International Journal of Multiphysics
Volume 18, No. 3, 2024
ISSN: 1750-9548

Green Rescheduling for Multi-Objective Hybrid flow Shop under Insertion Order Event

Bencong Kou, Tingxin Wen, Tingyu Guan*

School of Business Administration, Liaoning Technical University, Huludao, Liaoning, China

*Corresponding Author.

Abstract

With the increasing level of manufacturing and the rapid development of the market economy, enterprises need to improve the level of service to meet the individual needs of customers. Among them, urgent order is a common problem faced by order-oriented enterprises, but it brings additional profits to enterprises and also causes damage to the stability of production. To address this challenge, this paper takes the hybrid flow shop as the research object, considers the equipment switching strategy, AGV transportation device and the carrying capacity of the shop buffer, establishes a mathematical model with the optimization objectives of minimizing the total energy consumption, minimizing the maximum completion time and minimizing the production cost, and proposes the Improved Red-billed Blue Magpie Optimizer (IRBMO) based on the complete re-scheduling strategy for solving the problem. In this algorithm, according to the characteristics of the hybrid flow shop, a strategy approach based on workpiece-equipment two-layer encoding and decoding is adopted, and a Logistic Chaos Mapping backward learning strategy is proposed to generate the initial population; an adaptive cross-variance operator is introduced to balance the algorithm's search capability. For the emergency order insertion scenario, a comparison test with real workshop data is carried out through the extension of the algorithm, which verifies the feasibility of the IRBMO algorithm in complex combinatorial optimization problems and the superiority of the solving effect. It provides new ideas for the green and efficient production scheduling technology of manufacturing enterprises, and has practical application value for promoting the sustainable development of the production system.

Keywords: Rescheduling, Hybrid flow shop, Urgent orders, Adaptive cross-variation, Red-billed blue magpie optimizer, Multi-objective optimization.

1. Introduction

The “manufacturing sector” serves as the “primary engine” for driving new productivity advancements. The development of new quality productive forces will inevitably require further promotion of the greening of the manufacturing industry, accelerating “eco-friendly transformation” of this industry, bigger and stronger green and low-carbon industries, and provide strong support for the promotion of the new type of industrialization and Chinese-style modernization[1,2]. The development and implementation of a green shop scheduling system is crucial for reducing costs and promoting sustainability in the machinery manufacturing industry[3]. Therefore, optimizing production scheduling for energy efficiency from a shop floor perspective is a pressing challenge that must be addressed as the traditional machinery manufacturing industry undergoes transformation[4].

Green shop scheduling research focuses on both modeling and solution algorithms. The modeling involves two general directions: (1) the shop floor level, which is usually based on optimizing production hours to accomplish energy saving goals. Lu et al [5] constructed a flexible flow shop scheduling model with sequence correlation to improve productivity to reduce energy consumption. Cheng [6] proposed an automated hybrid flow shop energy efficient scheduling by optimizing forging tempering time. Sang et al [7] in the face of increasing customer requirements for product customization, manufacturing products with multi-species, small batch production, by improving the quality of production scheduling solutions to improve productivity and reduce energy consumption, the optimization scheme designed to solve the problem of high-dimensional multi-objective green flexible job shop scheduling, to enhance enterprise production capacity and processing quality. In recent years, scholars through the production of energy consumption for classification and optimization to achieve energy saving. Guo et al. [8], suggested an ultra-low idle state for devices, analyzing idle energy consumption in detail, and proposed setting state switching thresholds to adjust the idle states and reduce energy consumption.(2) Research at the machine and equipment level, including shop floor green scheduling for setting machines on/off and shop floor green scheduling for adjusting machine speeds. Mouzon et al. [9] demonstrated for the first time that the switching strategy saves 80% of the idle energy consumption. Meng et al. [10] designed a variety of on-off energy saving strategies and reduced production energy consumption by moving processes to concentrate idle time. Wei et al. [11] constrained the operation or stopping of equipment based on equipment changeover time with the goal of minimizing non-processing energy consumption, and used experiments to further demonstrate that making full use of the on-off strategy during idle periods reduces idle energy in the shop floor. Wu et al. [12] saved manufacturing energy from the production operation point of view, without violating the constraints, and realized the energy saving in the workshop through the adjustment of the machine speed. Foumani [13] achieved a balance between energy consumption and completion time by adjusting the machining speed to better reflect the actual production conditions of the machinery. Wu et al. [14] found that a low processing speed results in low energy consumption but at the expense of processing time, and a high processing speed is counterproductive; to address this phenomenon, an energy-efficient no-waiting-arrangement flow shop scheduling problem is considered, with the objective of minimizing the completion time and the total energy consumption, and building a dynamic model that can dynamically adjust the processing speed of the machine. Wang et al. [15] approached energy optimization by considering machine load and developing a multi-objective mathematical model that accounted for energy consumption, machine load factor, and completion time.

Solution methods for energy-efficient scheduling models: Fang et al. [16] resolved the replacement flow shop scheduling problem focusing on energy consumption and completion time through mathematical programming. Zhang [17] considered the machine shutdown strategy during idle time to reduce the machine energy consumption and using an efficient gene expression programming algorithm (eGEP) to solve the problem based on the gene expression programming algorithm designed by Ferreira [18]. Chen et al [19] proposed an adaptive simulated annealing genetic algorithm based on reinforcement learning for energy-efficient scheduling of workshops, which increases the reasonableness and accuracy of the solution, taking into account the constraints of complex parameter determination and poor local search capability faced by standard genetic algorithms when dealing with flexible job workshop scheduling.

In the process of developing new quality productivity and promoting the transformation and upgrading of the manufacturing industry, there is a great deal of uncertainty in the manufacturing process, whether in terms of production efficiency, production costs, cleanliness and environmental protection, or in terms of industrial services as the evaluation criteria. Therefore, the use of advanced technology and knowledge to control the uncertainty problem in the production scheduling system [20] is the key to improve the productivity, reduce the energy consumption and production cost of manufacturing enterprises nowadays, and it is also the inevitable trend of the development of the manufacturing industry [21].

Despite these advances, most studies on energy-efficient scheduling focus on static scheduling environments and neglect the influence of unexpected events on the shop floor, making them unsuitable for dynamic scheduling scenarios. Furthermore, common constraints in modern factories, such as transportation equipment and limited capacity buffer zones, are often overlooked in dynamic scheduling research. Consequently, existing methods for energy-saving optimization in shop floor rescheduling face limitations that hinder their applicability to real-world production systems.

Based on this, this paper proposes an improved red-billed blue magpie optimizer (IRBMO) for solving the emergency order insertion problem in a hybrid flow shop. This paper includes three main contributions:

(1) The paper introduces a dynamic scheduling model with the optimization objectives of minimizing the total energy consumption, minimizing the maximum completion time, and minimizing the production cost is established by taking the buffer carrying capacity, AGV transport operation, and equipment switching strategy as constraints.

(2) To achieve a more feasible search effect, the RBMO algorithm incorporates the reverse learning strategy of Logistic Chaos Mapping and an adaptive genetic cross-variance operator. This enhances the initial quality of the workpiece-device two-layer coding sequence, and potential optimized solutions can be generated by adjusting the workpiece-device operation order through the Adaptive Genetic Cross-Variation Operator, thus boosting the local optimization capabilities of the RBMO algorithm.

(3) Various arithmetic case comparisons demonstrate the IRBMO algorithm's superiority over other intelligent optimization methods. The real workshop data verifies that the IRBMO algorithm can efficiently respond to the reasonable planning of dynamic workpieces and realize the maximized productivity of the workshop.

The paper is structured as follows: Section 1, the background of the topic and the related literature of the relevant scholars on this problem are reviewed, and on this basis, the research content and structure of this paper are briefly introduced. Section 2, the rescheduling problem for a mixed flow shop with finite buffers facing order insertion is described; in Section 3, the solution flow of the improved red-billed blue magpie optimizer is described in detail; in Section 4, a detailed description of the example comparison experiments is presented; in Section 5, the analysis of the results of the example validation is demonstrated; and in Section 6, this paper is summarized and future research directions are given.

2. The Limited Buffer Hybrid Flow Shop Rescheduling Problem

2.1. Description of the Rescheduling Problem

In the actual production process, due to the occurrence of disturbing events in the production environment, the original scheduled plan is deviated, and it is difficult for the production line to continue to execute according to the established program. This paper addresses the Hybrid Flow shop Rescheduling Problem-Limited Buffer (HFRP-LB) with the introduction of AGV transportation equipment, while considering the impact of dynamic events of emergency order insertion on scheduling energy consumption, completion time and cost.

The HFRP-LB under the insertion order event can be specifically described as follows: n workpieces $J_i(i=1,2,\cdots n)$ need to be processed. Each workpiece J_i undergoes p processes, and in each process, there are M_j parallel devices with varying capabilities. Each process includes a buffer with a capacity C^k after the processing equipment. At time $T_{n'}$, n' urgent workpiece insertion orders occur. The machining sequence must be adjusted to develop an optimal rescheduling plan to mitigate the impact of these insertion events on shop floor production. Rescheduling objects in the production process can be categorized into three types [22]: processes for newly arrived insertion order workpieces, processes for workpieces not yet started, and processes for workpieces that have begun but are not yet completed.

Numerous factors influence the rescheduling process in an actual production plant. To standardize the scheduling problem, the following assumptions are made:

(1) A complete rescheduling approach is used to deal with order insertion events;

(2) Each processing stage can and can only select one parallel machine for processing;

(3) The machining process must proceed continuously once initiated and cannot be interrupted;

(4) The penalties for delayed delivery are the same for the initial and insertion order workpieces, with no priority;

(5) The original workpiece and equipment can start production at zero moment;

(6) An AGV (Automated Guided Vehicle) is capable of transporting only one workpiece at a time, and each workpiece can be transported by only one AGV;

(7) Same transport operation for each workpiece;

(8) The buffer maintenance cost includes only the AGV transportation cost and the buffer capacity management cost;

(9) The workpieces follow the first-in-first-out rule in the buffer zone.

To provide a precise description of the solution problem, the symbols and definitions of the relevant parameters used in this paper are presented in Table 1.

Table 1 Related Parameters.

Symbol	Definition
A	An infinite positive number
C^k	Buffer capacity of machine k
TE_i^j	Completion time of the jth process for workpiece i
tb^k	Time workpiece i spends in the buffer of machine k
n	Number of workpieces
p	Number of processes
m	Number of machines
W	Set of all workpieces
P	Set of all processes
M_j	Number of machines available for the jth process
d_i^d	Earliest delivery time for workpiece i
d_i^u	Latest allowable delivery date for workpiece i
q	unit time cost
e	Average industrial electricity prices

When an order insertion event occurs at time $T_{n'}$, different workpiece types are recognized and processed as shown in the table 2.

Table 2 Different types of workpiece discrimination processing methods.

Type of workpiece	discriminant	Treatment
Completed workpieces	Workpiece J_i last process O_p in process, i.e., $T_{n'} \geq TB_i^p$	Workpiece J_i is not a rescheduling object
Workpieces being processed	Workpiece J_i last process O_1 in process, i.e., $TB_i^1 \leq T_{n'}$	Processes in which workpiece J_i has not yet begun machining are the subject of rescheduling.
Workpieces not yet processed	The first process O_1 of the workpiece J_i is unmachined, i.e. $TB_i^1 > T_{n'}$	Workpiece J_i is a rescheduling object

2.2. Constructing energy-saving optimization model for HFRP-LB

2.2.1. Objective function establishment

In line with actual production needs, the objective function $f\left(T_{\text{object}}, E_{\text{object}}, Q_{\text{object}}\right)$ is formulated with the goals of reducing total energy consumption E_{object}, minimizing the maximum completion time T_{object}, and lowering production costs Q_{object}.

(1) Minimize total energy consumption

Consider the overall energy consumption E_p from machine tool operations, idle energy E_w, and transportation energy E_T needs for the hybrid flow workshop. At time $T_{n'}$, the occurrence of a single insertion event prompts the workshop to minimize total energy consumption, represented by the following objective function:

$$E_{object} = \min(E_p + E_w + E_T) \tag{1}$$

a) Total processing energy consumption

Total processing energy consumption pertains to the direct energy used by machine tools and equipment in the workshop. To provide a more precise analysis, focus on the main drive system's energy consumption and equipment in the cutting state of processing materials consumed by the cutting energy consumption E_c, the normal operation of the processing equipment, the rotating system in the cutting state of the additional losses generated by the additional loss of electrical losses and equipment losses, including the loss of E_a, and the workpiece does not arrive when the processing equipment E_{nl}, which is generated by the no-load energy consumption when the workpiece does not arrive.

Referring to the literature [23], the total input power, cutting power, additional loss power, and no-load power are expressed as P_{in}^{k}, P_{c}^{k}, P_{a}^{k}, and P_{nl}^{k}, respectively, then the energy consumption of the equipment processing has the following expression:

$$E_p^k = \sum_{k=1}^{m} \int_0^{t^k} P_{in}^k(t)dt = \sum_{k=1}^{m} \left(\int_0^{t^k} P_{nl}^k(t)dt + \int_0^{tc^k} P_c^k(t)dt + \int_0^{tc^k} P_a^k(t)dt \right) \tag{2}$$

For a machine tool, when the spindle speed is stabilized, its total input power and no-load power have only minor fluctuations, which can be considered as constant values [24], and the equation of energy consumption for machining of machine tools and equipment can be transformed into:

$$E_p^k = \sum_{k=1}^{m} \left(P_{nl}^k t^k + P_c^k tc^k + P_a^k tc^k \right) = \sum_{k=1}^{m} \left(P_{in} tc^k + P_{nl}^k \left(t^k - tc^k \right) \right) \tag{3}$$

The total energy consumption for production and processing can be expressed as:

$$\begin{aligned} E_p &= \sum_{i=1}^{n+n'} \sum_{j=1}^{p} \sum_{k=1}^{m} r_{ij} \left(P_{nl}^{k,i} t_i^k + P_c^{k,i} tc_i^k + P_a^{k,i} tc_i^k \right) B_i^k \\ &= \sum_{i=1}^{n+n'} \sum_{j=1}^{p} \sum_{k=1}^{m} r_{ij} \left(P_{in}^{k,i} tc_i^k + P_{nl}^{k,i} \left(t_i^k - tc_i^k \right) \right) B_i^k \end{aligned} \tag{4}$$

Where $P_{\mathrm{in}}^{k,i}$ and $P_{\mathrm{nl}}^{k,i}$ represent the input power and no-load power of the equipment k when machining the workpiece i, respectively, and tc_i^k and t_i^k represent the length of time the workpiece is cut on the equipment and the length of time it is machined.

b) Idle wait energy consumption

The waiting time tw^k of machine k is defined by the total machining time for all workpieces processed on this equipment, subtracting the total operational time of machine k:

$$tw^k = \sum_{i=1}^{n+n'} t_i^k B_i^k - MTE_k + MTB_k \tag{5}$$

where MTE_k is the end time of the last machining of device k and MTB_k denotes the start time of the first machining of device k.

After an insertion event at time $T_{n'}$, the waiting time expression is updated as follows:

$$tw_{n'}{}^{k} = \sum_{i=1}^{n+n'} \sum_{j=1}^{p} r_{ij} t_i^k B_i^k - MTE_k + T_{n'} \tag{6}$$

Where r_{ij} is a decision variable. r_{ij} = 1 when the jth process of workpiece i enters rescheduling after a dynamic event, 0 otherwise

The on-off strategy is utilized when machine k 's standby period is prolonged or when $E_w{}^k$ is considerably high. This strategy is executed only if the critical interval T_k^r, permitting shutdown or restart, exceeds or matches the duration required for the on-off procedure, and if the associated energy consumption meets or surpasses the energy used by machine k during each switch. The critical interval is defined as:

$$T_k^r = \max\left\{ t_k^{on} + t_k^{off}, \frac{E_k^{on} + E_k^{off}}{P_w{}^k} \right\}, \forall k \in M_m \tag{7}$$

Where t_k^{on} denotes the length of time used to turn on the device k, t_k^{off} denotes the length of time used to turn off the device k, E_k^{on} and E_k^{off} denote the energy consumption of the device to turn on and off the device, respectively. M_m is a collection of devices.

The E_w, i.e., the machine waiting energy consumption, is expressed as follows:

$$E_w = \sum_{k=1}^{m} P_w{}^k tw^k + \sum_{i=1}^{n+n'} \sum_{j=1}^{p} \sum_{k=1}^{m} r_{ij} (E_k{}^{on} + E_k{}^{off}) z_i{}^k \tag{8}$$

Where $z_i{}^k$ is a decision variable indicating whether to power on or restart when process i is machined on machine k ($B_i{}^k$ =1), by default machine k is on standby before machining starts, and remains on standby when $z_i{}^k$ =0; turn on strategy is applied when $z_i{}^k$ =1. $P_w{}^k$ denotes the standby power of machine k .

c) Transportation energy consumption

The energy consumption E_T for transportation mainly arises from the AGV moving workpieces between the back buffer of machine k and machine h ($h \in M_m$) in the subsequent stage. It is described as follows:

$$E_T = \sum_{i=1}^{n+n'} \sum_{j=1}^{p} \sum_{k=1}^{m} \sum_{h=1}^{m-1} r_{ij} P_{agv} t_{agv}{}^{k,h} B_i^{k,h} B_i^k \tag{9}$$

Where P_{agv} denotes the rated power of the AGV, and $t_{agv}{}^{k,h}$ denotes the length of time taken by the AGV to transfer the workpiece from the back buffer of the device k to the device h . $B_i^{k,h}$ is the decision variable, and the workpiece i is machined on the device k after the next stage of machining on the device h is 1, and otherwise is 0 for.

(2) Minimize the maximum completion time

Take all the workpieces to be processed last process end time for the completion time, then minimize the maximum completion time objective function is shown below:

$$T = \mathrm{Max}\left(MTE_k\right), \forall k \in M_m \tag{10}$$

$$T_{object} = Min(T) \tag{11}$$

(3) Minimizing production cost

Take the processing loss cost, buffer maintenance cost and power consumption cost of each equipment per hour as the production cost of the workpiece. Considering that the AGV transportation cost in the buffer maintenance cost is the AGV transportation energy cost, only the buffer management cost is taken into account, and the objective function of minimizing the production cost is shown as follows:

$$M_{Total} = \sum_{k=1}^{m} c^k M \tag{12}$$

$$T_{Total} = \sum_{i=1}^{n+n'} \sum_{j=1}^{p} \sum_{k=1}^{m} r_{ij} t_i^k B_i^k \tag{13}$$

$$Q_{object} = \min(qT_{Total} + eE_p + eE_w + eE_T + M_{Total}) \tag{14}$$

2.2.2. Constraints

For the multi-objective hybrid flow shop rescheduling problem under insertion of single events, the following constraints also need to be satisfied:

Constraints	Description of conditions	Equation number
$\exists M_j > 1, j \in P$	Parallel equipment exists on at least one process.	(15)
$\sum_{j=1}^{p} M_j = m, j \in P$	Sum of equipment on each process equals the total number of equipment.	(16)
$\sum_{k=1}^{M_j} B_i^k = 1, \forall i \in W + n', j \in P$	Each workpiece is assigned to only one machine at each stage.	(17)
$TB_i^{j+1} \geq TE_i^j, \forall i \in W + n', j \in P$	Workpieces must follow the sequence of process.	(18)
$TE_i^j = TB_i^j + t_i^k, \forall i \in W + n', j \in P$	The completion time of a workpiece in a process is the sum of the start time and the duration of the process.	(19)
$t_i^k = tc_i^k + t_{agv}^k + tb^k, \text{when } B_i^k = 1, \forall k \in M_m, i \in W + n'$	The machining k time of a workpiece i on a machine is the sum of the cutting time, handling time by the AGV, and resting time in the buffer.	(20)
$MTB_k \geq 0, \forall k \in M_m$	Each device is available at time 0.	(21)
$z_i^k = \begin{cases} 1, TB_i^j - TE_i^{j-1} \geq T_k^r \\ 0, TB_i^j - TE_i^{j-1} < T_k^r \end{cases}, \forall i \in W + n', k \in M_m, j \in P$	If the time interval between the first two subsequent processes exceeds the critical time T_k^r, the switching policy is applied ($z_i^k = 1$), otherwise not ($z_i^k = 0$).	(22)
$(E_k^{off} + E_k^{on}) z_i^k \leq E_w^k, \forall i \in W + n', k \in M_m$	The energy consumption for a single shutdown or restart is less than or equal to the standby energy consumption of the machine k .	(23)
$\sum_{i=1}^{n} B_i^k \leq C^k, \forall k \in M_m$	buffer capacity limit.	(24)
$r_{ij} = \begin{cases} 1, TB_i^j \geq T_{n'} \\ 0, TB_i^j < T_{n'} \end{cases}, \forall i \in W + n', k \in M_m, j \in P$	In the event of an order insertion at time $T_{n'}$, the workpieces not yet started will be included in the rescheduling ($r_{ij} = 1$), otherwise they will not participate in the rescheduling($r_{ij} = 0$).	(25)

2.3. Rescheduling Process Framework Based on Order Insertion Event.

In actual production, the target weights are adjusted according to the enterprise's policy tendency, and the improved Red-billed blue magpie optimizer (IRBMO) is used to generate and execute the scheduling program based on the current production status of the intelligent flow shop. When there is an order insertion, determine the order insertion point, the insertion point before the process has been processed and the process is being processed to remain unchanged, to find out the insertion point after the unprocessed process and the new insertion of the order with IRBMO for rescheduling, order insertion rescheduling of the specific process as shown in Figure 1.

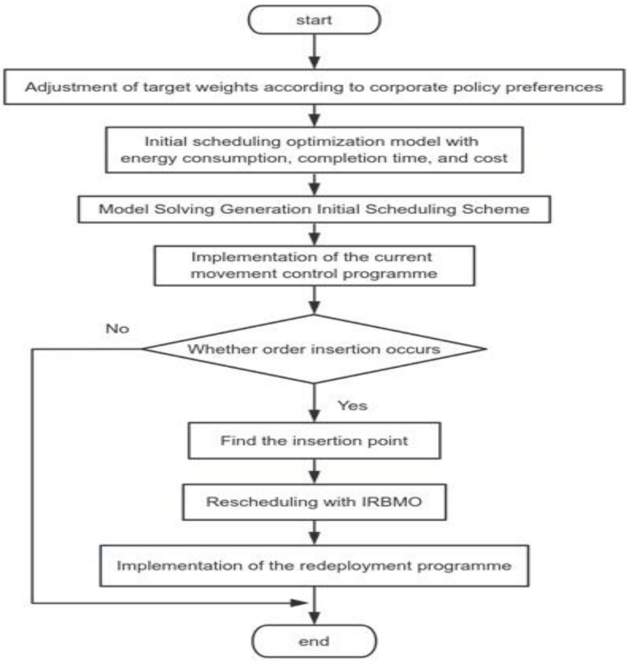

start
Adjustment of target weights according to corporate policy preferences
Initial scheduling optimization model with energy consumption, completion time, and cost
Model Solving Generation Initial Scheduling Scheme
Implementation of the current movement control programme
No
Whether order insertion occurs
Yes
Find the insertion point
Rescheduling with IRBMO
Implementation of the redeployment programme
end

Figure 1 This caption has one line so it is centred.

3. HFRP-LB Optimization Solution Based on Improved Red-Billed Blue Magpie Optimizer

In the emergency plug-and-play single-weight scheduling problem in a mixed flow shop, the multi-objective dynamic scheduling optimization model established by considering complex constraints such as buffer carrying capacity, AGV transport operation and equipment switching strategy improves the practical production application but also increases the complexity of problem solving, which makes it difficult for the previous methods to find a high-quality scheduling solution quickly in a complex production environment.

Red-billed blue magpie optimizer (RBMO), as a group intelligence algorithm imitating the foraging behavior of red-billed blue magpies and group collaborative foraging behavior, generates new solutions through the collaborative behavior and information sharing of red-billed blue magpies so that it possesses a strong global search capability and adaptability, and can effectively avoid local optimal solutions[25]. Secondly, its diverse foraging and predation mechanisms and dynamic weight adjustment function make it perform well in multi-objective optimization, which can balance the conflicts between different optimization objectives and find the Pareto optimal solution. In order to adapt to the complex production environment, this paper applies the RBMO algorithm to the rescheduling problem and optimizes the production scheduling process by considering the constraints of buffer capacity, switching strategy and transportation time. The algorithm consists of the following four main phases:

(1) Initialization phase

In the initialization phase, an initial population is generated and the fitness value of each individual is calculated to determine the initial optimal individual (food location). Each individual of the initial population represents a possible scheduling scheme that takes into account the state of each device (on or off), buffer capacity, and transportation time.

(2) Foraging behavior

Foraging behavior mimics the behavior of red-billed blue magpies during extensive search for food. This phase explores the solution space through cooperative foraging in small groups (2 to 5 individuals) and clusters (10 or more individuals). The algorithm adjusts the task allocation of each device during the foraging process to ensure that the buffer capacity is not overloaded, and takes into account the on/off state of the devices to optimize energy consumption and production costs.

(3) Predation behavior

Predation behavior simulates more refined search and capture behavior of red-billed blue magpies near known food locations. This phase exploits the solution space through cooperative predation in groups and clusters. During the predation process, the algorithm further optimizes the task allocation and switching strategy of the equipment to reduce energy consumption, ensure a smooth production process, and minimize completion time and production cost.

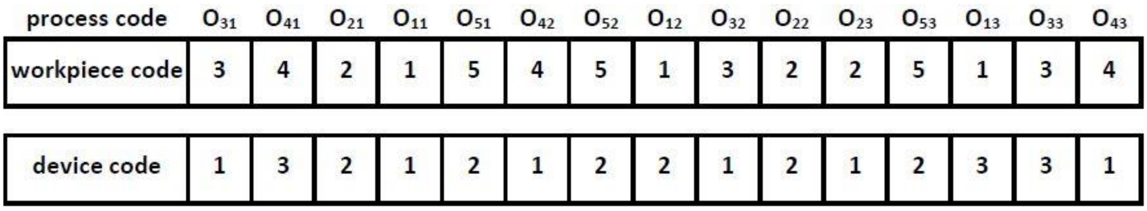
(4) Food storage behavior

The storage of food phase ensures that the currently found optimal solution is retained and updates the optimal individual in the current iteration. This phase checks the fitness value of each individual and updates the population based on the fitness value to ensure that the more optimal scheduling solution is retained. The fitness value is calculated taking into account not only the production completion time but also factors such as energy consumption and production cost.

While the traditional RBMO algorithm offers innovative approaches for complex scheduling issues, it has certain drawbacks. The random initialization of the population tends to be overly homogeneous, making the solution quality heavily reliant on the initial population's caliber. Additionally, the diversity within the population might diminish in later stages of iteration. To address these concerns, this paper proposes a Logistic chaotic mapping inverse learning strategy for initializing the population [26]. This approach ensures a discrete distribution of individuals across the solution space. Furthermore, it incorporates adaptive regulation, as well as crossover and mutation operators, to enhance the diversity of the target population and ultimately improve the solution quality.

3.1. Encoding and Decoding.

In the emergency plug-and-play single-weight scheduling problem of hybrid flow shop, a well-designed coding method can improve the expressive ability of the solution, adapt to the demand of multi-objective optimization, and simplify the processing of complex constraints, which significantly enhances the operability of the optimization algorithm and the solving effect. Therefore, this paper adopts a two-layer coding structure of workpiece code and equipment code. The first layer of coding is the workpiece code, which encodes the workpieces on each process according to the sequence of the process to ensure the rationality and consistency of the process; the second layer is the equipment code, which indicates the equipment number of the corresponding position of the process in the set of optional equipments, which optimizes the utilization rate of the equipment and reduces the switching time of the equipments. Each process can be selected {3, 2, 3} parallel devices, the corresponding device set are $\{[M_1, M_2, M_3], [M_4, M_5], [M_6, M_7, M_8]\}$. An example of a process code is shown in Figure 2:

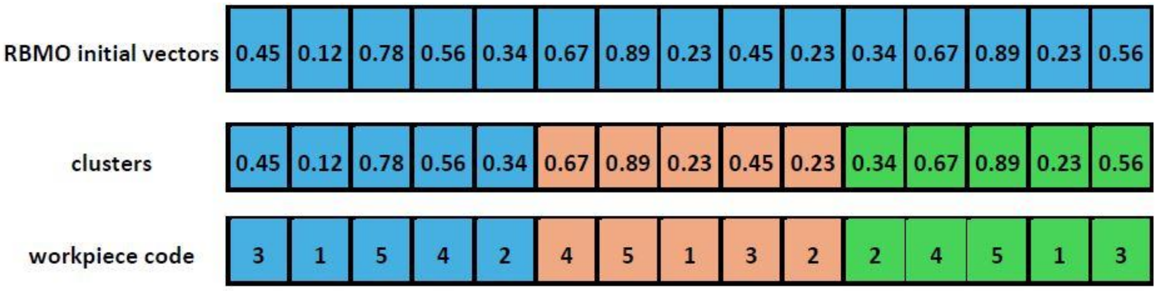


process code	O_{31}	O_{41}	O_{21}	O_{11}	O_{51}	O_{42}	O_{52}	O_{12}	O_{32}	O_{22}	O_{23}	O_{53}	O_{13}	O_{33}	O_{43}
workpiece code	3	4	2	1	5	4	5	1	3	2	2	5	1	3	4
device code	1	3	2	1	2	1	2	2	1	2	1	2	3	3	1

Figure 2 Schematic diagram of coding.

In the example, there are 5 workpieces that need to be processed through 3 processes, and the total length of the 3 process codes is 30. Taking the 1st process as an example, the corresponding processing order of the workpieces is No. 3, No. 4, No. 2, No. 1, and No. 5; the corresponding serial numbers of the processing equipments are [1,3,2,1,2], and based on the positional counterparts, it can be determined that the corresponding equipments of this process are in order of $\{M_1, M_3, M_2, M_1, M_2\}$, and so on.

Since the codes in the HFRP-LB problem are discrete integer variables, this paper adopts local ascending ordering to map the initial RBMO vectors to the workpiece codes. Firstly, the h-dimensional vectors generated by RBMO are divided into p groups with workpiece n as the interval; then the intra-group ordering is performed, and the intra-group order values are the workpiece codes of each process. The conversion process is shown in Figure 3. The steps of equipment coding and workpiece coding are the same.



RBMO initial vectors	0.45	0.12	0.78	0.56	0.34	0.67	0.89	0.23	0.45	0.23	0.34	0.67	0.89	0.23	0.56
clusters	0.45	0.12	0.78	0.56	0.34	0.67	0.89	0.23	0.45	0.23	0.34	0.67	0.89	0.23	0.56
workpiece code	3	1	5	4	2	4	5	1	3	2	2	4	5	1	3

Figure 3 Conversion of initial vector to artifact coding process.

Decoding is the process of converting the shop floor scheduling program according to the workpiece number and equipment serial number, and then calculating the corresponding machining time for each process, and projecting the energy consumption and cost based on the time.

3.2. Logistic Chaos Mapping Based Reverse Learning Strategy.

The population initialization method of the RBMO algorithm is based on random values of the upper and lower bounds of individual vector elements, and its performance is relatively homogeneous. In this paper, we propose a Logistic Chaos Mapping based inverse learning strategy to generate the initialized population, which enriches the diversity of the target population to a certain extent.

Firstly, the initial population with dimension d and population size $N/2$ is generated by Logistic Chaos Mapping, assuming that the individual position vector of the population is $(x_1^d, x_2^d, \cdots, x_{n/2}^d)$, then the position vector of the reverse learning strategy is $(x_1^{d\prime}, x_2^{d\prime}, \cdots, x_{n/2}^{d}{}^{\prime})$, the specific expression is as follows. Finally, the two initial populations are merged to form an initial population with dimension d and population size N.

$$R_i^d = A_i^d + B_i^d - x_i^d, i = 1,2,3,\cdots,n/2 \tag{26}$$

$$x_i^{d\prime} = R_{n/2+1-i}^d, i = 1,2,3,\cdots,n/2 \tag{27}$$

Where: A_i^d and B_i^d denote the upper and lower limits of the value range of the current individual position vector, respectively, x_i^d denotes the initial individual position vector, R_i^d denotes the reverse individual position vector, and $x_i^{d\prime}$ denotes the position vector of the reverse learning strategy.

3.3. Calculation of Adaptation.

The fitness assessment is used to measure the quality of each scheduling option. It combines the completion time, energy consumption, and cost to determine which schemes are better. The corresponding maximum value of completion time $T_{\max}$, minimum value of completion time $T_{\min}$, maximum value of energy consumption $E_{\max}$, minimum value of energy consumption $E_{\min}$, maximum value of cost $Q_{\max}$, and minimum value of cost $Q_{\min}$ are computed for each scheduling scenario generated after initialization. Given the different units of these metrics, a single objective function is formulated by de-meaning and weighted summation of the objectives. The combined objective evaluation function f is：

$$f\left(T_{\text{object}}, E_{\text{object}}, Q_{\text{object}}\right) = w_1\left(\frac{T_{\text{object}} - T_{\min}}{T_{\max} - T_{\min}}\right) + w_2\left(\frac{E_{\text{object}} - E_{\min}}{E_{\max} - E_{\min}}\right) + w_3\left(\frac{Q_{\text{object}} - Q_{\min}}{Q_{\max} - Q_{\min}}\right) \tag{28}$$

where is a weighting coefficient, m = 1. Adaptation of the firm to the target propensity can be realized by changing the size of w. The adaptation values corresponding to the initially generated n processing sequence numbers are compared to find the initial optimal individual (i.e., food location).

3.4. Location Update.

Search optimization is performed by mimicking the foraging behavior of red-billed blue magpies in the search space based on the value of the individual fitness function (i.e., the value of the objective function). Each magpie adjusts its position according to its own and the group's information to find a more optimal scheduling solution. During the foraging process, magpies in the group cooperate to capture the target, i.e., the scheduling solution is improved and the objective function value is reduced by combining local and global search. In addition, magpies will retain the current optimal solution during the search for an optimal solution and update it in each generation iteration to ensure that an optimal or near-optimal solution is finally obtained.

(1) Foraging behavior: In the process of searching for food, red-billed blue magpies usually act in small groups or clusters to improve the search efficiency. When the value of the objective function is less than ε the group mode of action is used, and the position is updated as in equation (29), and vice versa the cluster mode is used as in equation (30). In this paper, the value of ε is selected as 0.5.

$$X^i(t+1) = X^i(t) + (\frac{1}{p} \times \sum_{m=1}^{p} X^m(t) - X^*(t)) \times R \tag{29}$$

$$X^i(t+1) = X^i(t) + (\frac{1}{q} \times \sum_{m=1}^{q} X^m(t) - X^*(t)) \times R \tag{30}$$

Where $X^i(t+1)$ represents the next generation position obtained by the i th individual update; $X^i(t)$ represents the current position of the i th individual; p represents the number of red-billed blue magpie groups, which usually ranges from 2 to 5, and its value of 5 was chosen for this paper; q represents the number of red-billed blue magpie clusters, which usually ranges from 10 to N. In this paper, a value of $\lceil N/3 \rceil$ was chosen; $X^m(t)$ represents the position of the m th randomly selected individual in the t th generation; $X^*(t)$ represents the position of a randomly selected individual in the current iteration; R is a random number that conforms to a standard normal distribution.

(2) Predatory behavior: Red-billed blue magpies showed high hunting proficiency and cooperation when pursuing prey. The locations updated for small values of the target function (Eq. (31) and for large values of the target function (Eq. (32)) are as follows:

$$X^i(t+1) = X^{food}(t) + CF \times (\frac{1}{p} \times \sum_{m=1}^{p} X^m(t) - X^i(t)) \times R \tag{31}$$

$$X^i(t+1) = X^{food}(t) + CF \times (\frac{1}{q} \times \sum_{m=1}^{q} X^m(t) - X^i(t)) \times R \tag{32}$$

Where $X^{food}(t)$ is the location of the food, i.e., the optimal scheduling scheme in the current iteration; CF is the perturbation factor, which is gradually reduced in step size as the number of iterations increases to avoid extensive search at a later stage:

$$CF = (1 - (\frac{t}{T}))^{(2 \times \frac{t}{T})} \tag{33}$$

Where T denotes the maximum number of iterations and t denotes the current number of iterations.

(3) Food storage behavior: in addition to foraging and hunting, red-billed blue magpies store excess food for future consumption, ensuring a stable food supply in case of food shortage. The storing food phase ensures that the algorithm remembers the superior solution, i.e., when the fitness value after updating the position is superior to the fitness value of the current position, the new position will be retained.

$$X^i(t+1) = \begin{cases} X^i(t) & if \quad fitness_{old}^i > fitness_{new}^i \\ X^i(t+1) & else \end{cases} \tag{34}$$

Where $fitness_{old}^i$ and $fitness_{new}^i$ represent the fitness values of the i th red-billed blue magpie before and after the position update, respectively.

3.5. Crossover and Variation Operations.

In the iterative process of the red-billed blue magpie algorithm, the crossover and mutation operations of the hybrid genetic algorithm are introduced, and the crossover and mutation rates are adaptively adjusted based on the individual fitness values to improve the diversity of the solutions and the exploratory ability.

In this paper, the POX crossover operator from literature [27] is used to ensure the integrity of the code. Local crossover is performed on the artifact codes in terms of processes, assuming that the initial artifact set is $O=\{1,2,3,4,5\}$, and the artifact $O_a=\{3,4\}$ set is randomly divided into 2 sub-artifact sets and $O_b=\{1,2,5\}$. The specific operation process is shown in Figure 4, which mainly includes: copying each group of processes in the artifact set O_a from parent 1 to child 1 and from parent 2 to child 2, and keeping the positions unchanged; filling the positions of the processes in the artifact set O_b from parent 1 to the blanks of child 2 and from parent 2 to the blanks of child 1 in the same order. After completing the workpiece code crossover, the same crossover operation is performed for its corresponding equipment code.

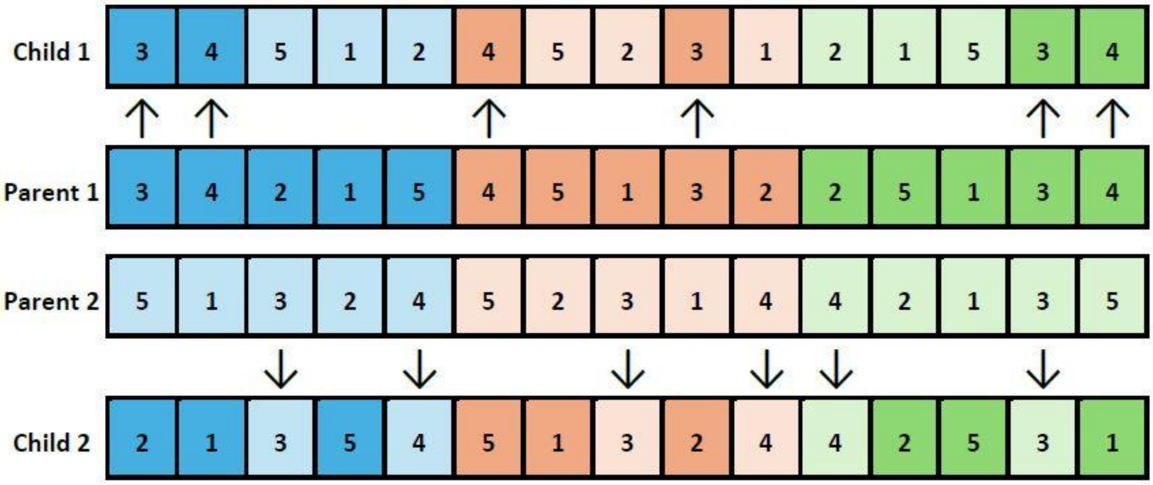


Figure 4 Schematic diagram of individual POX crossing.

Too high a crossover probability may destroy the structure of the superior individuals, while too low a crossover probability may cause the algorithm to fall into a local optimum. To balance this problem, adaptive crossover probability can be used, which dynamically adjusts the crossover probability according to the fitness of the individuals: smaller crossover probability for the better individuals and larger crossover probability for the worse individuals, as calculated in the following formula:

$$P_c = \max(P_{c\_min}, \min(1, P_{c\_max} - (P_{c\_max} - P_{c\_min}) \square \frac{f_{avg} - f}{f_{max} - f_{avg}})) \tag{35}$$

Where: P_c is the current individual crossover probability; P_{c_max}, P_{c_min} are the maximum crossover probability and minimum crossover probability, which are taken as 0.9 and 0.3 in this paper, respectively; f is the fitness of the current individual; and f_{max} and f_{avg} denote the maximum and average of the fitness values of all the individuals in the population, respectively.

In this paper, we use the local inverse order variation operator in terms of processes. This operator will produce large changes to the original individual, which can increase its ability to jump out of the local optimum. The specific operation process is shown in Figure 5, which is as follows: firstly, two coding variation positions are chosen arbitrarily by process unit, and then the coding between the two positions in the parent generation is inverted-order variation, thus obtaining a new child generation.

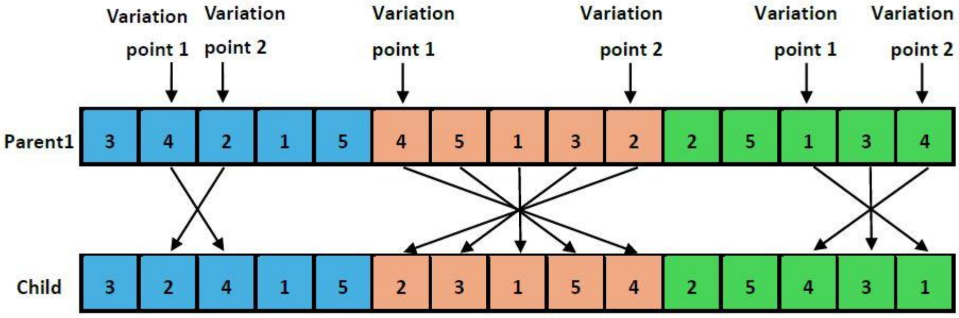


Figure 5 Schematic diagram of inverted mutation.

The mutation operation changes the value of one or more genes in an individual, and the execution of the operation relies on the setting of the mutation probability, which is adaptively adjusted with the following formula:

$$P_m = \max(P_{m\_min}, \min(1, P_{m\_max} - (P_{m\_max} - P_{m\_min})\square\frac{f_{avg} - f}{f_{max} - f_{avg}})) \tag{36}$$

Where: P_m is the current individual variation probability; P_{m_max} , P_{m_min} are the maximum variation probability and minimum variation probability, which are taken as 0.1 and 0.005 in this paper, respectively.

3.6. Algorithm Flow.

Based on the above improvement, the IRBMO flow for solving HFRP-LB is shown in Figure 6:

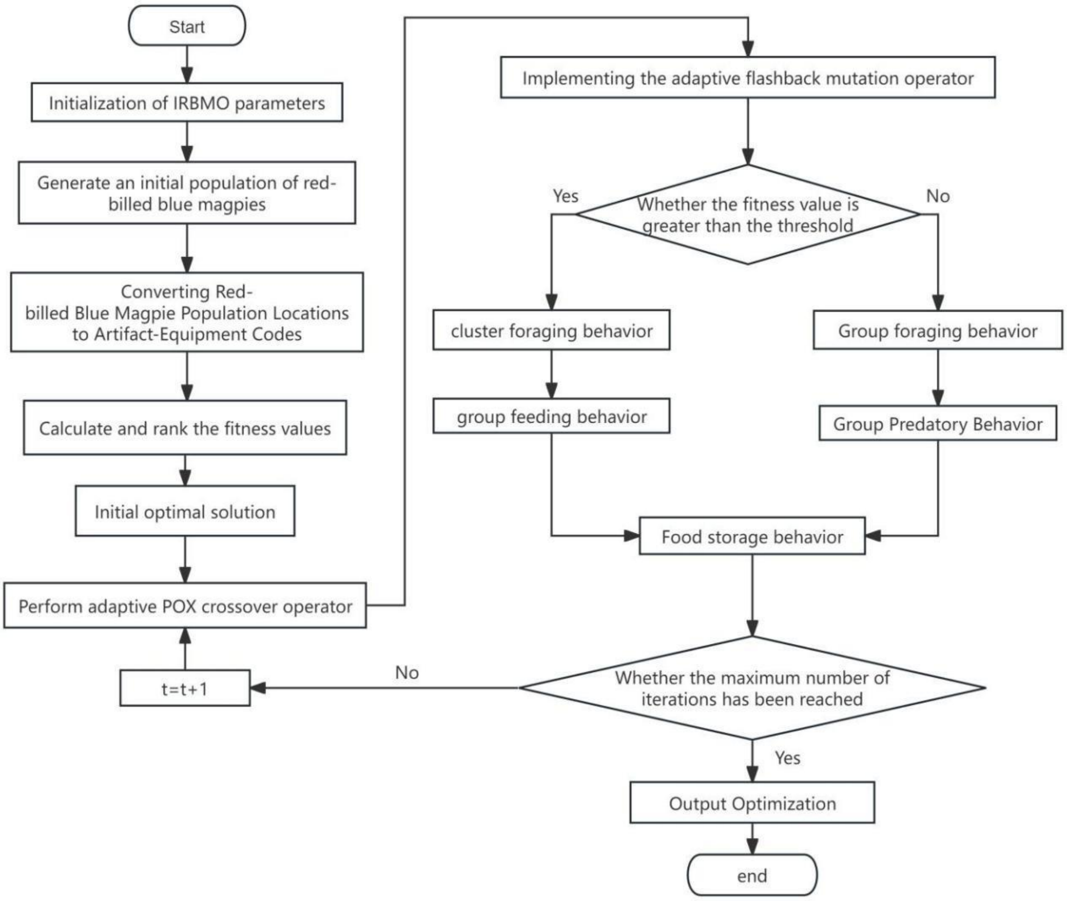

Start
Initialization of IRBMO parameters
Generate an initial population of red-billed blue magpies
Converting Red-billed Blue Magpie Population Locations to Artifact-Equipment Codes
Calculate and rank the fitness values
Initial optimal solution
Perform adaptive POX crossover operator
t=t+1
Implementing the adaptive flashback mutation operator
Whether the fitness value is greater than the threshold
Yes
No
cluster foraging behavior
group feeding behavior
Group foraging behavior
Group Predatory Behavior
Food storage behavior
Whether the maximum number of iterations has been reached
No
Yes
Output Optimization
end

Figure 6 IRBMO flow chart.

4. Simulation Experiment

In this paper, simulation experiments are simulated by MATLAB R2021a software with computer parameters: processor Intel(R) Pentium(R) Gold G5420 CPU @ 3.80GHz and 8GB of RAM. Simulation experiments are conducted on dynamic scheduling problem using complete rescheduling strategy and Improved Red-billed Blue Magpie Algorithm for the insertion of single event. The RBMO, traditional red-billed blue magpie algorithm, Multi-Objective Particle Swarm Optimization (MOPSO) [28, 29] and Multi-objective genetic algorithm (MOGA) [27] are combined to verify the feasibility of the rescheduling model and the effectiveness of IRBMO.

4.1. Arithmetic Example Production.

There are no existing benchmark data sets available for HFRP-LB, so we extended the arithmetic example found in the literature [30] (the extension information is shown in Table 3). The problem size of the arithmetic example is set as follows: the number of orders M ; the number of processe N ; the number of optional parallel equipment is the same for all processes, i.e., B, and the buffer capacity is V. For each arithmetic example, data is added to each arithmetic example in the first process time to add the insertion of orders for a different number of workpieces C, and arithmetic example The data is expressed as: X, for example, 20-3-5-4-2 means that the original plan to process 20 workpieces, after the insertion of 2 workpieces, all the workpieces are to be completed in the workshop 3 machining processes and parallel equipment for the 5, the workshop buffer capacity is 4.

Table 3 Rules for the values of the algorithm parameters.

parameters	Retrieve Value	parameters	Retrieve Value
Cutting time	[20,80]s	Rated power of AGV	1.89kw
Input power	[2.0,3.5]kw	Duration of transport	[30,50]s
Idle power	[2.0,3.5]kw	Workpiece resting time	45s
Standby power	[1.0,2.0]kw	Startup time	[50,150]s
Energy consumption for one switch	[10800,21600]kw·s	Power off time	[50,150]s
unit processing cost	[20,40]CNY/h	Average industrial electricity price	0.82 CNY/kW·h

4.2. Parameter Setting.

In this paper, the optimization objectives are to minimize total energy consumption, minimizing the maximum completion time, and production cost. The weights assigned to each optimization criterion are set at {0.5, 0.2, 0.3} respectively. The standard parameters for the algorithm include a population size of 100 and a maximum of 100 iterations. Based on existing literature concerning the multi-objective particle swarm algorithm and multi-objective genetic algorithm, this paper conducts several experiments with medium-sized datasets (20-3-5-4-2) using the grid search method. The final parameter configuration for the multi-objective particle swarm and multi-objective genetic algorithms, determined through testing, includes a crossover probability of 0.8, the variance probability of 0.2, and inertia values of 0.8. Both c1c1 and c2c2 are set to 2.

4.3. Experimental Results and Comparison.

Algorithm performance is evaluated from the perspective of algorithm effectiveness by minimizing the total energy consumption, minimizing the maximum completion time, and minimizing the production cost. Relative Percentage Deviation (RPD) [31,32] is used to measure the algorithm performance, which is calculated as equation (37).

$$RPD = \frac{C(A) - Cbest}{Cbest} \times 100\% \tag{37}$$

Where $C(A)$ is the value of the objective function obtained by the current algorithm and $Cbest$ is the best value obtained by all algorithms. In this paper, the relative deviation of energy consumption RPD_E , the relative deviation of completion time RPD_T and the relative deviation of cost RPD_Q are calculated respectively.

Different algorithms are used to solve the above example scale, and the average value of RPD of each algorithm is calculated, and the results of each algorithm are shown in Table 4.

Table 4 Comparison of runtime results for example data.

	IRBMO			RBMO			MOPSO			MOGA		
	RPD_E	RPD_T	RPD_Q	RPD_E	RPD_T	RPD_Q	RPD_E	RPD_T	RPD_Q	RPD_E	RPD_T	RPD_Q
10-3-2-5-1	2.20%	3.05%	2.77%	3.66%	5.58%	5.41%	13.95%	6.40%	10.44%	5.11%	10.12%	11.21%
10-3-2-5-2	3.22%	1.59%	1.64%	4.51%	3.58%	4.10%	19.15%	3.91%	7.57%	6.08%	6.69%	7.30%
10-3-2-5-3	2.38%	2.73%	1.67%	3.80%	5.43%	3.65%	15.96%	5.47%	6.90%	5.59%	10.37%	8.02%
10-3-3-4-1	2.03%	1.36%	2.67%	3.37%	3.11%	5.63%	13.22%	3.99%	9.73%	4.85%	6.14%	10.30%
10-3-3-4-2	2.58%	1.54%	3.27%	4.21%	3.99%	6.95%	16.83%	4.25%	10.99%	5.20%	6.28%	11.59%
10-3-3-4-3	2.22%	1.71%	2.98%	3.52%	3.73%	6.13%	14.29%	4.42%	10.24%	5.41%	7.54%	10.63%
10-3-4-3-1	1.36%	2.78%	2.73%	2.82%	5.67%	5.25%	11.13%	5.53%	10.67%	3.84%	9.80%	9.74%
10-3-4-3-2	2.05%	1.82%	2.77%	3.52%	4.36%	5.75%	13.49%	4.51%	10.65%	4.77%	7.82%	10.49%
10-3-4-3-3	2.53%	2.02%	1.31%	3.68%	3.98%	3.74%	15.96%	4.67%	7.13%	5.17%	7.94%	6.98%
20-5-2-10-1	2.07%	1.47%	2.07%	4.47%	7.08%	8.07%	21.17%	7.53%	16.43%	8.00%	13.14%	15.93%
20-5-2-10-2	3.52%	3.27%	2.13%	6.60%	10.53%	7.52%	30.22%	12.00%	14.58%	12.20%	20.16%	15.32%
20-5-2-10-3	1.80%	2.04%	3.57%	4.25%	7.88%	11.34%	18.68%	9.08%	23.63%	7.45%	14.81%	22.33%
20-5-3-7-1	2.21%	1.38%	2.80%	5.04%	5.53%	8.79%	22.97%	6.50%	17.96%	8.88%	12.92%	18.57%
20-5-3-7-2	3.18%	2.98%	1.95%	6.74%	10.00%	7.48%	27.61%	10.85%	13.95%	11.93%	18.63%	14.38%
20-5-3-7-3	1.60%	2.60%	2.96%	3.62%	9.28%	9.06%	18.50%	9.48%	20.12%	7.18%	18.32%	18.99%
20-5-4-5-1	2.35%	1.55%	3.53%	5.66%	6.25%	11.16%	23.43%	7.33%	22.23%	8.80%	11.70%	23.83%
20-5-4-5-2	3.75%	2.69%	2.18%	7.20%	9.83%	8.51%	31.78%	10.45%	16.47%	12.03%	16.97%	17.48%
20-5-4-5-3	3.02%	1.32%	1.63%	6.42%	5.30%	5.89%	26.39%	6.93%	14.04%	10.43%	12.06%	13.07%
50-5-4-13-5	2.77%	3.53%	3.47%	5.57%	10.35%	9.55%	20.71%	10.58%	23.24%	8.70%	24.81%	23.95%
50-5-4-13-10	4.48%	3.23%	3.88%	7.32%	9.58%	10.17%	29.29%	10.38%	25.20%	11.88%	22.46%	25.11%
50-5-4-13-16	4.62%	4.08%	4.20%	7.37%	10.53%	11.62%	30.80%	12.45%	29.01%	13.63%	25.48%	27.49%
50-8-4-13-5	4.36%	2.40%	2.15%	7.23%	8.40%	7.25%	29.99%	7.88%	18.26%	13.30%	17.19%	17.98%
50-8-4-13-10	4.43%	3.52%	3.60%	7.49%	9.88%	9.29%	31.20%	11.08%	25.87%	13.33%	23.46%	24.94%
50-8-4-13-16	4.20%	3.28%	3.61%	7.28%	10.15%	9.38%	28.54%	9.70%	24.98%	11.30%	22.18%	25.17%
100-5-4-25-16	3.11%	3.64%	2.06%	4.56%	11.67%	7.68%	27.92%	11.88%	24.24%	10.11%	31.28%	21.55%
100-5-4-25-25	2.28%	3.46%	3.70%	4.32%	11.10%	12.66%	25.12%	11.19%	33.76%	8.46%	29.52%	31.55%
100-5-4-25-34	2.18%	4.79%	4.65%	4.28%	13.38%	12.78%	22.88%	13.92%	38.00%	9.18%	38.00%	38.39%
100-8-	5.45%	4.09%	3.75%	7.68%	11.94%	11.01%	42.40%	13.20%	30.24%	14.91%	31.60%	31.64%

4-25-16												
100-8-4-25-25	5.10%	3.99%	3.49%	6.77%	11.82%	11.01%	39.52%	12.48%	30.64%	15.21%	33.36%	30.10%
100-8-4-25-34	3.42%	3.75%	3.58%	5.76%	11.61%	10.47%	29.44%	12.30%	29.36%	11.55%	33.20%	33.35%

As observed from Table 3, calculating the relative deviation rate of each algorithm across different test cases reveals that the IRBMO algorithm exhibits superior performance in optimizing energy consumption, completion time, and cost compared to other algorithms. This suggests that the initialization method of the IRBMO algorithm effectively generates high-quality initial solutions, and the adaptive cross-variable operator enhances solution quality.

The stability and adaptability of the model on problems of different complexity can be verified by testing under different sizes of workpieces, processes and equipment configurations. With the increase of order size, the relative deviation rate (RPD_E , RPD_T , RPD_Q) of each algorithm generally increases, indicating that the increase of order size increases the scheduling complexity and increases the algorithm solving difficulty. With the increase of buffer capacity, the relative deviation rate (RPD_E , RPD_T , RPD_Q) of each algorithm generally decreases, indicating that a larger buffer capacity helps to alleviate the impact of order insertion on scheduling and improve the algorithm solving effect. As the number of insertion orders increases, the relative deviation rate (RPD_E , RPD_T , RPD_Q) of each algorithm generally increases, indicating that the increase in the number of insertion orders increases the scheduling difficulty and affects the performance of the algorithm.

Despite these challenges, the model effectively optimizes the production plan when considering workpiece size, buffer capacity, and insertion orders. The IRBMO algorithm demonstrates better optimization ability than the traditional RBMO algorithm and other commonly used algorithms. Furthermore, it generates more effective initial solutions through the reverse learning strategy of Logistic Chaos Mapping and improves local optimization by utilizing the adaptive cross-variation operator, thereby enhancing solution quality.

In summary, the IRBMO algorithm effectively addresses the multi-objective HFRP-LB problem, accommodating insertion orders and buffer capacity changes, and adapting to various order sizes, making it an optimal scheduling choice.

5. Example Simulation Analysis.

To validate the feasibility of the optimization model and algorithm, a cutting shop example with a buffer capacity of 5 and an AGV power of 1.89 kW was used. This example included 3 processes, each with 3 parallel machines of varying machining capacities, as detailed in Tables 5-8. Before machining, the first 14 orders were arranged, and the 15th order was inserted to reorder at the 45th second after starting machining.

Table 5 Equipment processing time/s.

workpieces		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Process 1	Device 1	32	80	57	77	69	76	36	63	53	24	36	45	44	78	38
	Device 2	51	65	58	79	46	28	79	30	37	76	61	36	42	45	24
	Device 3	41	42	45	64	53	71	29	39	39	32	55	39	24	65	34
Process 2	Device 4	75	79	67	51	78	52	43	22	65	20	54	56	22	69	50
	Device 5	28	76	47	59	43	74	71	40	39	28	34	27	37	22	61
	Device 6	56	51	35	66	80	26	39	74	24	39	50	31	30	27	55
Process 3	Device 7	55	22	35	64	68	21	28	22	36	50	64	31	55	38	64
	Device 8	23	34	69	31	31	57	27	21	29	31	61	67	56	65	45
	Device 9	46	69	44	76	50	74	60	79	49	31	58	60	51	37	39

Table 6 Processing power/kW.

		Device 1	Device 2	Device 3	Device 4	Device 5	Device 6	Device 7	Device 8	Device 9
workpiece 1	Pin	3.1	3.1	3.5	3.4	2.5	2.1	2.7	3.4	2.8
	Pnl	3.3	2	2.2	2.9	3.5	2.7	2.1	2.4	2.7
workpiece 2	Pin	3.4	2	2.1	2.8	2.6	2	2.3	2.8	2
	Pnl	2.8	3.1	2.2	3	2.7	3.3	2.9	3.1	2.5
workpiece 3	Pin	3.5	2.2	3.3	2	2	2	2.8	2.8	2.2
	Pnl	3	2.3	2.4	2.5	2	2.9	2.6	2.8	2.3
workpiece 4	Pin	3.1	2.4	2.4	2.8	2.9	2.1	2.5	2.1	2.5
	Pnl	3	3.5	2.7	3	3.3	2.7	3.2	3.2	3.3
workpiece 5	Pin	3.3	3.1	2.1	2.4	2.5	3.2	2	2.5	2.5
	Pnl	3.3	2.1	3	3.4	2.1	2.8	2.6	2	2.4
workpiece 6	Pin	3.2	3.3	3.4	2.9	2	2.7	2.4	2.4	3.2
	Pnl	2.3	2.7	3.4	2.1	2.9	3.2	2.1	2.6	2.4
workpiece 7	Pin	2.9	2.5	2.6	2.4	3.2	2.4	2.9	2.2	2.7
	Pnl	3	2.5	2.3	3.3	2.6	2.6	2.2	3.5	2.5
workpiece 8	Pin	2.1	2.9	3.5	2.4	2	3.1	2.3	3	2
	Pnl	2.4	2.4	2.6	2	2.8	2.9	2.5	2.7	2.8
workpiece 9	Pin	2	2.2	2	2.8	2.8	2.7	2.2	2	2
	Pnl	2.2	2.6	3.1	2.2	2.6	2.4	2.3	3.4	2.8
workpiece 10	Pin	2.4	2.6	3.2	3.4	3	2.4	2	2.3	2.1
	Pnl	3.5	2.8	3.5	2.4	2.2	2.1	2	3.4	2.2
workpiece 11	Pin	3.1	2.1	2	2.8	3.2	3.3	2.3	2.7	3.4
	Pnl	3.1	2.3	3.3	2	3	3.3	2.9	3	3.2
Workpiece 12	Pin	2.5	2	2.5	2.4	3.4	3.2	3.1	2.2	3.5
	Pnl	3.2	3.3	2.2	3.5	3.1	2.9	3.1	2.1	3.1
Workpiece 13	Pin	2.2	3.4	2.5	2.6	3	3.4	2	2.6	3.4
	Pnl	2.2	2.7	2.9	2.9	2.5	2.5	2.9	2.3	3.2
Workpiece 14	Pin	2.4	2.4	2.5	2.1	3	2.4	2.2	3	3.4
	Pnl	2.7	2.2	3.1	3.2	2.7	3.4	3.4	2.7	2.9
Workpiece 15	Pin	3	2.3	2	2.6	2.5	2.2	2	2.7	3
	Pnl	3.1	3	2.5	2	3.5	2.5	2.7	2.2	3.4
Pw		1.6	1.8	1.7	1.6	1.9	1.7	1.2	1.5	1.7

Table 7 Device switching data.

		Startup time/s	Power off time/s	Start-up energy consumption/ kW·s	Shutdown Energy Consumption/ kW·s
Process 1	Device 1	81	105	21449	13921
	Device 2	123	125	11713	13384
	Device 3	51	65	12126	12549
Process 2	Device 4	89	112	20645	12505
	Device 5	101	108	19683	15199
	Device 6	129	98	14426	19483
Process 3	Device 7	99	124	19275	13534
	Device 8	72	99	16027	20969
	Device 9	116	114	20536	11777

Table 8 AGV movement time/s.

	Process 2			Process 3		
	Device 4	Device 5	Device 6	Device 7	Device 8	Device 9
Device1 back buffer	32	48	30	0	0	0
Device2 back buffer	50	48	34	0	0	0
Device3 back buffer	45	46	39	0	0	0
Device4 back buffer	0	0	0	36	30	34
Device5 back buffer	0	0	0	39	50	37
Device6 back buffer	0	0	0	34	32	32

The initial scheduling optimization model is established by minimizing the total energy consumption, minimizing the maximum completion time, and minimizing the production cost, and the initial scheduling scheme is obtained by solving IRBMO multiple experiments, and its Gantt chart is shown in Figure 7. Different colors in the figure represent different workpieces, and the processing order of workpieces is indicated by the numbers in the rectangular blocks. The maximum completion time is 1044 s and the total energy consumption is 8277.55 kW·s

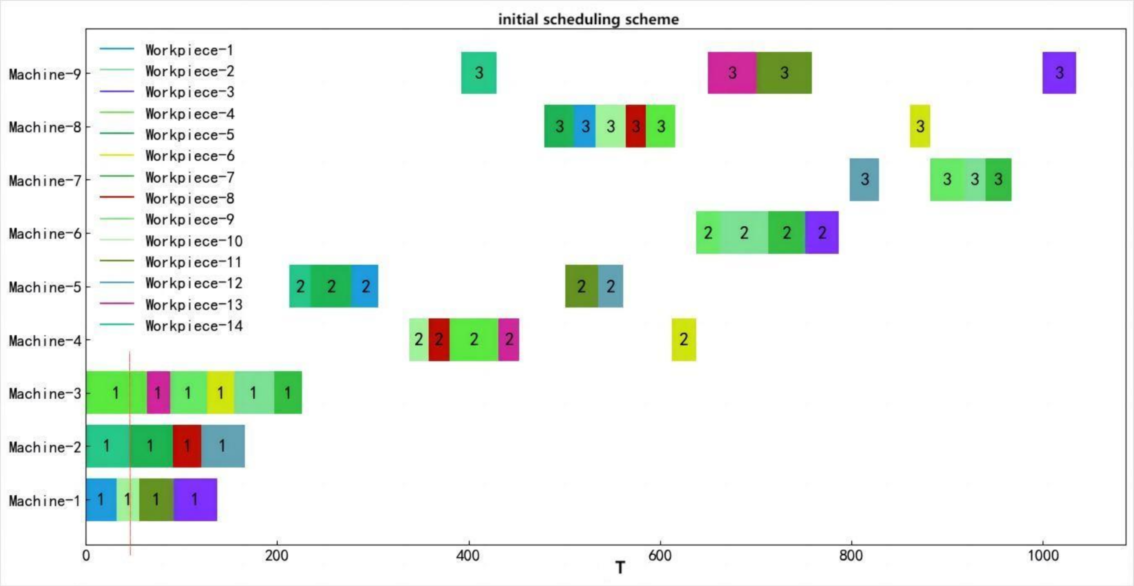


Figure 7 Gantt chart of initial scheduling scheme.

Inserting workpiece 15 at 45 s, the Gantt chart corresponding to the optimal scheduling scheme after rescheduling is shown in Figure 8, in which the maximum completion time is 1102 s, the total energy consumption is 9353.65 kW·s, and the total delay is 58 s. The rescheduling results show that the maximum completion time and the total delay are controlled within a better range, which ensures the stability and efficiency of the rescheduling system.

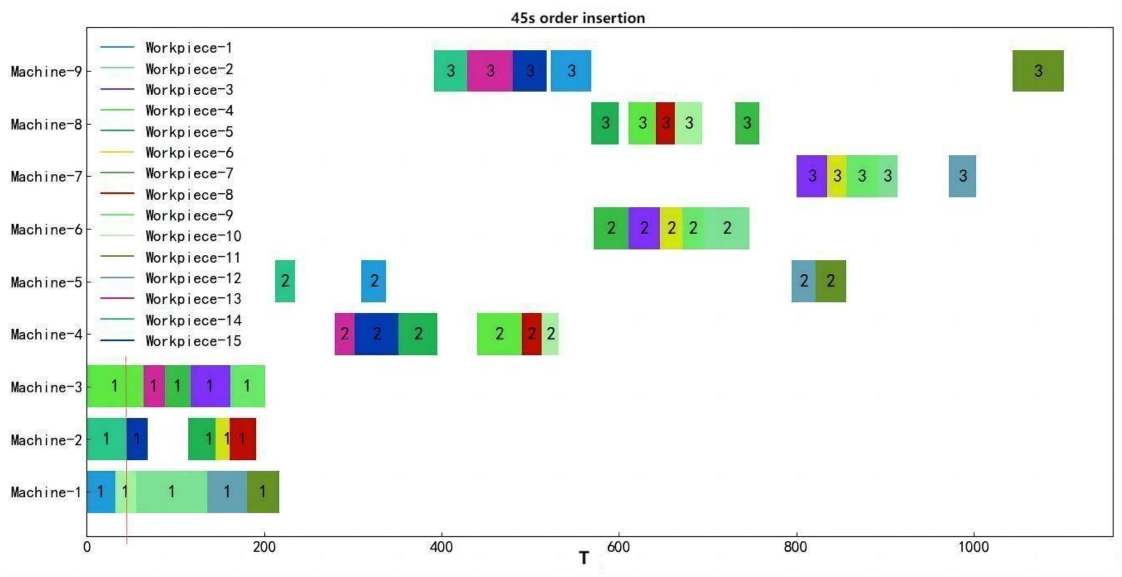


Figure 8 Dynamic scheduling Gantt chart under order insertion.

In order to further verify the effectiveness of IRBMO solving, the standard RBMO, MOGA and MOPSO are used to solve the above insertion order scenarios, respectively, and each comparison algorithm is run independently for 10 times to take the mean value for comparison, and the comparison of the insertion order event scheduling results is shown in Table 9.

Table 9 Comparison of Order Insertion Event Scheduling Results.

insertion order	scheduling algorithm	Production energy consumption/(kW·s)	Completion time/s	Production Cost/CNY
unplugged order	IRBMO	8277.55	1044	11.85
	RBMO	8858.48	1079	12.53
	MOGA	8962.67	1178	13.18
	MOPSO	8535.09	1151	12.83
Insert order at 45s	IRBMO	9353.65	1102	12.25
	RBMO	9625.76	1174	12.88
	MOGA	9872.52	1233	14.09
	MOPSO	11019.69	1169	13.71

As can be seen from Table 6, IRBMO algorithm performs well in the three dimensions of energy consumption, completion time and cost in the unplugged order scenario, followed by RBMO and MOPSO, and MOGA is less effective. The IRBMO algorithm reduces the production energy consumption by 6.56%, the completion time by 3.24%, and the cost by 5.43% as compared to the RBMO algorithm. In the order insertion scenario, the IRBMO algorithm reduces production energy consumption by 2.83%, 5.26%, and 15.12%, completion time by 6.13%, 10.62%, and 5.73%, and production cost by 4.89%, 13.06%, and 10.65%, respectively, in comparison with the other algorithms, which reflects the best adaptability of the IRBMO algorithm to the event of insertion order. Comparing the performance of each dimension before and after order insertion, it is found that the algorithms increase in the dimensions of energy consumption, completion time and cost, among which the production energy consumption of IRBMO and RBMO algorithms increase at a lower rate, which indicates that these two algorithms can control the energy consumption better under the event of order insertion, and MOPSO has the largest increase in the energy consumption, which indicates that it has poorer ability to deal with the event of order insertion; and the increase in the completion time of IRBMO is lower which shows that it can manage time more efficiently under order insertion events, MOPSO has the smallest increase, but its base completion time is higher, and its overall performance is not as good as that of IRBMO; IRBMO and RBMO have a smaller increase in cost, which shows that they have better economics under order insertion events, and MOGA has the largest increase in cost, which shows that it is less economical in dealing with order insertion events. Overall, IRBMO performs well in all three dimensions of energy consumption, completion time and cost, especially in the case of order insertion, where the increases are all smaller, showing its superior adaptability and robustness.

6. Conclusion

In this paper, addressing the emergency insertion single-heavy scheduling problem in hybrid flow shops, a dynamic scheduling optimization model is established. This model considers constraints such as buffer capacity, AGV transport operation, and equipment switching strategies, with the optimization goals of minimizing total energy consumption, maximum completion time, and production cost. The proposed model aims to help enterprises optimize their scheduling schemes, reduce energy consumption and costs, and enhance production efficiency.

To tackle the dynamic scheduling challenges of hybrid flow shops, various improvements are introduced to the red-billed blue magpie optimizer. This includes a Logistic chaotic mapping reverse learning strategy to initialize the population, enhancing the diversity of the algorithm's population and improving initial solution quality. The algorithm also incorporates adaptive genetic crossover and mutation operators, which balance the algorithm's search capabilities and increase the likelihood of escaping local optima. In this study, HFRP-LB algorithms of different scales are designed for computational experiments. By comparing the experimental results of IRBMO and RBMO, the effectiveness of the inverse learning strategy of Logistic Chaos Mapping and adaptive genetic crossover and mutation operators is demonstrated. The results show that IRBMO has the capability to solve the HFRP-LB problem. Finally, the effectiveness and feasibility of the IRBMO algorithm are confirmed through example simulation experiments.

The occurrence of order insertion events in the mixed flow shop belongs to the production disturbance problem, which shows complexity, diversity and uncertainty in the actual production line, so further research on dynamic scheduling is particularly important. The next step of the research has the following two points: 1) consider the events of workpiece withdrawal and mechanical failure during workpiece processing; 2) consider more complex situations such as personnel and multi-logistics delivery systems to establish a more efficient dynamic scheduling model.

Acknowledgements

This research was funded by the National Natural Science Foundation of China, grant number 71771111.

References

[1] N. Yu. Cultivating New Quality Productivity and Promoting the Greening of the Manufacturing Industry. China Ind. News, vol. 7, pp. 05-21(007), 2024.

[2] M. Ren. New Quality Productivity Theory - Leading the Ideological Revolution of New Industrialization in the Future, J. China Univ. Min. Technol. (Soc. Sci.), vol. 26, no. 3, pp. 1-12, 2024.

[3] Y. Feng, Z. Hong, Z. Li, Hao Z. J. Tan. Integrated intelligent green scheduling of sustainable flexible workshop with edge computing considering uncertain machine state, J. Cleaner Prod, vol. 246, pp. 119070-119087, 2020.

[4] N. Wan, L. Li, C. Ye, B. Wang. Risk Assessment in Intelligent Manufacturing Process: A Case Study of an Optical Cable Automatic Arranging Robot, IEEE Access, vol. 7, pp. 105892-105901, 2019.

[5] Y. Lu, J. Lu, T. Jiang. Energy-conscious scheduling problem in a flexible job shop using a discrete water wave optimization algorithm, IEEE Access, vol. 7, PP. 101561-101574, 2019.

[6] Q. Cheng, C. Liu, H. Chu, Z. Liu, W. Zhang, J. Pan. A New Multi-Objective Hybrid Flow Shop Scheduling Method to Fully Utilize the Residual Forging Heat, IEEE Access, vol. 8, 2169-3536, 2020.

[7] Y. Sang, J. Tan. Many-Objective Flexible Job Shop Scheduling Problem with Green Consideration, Energies, vol. 15, no. 5, pp.1884-1900, 2022

[8] J. Guo, L. Wang, L. Kong, X. Lv. Energy-efficient flow-shop scheduling with the strategy of switching the power statuses of machines, Sustainable Energy Technol. Assess., vol. 53, pp. 102649, 2022.

[9] G. Mouzon, M. Yildirim, J. Twomey. Operational methods for minimization of energy consumption of manufacturing equipment, Int. J. Prod. Res., vol. 45, no. 18-19, pp. 4247-4271, 2007.

[10] L. Meng, Z. Chaoyong, S. Xinyu, R. Yaping. MILP models for energy-aware flexible job shop scheduling problem. J. Cleaner Prod., vol. 210, pp. 710-723, 2019.

[11] H. Wei, S. Li, H. Quan, D. Liu, S. Rao, C. Li, J. Hu. Unified multi-objective genetic algorithm for energy efficient job shop scheduling, IEEE Access, vol. 9, pp. 54542-54557, 2021.

[12] X. Wu, Y. Sun. Flexible job shop green scheduling problem with multi-speed machine. Comput. Integr. Manuf. Syst., vol. 24, no. 4, pp. 862-875, 2018.

[13] M. Foumani, K. Smith-Miles. The impact of various carbon reduction policies on green flowshop scheduling, Appl. Energy, vol. 249, pp. 300-315, 2019.

[14] X. Wu, A. Che. Energy-efficient no-wait permutation flow shop scheduling by adaptive multi-objective variable neighborhood search, Omega, vol. 94, pp. 102117-102132, 2020.

[15] J. Wang, Y. Pan, R. Sun. Multi-objective flexible job shop green scheduling problem with self-adaptive Jaya algorithm, Control Decis., vol. 36, no. 07, pp. 1714-1722,2021.

[16] K. Fang, N. Han, F. Zhao, J. W.Sutherlan. Flow Shop Scheduling with Peak Power Consumption Constraints, Ann. Oper. Res., vol. 206, no. 1. pp.115-145, 2013.

[17] L. Zhang, Q. Tang, Z. Wu, F. Wang. Mathematical modeling and evolutionary generation of rule sets for energy-efficient flexible job shops, Energy, vol. 138, pp. 210-227, 2017.

[18] C. Ferreira. Gene Expression Programming in Problem Solving. In: Roy, R. (Ed.), Soft Computing and Industry: Recent Applications. Springer London, London, pp. 635-653, 2001.

[19] R. Chen, B. Yang, S. Li, S. Wang. A Self-Learning Genetic Algorithm based on Reinforcement Learning for Flexible Job-shop Scheduling Problem, Comput. Ind. Eng., vol. 149, no. 1993, pp. 106778, 2020.

[20] P. Li. Intelligent Manufacturing, Sci. Technol. Rev., vol. 5, pp. 59-60, 2019.

[21] F. Geng, X. Luo, Q. Jiang, J. Han, L. Wang. Developing Intelligent Manufacturing, Transforming the Manufacturing Industry. Software Integr. Circuit, vol. 1, pp. 64-67, 2019.

[22] C. Li, Y. Kou, Y. Lei, Q. Xiao, L. Li. Flexible job shop rescheduling optimization method for energy-saving base d on dynamic events. Comput. Integr. Manuf. Syst., vol. 26, no. 02, pp. 288-299. 2020.

[23] S. Hu, F. Liu, Y. He, T. Hu. An On-line approach for energy efficiency monitoring of machine tools, J. Cleaner Prod.,vol. 27, no. 1, pp. 133-140,2012.

[24] B. Gu, X. Tan, X. Tan, B. Xu, H. Li. Study on the Production Unit’s Carbon Emission Accounting Model in the Manufacturing System, Chin. J. Manage. Sci., vol. 26, no. 10, pp. 123-131, 2018.

[25] S. Fu, K. Li, H. Huang, C. Ma, Q. Fang, Y. Zhu. Red-billed blue magpie optimizer: a novel metaheuristic algorithm for 2D/3D UAV path planning and engineering design problems, Artif. Intell. Rev., vol. 57, no. 6, pp. 1-89, 2024.

[26] C. Rim, S. Piao S, G. Li, U. Pak. A niching chaos optimization algorithm for multimodal optimization, Soft Comput., vol. 22, no. 2, pp. 621-33, 2018.

[27] C. Zhang, Y. Rao, X. Liu, P. Li. An Improved Genetic Algorithm for the Job Shop Scheduling Problem, China Mech. Eng., vol. 15, no. 23, pp. 2149-2153, 2004.

[28] Y. Huang, Q. Pan, J. Huang, S. PN, G, Liang. An improved iterated greedy algorithm for the distributed assembly permutation flowshop scheduling problem, Comput. Ind. Eng., vol. 152, pp.107021-107031, 2021.

[29] H. Han, L. Zhang, Y. Hou, J. Qiao. Adaptive candidate estimation-assisted multi-objective particle swarm optimization, Sci. China(Tech. Sci.). vol. 65, no. 8, pp. 1685-1699,2022.

[30] Wang J, Wang L, Cai J, et al. Solution Algorithm of Multi-objective Hybrid Flow Shop Scheduling Problem. J. Nanjing Univ. Aeronaut. Astronaut, vol. 55, pp. 544-552,2023.

[31] B. Wang, K. Huang, T. Li. Permutation flow shop scheduling with time lag constraints and makespan criterion. Comput. Ind. Eng., vol. 120, no. 6. pp. 1-14, 2018.

[32] H. Tang, J. Liu. An Improved Cuckoo Algorithm for Distributed Flexible Flow-shop Scheduling Problem with Transport Time Consideration, Oper. Res. Manage. Sci., vol. 30, no. 11, pp. 76-83, 2021