



ARTICLE



## Research on Dynamic Scheduling Method for Hybrid Flow Shop Order Disturbance Based on IMOGWO Algorithm

Feng Lv\*, Huili Chu, Cheng Yang and Jiajie Zhang

School of Mechatronics Engineering, Henan University of Science and Technology, Luoyang, 471003, China

\*Corresponding Author: Feng Lv. Email: lvfeng1980@haust.edu.cn

Received: 06 September 2025; Accepted: 28 October 2025; Published: 12 January 2026

**ABSTRACT:** To address the issue that hybrid flow shop production struggles to handle order disturbance events, a dynamic scheduling model was constructed. The model takes minimizing the maximum makespan, delivery time deviation, and scheme deviation degree as the optimization objectives. An adaptive dynamic scheduling strategy based on the degree of order disturbance is proposed. An improved multi-objective Grey Wolf (IMOGWO) optimization algorithm is designed by combining the “job-machine” two-layer encoding strategy, the timing-driven two-stage decoding strategy, the opposition-based learning initialization population strategy, the POX crossover strategy, the dual-operation dynamic mutation strategy, and the variable neighborhood search strategy for problem solving. A variety of test cases with different scales were designed, and ablation experiments were conducted to verify the effectiveness of the improved strategies. The results show that each improved strategy can effectively enhance the performance of the IMOGWO. Additionally, performance analysis was conducted by comparing the proposed algorithm with three mature and classical algorithms. The results demonstrate that the proposed algorithm exhibits superior performance in solving the hybrid flow-shop scheduling problem (HFSP). Case validations were conducted for different types of order disturbance scenarios. The results demonstrate that the proposed adaptive dynamic scheduling strategy and the IMOGWO algorithm can effectively address order disturbance events. They enable rapid response to order disturbance while ensuring the stability of the production system.

**KEYWORDS:** Hybrid flow shop; order disturbance; dynamic scheduling; improved multi-objective Grey Wolf optimization

### 1 Introduction

The hybrid flow-shop scheduling problem (HFSP), a common type of shop scheduling, integrates traditional permutation flow shop production and parallel machine scheduling. It is widely applied in industries such as steelmaking, casting, chemical engineering, textile, machinery, and assembly [1]. During the manufacturing process of hybrid flow shops, various random factors and dynamic disturbances often exist. As a direct reflection of changes in external market demand, order disturbance is characterized by considerable suddenness and unpredictability [2]. Balancing production efficiency and stability in hybrid flow shops amidst order disturbances holds significant theoretical importance and practical application value. Dynamic scheduling is an anti-interference method for workshops, which replans schemes by integrating the current status, unprocessed tasks, and disturbances [3]. Due to its capability to handle production uncertainties, dynamic scheduling has become a research focus in the HFSP field.



In terms of shop scheduling models, multi-objective optimization is more in line with practical requirements in most cases. Yang et al. [4] constructed a mathematical model with maximum makespan and total delay time as objectives for the dynamic shop scheduling problem. Li et al. [5] addressed dynamic shop scheduling by formulating a model focused on makespan, worker fatigue, and scheduling deviation. Shi et al. [6] developed a mathematical model targeting makespan, total job delay time, overall machine idle time, and total energy use as its objective function for the problem of random job arrivals in the shop. Liu et al. [7] established a mathematical model aiming at minimizing makespan and total energy consumption for the HFSP.

Existing studies on dynamic scheduling strategies mainly focus on event-driven, periodic, and hybrid rescheduling. Wang et al. [8] proposed a hybrid adaptive rescheduling strategy to address uncertain disturbances. Chen et al. [9] adopted event-driven rescheduling to resolve dynamic disturbances in production. Zhu et al. [10] utilized a hybrid rescheduling approach to handle order cancellation disturbances. Wei et al. [11] adopted a hybrid rescheduling strategy to solve the shop scheduling problem with machine failures. Angel-Bello et al. [12] applied a periodic rescheduling strategy to cope with random order arrivals.

Currently, the most commonly used methods for solving HFSP are meta-heuristic algorithms, with swarm intelligence optimization algorithms and genetic algorithms as typical representatives. Zou et al. [13] proposed a multi-level subpopulation-based particle swarm optimization algorithm (MLPSO). Wang et al. [14] developed a variable-representation discrete ABC algorithm. Zeng et al. [15] designed an improved NSGA-III algorithm. The GWO algorithm is a new type of swarm intelligence optimization algorithm inspired by the hunting behavior and internal hierarchical system of wolf packs. This algorithm features a simple basic framework, a small number of parameters, and has been proven to be superior to other heuristic algorithms in terms of convergence and optimization performance [16]. Li et al. [17] designed a reasonable encoding and decoding scheme, developed four crossover operators to expand the search space, and proposed a local search strategy with the concept of key factories, thereby improving the exploitability of the IGWO algorithm. Lin et al. [18] used a reinforcement learning algorithm to optimize the parameters of the GWO algorithm, effectively enhancing the algorithm's effectiveness. Zheng and Chen [19] developed a Q-learning multi-objective grey wolf optimizer (QMOGWO) algorithm and designed several initialization strategies and local search strategies to address the distributed hybrid flow shop scheduling problem.

Overall, existing studies have conducted a series of discussions on dynamic shop scheduling problems, laying a foundation for the research in this paper, but still need to be expanded. First, regarding the construction of dynamic scheduling models, existing studies have mostly focused on factors such as time and energy consumption, while attention to the stability of scheduling systems remains insufficient. To address this, on the premise of selecting the maximum makespan (reflecting production efficiency) and the delivery time deviation (measuring the satisfaction of customer demands) as static optimization objectives, this paper further introduces the scheme deviation degree (related to the stability of scheduling systems) to construct a dynamic scheduling mathematical model. Second, concerning dynamic scheduling strategies, the lack of an effective driving mechanism may lead to excessive or insufficient rescheduling in actual production processes, which causes adverse effects such as a decrease in the stability of the production line or a lack of rapid response capability. Thus, this paper designs a dynamic scheduling strategy based on the order disturbance degree. Third, regarding solution algorithms, the encircling and hunting mechanism of the GWO algorithm ensures its high accuracy, fast convergence speed, few parameters, and strong optimization ability. Inspired by the algorithm improvements in the preceding literature, this paper designs an IMOGWO algorithm to solve the dynamic HFSP.

The main contributions of this paper are as follows: (1) A multi-objective dynamic scheduling model is constructed. With maximum makespan, delivery time deviation, and scheme deviation degree as the

optimization objectives, the model enables rapid response to order disturbances while ensuring the stability of the production system. (2) An adaptive dynamic scheduling strategy based on the order disturbance degree is designed. It adopts different scheduling strategies according to varying degrees of order disturbance, which avoids excessive or insufficient rescheduling frequency in the actual production process of the shop. (3) An IMOGWO algorithm is designed. This algorithm integrates the “job-machine” two-layer encoding strategy, the timing-driven two-stage decoding strategy, the opposition-based learning initialization population strategy, the POX crossover strategy, the dual-operation dynamic mutation strategy, and the variable neighborhood search (VNS) strategy. Experimental results show that this algorithm outperforms the existing algorithms commonly used for HFSP.

The structure of this paper is organized as follows: [Section 2](#) provides a detailed description of the problem and proposes a corresponding mathematical model. [Section 3](#) presents an adaptive dynamic scheduling strategy based on the order disturbance degree. [Section 4](#) focuses on algorithm design. [Section 5](#) conducts experimental simulation and case analysis. [Section 6](#) draws the conclusion and outlines future work.

## 2 Problem Description and Modeling

### 2.1 Problem Description and Assumptions

The dynamic scheduling problem of the hybrid flow shop studied in this paper can be described as follows: There are  $n$  jobs processed through  $j$  operations. The set of process stages is  $S = \{1, 2, \dots, j, \dots, s\}$ . Each stage  $j$  contains  $e_j$  parallel machines, corresponding to the processing machine set  $E_j$ , and at least one stage has 2 or more parallel machines. The set of processing equipment is  $E = \{1, 2, \dots, k, \dots, e\}$ . The initial order  $W = \{W_1, W_2, \dots, W_i, \dots, W_n\}$  includes  $n$  types of jobs, where the batch set of jobs  $W_i$  is  $X_i = \{1, 2, \dots, x, \dots, x_i\}$ .  $W_{i,x}$  denotes the  $x$ -th batch of jobs  $W_i$ , and  $W_{ij,x}$  represents the processing of jobs  $W_{i,x}$  at stage  $j$ . Its optional processing machine set is  $E_{ij} \subseteq E_j$ . On this basis, three types of order disturbance events, namely order expediting, order cancellation, and order insertion, are considered. At a random moment  $T_r$ , there exist  $W_I$  expedited jobs  $W_I = \{W_{I,1}, W_{I,2}, \dots, W_{I,i}, \dots, W_{I,n}\}$ ,  $W_{II}$  inserted jobs  $W_{II} = \{W_{II,1}, W_{II,2}, \dots, W_{II,i}, \dots, W_{II,n}\}$ ,  $W_{III}$  withdrawn jobs  $W_{III} = \{W_{III,1}, W_{III,2}, \dots, W_{III,i}, \dots, W_{III,n}\}$ . When a disturbance event occurs, different rescheduling schemes are generated according to different situations to minimize the impact of emergencies on production.

Based on the above description, the following assumptions are made in this paper:

- (1) Job production follows the batch production principle, and batch production is carried out according to the optimal batch size.
- (2) There is no sequence constraint between different batches of jobs to realize flexible scheduling.
- (3) The processing of the same batch of jobs cannot be interrupted;
- (4) The preparation time before job processing has been included in the actual processing time.
- (5) All parts can be served directly at the initial moment, and all processing equipment can be used for processing production activities at the initial moment.
- (6) There are infinite buffer storage areas between two consecutive processing operations.
- (7) When implementing the dynamic scheduling strategy, the completed jobs and the jobs in processing are not scheduled.

To facilitate modeling, the relevant parameter symbols and their meanings in the model are presented in [Table 1](#).

**Table 1:** Meaning of the symbol

Symbol	Meaning
$R_{ije}$	Processing time of one job of type $i$ on machine $e$ in stage $j$
$R_{ije,x}$	Processing time of the whole batch of jobs $W_{i,x}$ on machine $e$ in stage $j$
$B_{ije,x}$	Start of processing time of operation $W_{i,x}$ on machine $e$
$C_{ije,x}$	Completion time of work operation $W_{i,j,x}$ on machine $e$
$Tc_i$	Completion time of the $i$ th jobs
$Td_i$	Delivery time of the $i$ th jobs
$W_{i(j+1),e',x}^{ab} = 1$	0-1 decision variable, 1 if jobs $W_{a,b}$ are the immediate predecessor of jobs $W_{i,x}$ on machine $m$ at stage $j+1$ , 0 otherwise
$X_{ije,x}$	0-1 decision variable, 1 if jobs $W_{i,x}$ are processed on machine $e$ at stage $j$ , 0 otherwise

## 2.2 Modeling

To address order disturbances and rescheduling problems, this paper divides the scheduling problem into two phases: the first phase is the initial static scheduling before the order disturbance; the second phase is the dynamic optimization rescheduling of all unfinished jobs after the order disturbance.

### 2.2.1 Static Scheduling Optimization Model

The initial static scheduling problem before order disturbance is modeled to minimize the maximum makespan and minimize the delivery time deviation as the performance optimization objectives, and the objective function is established as shown below:

$$\min f_1 = \max \{C_{ije,x}\} \quad (1)$$

$$\min f_2 = \sqrt{\frac{1}{n} \sum_{i=1}^n (Td_i - Tc_i)^2} \quad (2)$$

The following constraints are established:

$$\sum_{x=1}^{X_i} \text{Sum}(W_{i,x}) = \text{Sum}(W_i) \quad \forall i \quad (3)$$

$$R_{ije,x} = R_{ije} \times \text{Sum}(W_{i,x}) \quad \forall i, j, \forall x \in \{1, 2, \dots, X_i\}, \forall e \in E_{ij} \quad (4)$$

$$C_{ije,x} = B_{ije,x} + R_{ije,x}, \text{when } X_{ije,x} = 1, \forall i, j, \forall x \in X_i, \forall e \in E_{ij} \quad (5)$$

$$Tc_i = \max \{C_{ije,x}\}, \forall j, e, x \quad (6)$$

$$B_{i(j+1)e',x} = \max \{C_{ije,x}, C_{a(j+1)e',b}\}, \text{when } X_{i(j+1)e',x}, X_{a(j+1)e',b}, W_{i(j+1),e',x}^{ab} = 1 \quad (7)$$

$$\forall i, a, j, \forall x \in X_i, \forall b \in X_a, \forall e \in E_{ij}, \forall e' \in E_{i(j+1)}$$

$$\sum_{i=1}^n \sum_{x=1}^{X_i} X_{ije,x} = 1, \forall t \in [B_{ije,x}, C_{ije,x}], \forall e \in E_j, \forall j \quad (8)$$

$$\sum_{e \in E_j} X_{ije,x} = 1, \forall t \in [B_{ije,x}, C_{ije,x}], \forall x \in \{1, 2, \dots, X_i\}, \forall i, j \quad (9)$$

$$R_{ije}, R_{ije,x} \geq 0, \forall x \in \{1, 2, \dots, X_i\}, \forall e \in E_{ij}, \forall i, j \quad (10)$$

$$B_{ije,x} \geq 0, \forall x \in \{1, 2, \dots, X_i\}, \forall e \in E_{ij}, \forall i, j \quad (11)$$

Constraints (3) and (4) are time constraints for job batch processing. Constraints (5) and (6) are processing time constraints. Constraint (7) is the mixed flow shop operation sequence constraint. It means that the start time of the next operation of the job is determined by the end time of the previous operation and the processing completion time of the machine where the next operation is performed. Constraints (8) and (9) are machine capacity constraints. It indicates that for any batch of jobs, they can be processed by only one machine in the same time period, and the same machine can process only the same batch of jobs at the same time. Constraint (10) denotes that the processing time of all jobs is positive. Constraint (11) means that all jobs can be processed at moment 0.

### 2.2.2 Dynamic Scheduling Optimization Model

#### (1) Description of order disturbance events

Complete rescheduling after all perturbation events in the production process can detrimentally impact the stability of the production system. Therefore, drawing upon the scheduling characteristics of a hybrid flow shop [20], we have categorized order disturbance events into varying degrees. Dynamic scheduling strategies are matched according to these levels. When the disturbance is minor, partial rescheduling or temporary non-rescheduling can be adopted to stabilize production. The specifics of this categorization are outlined in [Table 2](#).

**Table 2:** Order disturbance degree levels

Level status	Disturbance state	Impact results	Degree of impact
General	Order cancellation	Task reduction	Slight decline in scheduling effectiveness, little impact on production flow.
Medium	Order insertion	Task increase	Usable machine idle time, minor impact on production flow.
Severe	Order expediting	Shorter delivery	Current schedule fails to meet demands, significant impact on production flow.

#### (2) Objective functions and constraints

In actual production, various order disturbances can disrupt the originally formulated scheduling plan. To maintain the stability of the scheduling system and reduce adjustment costs, a stability indicator for minimizing the scheme deviation degree (including operation deviation cost and machine deviation cost) is introduced based on the static scheduling model. The constraints remain identical to those in the static scheduling optimization model, with the only modification being that the solution scope is shifted to the set of disturbance orders or the combination of disturbance orders and remaining unfinished jobs. The objective functions are formulated as follows:

$$\min f_3 = \min \left\{ \sum_{i=1}^n \sum_{j=1}^{n_i} p_j |B'_{ij} - B_{ij}| + \sum_{e=1}^k p_e y_e \right\} \quad (12)$$

**Operation Deviation Cost (ODC):** When the operation deviation cost fluctuates within a certain range, it significantly impacts the shop's resource consumption. For example, it may require rescheduling material

delivery timings, or even necessitate immediate delivery for certain materials, and so on.

$$\text{odc} = \sum_{i=1}^n \sum_{j=1}^{n_i} p_j |B'_{ij} - B_{ij}| \quad (13)$$

where  $n$  and  $n_i$  denote the number of unprocessed jobs and operations at the start of dynamic scheduling, respectively.  $B_{ij}$  and  $B'_{ij}$  represent the start times of the  $j$ th operation for the  $i$ th job in the rescheduling scheme and the original static scheduling scheme, respectively.  $p_j$  is the penalty coefficient associated with changes in the start time of processing at stage  $j$ , where  $p_j$  is set to 1. The calculation of ODC only considers jobs that exist in both the original scheduling plan and the rescheduling scheme.

**Machine Deviation Cost (MDC):** When an operation is added to or removed from a machine, it inevitably triggers changes in the machine's tooling or fixtures. This needs consideration of the machine deviation cost.

$$\text{mdc} = \sum_{e=1}^k p_e y_e \quad (14)$$

where  $y_e$  represents the total number of operation changes on machine  $e$  at the start of dynamic scheduling compared to the initial scheduling plan.  $p_e$  is the penalty coefficient of the machine caused by operation changes in rescheduling, where  $p_e$  is set to 1.

### 3 Adaptive Dynamic Scheduling Strategy Based on the Order Disturbance Degree

#### 3.1 Dynamic Scheduling Driving Mechanism

Dynamic scheduling driving mechanism, also known as the dynamic scheduling strategy, refers to a mechanism that replans or adjusts the original scheduling plan due to changes or the occurrence of specific events [11]. Currently, it mainly includes periodic scheduling strategy, event-triggered scheduling strategy, and hybrid scheduling strategy [21].

Considering the detrimental impacts of the previously mentioned three strategies, such as excessive or insufficient scheduling times, which may lead to reduced stability of the production line or inadequate rapid response capability, this paper proposes a method integrating the order disturbance degree with the hybrid scheduling strategy, thereby designing an adaptive dynamic scheduling strategy based on the order disturbance degree. The specific steps are as follows: construct a set  $D$  of  $n$  disrupted jobs, each containing  $h_d$  operations, and define the makespan of the initial scheduling plan as the current scheduling cycle time. Upon the occurrence of disturbance orders, evaluate the level of the current order disturbance event, and calculate the order disturbance degree  $S$  based on the impact level, operation processing time, and delivery date, etc. The detailed procedure is presented in Section 3.3. The order disturbance degree  $S$ , the coefficient of delivery date of the disturbed job, and the coefficient of disturbance level are calculated as shown in Eqs. (15)–(17).

$$S = \frac{\sum_{d=1}^n (\alpha_d \beta_d \sum_{k=1}^{h_d} \bar{T}_{d,k})}{T'} \times 100\% \quad (15)$$

$$\beta_d = 1 + \frac{1}{LT_d}, \forall d \in D \quad (16)$$

$$\alpha_d = \begin{cases} 1.0, & \text{General disturbance events} \\ 1.5, & \text{Medium disturbance events, } \forall d \in D \\ 2.0, & \text{Severe disturbance events} \end{cases} \quad (17)$$

where  $\alpha_d$  denotes the disturbance coefficient of the disturbed job  $d$ ,  $\beta_d$  denotes the delivery date coefficient of the disturbed job  $d$ ,  $\bar{T}_{d,k}$  denotes the average processing time of the  $k$ th operation of the disturbed job  $d$ ,  $T'$  denotes the remaining time of the current scheduling cycle, and  $LT_d$  denotes the delivery date of the disturbed job  $d$ .

### 3.2 Dynamic Scheduling Modes

After determining the dynamic scheduling strategy, it is necessary to determine the dynamic scheduling mode. This paper employs three dynamic scheduling modes based on different scenarios: complete rescheduling, partial rescheduling, and left-shift rescheduling [22].

#### (1) Complete rescheduling

After a disturbance occurs, re-optimize the remaining unprocessed jobs and disturbed jobs to ensure the global optimality of the scheduling scheme.

#### (2) Partial rescheduling

Only scheduling the disturbed job within the idle time periods of the initial scheme, it maintains a certain degree of similarity to the initial scheduling scheme and exhibits good stability.

#### (3) Left-shift rescheduling

Remove the operations of canceled jobs, shift the remaining operations to the left to adjust their start times while retaining the original equipment allocation. This approach maintains a certain degree of similarity to the initial scheduling scheme and is consistent with actual production scenarios.

In production scheduling, the degree of order disturbance  $S$  is used as the decision threshold: when  $S > 100\%$ , the disturbance is judged to exceed the original plan's capacity, triggering the dynamic scheduling mechanism. This paper only considers one type of disturbance occurring at the same moment. For order cancellation disturbances, directly adopts left-shift rescheduling; for non-cancellation disturbances, the feasibility of partial rescheduling is first evaluated, and complete rescheduling is executed if partial rescheduling is infeasible. When  $S \leq 100\%$ , the disturbance impact is within the system's controllable range, and the original scheduling plan is maintained to reduce adjustment costs. This process is iteratively executed until the periodic scheduling strategy automatically triggers dynamic scheduling. This forms a closed-loop response mechanism of "threshold judgment-type adaptation-strategy execution", ensuring the rapid response of scheduling while taking into account system stability.

### 3.3 Dynamic Scheduling Process

Before the disturbance occurs, jobs are processed according to the initial static schedule. Upon the emergence of a disturbance, the time point and mode of dynamic scheduling are determined based on the adaptive dynamic scheduling strategy designed in this paper. This strategy is then applied to the affected jobs. The specific process is as follows. The dynamic scheduling process based on the degree of order disturbance is shown in [Fig. 1](#).

Step 1: Establish a static scheduling model aiming to minimize both the maximum makespan and the delivery time deviation. Generate and implement the initial scheduling plan using the proposed IMOGWO.

Step 2: Determine whether order disturbance has occurred. If so, proceed to the next step; otherwise, proceed to Step 4.

Step 3: Evaluate the level of the current order disturbance event and calculate the order disturbance degree  $S$ . If  $S > 100\%$ , proceed to Step 5; otherwise, proceed to the next step.

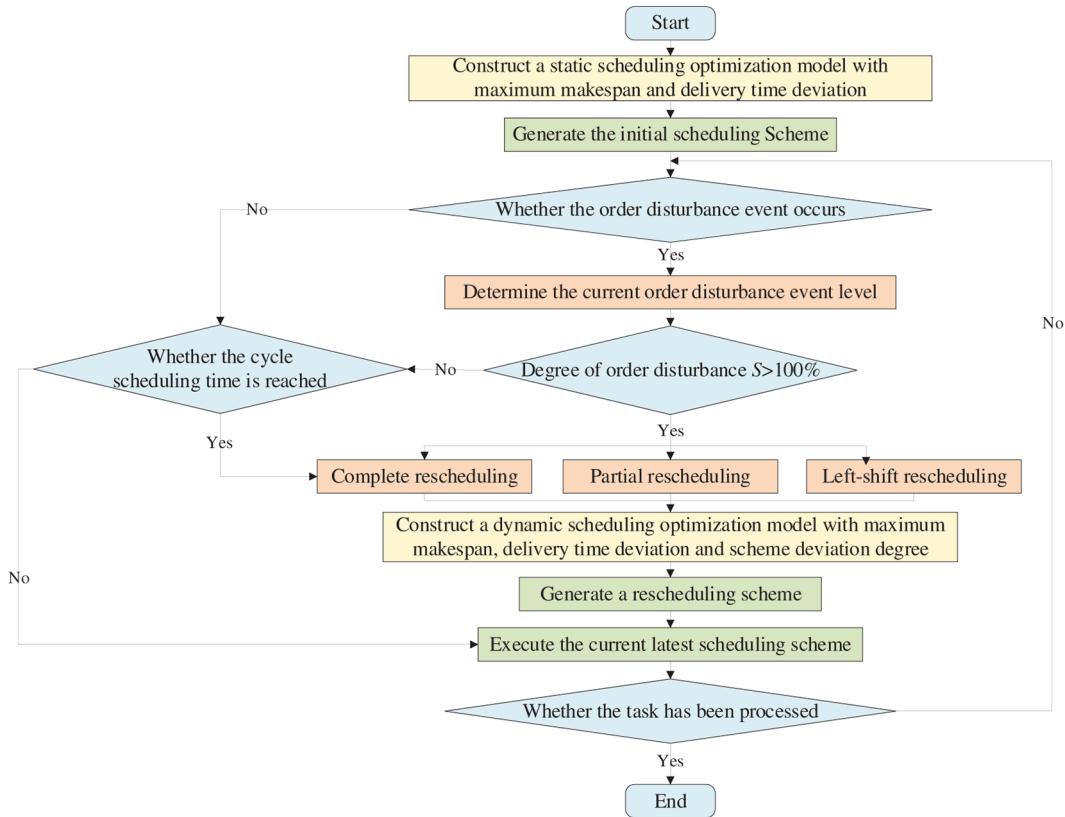
Step 4: Determine whether the scheduled cycle time has been reached. If so, execute the complete rescheduling strategy; otherwise, proceed to Step 7.

Step 5: Select an appropriate dynamic scheduling mode according to [Section 3.2](#) of this paper

Step 6: Establish a dynamic scheduling optimization model considering the maximum makespan, delivery time deviation, and scheme deviation degree. Generate and update the rescheduling plan using the proposed IMOOGWO.

Step 7: Execute the current latest scheduling scheme.

Step 8: Determine whether all tasks have been processed. If so, terminate the procedure; otherwise, return to Step 2.



**Figure 1:** Dynamic scheduling flowchart based on the degree of order disturbance

## 4 Algorithm Design

### 4.1 Gwo Algorithm

The Grey Wolf Optimizer (GWO) is a swarm intelligence optimization algorithm, simulating the collaborative hunting behavior of grey wolf packs [17]. In solving the optimization problem, the optimal three individuals in the current population are denoted as  $\alpha$ ,  $\beta$ , and  $\delta$ , called the leader wolves, and the other individual wolves (denoted as  $\omega$ ) continuously adjust their positions under the guidance of the positioning of the three leader wolves, and approach towards the optimal direction in the search space. Due to space constraints, the specific implementation process of the GWO algorithm is detailed in Reference [17].

## 4.2 Imogwo Algorithm

The Multi-Objective Grey Wolf Optimizer (MOGWO), proposed by Mirjalili et al. in 2016, extends the single-objective GWO by introducing an external archive to store current non-dominated optimal solutions and modifying the leader selection strategy [23]. However, when solving complex optimization problems, MOGWO still suffers from problems such as easy to falling into a local optimal solution and poor population diversity. Therefore, this paper makes the following improvements to the MOGWO algorithm in Reference [23].

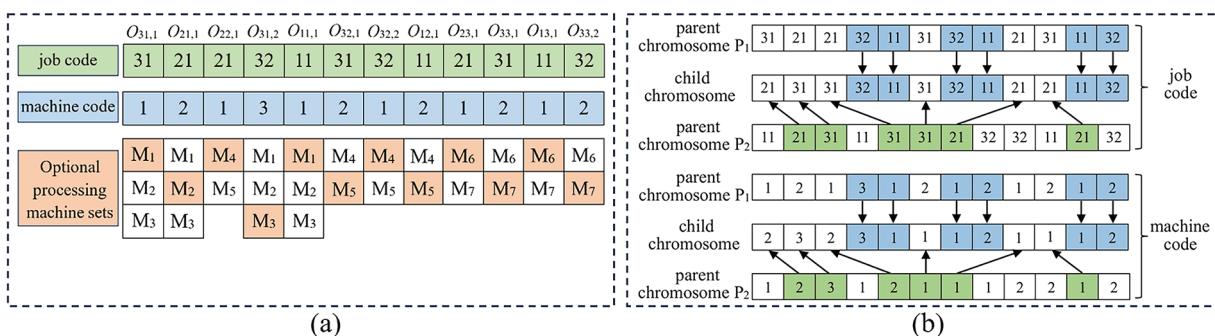
### 4.2.1 Encoding and Decoding Strategies

The GWO algorithm was originally proposed to solve continuous problems, but the HFSP is a discrete problem. To better integrate the GWO algorithm with HFSP, in terms of process and machine arrangement, they should be divided into two corresponding parts. These two parts form a chromosome, serving as a feasible encoding scheme for HFSP. Therefore, this paper designs a “job-machine” two-layer encoding strategy.

The first layer is the job code: the first digit represents the job number, the second digit represents the job's processing batch, and the occurrence frequency of the job number indicates the number of its operations.

The second layer is the machine code: each machine code denotes the index of the alternative processing machine set corresponding to the job in the first layer.

Each two-layer code corresponds to a scheduling scheme, specifying the processing machine selected for each operation of each job. The size and number of job subbatches are determined before applying the algorithm, so only the job batch needs to be reflected in the code. Take 3 jobs with 3 operations as an example, where the 3rd job is divided into 2 batches and the others are 1 batch each, as illustrated in Fig. 2a. The job code is [31, 21, 21, 32, 11, 31, 32, 11, 21, 31, 11, 32], and the machine code is [1, 2, 1, 3, 1, 2, 1, 2, 1, 2, 1, 2]. The represented operation sequence is  $\{O_{31,1}, O_{21,1}, O_{22,1}, O_{31,2}, O_{11,1}, O_{32,1}, O_{32,2}, O_{12,1}, O_{23,1}, O_{33,1}, O_{13,1}, O_{33,2}\}$ , where  $O_{32,1}$  denotes the 2nd operation of the 1st batch of job 3. The processing machine numbers corresponding to each operation are sequentially  $\{M_1, M_2, M_4, M_3, M_1, M_5, M_4, M_6, M_6, M_7, M_6, M_7, M_7, M_6, M_7, M_7\}$ .



**Figure 2:** Two-layer encoding and decoding (a) and POX crossover example (b)

During decoding, the designed time-driven two-stage decoding strategy is adopted. Following the logic of “determining the basic order first and then optimizing local efficiency”, the genes of the operation sequence are traversed from left to right to determine the processing sequence of all operations.

In the first stage, the job processing sequence and machine assignment are determined based on the encoding. By encoding to define the process and machine, the common infeasible solution of “reverse process

sequence" is avoided, ensuring that the solution complies with the hard timing constraints of HFSP. At the same time, it prevents the increase in computational complexity caused by the coupling of "machine selection" and "timing sequence" in the subsequent stages, reducing interference for the local optimization in the second stage.

In the second stage, the principle of "shortest processing completion time in the current stage" is followed for sorting. The specific process involves screening candidate operations, calculating the expected completion time, and determining the sorting within the stage. All operations of workpieces that have completed the previous stage processing and have not entered the current stage are taken as candidate operations. For each candidate operation, the expected completion time on the current machine is calculated by combining the current idle time of its dedicated machine (i.e., the completion time of the previous operation on the machine) and the processing time of the operation itself (expected completion time = machine idle time + operation processing time). They are then arranged in ascending order of expected completion time, and the operation that can make the machine complete its work faster is prioritized for initiation.

From the perspective of the quality improvement mechanism of the HFSP solution, this two-stage strategy first avoids unfeasible solutions through the determination of operations and machines in the first stage to ensure the basic quality of the solution. Second, it reduces in-stage congestion through local optimization in the second stage, shortening the global makespan.

#### 4.2.2 Opposition-Based Learning Initialization Population Strategy

The standard GWO algorithm utilizes a random initialization strategy, which often fails to ensure the diversity of the initial population. To address this issue, this paper introduces an opposition-based learning initialization strategy. The specific steps are as follows:

- (1) Randomly generate  $n$  grey wolf individuals denoted as  $x_{ij}$  within the decision space of the problem to form the initial population  $P_1$ .
- (2) Construct an opposite population  $P_2$  corresponding to  $P_1$  using the principle of opposition-based learning. The opposite individual  $x'_{ij}$  is defined as:  $x'_{ij} = s - x_{ij} + 1$ , where  $s$  is the total number of operations of the jobs.
- (3) Merge the initial population  $P_1$  and the opposite population  $P_2$  to form a combined population with  $2n$  grey wolf individuals. Sort the  $2n$  individuals using non-dominated sorting, rank each layer of Pareto solutions, and select the top  $n$  individuals with the best fitness as the final initial population.

#### 4.2.3 Crossover and Mutation Strategies

To enhance the global search capability of the MOGWO algorithm during the iteration process, crossover and mutation operations are incorporated to strengthen the algorithm's ability to escape local optima.

The crossover operation is the main means of the algorithm to generate new individuals. This study employs POX crossover to perform crossing on both job codes and machine codes, with the specific crossing process illustrated in [Fig. 2b](#). Firstly, the crossover is performed on the job code, two parent chromosomes  $P_1$  and  $P_2$  are randomly selected, assuming the initial job set  $W = \{11, 21, 31, 32\}$ . The job set is divided into two subsets  $W_1 = \{11, 32\}$  and  $W_2 = \{21, 31\}$ , and the genes containing  $W_1$  in  $P_1$  are copied to the child chromosomes without changing their positions in the process of copying; and then the genes containing  $W_2$  in  $P_2$  are moved to the child chromosomes sequentially in accordance with the order. After the job code crossover operation is completed, the machine code is crossover in accordance with the job code crossover

method. This is done to ensure that the processing machine assigned to each operation of every job remains as an optional processing machine.

Mutation operations are applied to the job codes and machine codes after POX crossing. A dual-operation dynamic mutation strategy is designed in this paper, encompassing two distinct mutation operations. A random selection of one of these operations is performed during the implementation process.

- (1) Job Code Mutation: Generate two positions randomly within the length of the solution, swap the job codes of the two positions, and synchronously exchange their corresponding machine codes to ensure machine feasibility.
- (2) Machine Code Mutation: Randomly select multiple positions within the solution length, reassign the machine code in the position, and carry out the mutation following the principle of large allocation probability for short processing time.

#### 4.2.4 Variable Neighborhood Search

To further enhance the local search capability of the algorithm, the local optimization ability is strengthened through a customized neighborhood structure; a VNS containing multiple neighborhood structures is designed [24]. Performing a local search on all population individuals would increase computational time and reduce efficiency. Therefore, to balance computation reduction and diversity preservation, only partial individuals ( $\alpha$ ,  $\beta$ , and  $\delta$ ) in the population are selected for VNS. The following three neighborhood search structures are employed in this paper.

N1: Randomly select an operation from the current solution. If the operation has multiple alternative machines, choose the machine with the minimum processing time from its alternative machine set.

N2: Find the machine with the largest load from the current solution, randomly select an operation to be processed on that machine, and randomly select a machine to replace it from its set of available machines.

N3: Randomly select two different jobs and an operation position, then swap the machine allocations of the two jobs at this position.

In conducting a VNS, the process begins by determining the current solution. Subsequently, the aforementioned operations are executed sequentially. If a more optimal solution is produced using the neighborhood structure N1, it replaces the current solution, and the process continues with the same neighborhood structure. However, if no improvements can be made to the current solution using N1, the search then progresses to the next neighborhood structure. This systematic shift continues until the predetermined termination iteration number has been reached.

#### 4.2.5 Specific Steps of IMOGWO Algorithm

The flowchart of the IMOGWO Algorithm is shown in [Fig. 3](#). The detailed steps are as follows:

Step 1: Input order information and initialize algorithm parameters.

Step 2: Population initialization. Generate the population using a random strategy with the two-layer coding method. Construct an opposite population via opposition-based learning, merge the two populations, decode using the time-driven two-stage decoding strategy, and calculate the individual fitness value.

Step 3: Perform non-dominated sorting and crowding distance calculation on the initial wolf pack. Select the top  $n$  individuals to form the initial population, and store the Pareto optimal solutions in an external archive.

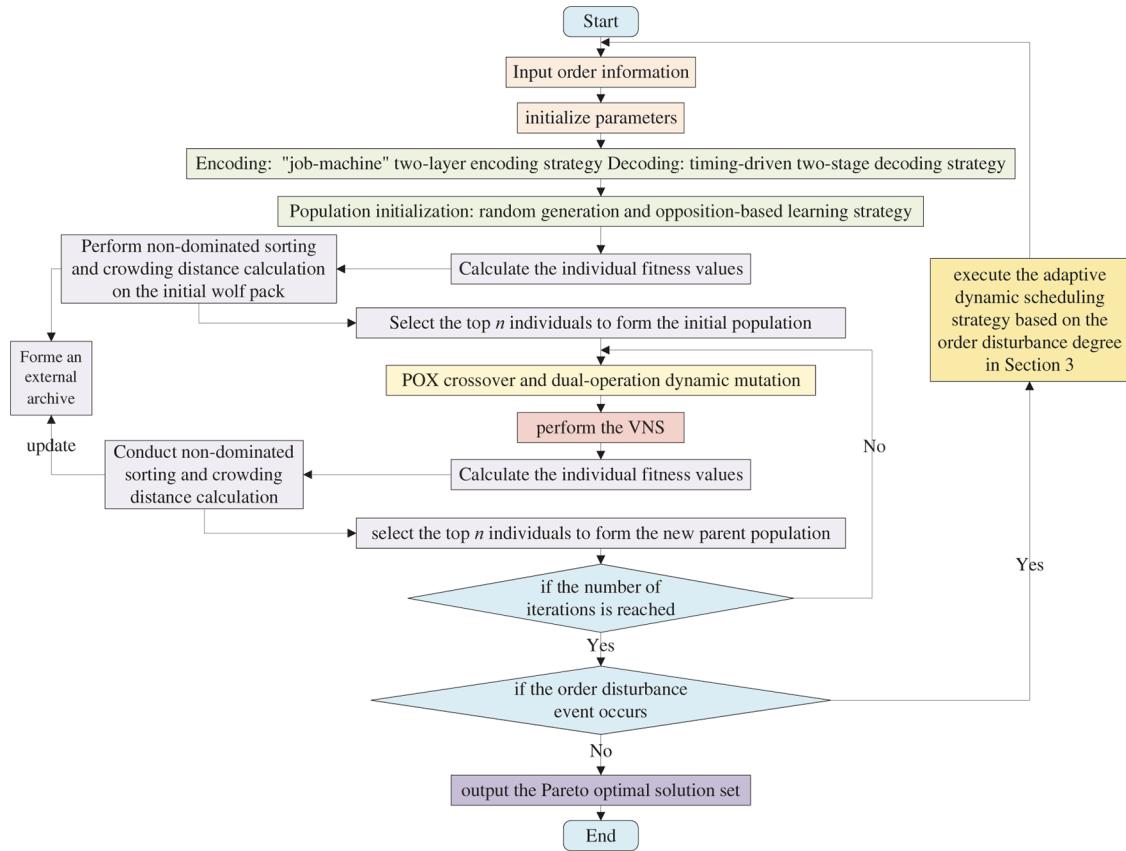
Step 4: Randomly select two parent chromosomes and perform POX crossover and dual-operation dynamic mutation.

Step 5: Partial individuals ( $\alpha$ ,  $\beta$ , and  $\delta$ ) in the population perform the VNS.

Step 6: Calculate the individual fitness values, conduct non-dominated sorting and crowding distance calculation, update the external archive, and select the top  $n$  individuals to form the new parent population.

Step 7: Judge if the number of iterations is reached; if so, execute step 8, otherwise, execute steps 4–6.

Step 8: Judge if the order disturbance event occurs; if so, execute the adaptive dynamic scheduling strategy based on the order disturbance degree in [Section 3](#); otherwise, output the Pareto optimal solution set.



**Figure 3:** Flowchart of IMOGWO algorithm

## 5 Experimental Simulation and Case Analysis

### 5.1 Experiment Design and Parameter Setting

The algorithm proposed in this paper is implemented in MATLAB R2022b, running on a Windows 11 operating system with an Intel(R) Core(TM) i5-12500H processor (2.50 GHz) and 16 GB RAM. To verify the effectiveness of the algorithms in scenarios of different sizes and complexities, 12 sets of test cases are designed. The parameters of the test cases are as follows: the number of jobs  $n \in \{10, 20, 30, 50\}$ , referring to the TAillard test set to cover different scales [25]; the number of operations  $j \in \{3, 5, 8\}$ , corresponding to the combination of less-to-many-stage processes, which is in line with the logic of stage division in the domain-classical test sets FT10 and Mk04 [26]; the processing data is generated randomly, where the number of parallel machines  $e$  is an integer between the interval [2, 3], covering the typical capacity configuration of a hybrid flow processing shop; the processing time of each operation is an integer between the interval

[1, 10], and the delivery date as an integer in the interval [5, 15] to simulate order urgency in tight, normal, and loose layers.

### 5.2 Selection of Evaluation Metrics

In multi-objective optimization, algorithm performance is primarily reflected in the convergence, uniformity, and distribution of the Pareto front solutions. This study employs the Inverted Generational Distance (IGD) and Hypervolume Indicator (HV) as performance evaluation metrics. IGD represents the average distance from each point on the true Pareto front to the nearest point on the obtained Pareto front, calculated by summing and averaging the distances between these points to evaluate the algorithm's convergence and diversity. Notably, lower IGD values suggest superior comprehensive performance [27]. HV denotes the volume of the objective space bounded by the non-dominated solution set and a dominated reference point (worst point), representing the coverage of non-dominated solutions in the objective space to comprehensively assess convergence and diversity. Here, larger HV values denote improved comprehensive performance [28]. The true Pareto front is constructed by combining all non-dominated solutions from all algorithms and removing dominated solutions from the combined set.

### 5.3 Validation of the Effectiveness of Improved Strategies

To effectively verify the improvement mechanisms of the IMOGWO algorithm, this paper conducts ablation experiments on the improved components of the algorithm, using IGD and HV values as evaluation metrics. In Tables 3 and 4, IMOGWO and three variant algorithms, along with their corresponding results, are listed. Among them, IMOGWO1, IMOGWO2, and IMOGWO3 correspond to the IMOGWO algorithm modified by removing the opposition-based learning initialization population strategy, the POX crossover and dual-operation dynamic mutation strategy, and the VNS strategy, respectively. In the experiments, the population size of the algorithm is set to 50, the maximum number of iterations is 200, and the capacity of the external archive is 50. Parameter A is linearly decreased from 2 to 0, and parameter C is randomly generated within the interval [0, 2]. The crossover rate and mutation rate for IMOGWO, IMOGWO1, and IMOGWO3 are 0.8 and 0.1, respectively. All other settings follow those in Reference [23]. To avoid the randomness of the algorithm, each test case is run 10 times in a loop, with the optimal values highlighted in bold.

**Table 3:** Comparison of IGD results in ablation experiments

Test cases n × j	Equipment layout	IMOGWO			IMOGWO1			IMOGWO2			IMOGWO3		
		Best	Ave.	Std.	Best	Ave.	Std.	Best	Ave.	Std.	Best	Ave.	Std.
10 × 3	2, 2, 3	0.0721	0.0753	<b>0.0025</b>	0.0853	0.0882	0.0028	0.0889	0.0917	0.0032	<b>0.0708</b>	<b>0.0731</b>	0.0026
10 × 5	2, 3, 2, 2, 2	<b>0.0984</b>	<b>0.1009</b>	<b>0.0022</b>	0.1147	0.1173	0.0025	0.1162	0.1195	0.0028	0.1109	0.1137	0.0024
10 × 8	2, 2, 3, 2, 2, 2, 3, 2	<b>0.1276</b>	<b>0.1302</b>	<b>0.0024</b>	0.1428	0.1458	0.0026	0.1379	0.1401	0.0026	0.1405	0.1439	0.0025
20 × 3	2, 3, 2	<b>0.0908</b>	<b>0.0936</b>	<b>0.0022</b>	0.1045	0.1079	0.0028	0.1057	0.1082	0.0027	0.1026	0.1053	0.0024
20 × 5	3, 2, 3, 2, 2	<b>0.1153</b>	<b>0.1184</b>	<b>0.0026</b>	0.1316	0.1341	0.0028	0.1301	0.1329	0.0027	0.1332	0.1362	0.0028
20 × 8	2, 3, 2, 2, 2, 3, 2, 2	<b>0.1493</b>	<b>0.1523</b>	<b>0.0028</b>	0.1732	0.1769	0.0032	0.1714	0.1742	0.0030	0.1756	0.1783	0.0029
30 × 3	3, 3, 2	<b>0.1072</b>	<b>0.1107</b>	<b>0.0026</b>	0.1261	0.1291	0.0028	0.1234	0.1263	0.0027	0.1246	0.1278	0.0027
30 × 5	3, 2, 2, 3, 2	<b>0.1337</b>	<b>0.1365</b>	<b>0.0026</b>	0.1502	0.1537	0.0030	0.1479	0.1502	0.0028	0.1519	0.1543	0.0028
30 × 8	2, 3, 3, 2, 3, 2, 2, 2	<b>0.1709</b>	<b>0.1731</b>	<b>0.0026</b>	0.1935	0.1964	0.0030	0.1980	0.2013	0.0032	0.1968	0.1998	0.0029
50 × 3	3, 2, 3	<b>0.1200</b>	<b>0.1238</b>	<b>0.0028</b>	0.1376	0.1403	0.0030	0.1358	0.1387	0.0030	0.1382	0.1412	0.0029
50 × 5	3, 3, 2, 3, 2	<b>0.1538</b>	<b>0.1562</b>	<b>0.0024</b>	0.1791	0.1824	0.0030	0.1774	0.1801	0.0029	0.1816	0.1847	0.0028
50 × 8	2, 2, 3, 2, 3, 2, 2, 3	<b>0.1971</b>	<b>0.2005</b>	<b>0.0028</b>	0.2234	0.2263	0.0030	0.2209	0.2238	0.0029	0.2253	0.2284	0.0029
Average value		<b>0.1280</b>	<b>0.1310</b>	<b>0.0025</b>	0.1468	0.1499	0.0029	0.1461	0.1489	0.0029	0.1460	0.1489	0.0027

**Table 4:** Comparison of HV results in ablation experiments

Test cases $n \times j$	Equipment layout	IMOGWO			IMOGWO1			IMOGWO2			IMOGWO3		
		Best	Ave.	Std.	Best	Ave.	Std.	Best	Ave.	Std.	Best	Ave.	Std.
10 × 3	2, 2, 3	0.0782	0.0751	<b>0.0025</b>	0.0768	0.0730	0.0028	0.0757	0.0726	0.0030	<b>0.0785</b>	<b>0.0754</b>	0.0027
10 × 5	2, 3, 2, 2, 2	<b>0.0735</b>	<b>0.0700</b>	<b>0.0022</b>	0.0706	0.0671	0.0025	0.0694	0.0666	0.0027	0.0703	0.0673	0.0024
10 × 8	2, 2, 3, 2, 2, 2, 3, 2	<b>0.0668</b>	<b>0.0629</b>	<b>0.0024</b>	0.0625	0.0594	0.0026	0.0639	0.0604	0.0028	0.0627	0.0597	0.0025
20 × 3	2, 3, 2	<b>0.0734</b>	<b>0.0705</b>	<b>0.0022</b>	0.0709	0.0677	0.0027	0.0706	0.0675	0.0029	0.0718	0.0680	0.0026
20 × 5	3, 2, 3, 2, 2	<b>0.0672</b>	<b>0.0646</b>	<b>0.0024</b>	0.0647	0.0613	0.0027	0.0642	0.0615	0.0028	0.0645	0.0612	0.0026
20 × 8	2, 3, 2, 2, 2, 3, 2, 2	<b>0.0609</b>	<b>0.0574</b>	<b>0.0025</b>	0.0578	0.0542	0.0028	0.0573	0.0545	0.0030	0.0576	0.0541	0.0027
30 × 3	3, 3, 2	<b>0.0685</b>	<b>0.0658</b>	<b>0.0023</b>	0.0653	0.0625	0.0026	0.0657	0.0628	0.0027	0.0659	0.0626	0.0025
30 × 5	3, 2, 2, 3, 2	<b>0.0624</b>	<b>0.0595</b>	<b>0.0024</b>	0.0596	0.0564	0.0027	0.0598	0.0567	0.0029	0.0592	0.0562	0.0026
30 × 8	2, 3, 3, 2, 3, 2, 2, 2	<b>0.0550</b>	<b>0.0524</b>	<b>0.0022</b>	0.0529	0.0496	0.0025	0.0520	0.0491	0.0027	0.0524	0.0491	0.0024
50 × 3	3, 2, 3	<b>0.0639</b>	<b>0.0600</b>	<b>0.0023</b>	0.0594	0.0569	0.0026	0.0605	0.0572	0.0028	0.0593	0.0569	0.0025
50 × 5	3, 3, 2, 3, 2	<b>0.0567</b>	<b>0.0539</b>	<b>0.0022</b>	0.0537	0.0508	0.0025	0.0535	0.0509	0.0027	0.0532	0.0506	0.0024
50 × 8	2, 2, 3, 2, 3, 2, 2, 3	<b>0.0496</b>	<b>0.0467</b>	<b>0.0021</b>	0.0465	0.0436	0.0024	0.0468	0.0438	0.0026	0.0463	0.0435	0.0023
Average value		<b>0.0647</b>	<b>0.0616</b>	<b>0.0023</b>	0.0617	0.0585	0.0026	0.0616	0.0586	0.0028	0.0618	0.0587	0.0025

From the comparison results in Tables 3 and 4, it can be observed that the optimal value, average value, and standard deviation of IGD obtained by IMOGWO are 0.1280, 0.1310, and 0.0025, respectively; meanwhile, the optimal value, average value, and standard deviation of HV achieved by IMOGWO are 0.0647, 0.0616, and 0.0023, respectively. All of these metrics are superior to those of the other three algorithms. In the small-scale  $10 \times 3$  test cases, the optimal values and average values of IGD and HV under the IMOGWO3 algorithm are superior to those of IMOGWO, while IMOGWO exhibits better stability. This is because the solution space of this instance is simple, and the global optimum is easily reached by the initial search, while the variable neighborhood search in IMOGWO fails to achieve optimal performance due to redundant computations and the risk of overfitting. On the whole, in most test cases, the optimal values, average values, and standard deviations of IGD and HV for the IMOGWO algorithm are all superior to those of the three variant algorithms with one improved strategy removed. These results verify the effectiveness of each improved strategy.

#### 5.4 Performance Analysis of IMOGWO Algorithm

To analyze the performance of the IMOGWO algorithm in solving the hybrid flow shop multi-objective scheduling optimization model, its operational results are compared with mature and classical MOGWO, MOPSO, and NSGA-II algorithms [23,29,30]. The parameter settings of these algorithms are presented in Table 5. To avoid algorithmic randomness, each test case is run 10 times in cycles, and the maximum, minimum, and average values of the indicators are taken as the final results. Table 6 presents the comparison results of IGD for the four algorithms. Table 7 presents the comparison results of HV for the four algorithms. The optimal value is displayed in bold. Fig. 4 presents the boxplots of IGD and HV for the four algorithms.

**Table 5:** Algorithm parameters

Algorithm	Population size	Maximum number of iterations	Parameters
IMOGWO	50	200	Parameter A is linearly decreased from 2 to 0; parameter C is randomly generated within the interval [0, 2]; the crossover rate is set to 0.8; the mutation rate is set to 0.1; and the capacity of the external archive is 50.

(Continued)

**Table 5 (continued)**

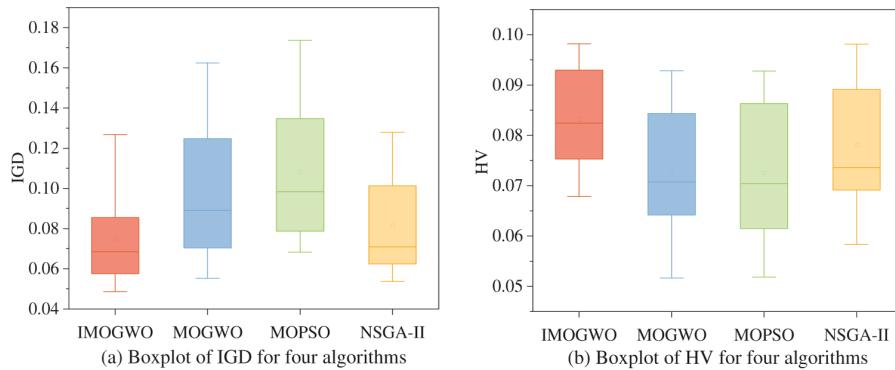
Algorithm	Population size	Maximum number of iterations	Parameters
MOGWO	50	200	Parameter A is linearly decreased from 2 to 0; parameter C is randomly generated within the interval [0, 2]; and the capacity of the external archive is 50.
NSGA-II	50	200	The crossover distribution index $\eta_c = 20$ ; the mutation distribution index $\eta_m = 20$ ; the crossover rate is 0.8; the mutation rate is 0.1.
MOPSO	50	200	The learning factors are set as $c_1 = 2$ and $c_2 = 2$ ; the inertia weight $w$ is linearly decreased from 0.9 to 0.4; and the capacity of the external archive is 50.

**Table 6:** Comparison of IGD values

Test cases n × j	Equipment layout	IMOGWO			MOGWO			NSGA-II			MOPSO		
		Best	Ave.	Std.	Best.	Ave.	Std.	Best	Ave.	Std.	Best	Ave.	Std.
10 × 3	2, 2, 3	<b>0.0467</b>	<b>0.0522</b>	<b>0.0043</b>	0.0571	0.0644	0.0057	0.0515	0.0583	0.0049	0.0624	0.0718	0.0064
10 × 5	2, 3, 2, 2, 2	<b>0.0432</b>	<b>0.0486</b>	<b>0.0036</b>	0.0508	0.0553	0.0046	0.0491	0.0537	0.0037	0.0619	0.0683	0.0053
10 × 8	2, 2, 3, 2, 2, 2, 3, 2	<b>0.0528</b>	<b>0.0590</b>	<b>0.0041</b>	0.0695	0.0784	0.0052	0.0560	0.0642	0.0043	0.0763	0.0881	0.0062
20 × 3	2, 3, 2	<b>0.0580</b>	<b>0.0653</b>	<b>0.0044</b>	0.0759	0.0852	0.0061	0.0645	0.0727	0.0047	0.0834	0.0954	0.0071
20 × 5	3, 2, 3, 2, 2	0.0693	0.0786	0.0052	0.0825	0.0929	0.0067	<b>0.0542</b>	<b>0.0613</b>	<b>0.0041</b>	0.0897	0.1014	0.0076
20 × 8	2, 3, 2, 2, 3, 2, 2	<b>0.0639</b>	<b>0.0715</b>	<b>0.0046</b>	0.0926	0.1057	0.0072	0.0793	0.0893	0.0056	0.0994	0.1135	0.0081
30 × 3	3, 3, 2	<b>0.0543</b>	<b>0.0615</b>	<b>0.0039</b>	0.0658	0.0733	0.0056	0.0602	0.0690	0.0048	0.0729	0.0824	0.0063
30 × 5	3, 2, 2, 3, 2	<b>0.0903</b>	<b>0.1025</b>	<b>0.0061</b>	0.1279	0.1456	0.0092	0.1024	0.1162	0.0074	0.1386	0.1581	0.0102
30 × 8	2, 3, 3, 2, 3, 2, 2	<b>0.0507</b>	<b>0.0561</b>	<b>0.0034</b>	0.0592	0.0675	0.0047	0.0569	0.0635	0.0041	0.0663	0.0750	0.0056
50 × 3	3, 2, 3	<b>0.0810</b>	<b>0.0924</b>	<b>0.0059</b>	0.1036	0.1180	0.0082	0.0867	0.0974	0.0071	0.1118	0.1266	0.0093
50 × 5	3, 3, 2, 3, 2	<b>0.0689</b>	<b>0.0773</b>	<b>0.0051</b>	0.1153	0.1315	0.0094	0.0917	0.1052	0.0079	0.1254	0.1428	0.0106
50 × 8	2, 2, 3, 2, 3, 2, 2, 3	<b>0.1114</b>	<b>0.1268</b>	<b>0.0079</b>	0.1437	0.1624	0.0101	0.1125	0.1279	0.0089	0.1526	0.1737	0.0112
Average value		<b>0.0659</b>	<b>0.0743</b>	<b>0.0049</b>	0.0870	0.0984	0.0069	0.0721	0.0816	0.0056	0.0951	0.1081	0.0078

**Table 7:** Comparison of HV values

Test cases n × j	Equipment layout	IMOGWO			MOGWO			NSGA-II			MOPSO		
		Best	Ave.	Std.	Best	Ave.	Std.	Best	Ave.	Std.	Best	Ave.	Std.
10 × 3	2, 2, 3	<b>0.1025</b>	<b>0.0982</b>	<b>0.0027</b>	0.0984	0.0928	0.0036	0.1017	0.0966	0.0033	0.0976	0.0918	0.0039
10 × 5	2, 3, 2, 2, 2	0.1003	<b>0.0956</b>	<b>0.0026</b>	0.0935	0.0842	0.0053	<b>0.1038</b>	0.0947	0.0046	0.1024	0.0928	0.0059
10 × 8	2, 2, 3, 2, 2, 2, 3, 2	<b>0.0967</b>	<b>0.0910</b>	<b>0.0031</b>	0.0739	0.0657	0.0054	0.0925	0.0835	0.0051	0.0732	0.0653	0.0053
20 × 3	2, 3, 2	<b>0.0958</b>	<b>0.0855</b>	<b>0.0043</b>	0.0805	0.0722	0.0051	0.0903	0.0817	0.0046	0.0587	0.0518	0.0041
20 × 5	3, 2, 3, 2, 2	0.1054	0.0950	<b>0.0036</b>	0.0983	0.0885	0.0049	<b>0.1086</b>	<b>0.0981</b>	0.0043	0.0986	0.0884	0.0056
20 × 8	2, 3, 2, 2, 3, 2, 2	<b>0.0985</b>	<b>0.0883</b>	<b>0.0041</b>	0.0947	0.0845	0.0053	0.0802	0.0721	0.0046	0.0942	0.0843	0.0051
30 × 3	3, 3, 2	<b>0.0865</b>	<b>0.0771</b>	0.0044	0.0583	0.0517	0.0041	0.0774	0.0693	<b>0.0043</b>	0.0786	0.0699	0.0046
30 × 5	3, 2, 2, 3, 2	<b>0.0826</b>	0.0735	<b>0.0041</b>	0.0753	0.0667	0.0051	0.0835	<b>0.0741</b>	0.0049	0.0794	0.0710	0.0053
30 × 8	2, 3, 3, 2, 3, 2, 2, 2	<b>0.0889</b>	<b>0.0787</b>	0.0043	0.0775	0.0693	0.0046	0.0770	0.0690	<b>0.0041</b>	0.0678	0.0598	0.0043
50 × 3	3, 2, 3	<b>0.0767</b>	<b>0.0683</b>	<b>0.0041</b>	0.0705	0.0627	0.0046	0.0748	0.0664	0.0043	0.0710	0.0632	0.0049
50 × 5	3, 3, 2, 3, 2	<b>0.0764</b>	<b>0.0678</b>	0.0043	0.0648	0.0568	0.0041	0.0653	0.0583	<b>0.0039</b>	0.0615	0.0546	0.0046
50 × 8	2, 2, 3, 2, 3, 2, 2, 3	<b>0.0890</b>	<b>0.0793</b>	<b>0.0046</b>	0.0876	0.0781	0.0053	0.0817	0.0731	0.0049	0.0864	0.0769	0.0056
Average value		<b>0.0916</b>	<b>0.0832</b>	<b>0.0039</b>	0.0811	0.0728	0.0048	0.0864	0.0781	0.0044	0.0808	0.0725	0.0049

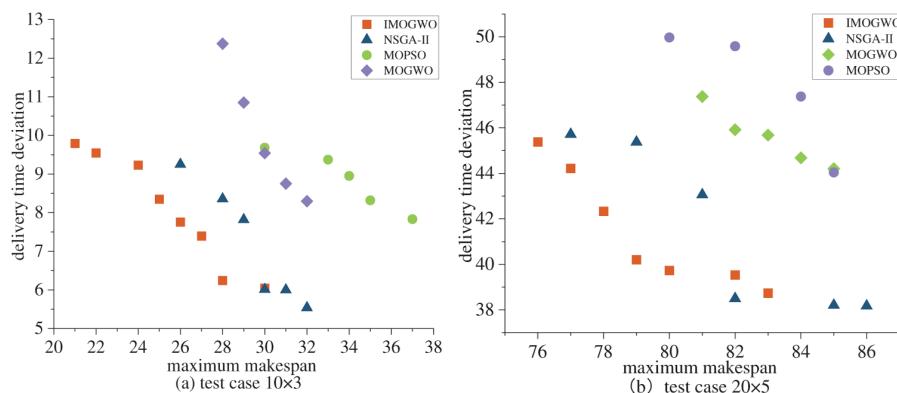


**Figure 4:** Boxplots of four algorithms

Analysis based on Tables 6 and 7 reveals that the optimal value, average value, and standard deviation of IGD obtained by IMOGWO are 0.0659, 0.0743, and 0.0049, respectively; meanwhile, the optimal value, average value, and standard deviation of HV achieved by IMOGWO are 0.0916, 0.0832, and 0.0039, respectively. All of these metrics are superior to those of the other three algorithms. NSGA-II exhibits a certain degree of competitiveness only in a small number of test cases, and its overall performance is slightly inferior to that of IMOGWO. Meanwhile, MOGWO and MOPSO perform poorly in terms of average IGD and average HV, and they also show insufficient stability. Thus, it can be concluded that the IMOGWO algorithm demonstrates superior comprehensive performance compared to the other three algorithms in solving the HFSP.

As shown in Fig. 4, the boxplots of IMOGWO are flatter than those of the other three algorithms, indicating smaller variance and better stability. The IGD boxplot reveals that IMOGWO has a lower median line than the other three algorithms. Although the difference from NSGA-II is marginal, IMOGWO's boxplot exhibits significantly greater flatness. The HV boxplot shows that IMOGWO has a notably higher median line than the other three algorithms. These indicate that the non-dominated solution set obtained by IMOGWO is of better quality.

A comparative analysis of the Pareto fronts for the four algorithms is conducted using a random selection of test cases from the 12 instances, as illustrated in Fig. 5. It can be seen that the non-dominated solution set obtained by IMOGWO is located in the lower left of the rest of the algorithms, which is closer to the approximate optimal Pareto front. This indicates that IMOGWO achieves superior accuracy than the comparative algorithms in solving the hybrid flow shop multi-objective scheduling problem.



**Figure 5:** Comparison of Pareto fronts for 4 algorithms

### 5.5 Case Analysis

The gear processing shop of a certain enterprise was taken as the research object for analysis. The initial product requirements for gears within a week are shown in [Table 8](#), and the processing times for each gear operation are listed in [Table 9](#). The rated processing time for each machine in the enterprise's production shop is 8 h per day, and the number of gear production batches and batch sizes are determined based on the shop's historical experience.

**Table 8:** Initial product requirements for gears

Order number	Order type	Number of requirements (pcs)	Delivery date (days)	First batch (pcs)	Second batch (pcs)	Third batch (pcs)
1	M001	100	5	50	50	0
2	M002	230	6	100	130	0
3	M003	280	6	100	100	80
4	M004	120	7	60	60	0
5	M005	270	6	150	120	0
6	M006	300	6	150	150	0
7	M007	240	6	100	140	0
8	M008	120	5	120	0	0
9	M009	355	7	160	195	0
10	M0010	140	7	70	70	0

**Table 9:** Processing times for each gear operation

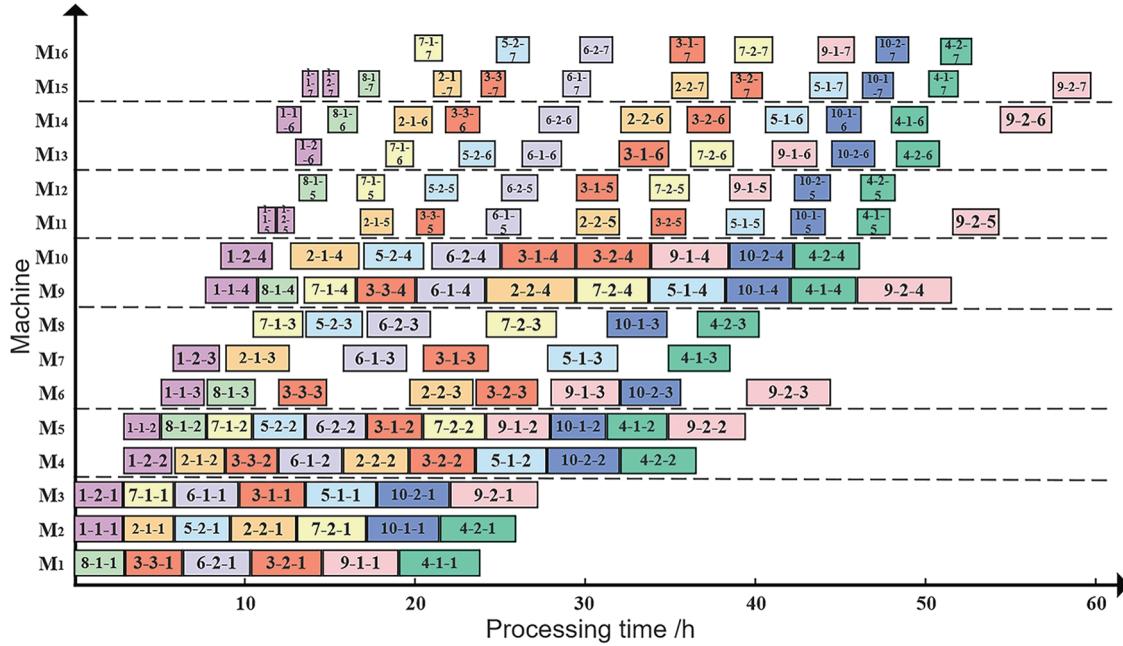
Order type	Rough turning (s/pc)					Finish turning (s/pc)					Hobbing (s/pc)			Chamfering (s/pc)		Gear shaping (s/pc)		Gear shaving (s/pc)		Gear honing (s/pc)	
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16					
M001	218	205.8	205.8	205.8	157.8	184.4	200	200	222.8	222.8	80	97.9	115.9	106.7	73.9	86					
M002	110.4	107.3	107.3	107.3	96.9	104.2	137.6	137.6	146.6	146.6	71.9	76.6	88.5	83.1	61	66					
M003	151	138.7	138.7	138.7	117.71	131.9	139.5	139.5	157.5	157.1	74.3	88.6	107.3	93.6	69.2	74.3					
M004	290.5	267.5	267.5	267.5	218.6	231.9	223.1	223.1	231.6	231.6	119.1	124.6	156.6	131.9	107.3	112.8					
M005	106.7	100.4	100.4	100.4	93.9	102.2	101.9	101.9	107.8	107.8	54.2	58.6	64.7	62.2	56.2	60.2					
M006	95.8	91.5	91.5	91.5	85.5	91.5	90	90	97.5	97.5	50.4	53.8	58.7	56.7	41	46.8					
M007	114.1	106.7	106.7	106.7	96.1	103	107	107	110.7	110.7	59.8	62.3	65.7	62.6	55.3	60.1					
M008	91.2	86.8	86.8	86.8	80.7	86.5	88.9	88.9	96	96	49.8	51.6	56.4	52.9	38.8	45.7					
M009	102.2	95.6	95.6	95.6	85.3	91.4	98.1	98.1	102.8	103.2	52.6	56.5	59.8	56.6	43	49.3					
M0010	242.5	219.6	219.6	219.6	170.4	184	187.2	187.2	195.7	195.7	107.2	112.7	132.7	108	95.4	100.9					

Based on the above information, the static scheduling problem in the first phase was solved. Considering the actual production needs of the enterprise, the initial scheduling Gantt chart is generated by selecting the scheme with the minimum maximum makespan as shown in [Fig. 6](#). The products of different orders are represented by different colors, the batch of the order is represented by the second digit, and the operation is represented by the third digit. For example, 6-2-1 indicates the 1st operation of the 2nd batch of jobs in order 6.

After scheduling using the model and algorithm proposed in this paper, all the operations are better scheduled. The maximum makespan is 59.8h, and the delivery time deviation is 12.81 h.

On this basis, the dynamic scheduling under order disturbance in the second phase was verified. Two order disturbance scenarios with different degrees, namely order insertion and order expediting, were

designed. Order cancellation is simply eliminated from the scheduling plan, and the remaining unfinished processes are shifted left, which is not shown here due to space constraints.



**Figure 6:** Initial scheduling Gantt chart

### (1) Order insertion

At the 23rd hour, new order 11 (product type M008, quantity 80, delivery date 32 h) and order 12 (product type M001, quantity 50, delivery date 32 h) are received. According to Eqs. (16) and (17), the results are  $\beta_1, \beta_2 = 1.03125$ ,  $\alpha_1, \alpha_2 = 1.5$ ,  $\sum_{k=1}^{h_1} \bar{T}_{1,k} = 11.15$ ,  $\sum_{k=1}^{h_2} \bar{T}_{2,k} = 15.183$ .  $\sum_{k=1}^{h_1} \bar{T}_{1,k}$  and  $\sum_{k=1}^{h_2} \bar{T}_{2,k}$  are respectively the sum of the average processing times of new order 11 and order 12 at each operation. The processing time data is derived from Table 8, and the time unit needs to be converted to hours. The order disturbance degree  $S$  calculated from Eq. (15) is as follows:

$$S = \frac{1.5 \times 1.03125 \times 11.15 + 1.5 \times 1.03125 \times 15.183}{59.8 - 23} \times 100\% = 110.7\% > 100\%$$

It shows that the order insertion disturbance at this time has a greater impact on the production line, and the original scheduling plan has been difficult to support the current production requirements. Thus, the event-triggered scheduling strategy is invoked to implement dynamic scheduling. To reduce unnecessary scheduling adjustment costs and maintain production stability, partial rescheduling is performed based on the original scheduling plan. The dynamic scheduling Gantt chart for job insertion is shown in Fig. 7a. Inserting new orders during machine idle time effectively addresses the disturbance caused by order insertion. The completion times of Order 11 and Order 12 are 52.9 and 55.3 h, respectively, with delivery times of 29.9 and 32.3 h. Order 11 meets the delivery date of 32 h, whereas Order 12 exceeds the delivery date by only 0.3 h. The maximum makespan of all orders is 59.8 h, identical to that of the initial scheduling scheme. The delivery time deviation is 11.71 h, and the schedule deviation degree is 14.

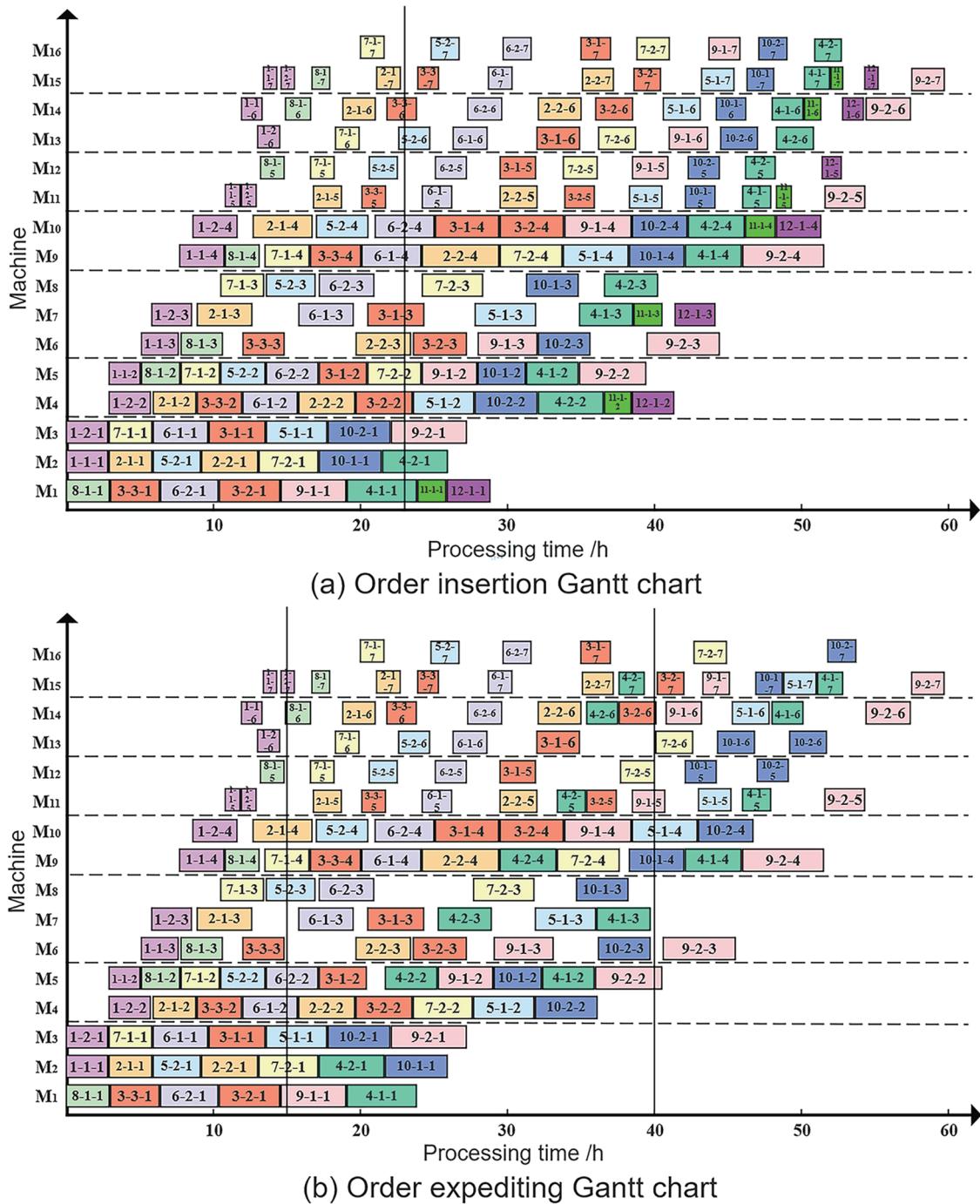


Figure 7: Order disturbance Gantt chart

## (2) Order expediting

At the 15th hour, the delivery date of 60 jobs in order 4 is shortened from 56 to 40 h. In the initial scheduling scheme, the completion time of order 4 is 52.8 h, which exceeds the new delivery date of 40 h.

According to Eqs. (16) and (17), the results are  $\beta_1 = 1.025$ ,  $\alpha_1 = 2$ ,  $\sum_{k=1}^{h_1} \bar{T}_{1,k} = 22.617$ .  $\sum_{k=1}^{h_1} \bar{T}_{1,k}$  represents the

sum of the average processing times of expediting order 4 across each operation. The processing time data were derived from [Table 6](#), with the time unit converted to hours. The order disturbance degree  $S$  calculated from [Eq. \(15\)](#) is as follows:

$$S = \frac{2 \times 1.025 \times 22.617}{59.8 - 15} \times 100\% = 103.5\% > 100\%$$

The order expediting disturbance at this stage indicates a significant impact on the production line. Similarly, the original scheduling plan has been difficult to support the current production requirements, so the event-triggered scheduling strategy is invoked to implement dynamic scheduling. Partial rescheduling is insufficient to meet the current production demands, necessitating complete rescheduling. The dynamic scheduling Gantt chart for the order expediting scenario is shown in [Fig. 7b](#). The delivery time of 60 jobs in order 4 is 39.4 h, satisfying the expediting requirement. The maximum makespan of all orders remains 59.8 h, consistent with the initial scheduling scheme. The delivery time deviation is 11.84 h, and the schedule deviation degree is 203.3.

The scheduling results comparison for order disturbance events is shown in [Table 10](#), with optimal solutions in boldface. The adaptive dynamic scheduling strategy proposed in this study is denoted as strategy A, and the complete rescheduling strategy is referred to as strategy B. Under order insertion disturbance, although the maximum makespan of the IMOGWO algorithm with strategy A is 0.34% higher than that of strategy B, the delivery time deviation and schedule deviation degree are both lower; specifically, the scheme deviation degree is 83.6% lower. When strategy B is adopted, the three objective function values under the IMOGWO algorithm are all smaller than those of the other three algorithms. Under order expediting disturbance, the IMOGWO algorithm combined with strategy A achieves the best values for all three objective functions compared with the other four comparison cases. Under both disturbance scenarios, due to the additional improved strategies incorporated into the IMOGWO algorithm compared to MOGWO, its runtime has slightly increased, but it still falls within the allowable range for actual shop scheduling. Moreover, in comparison with the other two algorithms, NSGA-II and MOPSO, its runtime is shorter.

**Table 10:** Comparison of order disturbance event scheduling results

Order disturbance	Scheduling method	Maximum makespan (h)	Delivery time deviation (h)	Scheme deviation degree	Runtime (s)
Order insertion	IMOGWO and strategy A	59.8	<b>11.71</b>	<b>14</b>	295.7
	IMOGWO and strategy B	<b>59.6</b>	12.24	85.2	312.5
	MOGWO and strategy B	62.5	13.17	97.4	<b>246.5</b>
	NSGA-II and strategy B	61.3	12.53	90.2	343.1
	MOPSO and strategy B	61.9	13.63	102.5	352.9
Order expediting	IMOGWO and strategy A	<b>59.8</b>	<b>11.84</b>	<b>203.3</b>	322.4
	IMOGWO and strategy B	60.2	12.35	210.5	330.3
	MOGWO and strategy B	61.3	14.26	226.9	<b>284.1</b>
	NSGA-II and strategy B	60.9	13.74	216.4	357.0
	MOPSO and strategy B	61.8	13.98	228.7	372.8

Case validations demonstrate that the adaptive dynamic scheduling strategy based on order disturbance degree and the IMOGWO algorithm proposed in this paper can effectively address order disturbance events, ensuring both rapid response to order disturbance and stability of the production system.

## 6 Conclusion and Future Works

This paper addresses the dynamic scheduling problem of hybrid flow shops under order disturbances and establishes a dynamic scheduling model by comprehensively considering three objectives: maximum makespan, delivery time deviation, and scheme deviation degree. To avoid excessive or insufficient scheduling times in actual production, an adaptive dynamic scheduling strategy based on order disturbance degree and an IMOGWO algorithm is proposed for the solution. Specifically, an opposition-based learning strategy is introduced to initialize the population to ensure population diversity; aiming at the problem that the algorithm tends to fall into local optima during solution, the POX Crossover strategy, dual-operation dynamic mutation strategy, and the VNS strategy are incorporated.

Experimental results demonstrate the efficacy and comprehensive performance of the IMOGWO algorithm and model in addressing the multi-objective HFSP. Instance verification for different order disturbance scenarios indicates that the IMOGWO algorithm, combined with the adaptive dynamic scheduling strategy and model, possesses significant advantages in addressing order disturbances.

However, current research exhibits limitations in verifying generalization capabilities. Experiments rely on randomly generated instances and single case studies, focusing solely on order disturbances within an overly idealized environment. Therefore, future research will expand the scale and scope of experimental data to thoroughly validate the effectiveness and comprehensive performance of models and algorithms in larger-scale, actual production with diverse configurations. We will further incorporate other dynamic disturbance events present in actual production, such as machine failures. Additionally, we will explore more complex constraints to better align with practical requirements, including transportation constraints, labor constraints, and others.

**Acknowledgement:** None.

**Funding Statement:** This work was funded by National Key Research and Development Program Projects of China under Grant No. 2020YFB1713500.

**Author Contributions:** The authors confirm contribution to the paper as follows: Conceptualization, Feng Lv and Huili Chu; methodology, Huili Chu; software, Huili Chu; validation, Huili Chu; formal analysis, Huili Chu; investigation, Feng Lv and Huili Chu; data curation, Huili Chu; writing—original draft preparation, Huili Chu; writing—review and editing, Feng Lv and Huili Chu; visualization, Huili Chu; supervision, Feng Lv, Cheng Yang and Jiajie Zhang; project administration, Feng Lv and Huili Chu; funding acquisition, Feng Lv. All authors reviewed the results and approved the final version of the manuscript.

**Availability of Data and Materials:** The authors confirm that the data supporting the findings of this study are available within the article.

**Ethics Approval:** Not applicable.

**Conflicts of Interest:** The authors declare no conflicts of interest to report regarding the present study.

## References

1. Ruiz R, Vázquez-Rodríguez JA. The hybrid flow shop scheduling problem. *Eur J Oper Res*. 2010;205(1):1–18. doi:10.1016/j.ejor.2009.09.024.

2. Duan J, Wang J. Robust scheduling for flexible machining job shop subject to machine breakdowns and new job arrivals considering system reusability and task recurrence. *Expert Syst Appl.* 2022;203(1):117489. doi:10.1016/j.eswa.2022.117489.
3. Branke J, Mattfeld DC. Anticipation and flexibility in dynamic scheduling. *Int J Prod Res.* 2005;43(15):3103–29. doi:10.1080/00207540500077140.
4. Yang Y-Y, Qian B, Hu R, Li Z, Zhang Z-Q, Jin H-P, et al. Deep reinforcement learning algorithm incorporating problem characteristics for dynamic multi-objective permutation flow-shop scheduling problem. *Swarm Evol Comput.* 2025;96(3):101973. doi:10.1016/j.swevo.2025.101973.
5. Li X, Guo A, Yin X, Tang H, Wu R, Zhao Q, et al. A Q-learning improved differential evolution algorithm for human-centric dynamic distributed flexible job shop scheduling problem. *J Manuf Syst.* 2025;80(4):794–823. doi:10.1016/j.jmsy.2025.04.001.
6. Shi Z, Si J, Zhang J, Pang Z, Chen H, Ding G. A deep reinforcement learning method based on Hindsight experience replay for multi-objective dynamic job-shop scheduling problem. *Expert Syst Appl.* 2025;284(2–3):127989. doi:10.1016/j.eswa.2025.127989.
7. Liu F, Li X, Lu C, Gong W. Adaptive knowledge-based multi-objective evolutionary algorithm for hybrid flow shop scheduling problems with multiple parallel batch processing stages. *Swarm Evol Comput.* 2025;95(1):101929. doi:10.1016/j.swevo.2025.101929.
8. Wang Z, Liao W, Zhang Y. Rescheduling optimisation of sustainable multi-objective fuzzy flexible job shop under uncertain environment. *Int J Prod Res.* 2024;62(24):8904–20. doi:10.1080/00207543.2024.2354830.
9. Chen R, Wu B, Wang H, Tong H, Yan F. A Q-Learning based NSGA-II for dynamic flexible job shop scheduling with limited transportation resources. *Swarm Evol Comput.* 2024;90(3):101658. doi:10.1016/j.swevo.2024.101658.
10. Zhu N, Gong G, Lu D, Huang D, Peng N, Qi H. An effective reformative memetic algorithm for distributed flexible job-shop scheduling problem with order cancellation. *Expert Syst Appl.* 2024;237(3):121205. doi:10.1016/j.eswa.2023.121205.
11. Wei L, He J, Guo Z, Hu Z. A multi-objective migrating birds optimization algorithm based on game theory for dynamic flexible job shop scheduling problem. *Expert Syst Appl.* 2023;227(1):120268. doi:10.1016/j.eswa.2023.120268.
12. Angel-Bello F, Vallikavungal J, Alvarez A. Fast and efficient algorithms to handle the dynamism in a single machine scheduling problem with sequence-dependent setup times. *Comput Ind Eng.* 2021;152(3):106984. doi:10.1016/j.cie.2020.106984.
13. Zou Y, Lu C, Yin L, Wen X. Multi-level subpopulation-based particle swarm optimization algorithm for hybrid flow shop scheduling problem with limited buffers. *Comput Mater Contin.* 2025;84(2):2305–30. doi:10.32604/CMC.2025.065972.
14. Wang Z-C, Pan Q-K, Gao L, Miao Z-H, Sang H-Y. A variable-representation discrete artificial bee colony algorithm for a constrained hybrid flow shop. *Expert Syst Appl.* 2024;254(3):124349. doi:10.1016/j.eswa.2024.124349.
15. Zeng L, Shi J, Li Y, Wang S, Li W. A strengthened dominance relation NSGA-III algorithm based on differential evolution to solve job shop scheduling problem. *Comput Mater Contin.* 2024;78(1):375–92. doi:10.32604/CMC.2023.045803.
16. Liu Y, As'arry A, Hassan MK, Hairuddin AA, Mohamad H. Review of the grey wolf optimization algorithm: variants and applications. *Neural Comput Appl.* 2024;36(6):2713–35. doi:10.1007/s00521-023-09202-8.
17. Li X, Xie J, Ma Q, Gao L, Li P. Improved gray wolf optimizer for distributed flexible job shop scheduling problem. *Sci China Technol Sci.* 2022;65(9):2105–15. doi:10.1007/s11431-022-2096-6.
18. Lin C, Cao Z, Zhou M. Learning-based grey wolf optimizer for stochastic flexible job shop scheduling. *IEEE Trans Autom Sci Eng.* 2022;19(4):3659–71. doi:10.1109/TASE.2021.3129439.
19. Zheng J, Chen S. A Q-learning multi-objective grey wolf optimizer for the distributed hybrid flowshop scheduling problem. *Eng Optim.* 2025;57(9):2609–28. doi:10.1080/0305215X.2024.2410844.
20. Tosun Ö, Marichelvam MK, Tosun N. A literature review on hybrid flow shop scheduling. *Int J Prod Res.* 2020;12(2):156–94. doi:10.1504/IJAOM.2020.108263.

21. Zhou L, Zhang L, Fang Y. Comparison of production-triggering strategies for manufacturing enterprises under random orders. *Engineering*. 2021;7(6):798–806. doi:10.1016/j.eng.2021.03.012.
22. Cui H, Li X, Gao L, Xiang Z, Zhou W, Yu Y et al. Cooperation-based bi-level rescheduling method for multi-objective distributed hybrid flow shop with unrelated parallel machines under multi-type disturbances. *Int J Comput Integr Manuf*. 2025:1–23. doi:10.1080/0951192X.2025.2478005.
23. Mirjalili S, Saremi S, Mirjalili SM, Coelho LdS. Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization. *Expert Syst Appl*. 2016;47(6):106–19. doi:10.1016/j.eswa.2015.10.039.
24. Alkhateeb F, Abed-alguni BH, Al-rousan MH. Discrete hybrid cuckoo search and simulated annealing algorithm for solving the job shop scheduling problem. *J Supercomput*. 2022;78(4):4799–826. doi:10.1007/s11227-021-04050-6.
25. Taillard E. Benchmarks for basic scheduling problems. *Eur J Oper Res*. 1993;64(2):278–85. doi:10.1016/0377-2217(93)90182-M.
26. Jiang T, Zhang C. Application of grey wolf optimization for solving combinatorial problems: job shop and flexible job shop scheduling cases. *IEEE Access*. 2018;6:26231–40. doi:10.1109/ACCESS.2018.2833552.
27. Fu Y, Gao K, Wang L, Huang M, Liang Y-C, Dong H. Scheduling stochastic distributed flexible job shops using an multi-objective evolutionary algorithm with simulation evaluation. *Int J Prod Res*. 2025;63(1):86–103. doi:10.1080/00207543.2024.2356628.
28. Zheng J, Zhao J, Zhao S. A heuristic method for multi-objective hybrid flow shop scheduling problem with parent-child relationships and space constraints. *Int J Prod Res*. 2025;63(7):2431–55. doi:10.1080/00207543.2024.2403126.
29. Shao X, Liu W, Liu Q, Zhang C. Hybrid discrete particle swarm optimization for multi-objective flexible job-shop scheduling problem. *Int J Adv Manuf Technol*. 2013;67(9):2885–901. doi:10.1007/s00170-012-4701-3.
30. Yu W, Zhang L, Ge N. An adaptive multiobjective evolutionary algorithm for dynamic multiobjective flexible scheduling problem. *Int J Intell Syst*. 2022;37(12):12335–66. doi:10.1002/INT.23090.