 2.1 Problem description

We proposed a MIP model for the integrated production and scheduling problem in a hybrid flow shop with maintenance constraints, in a multi-product, multi-period production system. The simultaneous production and scheduling problem in a hybrid flow shop S-stage involves determining the production quantity and sequence of $n$ jobs in the series systems that require more than one machine for processing at least in one stage. The hybrid flow shop system is defined by processing stages. At each stage $s$, $s \in S$, has a set $M = 1$ of $M_s$ Machines. The set $J = 1$ of $n$ independent jobs must undergo processing on the machines. Each job $j$, $j \in J$, has its processing time and a due date $d_j$.

The assumptions, parameters, decision variables, and processing restrictions of the jobs considered in this article are as follows:

- Each machine can process only one job at a time, implying that no machine can perform more than one operation simultaneously.
- In the HFS problem, each stage has at least one machine, and at least one stage must have more than one machine.
- The model considers the time required to prepare a maintenance facility, as well as the duration of maintenance activities.
- The problem is both multi-product and multi-period. Each product has a specific delivery time.
- All jobs are available for processing at time zero.
- Each job can only be performed on one machine with one mold.

- Each job will be processed in a serial batch according to its order quantity, with the possibility of splitting the batch.
- Two maintenance policies for machines are considered: imperfect and perfect maintenance activity.
- Every job is processed independently of others and every machine operates independently of the other machines.
- When a maintenance activity is carried out on a machine or mold, processing operations are halted on that machine.
- The maintenance policy for the job molds is based on preventive replacement; which means that each time a job begins processing; the mold will be replaced with a new one.
- Each operation requires one machine that is selected from a set of available machines.
- The processing time of an operation on machine $k$ is pre-determined.
- Preemption is permitted, i.e., Once an operation is commenced, it can be interrupted.
- Problem parameters are deterministic.

## 2.2 Mathematical programming

The important notations used throughout this article are defined as follows:

*Index list*

| | |
|---|---|
| $i, j$ | Job |
| $K$ | The sequence position in the product's process schedule |
| $S$ | Stage number |
| $M$ | Machine |
| $T$ | Planning period |

*Variable list*

| | |
|---|---|
| $x_{tsjm}$ | $\begin{Bmatrix} 1 \text{ if job } j \text{ is processed on machine } m \text{ at period } t; \\ 0 \text{ otherwise} \end{Bmatrix}$ |
| $y_{ikjk'm}$ | $\begin{Bmatrix} 1 \text{ if posion } k \text{ at job } i \text{ is processed before posion } k' \text{ at job } j \text{ on machine } m; \\ 0 \text{ otherwise} \end{Bmatrix}$ |
| $h_{sm}$ | $\begin{Bmatrix} 1\ if\ Imperfect\ maintenance\ occurs\ at\ machine\ m\ in\ stage\ s; \\ 0 \text{ otherwise} \end{Bmatrix}$ |
| $g_{sm}$ | $\begin{Bmatrix} 1 \text{ if perfect maintenance occurs at machine } m \text{ in stage } s; \\ 0 \text{ otherwise} \end{Bmatrix}$ |
| $Q_{tj}$ | Quantity of product $j$ produced at period $t$ |
| $\mathrm{RFI}_{sm}$ | Imperfect maintenance reliability of machine $m$ at stage $s$ |
| $\mathrm{RFP}_{sm}$ | Perfect maintenance reliability of machine $m$ at stage $s$ |
| $\mathrm{Re}\ l_{sm}$ | Reliability of machine $m$ at stage $s$ |
| $b_{tj}$ | Backorder of product $j$ at period $t$ |
| $I_{tj}$ | Inventory of product $j$ at period $t$ |
| $c_t^{\max}$ | Maximum completion time of job $j$ at period $t$ |
| $c_{tj}^{late}$ | Lateness of the $j^{th}$ job at period $t$ |
| $c_{tsm}^{"}$ | End time of machine $m$ in stage $s$ in period $t$ |

Parameter list

| | |
|---|---|
| $D_{tj}$ | External demand for the $j^{th}$ product in period $t$ |
| $dd_{tj}$ | Due date of product $j$ at period $t$ |
| $\mathrm{cost}_{RFPsm}$ | Cost of perfect maintenance of machine $m$ at stage $s$ |
| $HC_{tj}$ | Holding cost of product $j$ at period $t$ |
| $BC_{tj}$ | Backorder cost of product $j$ at period $t$ |
| $PC_{tj}$ | Production cost of product $j$ at period $t$ |
| $\mathrm{cost}_{RFIsm}$ | Cost of imperfect maintenance the machine $m$ at stage $s$ |
| $P_{jms}$ | Processing time of job $j$ at machine $m$ at stage $s$ |
| $pp^{/}_{jkms}$ | Processing time of position $k$ of job $j$ on machine $m$ at stage $s$ |
| $st_{sjkjk^{/}m}$ | Sequence-dependent setup time of switching from position $k^{/}$ job $j$ to position $k$ job $i$ on machine $m$ |
| $zp_{sm}$ | Perfect maintenance time of machine $m$ at stage $s$ |
| $zi_{sm}$ | Imperfect maintenance time of machine $m$ at stage $s$ |
| $zm_{sjm}$ | Mold replacement time of job $j$ in machine $m$ at stage $s$ |
| $zs_{jm}$ | Maximum time to use the $j^{th}$ mold of machine $m$ |
| $M$ | A very large number |
| $zt_{sm}$ | Repair time of machine $m$ on stage $s$ |
| $pro_{sm}$ | Probability of machine failure at stage $s$ |
| $UBRFI_{sm}$ | Upper bound of imperfect maintenance reliability of machine $m$ at stage $s$ |
| $LBRFP_{sm}$ | Lower bound of perfect maintenance reliability of machine $m$ at stage $s$ |
| $UBRFI_{sm}$ | Upper bound of imperfect maintenance reliability of machine $m$ at stage $s$ |
| $LBRFP_{sm}$ | Lower bound of perfect maintenance reliability of machine $m$ at stage $s$ |
| $capC^{\max}$ | Capacity of maximum time of production in all periods |

The mixed-integer linear formulation for the problem under study is presented below:

$$\min F_1 = \sum c_t^{\max} + \sum_t \sum_j c_{t,j}^{late}$$
$$\min F_2 = \sum_s \sum_m \mathrm{cost}_{RFPsm} \times RFP_{sm} + \sum_s \sum_m \mathrm{cost}_{RFIsm} \times RFI_{sm}$$
$$+ \sum_t \sum_j PC_{tj} \times Q_{tj} + \sum_t \sum_j HC_{tj} \times I_{tj} + \sum_t \sum_j BC_{tj} \times B_{tj} \tag{1}$$

S.T:

$$C_{tsjkm} \geq h_{sm} \times 0.5 \times pro_{sm} \times zt_{sm} + pp^{/}_{jkms} + st_{sjkj^{/}m}$$
$$+ \sum_{s^{/}\leq 1} \sum_{m^{/}} zp_{s^{/}m^{/}} \times g_{s^{/}m^{/}} + zi_{s^{/}m^{/}} \times h_{s^{/}m^{/}} - M \times (1 - y_{tsikjkm})$$
$$\forall s = 1, j = i, m \leq M(s), K = 1, \forall t, i, j \tag{2}$$

$$C_{tsjk^{/}m} \geq C_{tsikm} + pp^{/}{}_{jk^{/}ms} + st_{sjkj^{/}m} - M(1 - y_{tsikjk^{/}m})$$
$$\forall K^{/} = 2, K = 1, j = i, m \leq M(s), \forall t, s, i, j \tag{3}$$

$$C_{ts^{/}jkm} \geq st_{s^{/}ikjk'm} + CPP_{ts^{/}m} + h_{s^{/}m} \times 0.5 \times pro_{sm} \times zt_{sm}$$
$$+ pp^{/}{}_{jkms^{/}} - M(1 - y_{ts^{/}ikjk'm})$$
$$\forall K \leq K^{/}, m \leq M(s), \forall t, s, s', i, j \tag{4}$$

$$C_{ts^{/}jk^{/}m} \geq C_{ts^{/}-1j} + pp^{/}{}_{jk^{/}ms^{/}} + st_{s^{/}ikjk^{/}m} - M(1 - y_{ts^{/}ikjk^{/}m})$$
$$\forall j \neq i, m \leq M(s^{/}),$$
$$\forall t, s^{/}, i, j, k, k^{/} \tag{5}$$

$$CC_{tsj} \geq C_{tsjkm} \quad \forall t, s, j, k, m \tag{6}$$

$$CPP_{tsm} \geq \sum_{s^{/}} \sum_{m^{/}} zp_{s^{/}m^{/}} \times g_{s^{/}m^{/}} + zi_{s^{/}m^{/}} \times h_{s^{/}m^{/}}$$
$$+ \sum_{m^{//}} zp_{sm^{//}} \times g_{sm^{//}} + zi_{sm^{//}} . h_{sm^{//}}$$
$$\forall t, s \succ 1, m \tag{7}$$

$$CPP_{tsm} \geq CC_{t(s-1)j} + zp_{sm} \times g_{sm}$$
$$+ zi_{sm} \times h_{sm} - M(1 - y_{tsjkjkm})$$
$$\forall k = 1, t, s \succ 1, m, j \tag{8}$$

$$LBRFP_{sm} \leq RFP_{sm} \leq UBRFP_{sm} \ \forall s, m \tag{9}$$

$$LBRFI_{sm} \leq RFI_{sm} \leq UBRFI_{sm} \ \forall s, m \tag{10}$$

$$\mathrm{Re}l_{sm} \leq RFP_{sm} - M(g_{sm} - 1) \ \forall s, m \tag{11}$$

$$h_{sm} + g_{sm} = 1 \ \forall s, m \tag{12}$$

$$\sum_{m \leq M(s)} X_{tsjm} = 1 \quad \forall t, s, j \tag{13}$$

$$y_{tsikjk^{/}m} = 0 \quad \forall K \succ K^{/}, m \leq M(s), t, s, i, k, k^{/}, j, m \tag{14}$$

$$\sum_i \sum_{k=1} y_{tsikikm} = 1 \quad \forall, m \leq M(s), t, s, m \tag{15}$$

$$\sum_j \sum_{k^{l}} y_{tsikjk^{l}m} \leq 2 \quad \forall m \leq M(s), k = 1, \forall t, s, i, \tag{16}$$

$$y_{tsikikm} = 0 \quad \forall K = 2, m \leq M(s), t, s, i, k, m \tag{17}$$

$$\sum_i \sum_k y_{tsikjk^{l}m} = X_{tsjm} \quad \forall m \leq M(s), t, s, k^{/}, j, m \tag{18}$$

$$\sum_i \sum_k y_{tsjk^{l}ikm} \leq X_{tsjm} \quad \forall m \leq M(s), t, s, k^{/}, j, m \tag{19}$$

$$\sum_j \sum_{k^{l}} y_{tsikjk^{l}m} \leq 1 \quad \forall m \leq M(s), k = 1, \forall t, s, k^{/}, k, j, m \tag{20}$$

$$PP_{jkms} \geq 1 \ \forall k = 1, s, m, j \tag{21}$$

$$\sum_k PP_{jkms} = P_{jms} Q_{tj} \quad \forall t, s, m, j \tag{22}$$

$$PP^{l}_{jkms} = PP_{jkms} + zm_{sjm} \ \forall s, k, j, m \tag{23}$$

$$PP^{/}{}_{jkms} = PP_{jkms} \quad \forall s, k, j, m \tag{24}$$

$$PP_{jkms} \leq zs_{jm} \quad \forall s, k, j, m \tag{25}$$

$$CC^{\max}_{tj} \geq C_{tsjkm} \quad \forall t, s, k, j, m \tag{26}$$

$$C^{\max}{}_t \geq CC^{\max}_{tj} \quad \forall t, j \tag{27}$$

$$C^{late}{}_{tj} \geq CC^{\max}_{tj} - dd_{tj} \quad \forall t, j \tag{28}$$

$$\mathrm{Re}l^{/}_{sm} = \mathrm{Re}l^{i}_{sm}(1 - r_2^{\sum_j X_{tsjm}}) \quad \forall t, s, m \tag{29}$$

$$I_{tj} - B_{tj} = Q_{tj} - D_{tj} - B_{(t-1)j} + I_{(t-1)j} \ \forall t \succ 1, j \tag{30}$$

$$C_t^{\max} \leq capC^{\max} \quad \forall t \tag{31}$$

The first objective function (F1) minimizes the sum of the maximum compellation time and lateness time of products. The second objective function (F2) minimizes the costs associated with imperfect and perfect maintenance, production, inventory, and backorder. Eqs. 2-8 determine the completion time of jobs. Eq. 2 determines the completion time of a first job at the first stage. Eq. 3 computes the completion time of a job if there should be only one machine at each station. Eq. 4 calculates the completion time of a job from one stage to the next stage. Eqs. 8 and 9 ensure that perfect maintenance reliability of machines and imperfect maintenance reliability of machines are considered lower bound and upper bound. Eq. 11 expresses the reliability of machines. Eq. 12 ensures that the maintenance activity is performed on one machine at a time, either perfect maintenance or imperfect maintenance. Eqs. 13-20 determine the sequence of products in a production environment in the hybrid flow shop. Eq. 13 guarantees that each job is processed on at least one machine. Eq. 14 shows that the second position of the job does not start earlier than the first position. Eqs. 21-25 compute the completion time of job j of machine m in stage s. Eqs. 26-28 determine the lateness of each job. Eq. 29 expresses the final reliability of machines while considering the volume of jobs processed on the machine. Finally, Eq. 30 is the usual flow balance constraint to satisfy the demand for each product in each period.

## 3. Proposed optimization algorithms

A major issue regarding integrated production and scheduling problems is selecting the solution method. Due to the complexity of problems, in most studies, different heuristic and metaheuristic methods are applied to find solutions. Since the respective problem is NP-hard, to solve a problem with larger dimensions by exact procedures such as simplex, dynamic programming, or branch and bound, one may not be able to reach the optimum answer within a reasonable time. Hence, due to the intrinsic complexity of discrete optimization problems, particularly PS problems, metaheuristic methods yield better performance in solving the problem and providing an acceptable solution within an acceptable time [14]. Multi-objective problems are another type of operations research problem that includes a vector of objectives instead of a single objective. The main goal of multi-objective optimization techniques is to determine a set of optimal solutions, especially when the objectives are conflicting.

### 3.1 ε-constraint method

Exact solution methods are used to solve small-scale problems. One of the most important exact solution methods is the Epsilon constraint method, which is frequently employed to solve multi-objective problems.

In this method, one of the objectives is optimized at each stage, while the other objectives are considered as constraints with the upper bound of Epsilon (ε). this method converts the multi-objective optimization problem into a single-objective one.

$$\min f_1(x) \qquad x \in X \tag{32}$$

$$f_2(x) \le \varepsilon_2$$
$$\ldots$$
$$f_n(x) \le \varepsilon_n \tag{33}$$

The steps of the Epsilon constraint method are:

(1) In each step, one of the objective functions is selected as the main objective; the problem is solved with this objective; and the optimal value for this objective function is obtained.
(2) Other objective functions become sub-objectives; the distances between the optimal values of the sub-objectives are divided into a predetermined number; and a table for these values is created.
(3) Other objective functions become extraneous objectives; intervals between optimum values of extraneous objectives are divided into a prespecified number; and $\varepsilon_2, \ldots, \varepsilon_n$ values are detected.
(4) At any given time, the major objective function is solved for each value $\varepsilon_2, \ldots, \varepsilon_n$ to yield Pareto solutions [15].

### 3.2 NSGA-II

NSGA-II is the latter version of the popular "non-dominated sorting GA" developed in [16] to solve non-convex and non-smooth single and multi-objective optimization problems. Compared to NSGA, it is a useful algorithm that has an improved mating mechanism dependent upon the crowding distance. In the NSGAII algorithm, there are some parameters, including population size ($N_{pop}$), mutation rate ($P_m$), crossover rate ($P_c$), and maximum iteration ($max_{it}$). Population initialization is conducted as before. Initially, a zero level is allocated to all non-dominated individuals. During the elimination of the individuals from the population, the primary non-dominated solutions are allocated to level one. This procedure continues until all solutions are allocated to a non-domination level. The parent's selection process is carried out using binary tournament selection based on a less rank and greater crowding distance. The next step includes off-spring generating from the selected population using crossover and mutation operators, which will be discussed subsequently. Finally, the present off-springs and population are once again sorted based on the non-domination rule, and only the best individuals remain as the population size (*P*).

### 3.3 MOPSO

Swarm intelligence (SI) is mainly defined as the behavior of natural/artificial self-organized, decentralized systems. Swarms interact locally with each other or with external agents, i.e., the environment, and can be in the form of bird flocks, ants, bees, etc. The PSO algorithm, Introduced by Srinivas and Deb [17] to optimize continuous nonlinear functions, the PSO algorithm is an optimization technique based on the bird migration pattern. Moreover, the best bird (with the smallest distance from the food) is called the global best ($G_{best}$), and the best position of a bird ever is called the particle best ($P_{best}$)[18]. MOPSO was proposed by Moore et al [19] to optimize more than one objective function. In MOPSO, instead of a single solution, a set of solutions are determined, also known as the Pareto optimal set. There are five parameters explained as follows: 1) The number of iterations, which is the maximum number of iterations before the best Pareto frontier is achieved; 2) the number of particles, which is the number of initial solutions for the next iteration (similar to the population size in the NSGA-II); 3) *W*, which is known as the inertia weight and determines the willingness of a particle to keep its velocity from the previous iteration; 4) $C_1$, which is known as the personal influence weight and determines the willingness of a particle to move to one of its best positions; and 5) $C_2$, known as the social influence weight, determining the willingness of a particle to move to one of the Pareto frontier solutions.

### 3.4 Harmony search algorithm

The harmony search algorithm (HS) is considered to be one of the simplest metaheuristic methods. The optimal responding search process in optimization problems is inspired by the playing process of an orchestra. This solution method has been presented by [20]. This algorithm is composed of three stages: (1) providing the algorithm parameters and harmony memory with initial values, (2) generating a New Harmony vector, and (3) updating the algorithm memory. A new harmony vector can be generated by one of the following methods:

(1) Considering solutions available in the memory: This method ensures that the best solutions remain in the memory during the optimization process. This operator is controlled by the harmony memory consideration rate (*HMCR*). The smaller this rate, the slower the algorithm's convergence due to the existence of few selected harmonies; the larger this rate, the more available harmony in the memory is selected, and the algorithm is entrapped in the local optimum. The application of this operator is similar to the elite operator in a GA.

(2) Step adjustment (or pitch adjustment): Pitch adjustment means a change in the structure of frequencies, in which a rate, namely pitch adjustment rate, is used to control this operator. The larger this rate, the greater the diversity in the algorithm; this increase makes the algorithm operate as a random search method. Using this operator is equivalent to the neighborhood solutions generation in optimization algorithms and similar to a mutation in a GA.

(3) Random selection: It randomly generates solutions and adds them to the population.

#### 3.4.1 Multi-objective harmony search algorithm (MOHS)

The steps for implementing the MOHS are:

(1) Set the initial values of the problem and the parameters of the algorithm.

(2) Create a harmonic memory and set its initial values. All the variables must be randomly quantified in their maximum and minimum ranges.

(3) Evaluate the objective functions for each solution vector in the harmony memory.

(4) Create a new harmonic memory according to the previously mentioned description.

(5) Update the harmonic memory and select the best harmonic memory from the combination vectors.

(6) Repeat the fourth and fifth steps until the end condition is met or the repetitions are completed.

Harmony memory upgrades in HS for multi-objective problems differ from simple HS. In this study, Pareto optimal solutions are determined using the idea of the NSGA method. The new harmony memory described above and the existing harmony memory comprise two *HMS* solution vectors (integrated harmony memory). Then, the non-dominated ranking and sorting approach is used for it. When ranking is performed for all solution vectors in the integrated harmony memory, a diversity rank is devoted to solution vectors and used for a non-dominated batch of a crowded distance. The crowded distance expresses the distance among solution vectors surrounded by a specific solution vector. Its scale is the mean distance between two solution vectors on the two sides of a solution vector according to each objective function. Therefore, the best harmony memory (equivalent to *HMS*) is selected from the integrated harmony memory [21].

### 3.4.2 Improved multi-objective harmony search algorithm (IMOHS)

This algorithm was proposed after extensive research on the methods reviewed in the literature on the HS method. In this method, innovations in parameter setting and changes in the trend of *HMCR* and *PAR* parameter values during algorithm iterations are presented compared to the basic HS.

Searching for optimal solutions is a more efficient way to consider the diversity of the entire solution space at the beginning of the search (exploration), that is, to identify the entire solution space and move more toward the spaces that are most likely to be optimized. Then, two types of optimization may be encountered: local optimization and global optimization. If the algorithm becomes trapped in the local optimum, it may be separated from that area through local search techniques. The solutions reach the objective value faster. Accordingly, the HS algorithm may identify the high functional areas in the solution space within an appropriate time, but it is not efficient in the implementation of local search in the hybrid optimization problems, and such cases are situated in the local optimum. A technique used for controlling this problem is a restart phase. This technique involves applying a shock to the location where the algorithm is presently situated in the problem-solving space, increasing its dispersal. In this method, when the best value is not improved after some iterations, it acts as follows: (1) The harmony memory is sorted in ascending order based on the best objective function value. (2) The first memory list solutions are sorted and stored equal to the *restart phase consideration rate%*; then, for (1– (*Restart phase consideration rate*)) of the remaining memory, it selects a half by a one-point mutation operator on the regenerate restart phase rate and determines the other half randomly based on the range of each solution. In IMOHS, there are some parameters, including the number of algorithm memory vectors (*HMS*), minimum pitch adjustment rate ($PAR_{min}$), maximum pitch adjustment rate ($PAR_{max}$), minimum memory consideration rate ($hmcr_{min}$), and maximum memory consideration rate ($hmcr_{max}$). In the proposed algorithm, there are two types of parameter adjustment: dynamic adjustment and sequential adjustment based on the Taguchi method. Pitch adjustment rate parameters (*PAR*) and the *HMCR* are varied linearly and dynamically during the search process presented by a mathematical formula in Eqs. (34), (35). In these parameters, different values of parameters are appropriate during the search process.

$$PAR(t) = PAR_{\max} - \frac{PAR_{\max} - PAR_{\min}}{T} \times t \tag{34}$$

$$HMCR(t) = HMCR_{\min} + \frac{HMCR_{\max} - HMCR_{\min}}{T} \times t \tag{35}$$

## 3.5 Sample problems

To evaluate and compare the performance of the metaheuristic algorithms, 10 problems with varying numbers of jobs, machines, stages, and periods are considered. Table 1 presents the dimensions of the instances. Parameters were randomly generated within the ranges defined in Table 2. The sample problems were designed according to the reference provided [22].

## 3.6 Solution representation

One of the most important decisions in the coding of the metaheuristic method is how to represent problem solutions and how to transform them into real solutions available in the problem structure. The solution presentation must be such that it decreases the costs and time of using the algorithm. In this article, the vector is composed of four sections (Fig. 2). The first section is related to the part sequence jobs on the machines available in site one at stage one. The second section is related to the machine sequence. The third section of the chromosome shows the stage sequence, and the fourth section belongs to the part sequence jobs. Each component is obtained according to random values in [0, 1],

The chromosome is encoded from the permutation of jobs by the largest position value (LPV) rule (Fig. 3).

**Table 1.** Dimensions of each instance

| | | | Number of instances | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Indices | $m$ | Number of machines | 2 | 3 | 4 | 5 | 6 | 6 | 7 | 7 | 10 | 15 |
| | $s$ | Number of stages | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 4 | 6 |
| | $i, j$ | Number of jobs | 3 | 20 | 40 | 50 | 80 | 100 | 120 | 150 | 200 | 500 |
| | $t$ | Number of periods | 1 | 2 | 3 | 3 | 3 | 5 | 5 | 5 | 5 | 5 |

**Table 2.** Ranges of parameters for generating random instances

| Parameter | Parameter range | Parameter | Parameter range | Parameter | Parameter range |
|---|---|---|---|---|---|
| $D_{tj}$ | ~ U[1,5] | $BC_{tj}$ | ~ U[4000,5000] | $zm_{sjm}$ | ~ U[1,6] |
| $dd_{tj}$ | ~ U[400,800] | $PC_{tj}$ | ~ U[12,40] | $zs_{jm}$ | ~ U[1,5] |
| $\text{cost}_{RFPsm}$ | ~ U[12000,13000] | $\text{cost}_{RFIsm}$ | ~ U[8000,12000] | $zt_{sm}$ | ~ U[2,4] |
| $HC_{tj}$ | ~ U[20,24] | $P_{jms}$ | ~ U[20,35] | $pro_{sm}$ | ~ U[0.4,.09] |
| $pp^{/}_{jkms}$ | ~ U[20,35] | $zp_{sm}$ | ~ U[1,5] | $UBRFI_{sm}$ | ~ U[0.8,0.85] |
| $st_{sjkj^{/}m}$ | ~ U[10,15] | $zi_{sm}$ | ~ U[1,5] | $LBRFP_{sm}$ | ~ U[0.6,0.7] |
| $UBRFP_{sm}$ | ~ U[0.8,0.9] | $LBRFI_{sm}$ | ~ U[0.5,0.65] | $capC^{\max}$ | 1000 |

| Job(i) | | | | | | Machine(m) | | | Stage(s) | | Job(j) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.3 | 0.2 | 0.5 | 0.8 | 0.1 | 0.4 | 0.1 | 0.8 | 0.4 | 0.6 | 0.1 | 0.6 | 0.4 | 0.2 | 0.3 | 0.1 | 0.5 |

**Figure 2.** Chromosome representation

| Job(i) | | | | | | Machine(m) | | | Stage(s) | | Job(j) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 3 | 6 | 1 | 2 | 5 | 2 | 3 | 1 | 1 | 2 | 1 | 6 | 2 | 4 | 3 | 5 |

**Figure 3.** Chromosome representation after applying the LPV rule

## 3.7 Design of Experiments (DoE)

Since the parameters of an optimization algorithm play a key role in the quality of the solution, the Taguchi method is used for tuning. An optimum DoE provides the data required for analysis and achievement of optimal conditions with the least number of experiments. to determine the significant factors affecting a response (here, the solution), Taguchi developed a special type of fractional factorial experiment to reduce the large number of experiments required in a full factorial experiment [23].

As there are three candidate levels of each parameter, altogether there are 81 parameter hybrids for implementation of each problem. Considering the sample number of problems (10) and three iterations for each experiment, the total number of experiments will be 30×3×81 = 7290. Specifying the best hybrid by this number of experiments requires a lot of time and is not reasonable. Thus, the Taguchi DoE technique is used.

### 3.7.1 Adjustment of factors and levels of each factor of the model-solving algorithm

In the first step, to implement the Taguchi method and adapt the appropriate orthogonal array, it is necessary to compute the required degrees of freedom. In this problem, one degree of freedom for the total mean and two degrees of freedom for three-level factors are required. Thus, an array that includes at least seven lines must be selected. Considering the standard orthogonal Taguchi arrays, it is concluded that these conditions are applicable in orthogonal array L9, and considering the level of the factor, array L9 is selected. The total number of problem implementations using the Taguchi method will be 30×3×9=810 times, whereas utilizing the full factorial DoE would require 7290 implementations. This means that 6480 experiments have been saved in terms of time and cost.

In the IMOHS, there are four two-level factors (minimum pitch adjustment rate, maximum pitch adjustment rate, minimum algorithm memory consideration rate, and maximum algorithm consideration rate) and a three-level factor (harmony memory size). The factors and their levels are listed in Table 3. In this problem, one degree of freedom for the total mean, 2 degrees of freedom for the three-level factor, and 1 degree of freedom for each two-level factor are required. Therefore, the total required degree of freedom would be equal to 1+ (1×4) +2=7. Thus, an array that includes at least seven lines must be selected.

Considering the standard orthogonal Taguchi arrays, it is concluded that these conditions are applicable in orthogonal array L12, and array L12 is selected based on the level of the factors.

### 3.7.2 Selection of optimum factors of solution algorithms

In each experiment implementation, the objective function obtained must be converted according to the Taguchi method and proportional to the signal-to-noise ratio, and analyzed based on the S/N ratio variations. The objective of each implementation is to minimize the objective function; thus, the signal-to-noise type is selected for Eq. (36).

$$S/N_s = -10\log\left(\frac{1}{n}\sum_{i=1}^{n} y_i^2\right) \tag{36}$$

In this study, considering the selected S/N ratio, proportional to the nature of this problem, the lowest S/N ratio for each factor in each algorithm is selected as the optimum factor. All the results of solution algo-

**Table 3.** Factors and their levels in the model-solving algorithm

| Algorithm | Parameter | Description of each parameter | Levels | | | |
|---|---|---|---|---|---|---|
| | | | Number | Size of each level | | |
| MOPSO | $N_{par}$ | Number of particles | 3 | 10 | 20 | 30 |
| | $W$ | Inertia weight | 3 | 0.2 | 0.5 | 0.75 |
| | $C_1$ | Personal influence coefficient | 3 | 0.5 | 0.75 | 0.9 |
| | $C_2$ | Social influence coefficient | 3 | 0.2 | 0.5 | 0.75 |
| | $It_m$ | Maximum iteration | 1 | 100 | | |
| NSGAII | $N_{pop}$ | The population size | 3 | 15 | 30 | 50 |
| | $P_m$ | Mutation rate | 3 | 0.2 | 0.3 | 0.4 |
| | $P_c$ | Crossover rate | 3 | 0.5 | 0.75 | 0.95 |
| | $It_n$ | Maximum iteration | 1 | 100 | | |
| MOHS | $HMS$ | Harmony memory size | 3 | 5 | 10 | 15 |
| | $HMCR$ | Harmony memory consideration rate | 3 | 50% | 80% | 99% |
| | $PAR$ | Pitch adjustment rate | 3 | 10% | 50% | 90% |
| | $BW$ | Band width | 3 | 0.2 | 0.5 | 0.99 |
| IMOHS | $HMS$ | Harmony memory size | 3 | 5 | 10 | 15 |
| | $HMCR_{min}$ | Minimum of harmony memory consideration rate | 2 | 20% | 50% | |
| | $HMCR_{max}$ | Maximum of harmony memory consideration rate | 2 | 80% | 99% | |
| | $PAR_{min}$ | Minimum of pitch adjustment rate | 2 | 20% | 40% | |
| | $PAR_{max}$ | Maximum of pitch adjustment rate | 2 | 50% | 90% | |

rithms implemented by Taguchi DoE for parameters' adjustment are presented as the S/N ratio in Figure 4.

Table 4 lists the best factors for the final implementation of algorithms to solve the model.

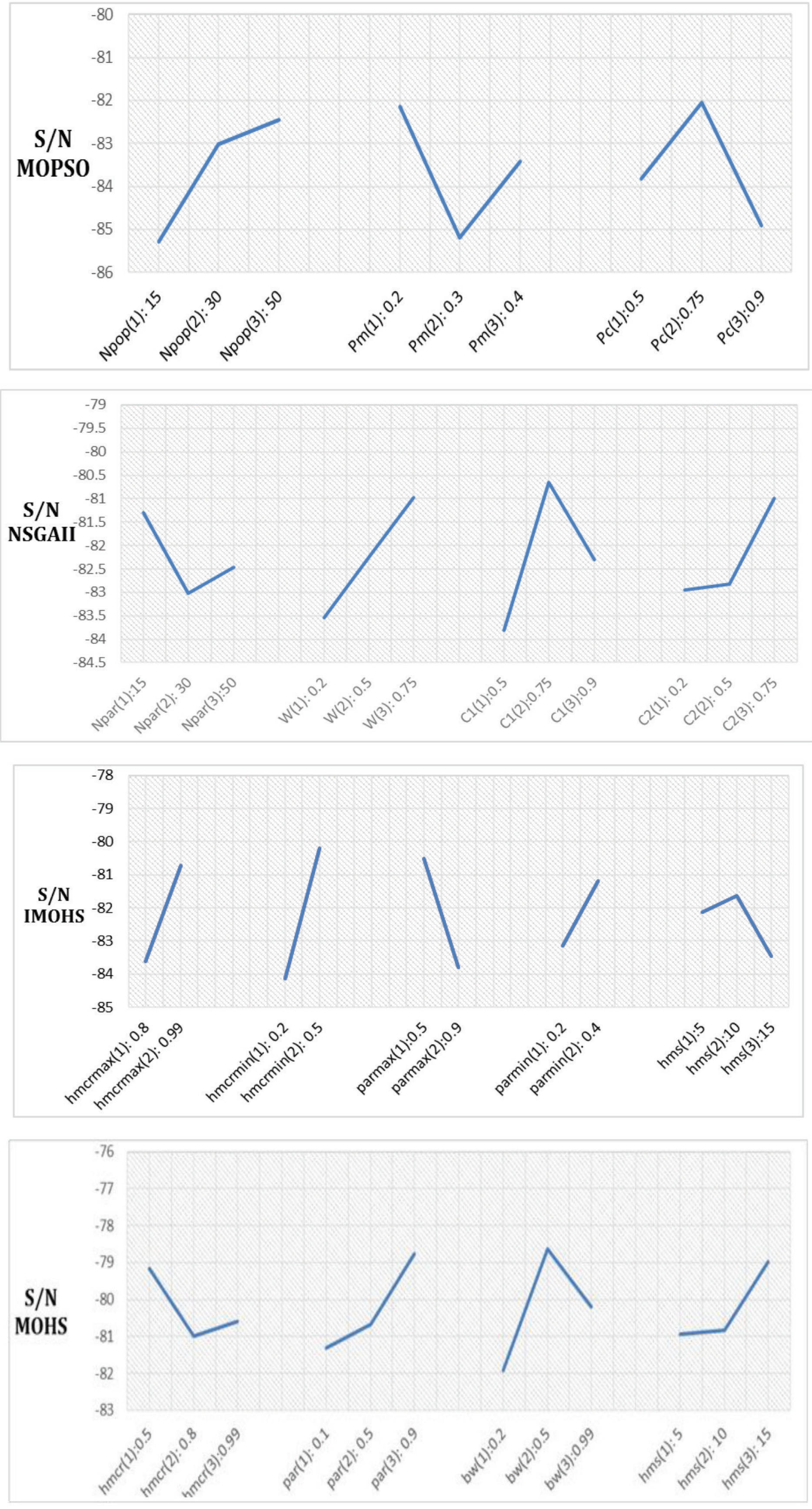



**Figure 4.** Mean S/N diagram for each level of the algorithms' factors

**Table 4.** Optimum operators of the algorithms

| Optimum level | Parameter | Algorithm |
|---|---|---|
| MOPSO | 30 | $N_{par}$ |
| | 0.2 | $W$ |
| | 0.5 | $C_1$ |
| | 0.2 | $C_2$ |
| | 100 | $It_m$ |
| NSGAII | 15 | $N_{pop}$ |
| | 0.3 | $P_m$ |
| | 0.9 | $P_c$ |
| | 100 | $It_n$ |
| MOHS | 5 | $HMS$ |
| | 0.8 | $HMCR$ |
| | 0.1 | $PAR$ |
| | 0.2 | $BW$ |
| IMOHS | 15 | $HMS$ |
| | 0.2 | $HMCR_{min}$ |
| | 0.8 | $HMCR_{max}$ |
| | 0.2 | $PAR_{min}$ |
| | 0.9 | $PAR_{max}$ |

## 4. Experimental study

To evaluate the algorithms' performance against the exact solution, the problems were solved using the GAMS commercial program (CPLEX solver). To this end, GAMS is executed for small instances and the outputs are presented in Table 5. The GAMS found the optimal solution for the first test problem; however, due to the multitude of decision variables and problem complexity, it could not solve large instances even after 32400000 seconds of computational time. Therefore, algorithms that can solve the model rapidly and with reasonable costs are proposed to tackle the large-sized problem since it is NP-hard. Initially, the sample problems generated were solved multiple times by each algorithm, utilizing the optimum parameter values obtained through the Taguchi method.

There are some measures defined for evaluating multi-objective metaheuristics algorithms. to compare the results achieved by the algorithms. some comparison metrics of metaheuristic algorithms were introduced by Tavakkoli-Moghaddam, R et al. [24]. Herein, four performance metrics were considered:

- The number of Pareto solutions (NPS): This measure presents the number of Pareto optimal solutions in each algorithm. The higher the NPS of an algorithm, the more utility it shows.
- Mean ideal distance (MID): The mean of Pareto solutions' distance from the coordinate origin (zero) is calculated by this metric. Algorithms with a smaller MID index are more effective.

$$MID = \frac{\sum_{i=1}^{n} c_i}{n} \tag{37}$$

$$c_i = \sqrt{f_{1i}^2 + f_{2i}^2} \tag{38}$$

$f_{1i}$: First objective function $f_{2i}$: Second objective function

- Spacing metric (SNS): This metric presented enables us to measure the uniformity of the point spread within the solution set. A smaller value for this metric indicates a better algorithm. This metric is given by Eq. (39):

$$y = \left[\frac{1}{N-1} \times \sum_{i=1}^{N}\left(MID - c_i\right)^2\right]^{\frac{1}{2}} \tag{39}$$

- Quality index (QM): This index compares the quality of the Pareto solutions obtained by each algorithm. It ranks all Pareto solutions obtained by each algorithm and determines what percentage of the best solutions belong to each algorithm. The higher the percentage, the better the algorithm quality.

**Table 5.** Result of different approaches for test problems in the small test problem

| Test problems | ∈-c | | | | MOHS | | | | IMOHS | | | | MOSPO | | | | NSGA-II | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NPS | MID | SNS | T(s) | NPS | MID | SNS | T(s) | NPS | MID | SNS | T(s) | NPS | MID | SNS | T(s) | NPS | MID | SNS | T(s) |
| 1 | **2** | **6881** | **716** | **4.4** | **2** | **6881** | **716** | **2.9** | **2** | **6881** | **716** | **2.6** | **2** | **6881** | **716** | **2.9** | **2** | **6881** | **716** | **2.4** |
| 2 | - | - | - | - | **7** | **51235** | **4205** | **8** | **12** | **49856** | **4158** | **9** | **7** | **63288** | **11364** | **8** | **12** | **60189** | **4267.8** | **7** |
| 3 | - | - | - | - | **12** | **63479** | **51284** | **18** | **9** | **60125** | **36542** | **21** | **9** | **93471** | **87551** | **15** | **15** | **86522** | **63401** | **12** |

The metrics of the proposed algorithms were compared to compare their results.

### 4.1 Computational time complexity

Computational time is an important characteristic of an optimization algorithm. The graph of convergence of meta-heuristic algorithms over time is as follows. According to Figure 5, the time to obtain a good solution increases exponentially with increasing problem size, which can be roughly considered close to an exponential distribution. Regarding the whole processing time (from the initial solution to the final solution), the comparisons have revealed a faster run in IMOHS.

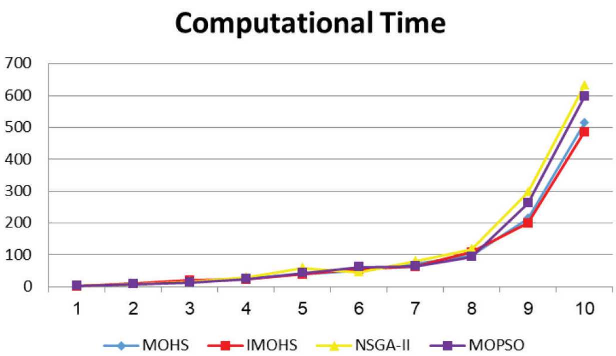



**Figure 5.** Computational time complexity

### 4.2 Implementation of algorithms

To implement the experiments, sample problems were devised, and the problem dimensions were changed, such as the number of jobs and machines. The algorithms were compared using the relative percentage deviation (RPD) rate because the objective function values were of different and unequal scales. This comparison was made [25].

$$RPD = \frac{ALG_{sol} - Min_{sol}}{Min_{sol}} \times 100 \quad (40)$$

$ALG_{sol}$ is the algorithm solution and $Min_{sol}$ is the minimum value of the solutions. In this ratio, the lower the RPD, the high the solution's quality, and the better the algorithm's performance. In the Results section, the performance of the algorithms is compared in terms of the problem size, which is varied based on the increased number of jobs, machines, and stages. The RPD results obtained from the algorithms' implementation are given in Table 6.

As mentioned before, the SNS metric is the most important metric for comparing multi-objective optimization algorithms. In 40% of all test problems, the IMOHS performed better in terms of the diversity metric. The MID metric of the NSGA-II algorithm

**Table 6.** The normalized outputs of different approaches for test problems

| | | Test problems | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| MOHS | MID | 0 | 0.0536 | 0.0785 | **0.0665** | 0.1152 | 0.0711 | 0.0945 | 0.1342 | 0.2652 | 0.2169 |
| | NPS | 0 | 0.0514 | 0.0846 | 0.1025 | 0.0485 | 0.0254 | 0.0107 | 0.0165 | 0.0314 | 0.0125 |
| | SNS | 0 | 0.1124 | 0.1412 | 0.1139 | 0.1854 | 0.1574 | 0.2173 | 0.221 | 0.2245 | **0.1923** |
| | QM | 0.25 | 0.28 | 0.27 | 0.19 | 0.23 | 0.28 | **0.34** | 0.22 | 0.18 | 0.12 |
| IMOHS | *MID* | 0 | **0.0263** | **0.0652** | 0.0747 | 0.1547 | 0.0262 | **0.0751** | 0.7469 | 0.1867 | **0.1625** |
| | *NPS* | 0 | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| | *SNS* | 0 | **0.0942** | **0.0592** | 0.1443 | 0.1237 | 0.2147 | 0.0589 | **0.1418** | **0.1685** | 0.2457 |
| | *QM* | 0.25 | **0.30** | 0.26 | **0.31** | **0.27** | **0.35** | 0.24 | **0.38** | **0.45** | **0.48** |
| MOPSO | *MID* | 0 | 0.04352 | 0.08136 | 0.09482 | 0.15524 | 0.05912 | 0.08055 | 0.07942 | **0.18333** | 0.23658 |
| | *NPS* | 0 | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| | *SNS* | 0 | 0.1355 | 0.16771 | 0.18506 | **0.07026** | 0.15967 | **0.03713** | 0.24132 | 0.20485 | 0.25431 |
| | *QM* | 0.25 | 0.23 | 0.18 | 0.23 | 0.25 | 0.12 | 0.16 | 0.14 | 0.21 | 0.23 |
| NSGA-II | *MID* | 0 | 0.03115 | 0.07073 | 0.08467 | **0.11497** | **0.01623** | 0.11011 | **0.04692** | 0.21367 | 0.21021 |
| | *NPS* | 0 | 0.07107 | 0.09572 | 0.13714 | 0.05688 | 0.01739 | 0.02069 | 0.02354 | 0.02143 | 0.01084 |
| | *SNS* | 0 | 0.11232 | 0.15954 | **0.07679** | 0.08373 | **0.13462** | 0.09839 | 0.15424 | 0.18985 | 0.26882 |
| | *QM* | 0.25 | 0.19 | **0.29** | 0.27 | 0.25 | 0.25 | 0.26 | 0.26 | 0.16 | 0.17 |

was better than the MOPSO in all test problems, while in 80% of the test problems, the MOPSO outperformed a number of Pareto solutions. As shown in Table 6 and Figure 6, in 70% of the test problems, the response quality generated by the IMOHS algorithm was higher than the other algorithms.

To avoid redundant figures, only one test problem has been selected. Figure 7 illustrates sample Pareto frontiers achieved by metaheuristic algorithms in test problem 4, in which there are seven solutions for MOHS, fourteen solutions for IMOHS, seven solutions for MOPSO, and eleven solutions for NSGA-II. The Pareto fronts of the optimal solutions in Figure 7 present the proper diversity of the generated solutions for algorithms.

Based on Table 6 and Figure 6, the best performance belonged to the IMOHS, MOHS, MOPSO, and NSGA-II, respectively, in terms of the SNS index. Furthermore, the IMOHS, NSGA-II, MOHS, and MOPSO, respectively, had the best performance in terms of the MID index. According to the quality index, the IMOHS, NSGA-II, MOHS, and MOP-

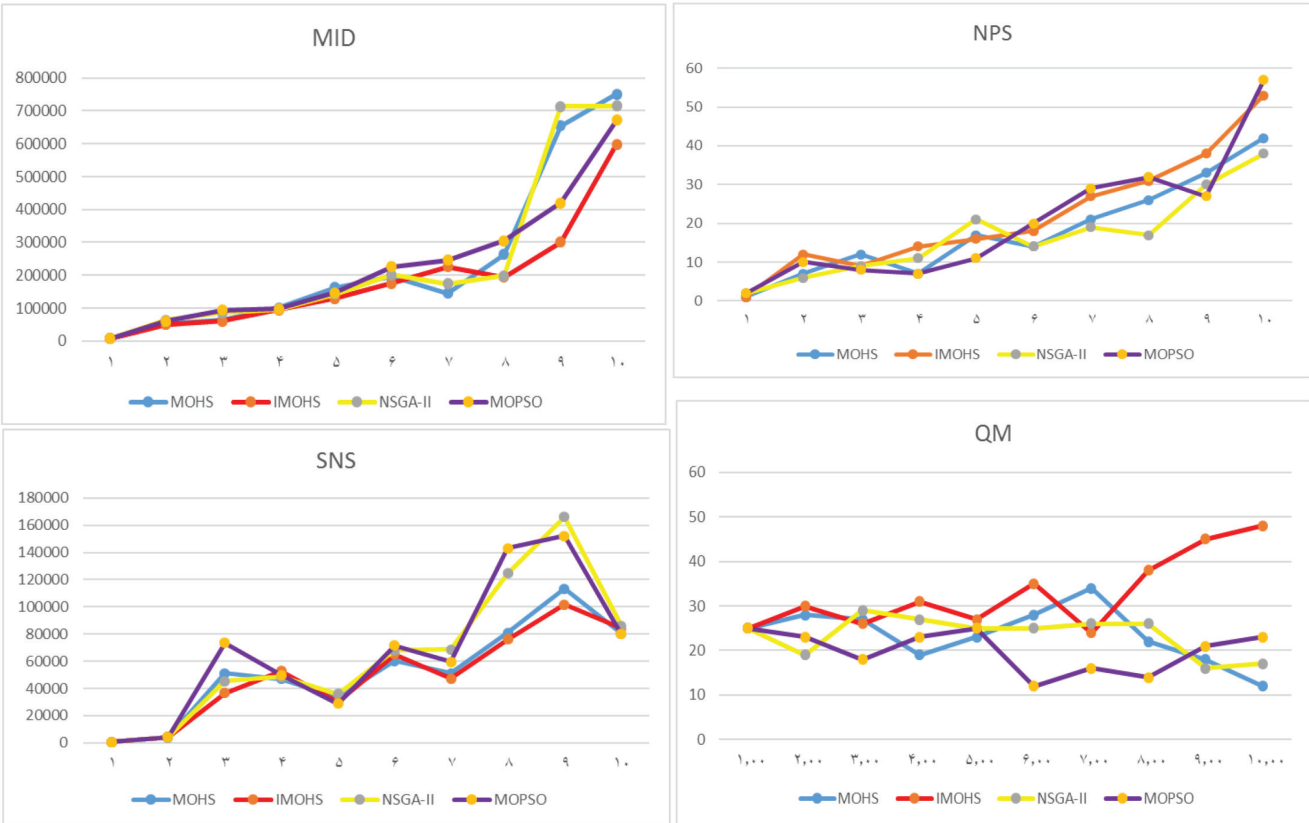



**Figure 6.** Performance measures of the algorithms

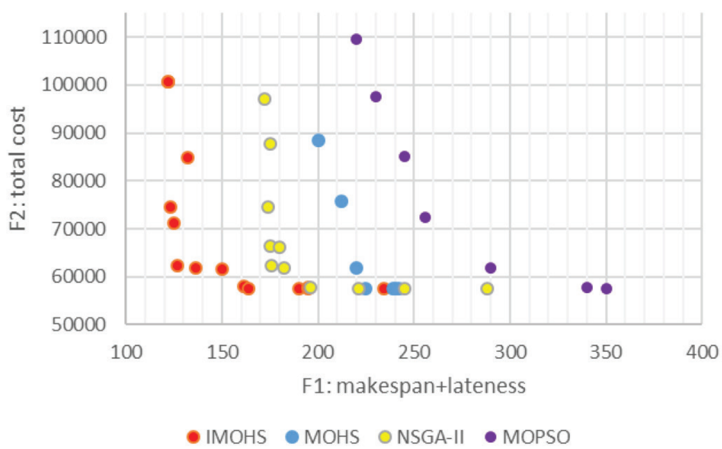



**Figure 7.** Pareto frontiers achieved by metaheuristic algorithms in test problem 4

SO respectively had the best performance. Regarding the dispersion and uniformity indices, in all cases, the IMOHS algorithm displayed the greatest potential to search the response space and obtained the best and near-optimal solutions. As for the three algorithms, the NSGA-II had the highest potential to achieve Pareto solutions.

## 5. Conclusions

This article presents a new model for integrating production planning and scheduling in a hybrid flow shop to simultaneously minimize two conflicting objectives: the time function, including makespan and lateness, and the cost function, including production, backorder, inventory, and maintenance. This model can be applied to most industries. In the proposed model, maintenance constraints were considered to obtain maintenance activities. This problem is also applicable in industries that deal with multi-source maintenance. Besides the machine as the main source of maintenance, molds were considered another source. Another novelty of this problem is its approximation to real-life situations by using the maintenance activity type as perfect or imperfect activity.

The literature attests to the difficulty of this problem; thus, the presented model is solved using GAMS. However, for larger problems, GAMS may increase the computational time. Therefore, a metaheuristic could be more effective. To address this, a multi-objective algorithm is proposed to solve the proposed model. To compare the results in terms of different comparison metrics, four well-known multi-objective algorithms, namely improved multi-objective harmony search (IMOHS), multi-objective harmony search (MOHS), multi-objective particle swarm optimization (MOPSO), and NSGA-II were applied. Due to the sensitivity of these metaheuristics to their parameters, the Taguchi method is employed to set the algorithms' parameters and find the best level for each parameter. Instances are randomly generated to compare the capability of the four algorithms. To compare the four algorithms, MID, NPS, SNS, and QM were utilized as measures. Based on these criteria, IMOHS outperformed the other algorithms. The computational results show that the metaheuristic algorithms produce good solutions within a reasonable computational time.

As a topic for future research, other objective functions could be added to the present objective functions to cover real-world problems. Foreseeing the uncertainty of real-world problems can be helpful for this purpose. Novel metaheuristic algorithms and the model can also be executed in a job-shop environment by researchers interested in this field.

## Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

## References

[1] S. Dellagi, A. Chelbi, and W. Trabelsi, "Joint integrated production-maintenance policy with production plan smoothing through production rate control," J. Manuf. Syst., vol. 42, pp. 262–270, 2017, doi: 10.1016/j.jmsy.2016.12.013.

[2] F. D'Amico, D. A. Rossit, and M. Frutos, "Lot streaming Permutation Flow shop with energy awareness," Int. J. Ind. Eng. Manag., vol. 12, no. 1, pp. 25–36, 2021, doi: 10.24867/IJIEM-2020-1-274.

[3] C. T. Maravelias and C. Sung, "Integration of production planning and scheduling: Overview, challenges and opportunities," Comput. Chem. Eng., vol. 33, no. 12, pp. 1919–1930, 2009, doi: 10.1016/j.compchemeng.2009.06.007.

[4] B. S. Purohit and B. Kumar Lad, "Production and maintenance planning: an integrated approach under uncertainties," Int. J. Adv. Manuf. Technol., vol. 86, no. 9–12, pp. 3179–3191, 2016, doi: 10.1007/s00170-016-8415-9.

[5] C. S. Wong, F. T. S. Chan, and S. H. Chung, "A joint production scheduling approach considering multiple resources and preventive maintenance tasks," Int. J. Prod. Res., vol. 51, no. 3, pp. 883–896, 2013, doi: 10.1080/00207543.2012.677070.

[6] S. Dauzère-Pérès and J.-B. Lasserre, "On the importance of sequencing decisions in production planning and scheduling," Int. Trans. Oper. Res., vol. 9, no. 6, pp. 779–793, 2002, doi: 10.1111/1475-3995.00388.

[7] S. Dauzère-Péres and J.-B. Lasserre, "Integration of lotsizing and scheduling decisions in a job-shop," Eur. J. Oper. Res., vol. 75, no. 2, pp. 413–426, 1994, doi: 10.1016/0377-2217(94)90085-X.

[8] X. Zhou, K. Huang, L. Xi, and J. Lee, "Preventive maintenance modeling for multi-component systems with considering stochastic failures and disassembly sequence," Reliab. Eng. Syst. Saf., vol. 142, pp. 231–237, 2015, doi: 10.1016/j.ress.2015.05.005.

[9] H.-S. Yan, X.-Q. Wan, and F.-L. Xiong, "Integrated production planning and scheduling for a mixed batch job-shop based on alternant iterative genetic algorithm," J. Oper. Res. Soc., vol. 66, no. 8, pp. 1250–1258, 2015, doi: 10.1057/jors.2014.88.

[10] Y. Chu, F. You, J. M. Wassick, and A. Agarwal, "Integrated planning and scheduling under production uncertainties: Bi-level model formulation and hybrid solution method," Comput. Chem. Eng., vol. 72, pp. 255–272, 2015, doi: 10.1016/j.compchemeng.2014.02.023.

[11] J. Han, Y. Liu, L. Luo, and M. Mao, "Integrated production planning and scheduling under uncertainty: A fuzzy bi-level

decision-making approach," Knowledge-Based Syst., vol. 201–202, 2020, doi: 10.1016/j.knosys.2020.106056.

[12] K. C. Bhosale and P. J. Pawar, "Material Flow Optimisation of Flexible Manufacturing System using Real Coded Genetic Algorithm (RCGA)," Mater. Today Proc., vol. 5, no. 2, Part 2, pp. 7160–7167, 2018, doi: 10.1016/j.matpr.2017.11.381.

[13] M. Alimian, V. Ghezavati, and R. Tavakkoli-Moghaddam, "New integration of preventive maintenance and production planning with cell formation and group scheduling for dynamic cellular manufacturing systems," J. Manuf. Syst., vol. 56, pp. 341–358, 2020, doi: 10.1016/j.jmsy.2020.06.011.

[14] A. Azzouz, M. Ennigrou, and L. Ben Said, "A hybrid algorithm for flexible job-shop scheduling problem with setup times," Int. J. Prod. Manag. Eng., vol. 5, no. 1 SE-Papers, pp. 23–30, Jan. 2017, doi: 10.4995/ijpme.2017.6618.

[15] G. Mavrotas, "Effective implementation of the ε-constraint method in Multi-Objective Mathematical Programming problems," Appl. Math. Comput., vol. 213, no. 2, pp. 455–465, 2009, doi: 10.1016/j.amc.2009.03.037.

[16] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 1917, pp. 849–858, 2000, doi: 10.1007/3-540-45356-3_83.

[17] N. Srinivas and K. Deb, "Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms," Evol. Comput., vol. 2, no. 3, pp. 221–248, 1994, doi: 10.1162/evco.1994.2.3.221.

[18] S. K. Karimi, S. J. Sadjadi, and S. G. J. Naini, "A bi-objective production planning for a flexible supply chain solved using NSGA-II and MOPSO," Int. J. Ind. Eng. Manag., vol. 13, no. 1, pp. 18–37, 2022, doi: 10.24867/IJIEM-2022-1-298.

[19] J. Moore, R. Chapman, and G. Dozier, "Multiobjective particle swarm optimization," in Proceedings of the Annual Southeast Conference, 2000, vol. 2000-April, pp. 56–57, doi: 10.1145/1127716.1127729.

[20] Z. W. Geem, J. H. Kim, and G. V Loganathan, "A New Heuristic Optimization Algorithm: Harmony Search," Simulation, vol. 76, no. 2, pp. 60–68, 2001, doi: 10.1177/003754970107600201.

[21] S. Sivasubramani and K. S. Swarup, "Multi-objective harmony search algorithm for optimal power flow problem," Int. J. Electr. Power Energy Syst., vol. 33, no. 3, pp. 745–752, 2011, doi: 10.1016/j.ijepes.2010.12.031.

[22] R. Ramezanian, M. Saidi-Mehrabad, and P. Fattahi, "MIP formulation and heuristics for multi-stage capacitated lot-sizing and scheduling problem with availability constraints," J. Manuf. Syst., vol. 32, no. 2, pp. 392–401, 2013, doi: 10.1016/j.jmsy.2013.01.002.

[23] D. C. Montgomery, Design and analysis of experiments. New Yor, NY, USA: Wiley, 2017.

[24] R. Tavakkoli-Moghaddam, A. Rahimi-Vahed, and A. H. Mirzaei, "A hybrid multi-objective immune algorithm for a flow shop scheduling problem with bi-objectives: Weighted mean completion time and weighted mean tardiness," Inf. Sci. (Ny)., vol. 177, no. 22, pp. 5072–5090, 2007, doi: 10.1016/j.ins.2007.06.001.

[25] I. Rastgar and R. Sahraeian, "New Formulation and Solution in PCB Assembly Systems with Parallel Batch processors," J. Qual. Eng. Prod. Optim., vol. 2, no. 1, pp. 27–46, 2017, doi: 10.22070/jqepo.2016.348.