



Research on flexible job-shop scheduling problem in green sustainable manufacturing based on learning effect

Zhao Peng¹ · Huan Zhang¹ · Hongtao Tang¹ · Yue Feng¹ · Weiming Yin¹

Received: 29 December 2019 / Accepted: 10 November 2020 / Published online: 22 March 2021
© Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

As one of the manufacturing industries with high energy consumption and high pollution, sand casting is facing major challenges in green manufacturing. In order to balance production and green sustainable development, this paper puts forward man–machine dual resource constraint mechanism. In addition, a multi-objective flexible job shop scheduling problem model constrained by job transportation time and learning effect is constructed, and the goal is to minimize processing time energy consumption and noise. Subsequently, a hybrid discrete multi-objective imperial competition algorithm (HDMICA) is developed to solve the model. The global search mechanism based on the HDMICA improves two aspects: a new initialization method to improve the quality of the initial population, and the empire selection method based on Pareto non-dominated solution to balance the empire forces. Then, the improved simulated annealing algorithm is embedded in imperial competition algorithm (ICA), which overcomes the premature convergence problem of ICA. Therefore, four neighborhood structures are designed to help the algorithm jump out of the local optimal solution. Finally, an example is used to verify the feasibility of the proposed algorithm. By comparing with the original ICA and other four algorithms, the effectiveness of the proposed algorithm in the quality of the first frontier solution is verified.

Keywords Green sustainable development · Man–machine dual resource constraint mechanism · FJSP · Learning effect · HDMICA · Improved simulated annealing

Introduction

With the continuous advancement of science and technology, manufacturing is no longer limited to machine. As big data (Wang et al. 2016), mobile internet (Wan et al. 2016), and intelligent decision (Gen and Lin 2014) continue to integrate manufacturing and information technology, the “In-

dustrial 4.0” strategy came into being (Zhang et al. 2019). In order to achieve the development from a large industrial manufacturing country to a strong industrial manufacturing country, the Chinese version of the “Industry 4.0”: “Made in China 2025” was proposed (He and Pan 2015). In addition, as the link between production and sustainable development is getting closer, the goal of manufacturing is not only the minimization of processing time, but also to influence the environment and effectively use resources in the production process have become the target factors that cannot be ignored. Nowadays, the production and operation mode of enterprises is changing to an information-based and intelligent mode, and traditional enterprises are facing major challenges (Ji 2015). Network manufacturing system is the vision of future manufacturing system, a concept that depicts advanced manufacturing systems combined with technologies such as the internet of things, cloud computing, sensor networks, and machine learning (Wu et al. 2019). With the popularization of artificial intelligence (AI), the production mode of manufacturing industry has gradually shifted to automation and intelligence. The intelligent optimization algorithm used to

✉ Hongtao Tang
tanghongtaozc@163.com

Zhao Peng
pz@whut.edu.cn

Huan Zhang
2932532969@qq.com

Yue Feng
2662133749@qq.com

Weiming Yin
528566385@qq.com

¹ Hubei Key Laboratory of Digital Manufacturing, School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan 430070, China

solve the job-shop scheduling problem (JSP) is an embodiment (Hajiaghahi-Keshteli and Fathollahi-Fard 2018). JSP is a kind of basic scheduling problem, which can be defined as job scheduling in a certain sequence (Wang et al. 2018), where each job can be processed on a specific machine. The flexible job-shop scheduling problem (FJSP) is an extension of JSP.

Sand casting is one of the industries with high energy consumption and high pollution (Hans and van de Velde 2011). Due to the complexity of the production process, there is little research in this area. Bewoor et al. (Bewoor et al. 2018) modeled the casting shop scheduling problem as a flow shop scheduling problem (FSP) with no waiting constraints. And the hybrid immune algorithm (HIA) is used to solve the multi-objective flexible job shop problem (Shivasankaran et al. 2015). There are high material and power consumption, complex process, and diverse equipment required in the casting production process. High-temperature, high-dust and high-noise production environment directly affects the health of workshop workers. Therefore, controlling the dust and noise in the foundry and saving energy consumption as much as possible is one of the tasks of modern foundry production (Coca et al. 2019). An energy consumption model is proposed to calculate the energy consumption, and an improved genetic algorithm (GA) used non-dominated solution ranking to optimize multi-objective problems (Wu and Sun 2018). Almost all machinery and manufacturing processes are accompanied by the by-product of noise during production. And the most harmful industrial noise can generally be divided into four categories: Continuous mechanical noise, strong sound generated by high-speed repetitive motion, caused by flow noise and noise produced by punching and impacting the job during machining. The combustion process related to the furnace, the impact noise related to the punching process, motors, generators and other electromechanical equipment are typical examples of vibration sources that generate noise in industrial environments (Olayinka and Abdullahi 2009). Aiming at the topic of green manufacturing, a new mixed integer programming model was proposed, taking carbon emissions and industrial noise as improvement targets, and a multi-objective GA based on simplex lattice design for optimization model (Yin et al. 2017; Lu et al. 2018).

Although intelligent and automation have been popularized in the manufacturing industry, not all industries in all fields can be separated from the role of humans. For this influencing factor of humans, a mixed integer programming model for green production scheduling considering dual flexibility is proposed (Gong et al. 2018; Wirojanagud et al. 2007). In the actual production process, the processing time of the job will be affected by people. Aiming at the constraints of machines and workers, a hybrid particle swarm optimization (HPSO) algorithm combining improved particle swarm optimization (PSO) and variable neighborhood

simulated annealing (SA) is used to solve the dual resource constraint (Zhang 2014). For human factors, a GA combines variable domain search and iterative local search to solve the learning effect and the FJSP for set time (Azzouz et al. 2017). Considering the heterogeneity of employees and jobs, a whale algorithm with chaotic local search strategy was used to solve the FJSP with AGV (Yunqin et al. 2019).

Imperial competition algorithm (ICA) is an intelligent optimization algorithm (Atashpaz-Gargari and Lucas 2007), which has effective global optimization capabilities and certain neighborhood search capabilities. Therefore, this algorithm has certain advantages over other algorithms such as GA (Lei et al. 2018). A new multi-objective ICA based on Pareto-dominated effective constraint processing strategy is proposed to solve the multi-objective low-carbon parallel machine scheduling problem (Cao and Lei 2019).

Based on the above background, this article makes the following research in combination with the characteristics of the foundry industry:

- (1) Considering human and machine dual resource constraints, a multi-objective FJSP model with learning effect based on similarity theory is established.
- (2) According to the sequence of process, machine selection and worker selection, a new initialization method is proposed to improve the quality of the initial solution.
- (3) In the global search process of ICA, the collaborative optimization of colonial countries and colonies is considered, and different intersection methods are designed for process sequencing, machine selection and worker selection respectively, which accelerates the convergence speed of the algorithm.
- (4) Combining the SA with ICA, and using the search ability of SA to perform partial search based on four search mechanisms for some individuals in the solution space, improving the local optimization ability of the algorithm.

Problem statement and mathematical model

Problem statement

Sand casting is a metal forming method that uses sand as the main molding material and various metal raw materials as input finished or semi-finished jobs as output. The process can be divided into three parts: modeling and core making, smelting and pouring, sand treatment and quality inspection. The research process description of this article is shown in Fig. 1.

The classic FJSP can be described as: n jobs $J = \{J_1, J_2, \dots, J_n\}$ are processed on m machines $M = \{M_1, M_2, \dots, M_m\}$, each job has n_i processes $O_i = \{O_{i1}, O_{i2}, \dots, O_{ini}\}$,

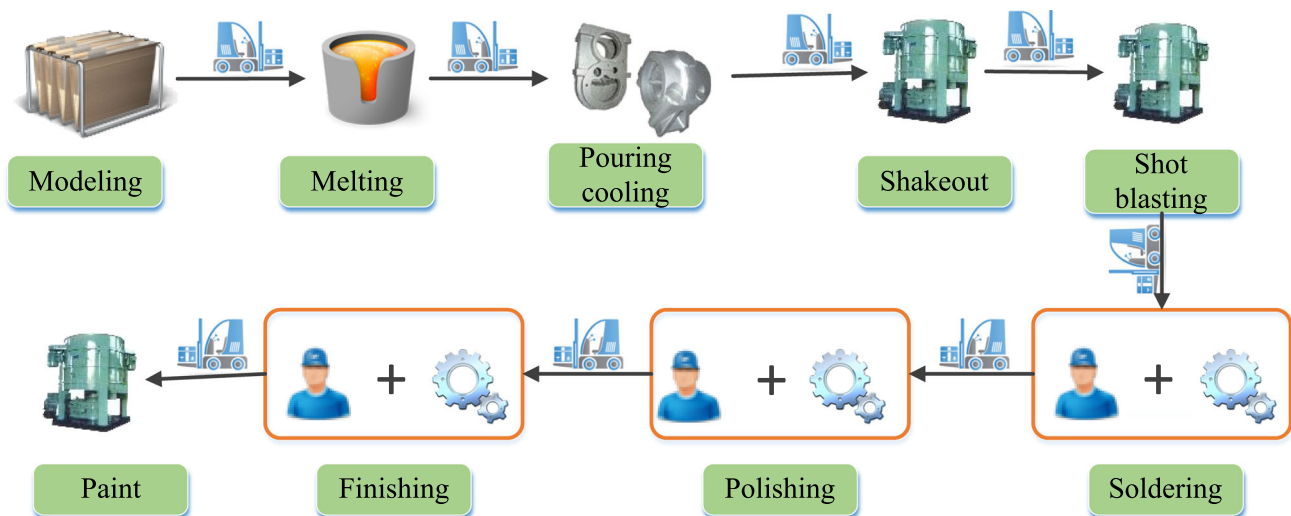


Fig. 1 Sand casting process flow chart

where there is no priority for the processing of each job, but the processing sequence of each job process is determined, and each operation has an optional processing machine set CM .

Dual resource constraint problem

On this basis, considering that in the actual machining process, some parts of the processes are not completed by pure machines, and some processes require human–machine cooperation. Because each person has different working abilities, the processing time for different people operating the same machine is different for the same process. That is to say, there is not only an optional processing machine set CM for each processing operation, but also an optional artificial resource set CP . By selecting the appropriate processing equipment for the process and rationally assigning workers to the processing equipment, the job processing sequence is appropriately optimized to achieve the target value optimization. In general FJSP contains three sub-problems:

1. Process sequencing: Sort the processing sequence of each process of each job.
2. Machine selection: Select one machine in a set of optional processing machines for each process.
3. Worker selection: For part of the process of the role of people cannot be ignored, according to the professional quality of each person operating different machine processing efficiency is different, a process needs to choose a worker resources for processing.

Job transportation time

The starting processing time of the operation theoretically depends on the completion time of the preceding process and the idle time of the processing machine. But since there is a certain distance between two adjacent processes, the actual processing time should consider the job transportation time (JTT), according to which the directed acyclic graph (DAG) of the transport time can be drawn. As shown in Fig. 2. The node number in the DAG indicate the processing machine number. The one-way arrow represents the transportation between the two machines, and the weight on the arrow is the transportation time (h). For example, the number 1 to 5 indicate that the transportation time for transporting the job from the machine 1 to the machine 5 is 0.2(h).

Before the model is established, the following assumptions are made: (1) All machines are idle when starting processing; (2) The processing sequence of the job process cannot be changed; (3) Each operation can only be processed continuously on one machine at any one time, and each machine can only process at most one operation; (4) All machines are switched on and off only once at the start of processing and at the end of processing; (5) Single shipment each time, there is no bulk transport this situation; (6) The calculation of energy consumption only considers the energy consumption of processing equipment, dust removal equipment and transportation equipment during processing; 7) Only the noise generated by processing equipment and dust removal equipment is calculated.

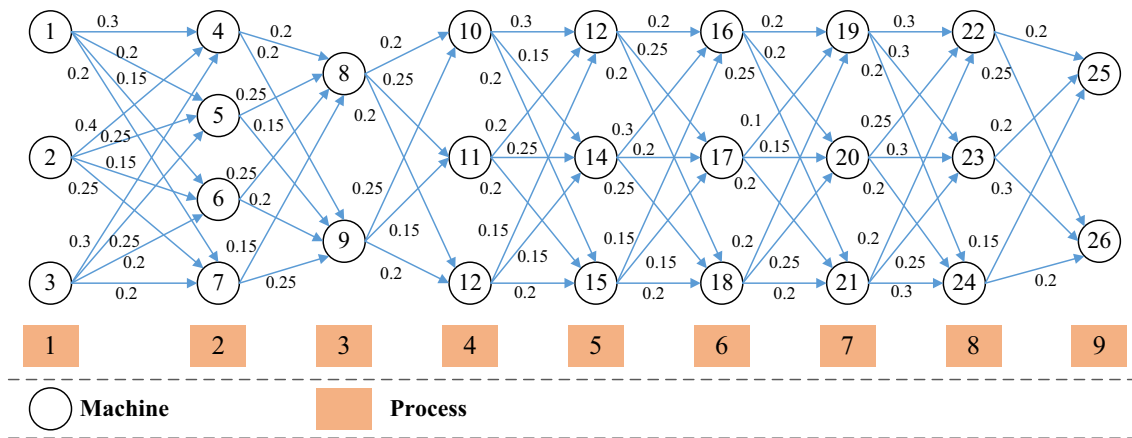


Fig. 2 Directed acyclic graph of transit time

Mathematical model

Job feature similarity

Taking into account the production characteristics of multiple varieties and small batches of the casting process, the number of each type of job will not be very large. If only considering the influence of the cumulative number of the same parts on the processing time, the learning effect will not be obvious, and the learning effect will have little effect. In order to solve this problem, the theory of job feature similarity is proposed that there is a similar relationship between the components of two jobs (Boutsinas 2013). From the following aspects of job feature similarity calculation:

1. Similarity of job types

The castings produced by the foundry in the actual production process may not be exactly the same, but there are types of division. For example, the wheel hub and steel rim are divided into two categories. The type similarity can be expressed by the following formula:

$$u_1(a, b) = \begin{cases} 1, & a \text{ and } b \text{ are of the same type} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

2. Material type similarity

The phenomenon that the processing resources required for jobs of the same type with different materials may be different. The material of the job will seriously affect the proficiency of worker in processing. Therefore, the material similarity cannot be ignored when calculating the feature

similarity of the job product characteristics. The calculation formula is as follows:

$$u_2(a, b) = \begin{cases} 1, & a \text{ and } b \text{ are the same material} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

3. Geometrical similarity

The difficulty of the part geometry will affect the processing time of the job surface. In order to calculate the similarity of the job geometry and simplify the job, only the correlation between the length, width and height of the job is considered (θ_i is the ratio of geometric attributes. For example, if the geometric attribute is length, width, and height, then it is the ratio of length, width, and height. At this case, $i = 3$).

$$u_3(a, b) = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^n \theta_i} \quad (3)$$

4. Feature similarity of job topology

If the job is detachable, no matter how complicated the job can be disassembled into multiple basic shapes such as cylinders, cuboids or prisms, the job can be regarded as consisting of several simple basic shapes. Based on the spatial position relationship of each plane of solid geometry, the topological structure characteristics of the job that represent the spatial positional relationship between the shape features are defined. According to the relative degree of the shape and position relationship, the topological relationship can be divided into adjacency relationship and inclusion relationship. The adjacency relationship refers to the phenomenon of two characteristics coplanar in contact, and the space occupied by the two is a superimposed relationship. While, the inclusion relationship, as the name implies, is the spatial relationship of one feature inside another, and the relationship

of the two pairs is the difference relationship. If the shape feature numbers of a and b are Q_f and Q_r , respectively, the corresponding shape feature number is m ; the topological relationship numbers of a and b are P_f and P_r , and the corresponding topological relationship number is n . Then the similarity of the topological relationship between the jobs a and b can be expressed by the following formula:

$$u_4(a, b) = \frac{1}{M + N} \cdot \left(\sum_{i=1}^m \alpha_i \cdot f_i + \sum_{j=0}^n \beta_j \cdot f_j \right) \tag{4}$$

Among them, $M = Q_f + P_f - m$, $N = Q_r + P_r - n$, f_i and f_j respectively represent the similarity attribute value of the shape feature type and the similarity attribute value of the topological relationship, which are equally important, so $f_i = f_j = 1$; α_i and β_j represent the similarity coefficients of shape features and topological relations, respectively. If the shape types are the same, $\alpha_i = 1$, otherwise α_i is the reciprocal of the number of all shape types, and the value of β_j is the same as the value of α_i .

For example, Fig. 3 is the topological relationship diagram of the jobs a and b , and the similarity of the topological relationship between a and b can be calculated:

$$u_4(a, b) = \frac{(1 \times 1 + 1 \times 1) + (1 \times 1)}{(4 + 3 - 2) + (4 + 2 - 1)} = 0.30$$

5. Process type feature similarity

In the FJSP model, the processing process types of the job are not necessarily identical according to the actual needs, and the similarity of the process types will also have an impact on the processing proficiency of the workers. On the basis of the principle of Jaccard similarity coefficient (Yu 2016), the characteristic similarity of process types can be determined by the following formula ($O_{a,b}$ represents the number of processes that both job a and b need to be processed. O_a represents the number of processes that job a needs to be processed but job b does not. While, O_b represents the number of processes that job b needs to be processed but job a does not):

$$u_5(a, b) = \frac{O_{a,b}}{O_{a,b} + O_a + O_b} \tag{5}$$

6. Similarity of processing time

The processing time for the same process of two jobs also affects the proficiency of workers. The closer the processing

Table 1 Importance scale table

Factor u_i ratio factor u_j	Scaling
Equally important	$Z_{ij} = 1$
Slightly important	$Z_{ij} = 3$
Obviously important	$Z_{ij} = 5$
Strongly important	$Z_{ij} = 7$
Extremely important	$Z_{ij} = 9$
Tradeoff between adjacent scales	$Z_{ij} = 2, 4, 6, 8$
Similar element comparison value	$Z_{ij} = Z_{ij}^{-1}$

time is, the easier it is to learn. The similarity of processing time can be expressed by the following formula (Yu 2016):

$$u_6(a, b) = \sum_{i=1}^O w_i \cdot h_i(a, b) \tag{6}$$

$$w_i = (T_i^a + T_i^b) / \sum_{i=1}^O (T_i^a + T_i^b) \tag{7}$$

$$h_i(a, b) = 1 - \frac{|T_i^a - T_i^b|}{\max(T_i^a, T_i^b)} \tag{8}$$

where $h_i(a, b)$ is the processing time similarity function of job a and job b on process i . T_i^a and T_i^b represent the processing time of the i th process of job a and job b respectively, and the total number of processes is O .

Feature weight coefficient

The weight distribution of the feature similarity of the job directly affects the value of the objective function, and it is difficult to evaluate the importance of each feature to the problem model by unreasonable weighting. Therefore, it is very important to choose the method of weighting scientifically. There are two kinds of common weighting methods: subjective weighting and objective weighting. Among them, subjective weighting method is a method of weighting based on the information of the decision-maker, including analytic hierarchy process (AHP), expert survey method, etc. While, objective weighting method is a method of quantitative weighting based on statistics or mathematics, mainly including entropy method, correlation coefficient method and principal component method.

In this paper, AHP (Hosseini and Al 2019) is adopted to aggregate and combine the workpiece features according to their correlation influence and membership relationship (Table 1) on different levels to form a multi-level analysis structure model, which finally transforms the problem into the importance of the lower level relative to the higher level. The specific steps to determine the weight coefficient according to the judgment matrix Z can be described as:

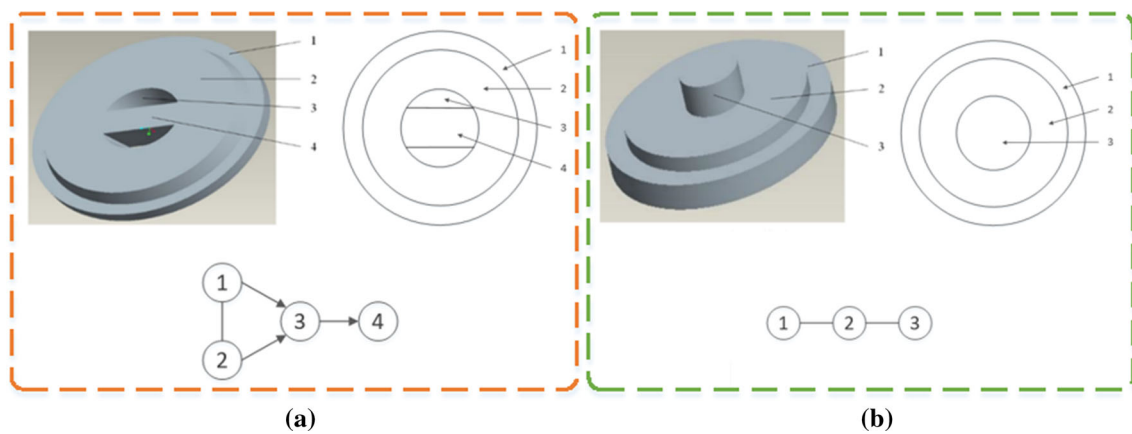


Fig. 3 Job topology diagram

Step 1: Establish a judgment matrix Z based on the factor level table (z_{ij} represents the importance of feature i relative to feature j , with n features in total).

$$Z = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1n} \\ z_{21} & z_{22} & \cdots & z_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \cdots & z_{nn} \end{bmatrix} \quad (9)$$

Step 2: Normalize each column of matrix Z by Eq. (10).

$$\bar{Z}_{ij} = Z_{ij} / \sum_{k=1}^n Z_{kj} \quad (10)$$

Step 3: Add the standardized judgment matrix to each row on the basis of Eq. (11) to obtain a column matrix $D = (\bar{d}_1, \bar{d}_2, \dots, \bar{d}_n)$.

$$\bar{d}_i = \sum_{k=1}^n \bar{Z}_{ik} \quad (11)$$

Step 4: The matrix D is normalized to obtain eigenvectors $d = (d_1, d_2, \dots, d_n)$.

Step 5: Make consistency test according to Eq. (12) to the rationality of weight distribution.

$$CI = \frac{\lambda_{\max} - n}{n} \quad (12)$$

Among them, λ_{\max} is the maximum eigenvalue of the judgment matrix Z , which can be obtained from Eq. (13). And n is the order of the judgment matrix Z .

$$\lambda_{\max} = \sum_{i=1}^n \frac{(Z \cdot d^T)_i}{n \cdot d_i} \quad (13)$$

The smaller the consistency index CI is, the smaller the judgment error is, indicating a better consistency. In order

to measure the size of CI , a random consistency indicator RI is introduced. While, in consideration of the consistency deviation caused by random reasons, when the judgment matrix is satisfied with the consistency test, the CI and RI are compared to obtain the test coefficient CR . And if $CR < 0.1$, the judgment matrix is considered to satisfy the consistency check. The RI values are shown in Table 2.

$$CR = CI / RI \quad (14)$$

FJSP model based on learning effect

The learning effect refers to that when workers process the job, their proficiency gradually increases with the increase of the number of processing, thus shortening the processing time. In addition, due to the heterogeneity of people, the learning effect of each person is different, which leads to the differentiation of processing time. In view of this phenomenon, it is necessary to consider the impact of the learning effect on processing time. In this paper, The DeJong Model based on the logarithm model is adopted to study the problem (Zhao et al. 2017). And the model is as follows:

$$P_{jr} = P_j [M + (1 - M) \cdot r^a] \quad (15)$$

Among them: P_j is the standard processing time, P_{jr} is the actual processing time considering the learning effect, r is the processing position, which is the number of jobs, $a = \lg/l\lg 2 < 0$ is the learning factor, l is the learning rate, and M ($0 \leq M \leq 1$) indicates the proportion of man-machine. All operations are performed manually when $M = 0$, namely Biskup Model (Biskup 1999). When $M = 1$, all operations are performed by the machine, and the processing process does not consider learning effects.

The learning effect of workers is affected by four factors: the initial ability of the worker, the learning ability of the worker, the difficulty of the work, and the number of

Table 2 Mean random consistency index

Matrix order (<i>n</i>)	1	2	3	4	5	6	7	8	9	10
<i>RI</i>	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

repetitions of the work (Yunqin et al. 2019) Where, the initial ability and learning ability of employees are related to the workers themselves, while the difficulty and the number of repetitions of job are related to the jobs themselves. Pre-job training, work experience, etc. will affect the initial capacity of employees. For the differences in the initial personal abilities of workers, the initial processing capacity *P* of the workers is determined by the ratio of the processing time to the standard processing time, and an initial processing capacity matrix *Pst* is constructed (where *P_{ij}* is the initial processing capacity of worker *i* to process the job *j*, and there are *m* workers processing *n* jobs):

$$P^{st} = \begin{bmatrix} P_{11}^{st} & P_{12}^{st} & \cdots & P_{1n}^{st} \\ P_{21}^{st} & P_{22}^{st} & \cdots & P_{2n}^{st} \\ \vdots & \vdots & \ddots & \vdots \\ P_{m1}^{st} & P_{m2}^{st} & \cdots & P_{mn}^{st} \end{bmatrix} \tag{16}$$

In addition to the personal learning ability of workers affected by the cumulative number of processed jobs, observation, understanding, memory and other factors of themselves will affect the learning ability of workers, expressed by the learning rate *l*, and the smaller *l* means the stronger the learning ability. The worker learning rate matrix *L* can be expressed as (where *l_{ij}* is the learning rate of worker *i* processing job *j*, and there are *m* workers processing *n* jobs):

$$L = \begin{bmatrix} l_{11} & l_{12} & \cdots & l_{1n} \\ l_{21} & l_{22} & \cdots & l_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ l_{m1} & l_{m2} & \cdots & l_{mn} \end{bmatrix} \tag{17}$$

The processing time of the job by the workers will not be infinitely reduced or even reached zero. With the increase of the processed job, the processing time tends to stabilize, and the ability of worker reaches an upper limit. This upper limit is related to the difficulty of the job and the workers themselves, and is expressed by the maximum capacity matrix *p^{end}*:

$$p^{end} = \begin{bmatrix} P_{11}^{end} & P_{12}^{end} & \cdots & P_{1n}^{end} \\ P_{21}^{end} & P_{22}^{end} & \cdots & P_{2n}^{end} \\ \vdots & \vdots & \ddots & \vdots \\ P_{m1}^{end} & P_{m2}^{end} & \cdots & P_{mn}^{end} \end{bmatrix} \tag{18}$$

Therefore, The DeJong Model can be modified according to the four major factors that affect the learning rate and the similarity theory of the job:

$$P_{ijr} = P_{ij} \cdot \max \left\{ P_{ij}^{st} \cdot \left(\sum_{r=1}^j S(j, r) \right)^a, P_{ij}^{end} \right\} \tag{19}$$

Combined with the similarity of the job and learning effect, the entire learning effect flowchart is shown in Fig. 4:

(1) Mathematical notation

- n*: Total number of jobs
- m*: Total number of processing machines
- m'*: Total number of workers
- n_i*: Number of steps for job *i*
- Ω*: Total machine set
- Ω_{ij}*: Optional processing resource set for the *j*th operation of the job *i*
- i, i'*: Index of the job, *i, i' = 1, 2, ..., n*
- j, j'*: Process index, *j, j' = 1, 2, ..., n_i*
- h*: Processing machine index, *h = 1, 2, ..., m*
- h'*: Worker index, *h' = 1, 2, ..., m'*
- m_{ij}*: Number of optional processing machines for the *j*th operation of job *i*
- O_{ij}*: The *j*th process of job *i*
- b*: Total number of dust removal equipment
- y*: Total number of transportation equipment
- k*: *k*th dust removal equipment, *k = 1, 2, ..., m'*
- c*: The *c* transport equipment, *c = 1, 2, ..., y*
- C_{ijh}*: The theoretical processing time of the *j*th process of job *i* on processing equipment *h*
- TC_{ijh}*: Actual processing time of the *j*th process of job *i* on processing equipment *h*
- C_i*: Earliest completion time of job *i*
- T_{ijk}*: Dust removal time of the *j*th process of job *i* on the dust removal equipment *k*
- T_{ij(j+1)h}*: Transport time of the process *j* to the process (*j*+1) of job *i*
- Power_h*: Working power of processing equipment *h*
- Power0_k*: Working power of dust removal equipment *k*
- P_c*: Transportation equipment *c* consumption of gasoline per unit time when transporting job
- St_{ij}*: Start time of the *j*th operation of job *i*
- Et_{ij}*: End time of the *j*th process of job *i*
- Be*: Standard coal conversion coefficient of electricity, 0.1229 kgce/(kW·h)

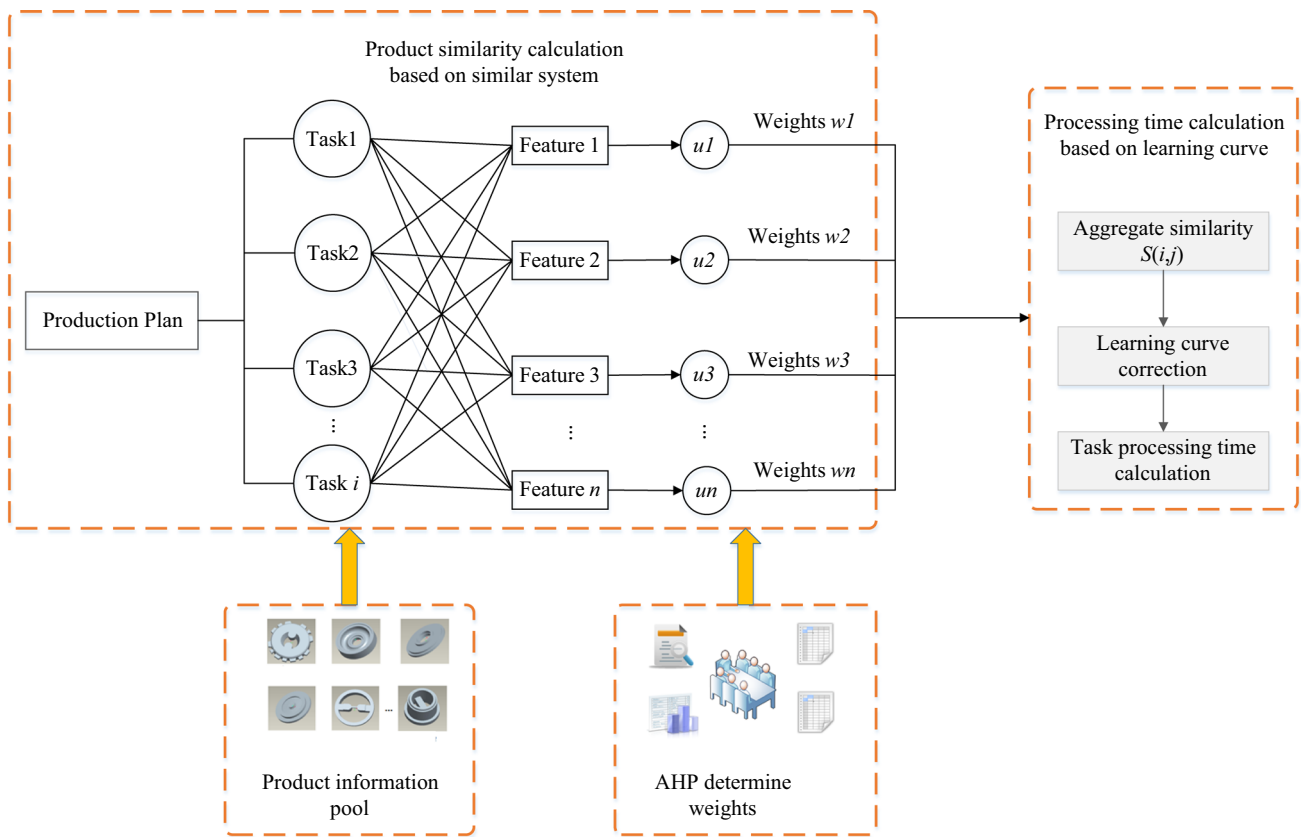


Fig. 4 Learning effect model flow chart

Bq : Standard coal conversion coefficient of gasoline, 1.4714 kgce/kg

V_{ijh} : The instantaneous A-weighted sound level of machine h for the j th process of job i

N : A large enough positive number

(2) Decision variables

$$\begin{aligned}
 x_{ijh} &= \begin{cases} 1, & \text{Process } O_{ij} \text{ is processed on machine } h \\ 0, & \text{Otherwise} \end{cases} \\
 x_{hijj'j'} &= \begin{cases} 1, & \text{Processing } O_{ij} \text{ is processed on machine } h \text{ before } O_{i'j'} \\ 0, & \text{Otherwise} \end{cases} \\
 x_{ij(j+1)c} &= \begin{cases} 1, & \text{Process } O_{ij} \text{ to process } O_{i(j+1)} \text{ require transport machine } c \\ 0, & \text{Otherwise} \end{cases} \\
 y_{ijh} &= \begin{cases} 1, & \text{Process } O_{ij} \text{ requires dust removal equipment } h \\ 0, & \text{Otherwise} \end{cases} \\
 z_{ijh'} &= \begin{cases} 1, & \text{Process } O_{ij} \text{ requires worker } h' \\ 0, & \text{Otherwise} \end{cases} \\
 M &= \begin{cases} 1, & \text{Process } O_{ij} \text{ has only machine resources} \\ 0, & \text{Otherwise} \end{cases} \\
 y_{ij} &= \begin{cases} 1, & t_i < S_{t_i(j+1)} - S_{t_{ij}} \\ 0, & \text{Otherwise} \end{cases}
 \end{aligned}$$

(3) Constraint

$$S_{t_{ij}} + x_{ijh} \cdot TC_{ijh} \leq Et_{ij} \tag{20}$$

$$Et_{ij} + T_{ij(j+1)} \leq S_{t_{i(j+1)}} \tag{21}$$

$$S_{t_{ij}} + x_{ijh} \cdot TC_{ijh} \leq S_{t_{i'j'}} + N \cdot (1 - x_{hijj'j'}) \tag{22}$$

$$Et_{ij} + T_{ij(j+1)} \leq S_{t_{i(j+1)}} + N \cdot (1 - x_{hijj'j'}) \tag{23}$$

$$\sum_h^m x_{ijh} = 1 \tag{24}$$

$$Et_{ij} = S_{t_{ij}} + x_{ijh} \cdot TC_{ijh} \tag{25}$$

$$\sum_h^m \sum_{h'}^{m'} x_{ijh} \cdot x_{ijh'} \leq 1 \tag{26}$$

Equations (20) and (21) constraint the process route of each job cannot be changed; Eqs. (22) and (23) indicate that a machine can only process one process at a time; Eq. (24) represents that each process can only be processed on one machine; Eq. (25) ensures that no interruption is allowed once each process is started; Eq. (26) denotes that a machine can only be operated by one worker when processing a job.

(4) Objective functions

Table 3 Standard coal conversion coefficients for different energy

Types of energy	Standard coal conversion coefficient	Types of energy	Standard coal conversion coefficient
Electricity	0.1229 kgce/(kW h)	Natural gas	1.2143 kgce/m ³
Raw coal	0.7143 kgce/kg	Coke	0.9714 kgce/kg
Crude	1.4286 kgce/kg	Liquefied petroleum gas	1.7143 kgce/kg
Gasoline	1.4714 kgce/kg	Kerosene	1.4714 kece/kg

(1) Minimize the makespan

The minimum time required to complete all job processing. This time includes not only the total processing time of each process but also the JTT between two adjacent processes.

$$f_1 = \min(C + T) \tag{27}$$

$$C = \sum_{i=1}^n \sum_{j=1}^{n_i} \sum_{h=1}^m x_{ijh} \cdot TC_{ijh} \tag{28}$$

$$T = \sum_i^n \sum_j^{n_i-1} \sum_c^y x_{ij(j+1)c} \cdot T_{ij(j+1)c} \tag{29}$$

(2) Energy consumption

The energy required to support the operation of the machine during processing, including the energy consumed by processing machines and the energy consumption of dust removal equipment and transportation equipment. Different production processes consume different types of energy. For example, the smelting process mainly consumes coke, and the molding consumes raw coal. For example, the smelting process mainly consumes coke, the molding core consumes raw coal, and the energy consumed when a job is transported by a forklift is gasoline. Due to the different types of energy consumed at each stage, the concept of standard coal conversion coefficient is introduced for the convenience of calculation. According to the current national standards of various energy conversion standard coal reference coefficients, common coal conversion coefficients for different energy standards are shown in Table 3:

$$f_2 = \sum_{i=1}^n \sum_{j=1}^{n_i} \sum_{h=1}^m (x_{ijh} \cdot Power_h + y_{ijh} \cdot Power_{0h}) \cdot TC_{ijh} \cdot Be + T \cdot Pc \cdot Bq \tag{30}$$

(3) Minimize noise

Sound pressure and sound pressure level, sound intensity and sound intensity level, sound power and sound power level can all be used to describe the size of sound. Generally, sound pressure level, or decibel (dB), is used to describe sound size. Decibel represents the ratio of two physical quantities

and describes the size of the physical quantity as a numerator relative to the reference value of the denominator. For foundry, the main sources of noise are the noise generated by the operation of the machine, the noise caused by the contact between the job and the machine during the processing of the job. For these structural noises (Tandon 2000), when the machine is working with a certain power, the noise is radiated in a stable wide band. According to the “Environmental Noise Emission Standard for Industrial Boundaries of Industrial Enterprises”, this article uses A-weighted sound level to evaluate the radiation of processing noise. A-weighted sound level is the noise level measured by a sound level meter or equivalent measuring instrument through the A weighting network. And the equivalent sound level can be expressed as:

$$Leq = 10 \cdot lg \left(\frac{1}{T} \cdot \int_0^T 10^{0.1 \cdot l_i} dt \right) \tag{31}$$

where l_i —the instantaneous A-weighted sound level at time t ; T —the measurement period.

For the convenience of calculation, when the machine is processing with a certain power, the radiation sound level of noise at each stage is considered to be definite, so the equivalent sound level can be expressed as:

$$Leq = 10 \cdot lg \left(\frac{\sum_i 10^{0.1 \cdot l_i} T_i}{\sum_i T_i} \right) \tag{32}$$

where l_i —Instantaneous A-weighted sound level for i paragraphed time, T_i — i paragraphed time.

In the problem studied in this paper, considering the different instantaneous A sound level of the sound sources of various machines, for different sound pressure levels corresponding to different machines, the noise generated during the entire processing can be calculated by the following formula:

$$f_3 = 10 \cdot lg \left(\frac{\sum_{h=1}^m \sum_{i=1}^n \sum_{j=1}^{n_i} 10^{0.1 \cdot V_{ijh}} \cdot C_{ijh} \cdot x_{ijh}}{\sum_{h=1}^m \sum_{i=1}^n \sum_{j=1}^{n_i} C_{ijh} \cdot x_{ijh}} \right) \tag{33}$$

Algorithm design

Background

(1) ICA

ICA is a kind of intelligent optimization algorithm inspired by empire competitive behavior. Each individual in the population represents a country. ICA will select the best individuals as colonial countries on the grounds of the size relationship of the fitness value of each individual, and other individuals will be assigned to colonial nations as colonies. Colonial nations and colonies will be updated through assimilation, revolution, and internal competition of the empire, through empire assimilation, revolution, and internal competition to update the colonial countries and colonies.

(2) Pareto non-dominated sort

Single objective optimization problem by directly comparing the target values can determine the solution to the pros and cons. However, for multi-objective optimization problems, because there are more than one target value, when the objective functions are in a state of conflict, there may not be an optimal solution that achieves the maximum or minimum of all objectives at the same time, and it is difficult to compare the advantages and disadvantages of the two solutions. Non-dominated solution sorting as an important sorting method, can sort solutions according to the quality of the solution. Solutions with a higher Pareto Rank dominate other solutions, and solutions with a Rank of 1 are called non-dominated solutions. This paper uses the fast non-dominated solution ranking method proposed in NSGA-II (Deb et al. 2000).

(3) Congestion distance calculation

The crowding distance indicates the denseness of the target space. The smaller the value is, the denser the target space is and the easier it is to fall into the local optimal solution; the larger the value is, the looser the target space, and the richer the population diversity. Therefore, solutions with larger crowding distances can be chosen when they are at the same Rank. The crowding distance of particle i can be calculated by the following formula:

$$d_i = \begin{cases} \infty, & i = 1 \text{ or } i = l \\ \sum_{k=1}^n |f_k^{i+1} - f_k^{i-1}|, & \text{otherwise} \end{cases} \quad (34)$$

Among them, n represents the number of targets, and f_k^{i+1} and f_k^{i-1} represent the target values of adjacent particles of particle i on target k , and l is the number of particles.

(4) SA

SA is an intelligent optimization algorithm that is widely used in the field of combinatorial optimization (Xinchao 2011). The algorithm is derived from the principle of solid annealing. The concept of slow cooling implemented in SA can be understood as the slow reduction of the probability of accepting a poor solution when exploring the solution space, which is of great significance for increasing exploration to avoid local optimization (Tang et al. 2019).

Coding and initialization based on multiple rules

In order to effectively utilize the intelligent optimization algorithm, the problem to be solved needs to be converted into a data type that can be recognized by the computer. For the FJSP model, this paper adopts the MSOS integer coding method, which designs three-stage coding rules for the three problems of process sequencing, machine selection, and worker selection involved in the model. For the individual $S = [SP \mid SM \mid SN]$, the SP segment indicates the process sequencing segment, SM indicates the machine selection segment, and SN indicates the worker selection segment. For example, individual $S = [2, 1, 1, 3, 2, 1, 3, 2, 3 \mid 1, 3, 1, 1, 2, 1, 2, 2, 3 \mid 0, 0, 1, 0, 3, 2, 1, 2, 1]$. The encoding scheme for this solution is shown in Fig. 5. Process sequencing part $SP = [1, 1, 1, 2, 2, 2, 3, 3, 3]$, the element value in SP represents the job number, and the number of occurrences of each number represents the number of processes. For instance, the second occurrence of the fifth number 2 indicates the second process of job 2 (O_{22}). Machine selection part $SM = [1, 3, 1, 1, 2, 1, 2, 2, 3]$, the element value in SM represents the index of the machine selected in the set of alternative machines of the corresponding operation, such as, the fifth number 2 represents the second machine in the set of machines for which the corresponding process is optional. Worker selection part $SN = [0, 0, 1, 0, 3, 2, 1, 2, 1]$, the element value in SN represents the index of the worker selected in the set of alternative workers of the corresponding operation. The position marked with zero indicates that the corresponding process requires no workers for pure machine processing, and the non-zero position is the processing worker serial number, for example, the fifth number 3 indicates the corresponding the third worker in the set of optional processing workers.

Since the initial population is the premise of algorithm optimization, its quality will directly affect the performance of the algorithm. Although the initial population generated by the random coding mechanism is high in diversity, the population quality is poor, so that it is more difficult to find iteratively in the process of iterative optimization. Many researches on this issue focus on simple heuristic rules, active scheduling, or a combination of heuristic rules and active scheduling. This paper combines active scheduling with

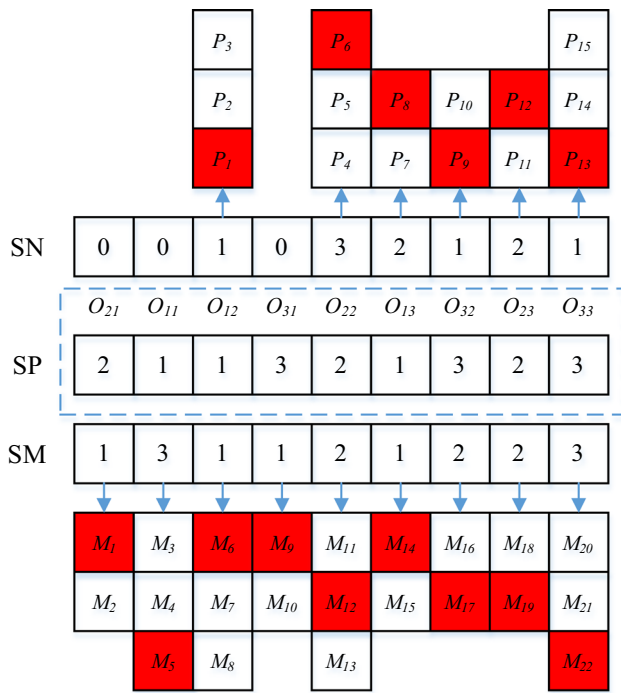


Fig. 5 The encoding scheme

heuristic rules, and adopts the shortest processing time (SPT) rule to optimize the minimum completion time during population initialization. In order to achieve the comprehensive optimization of other goals, the random initialization method is retained, which not only preserves population diversity but also improves population quality. The population was initialized with a 1:1 ratio. Figure 6 shows the specific process of population initialization. The specific steps are as follows:

Step 1: Set $S = \{ \}$, U as the collection of the first phase.

Step 2: The minimum value of the earliest possible completion time in U is recorded as T_s , and a machine M with the earliest idle time of T_s is selected.

Step 3: From U , select the process that can be processed on M and the idle time of the job is not greater than T_s . If there are multiple processes, generate a random number R between $(0,1)$. If $R < 0.5$, select SPT rule, otherwise, randomly initialize.

Step 4: Put the selected operation into the scheduled process set S , and update the schedulable process set U . If there are unscheduled operation in U , go to Step2, otherwise, end.

Where, S is the set of scheduled processes, U is the set of schedulable operations; T_s represents the earliest start-up time of the process in the set U . When U is empty, it means that all processes have been scheduled and completed, and the

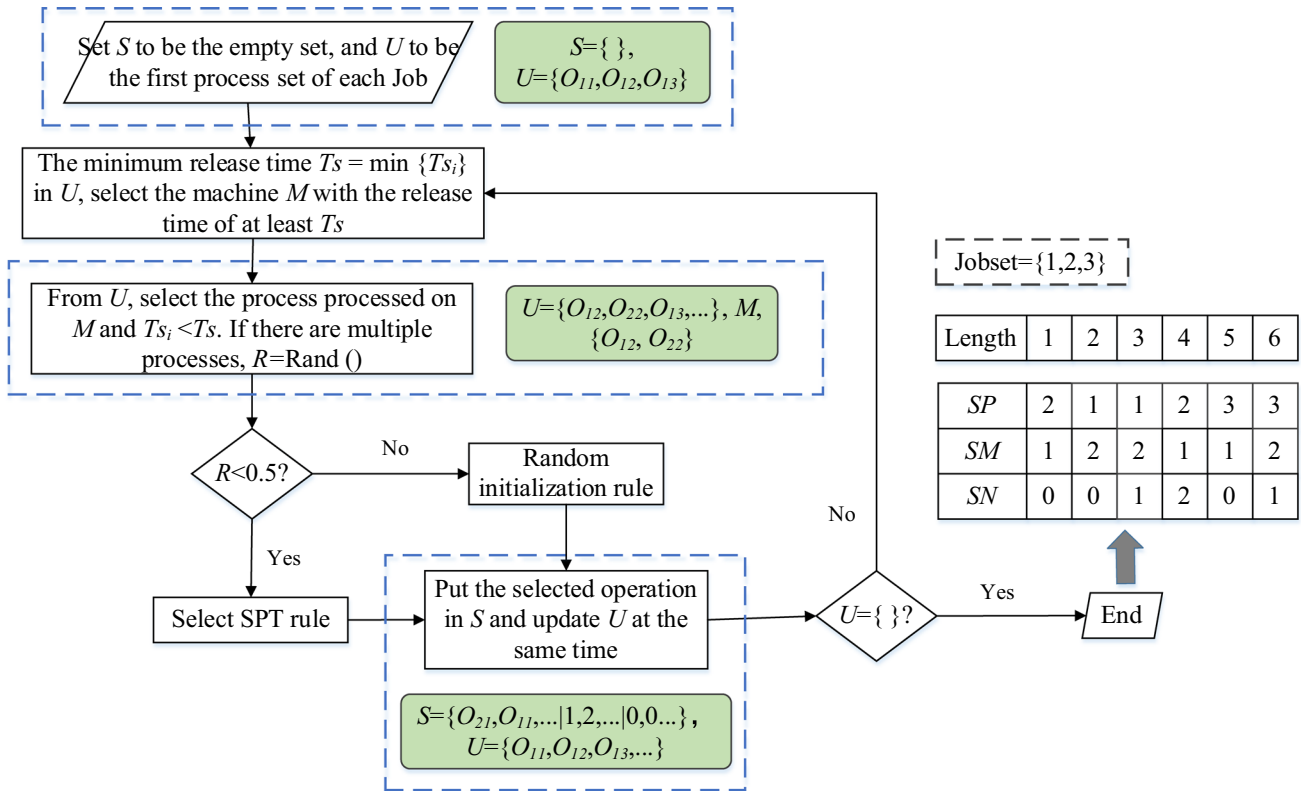


Fig. 6 Flow chart of population initialization

sequence of processes in S is the final obtained chromosome gene.

Initialize empire

As the first stage of hybrid discrete multi-objective Imperial competition algorithm (HDMICA), initializing the empire is the most critical step. A reasonable division of the country will affect the optimization performance of the entire population. For the multi-objective optimization model, this paper uses a Pareto non-dominated sorting algorithm to find a non-dominated solution set of number η in a population P of size N . If η is less than the number N_{emp} of colonial countries, the population P is updated, and the resulting high-quality individuals Pt are put into P as external populations until η is not less than N_{emp} . The former N_{emp} non-dominated solutions are set as colonial countries, and Rank of the remaining individuals are sorted in ascending order and assigned to colonial countries in a certain proportion as colonies. In this way, the overall value of each empire tends to be consistent, so that later empire competition will fully play its role.

Step 1: Sort the population P Pareto non-dominated solutions to obtain a non-dominated solution set π of size η .

Step 2: Determine the size relationship between η and N_{emp} . If $\eta < N_{emp}$, then perform a local search on π to generate an external population Pt , put Pt into the population P , and execute Step 1; otherwise, execute Step 3;

Step 3: Select N_{emp} non-dominated solutions in π as the colonial countries, and calculate the number of colonies allocated to each empire, $Col_i = [(N - N_{emp}) / N_{emp}] + \alpha$, $\alpha \in \{-1, 1\}$, $i = 1, 2, \dots, N_{emp} - 1$, α is a random number that takes an integer between $\{-1, 1\}$, and the number of colonies in the last empire is $Col_{N_{emp}} = (N - N_{emp}) - \sum_{i=1}^{N_{emp}-1} Col_i$.

Step 4: The Rank of the remaining $N - N_{emp}$ individuals in P are sorted in ascending order, and each empire is assigned in order.

Global search

Assimilation and revolution

The so-called assimilation, as its name suggests, makes two or more individuals tend to be similar. The assimilation process here is to bring the colony closer to the colonial country, which is usually achieved by crossover of the colony and the colonial country (Karimi et al. 2017). However, colonial countries, as elite individuals, will have a greater probability of producing excellent individuals through mutual learning. In response to this problem, this article gives a new way of assimilation. For the three sub-problems of the scheduling workshop, three crossover methods are proposed. The first

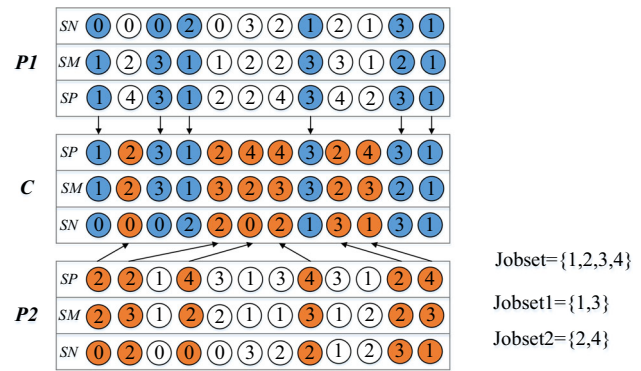


Fig. 7 IPOX diagram

crossover adopts IPOX improved POX crossover in KacemI to act on the process segment, which can be described as: The job set $Jobset = \{J_1, J_2, J_3, \dots, J_n\}$ is divided into two parts, Jobset1 and Jobset2, then the jobs belonging to Jobset1 in the parent $P1$ chromosome are copied to the child C , and while the ones belonging to Jobset2 in the parent $P2$ chromosome are kept in the same order and added to the C vacant position in order. The specific crossover operation is shown in Fig. 7.

The second type of crossover acting on the machine segment, using RPX cross, can be described as: generate a vector whose elements are composed of random numbers between (0,1), the size of the vector is determined by the length of the machine segment, and the elements in the vector are correspond to the machine code one by one. First, copy the process section and worker section from the parent $P1$ to the child C . Then, the machine code greater than pf in parent $P1$ is copied to child C , and the index of the remaining machine code is recorded. Finally, the machine code corresponding to the index in parent $P2$ and the process segment is found and copied to the vacant position in C . The pf varies with the number of iterations, and is calculated using the following formula:

$$pf = pf_{max} - \frac{pf_{max} - pf_{min}}{It} \times MaxIt \tag{35}$$

pf_{max} and pf_{min} are the maximum and minimum values of the self-adaptation degree, the total number of iterations is $MaxIt$. It is the current algebra (The parameter of measuring the number of iterations in the algorithm iteration process). And the Fig. 8 is a crossover diagram when $pf = 0.5$.

The third crossover method adopts a two-point crossover method to act on the worker selection segment: First, copy $P1$ and $P2$ to $C1$ and $C2$, respectively. Then, a job is randomly selected, and the machine number and worker number corresponding to the job are interchanged in the two children. As shown in Fig. 9.

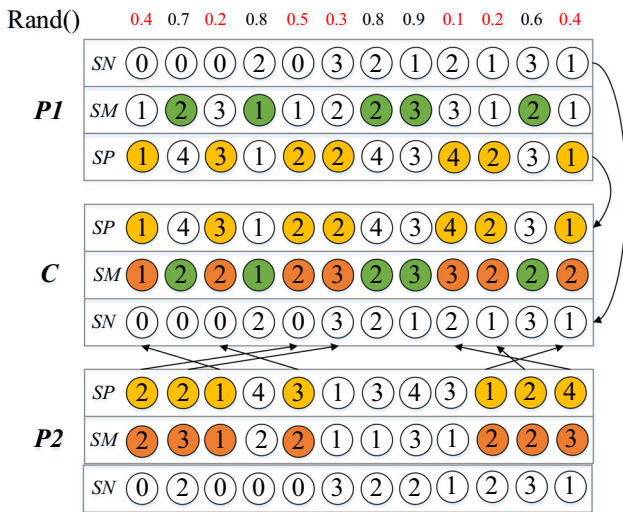


Fig. 8 RPX diagram

Empire competition

Colonial and colonial countries, colonial countries and colonial countries have undergone tremendous changes through assimilation and revolutionary empires. The weaker empire will lose its weakest colony. Other empires compete and occupy the colony with a certain probability, and usually the empire with a larger power has a higher probability of occupying the colony, until the weaker empire colony becomes zero and the colonial country is occupied as a colony, then the entire empire dies. For carrying out empire competition, an empire cost calculation method is defined. The cost c_i of country i can be expressed by the inverse of the weighted sum of each target value:

$$c_i = 1 / \sum_{j=1}^n (w_j \cdot f_j) \tag{36}$$

where w_j is the weight of the target j calculated by the principal component analysis (PCA) (Liu et al. 2019), and f_j is the target value of the target j . The total power of the empire can

be expressed by the weighted sum of the cost of the colonial country and the average cost of the colony. This article takes $\xi = 0.2$:

$$TC_k = c_k + \xi \cdot \text{mean}\{c(\text{colonies of empire } k)\} \tag{37}$$

Local search mechanism based on SA

In the hybrid optimization algorithm designed in this paper, SA is used for local search. Each time the algorithm moves, a new neighborhood solution is generated around the current solution. The effective neighborhood structure can greatly improve the local optimization ability, so four kinds of neighborhood structures are designed in this paper. As shown in Fig. 10.

NS1: Generate two positions randomly and reverse the operation between the two positions.

NS2: Critical path based switching operations. The critical path refers to the longest route from the first operation to the end of the last operation, which determines the completion time of the schedule. This paper adopts the N5 neighborhood structure proposed by Nowicki and Smutnicki (1996) to randomly select a critical path, and swaps the first two (and the last two) operations in each key block to achieve.

NS3: Process Insertion Operation: Randomly select two processes a and b from the process segment, then insert process b before process a .

NS4: Resource reallocation: Randomly select an element in the resource allocation segment, and then reselect the resource from the candidate resource set to replace the current resource.

The improved HDMICA algorithm process is as follows:

Step 1: Generate population P based on rule-based population initialization;

Step 2: The non-dominated solution set π of size η is obtained by non-dominated sorting. If $\eta < N_{emp}$, execute Step 3; otherwise, execute Step 4;

Step 3: Perform a global search on π to obtain the external population P_t , and merge P_t and P into P ;

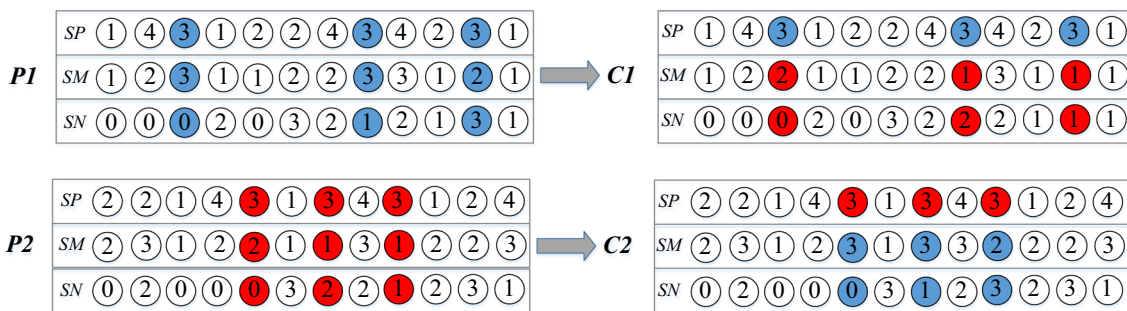


Fig. 9 Two-point crossover diagram

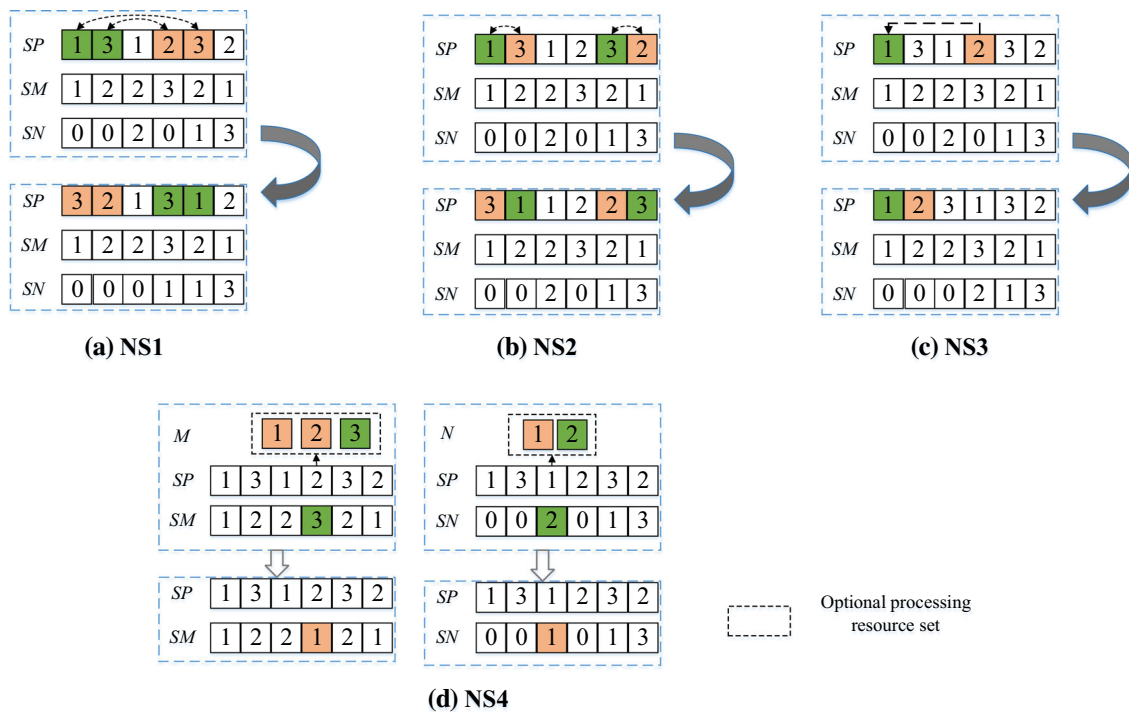


Fig. 10 Schematic diagram of four neighborhood structures

Step 4: Initialize the empire based on Pareto according to the number N_{emp} of empires;

Step 5: Assimilation and revolution of colonial countries and colonies, colonial countries and colonial countries;

Step 6: Randomly select some individuals P_{cal} from population P for SA local search;

Step 7: Update the power of the empire. The weaker countries are gradually annexed by other powerful countries. If only one country remains or the maximum number of iterations is reached, go to Step 5; otherwise, the algorithm ends.

The overall flowchart of the algorithm is shown in Fig. 11.

Case study

Case description

Existing production data of a foundry, an order of 16 jobs, each job has 8 or 9 procedures, requires processing on 26 machines. Each process has a set of processing machines. According to actual needs, some processes require workers to participate in processing, and some processes require dust removal equipment. The specific data is shown in Table 4, where *People* indicates workers, *PowerI* indicates the power of the processing machine, and *Power0* indicates the power of the dust removal equipment. *Noise* is the A sound level produced during processing. Table 5 shows various processing parameters. The second column in Table 5 shows the

material (1, 2, 3) and size (length, width and height) of the job. L is the learning rate of employees, Pst is the initial processing capacity of workers, $Pend$ is the maximum capacity of employees, and T is the theoretical processing time.

Based on the field investigation of the foundry enterprise, according to the AHP, the weight coefficients between the six features can be obtained. The discriminant matrix Z constructed is shown in the following formula.

$$Z = \begin{bmatrix} 1 & 3 & 5 & 4 & 5 & 2 \\ \frac{1}{3} & 1 & 3 & 2 & \frac{1}{4} & \frac{1}{3} \\ \frac{1}{5} & \frac{1}{3} & 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{5} \\ \frac{1}{4} & \frac{1}{2} & 2 & 1 & \frac{1}{5} & \frac{1}{4} \\ \frac{1}{5} & 4 & 3 & 5 & 1 & \frac{1}{3} \\ \frac{1}{2} & 3 & 5 & 4 & 3 & 1 \end{bmatrix}$$

Evaluation criteria

This paper proposes three metrics to evaluate the quality and diversity of non-dominated solutions (Behnamian and Ghomi 2011). The specific methods are as follows:

- (1) Mean ideal distance (MID). Shows the distance between the non-dominated solution and the ideal point (0, 0). The smaller the value, the closer the non-dominated solution is to the ideal value, the better the result.

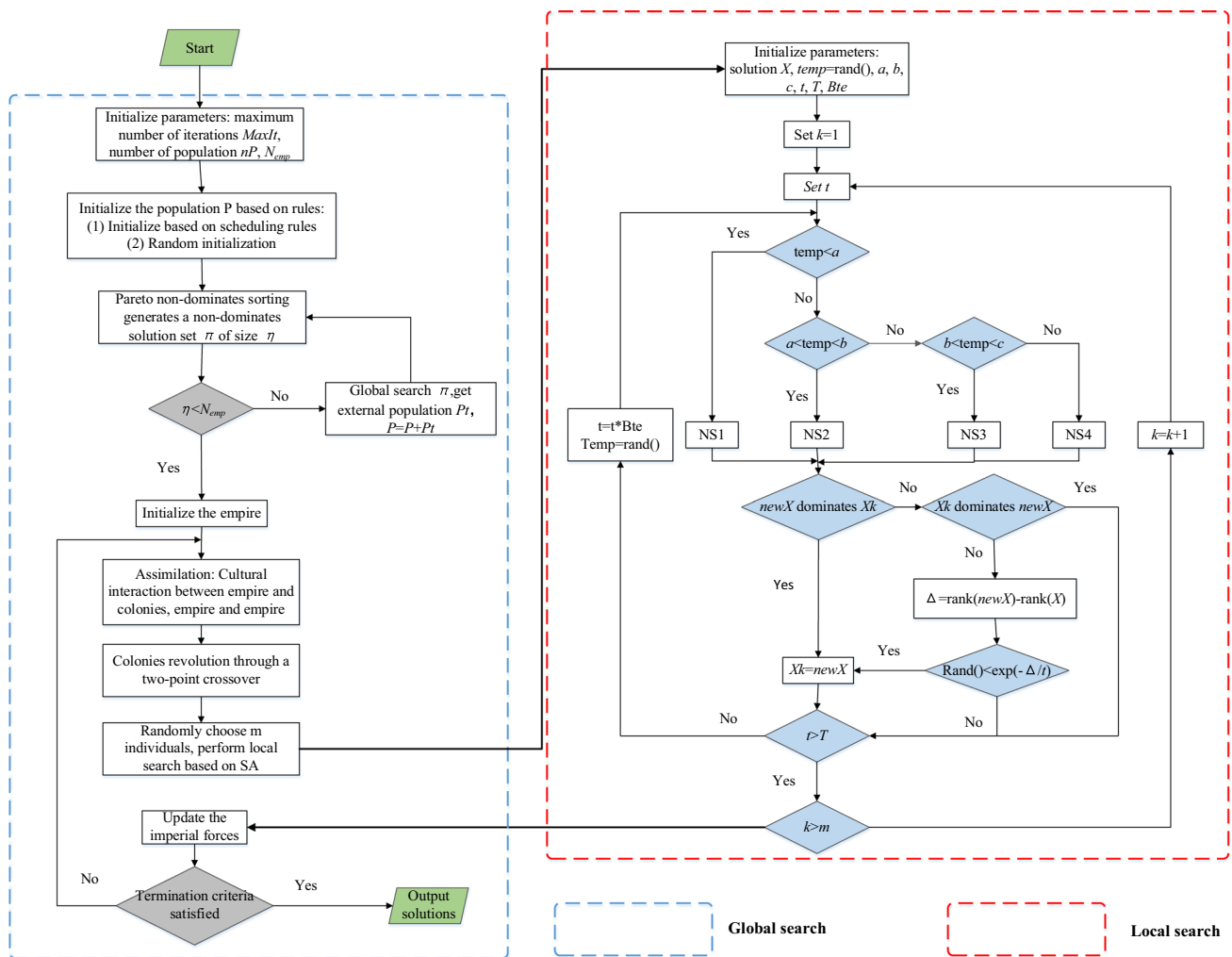


Fig. 11 Flow chart of HDMICA

Table 4 Processing resources for each process

	Operations								
	Modeling	Melting	Pouring cooling	Shakeout	Shot blasting	Soldering	Polishing	Finishing	Paint
People	–	–	–	–	–	$P_1 \sim P_3$	$P_4 \sim P_6$	$P_7 \sim P_9$	–
Machine	$M_1 \sim M_3$	$M_4 \sim M_7$	$M_8 \sim M_9$	$M_{10} \sim M_{12}$	$M_{13} \sim M_{15}$	$M_{16} \sim M_{18}$	$M_{19} \sim M_{21}$	$M_{22} \sim M_{24}$	$M_{25} \sim M_{26}$
Power I (kW)	2, 2.1, 1.9	3, 3.5, 3.5, 3.3	1.3, 1.5	1.3, 1.4, 1.5	2, 1.8, 1.2	2, 2.2, 1.9	1, 1.2, 1	2, 2.3, 1.8	2.1, 2
Power O (kW)	–	–	–	2, 2.5, 3	1, 0.9, 0.8	–	0.9, 0.8, 0.9	–	–
Noise (dB)	84, 83, 85	83, 82, 84, 80	–	88, 85, 80	82, 80, 79	83, 80, 82	80, 79, 81	70, 78, 75	75, 70

$$MID = \frac{1}{n} \cdot \sum_{i=1}^n c_i \tag{38}$$

where $c_i = \sqrt{f_{1i}^2 + f_{2i}^2 + f_{3i}^2}$ is the square root of the sum of the squared target values, and n is the number of non-dominated solutions.

(2) Spread of non-dominated solution (SNS). As a measure of the diversity of non-dominated solutions, the larger the SNS, the richer the diversity of the non-dominated solutions, and the better the quality of the solution can be expressed as:

Table 5 Practical production data of jobs

Job	Process										
	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}
J ₁	1	<i>L</i>	–	–	–	–	–	0.75	0.8	0.9	–
	350	<i>Pst</i>	–	–	–	–	–	[0.7, 0.9, 0.8]	[0.7, 0.7, 0.8]	[0.8, 0.9, 0.7]	–
	350	<i>Pend</i>	–	–	–	–	–	[0.7, 0.8, 0.6]	[0.8, 0.8, 0.6]	[0.7, 0.8, 0.7]	–
	150	<i>T</i>	[5, 6, 5]	[8, 10, 11, 9]	[13, 15]	[5, 5, 4]	[6, 8, 7]	[6, 6, 5]	[6, 6, 7]	[5, 5, 6]	[3, 5]
J ₂	1	<i>L</i>	–	–	–	–	–	0.75	0.8	0.9	–
	550	<i>Pst</i>	–	–	–	–	–	[0.7, 0.9, 0.7]	[0.8, 0.7, 0.7]	[0.9, 0.8, 0.9]	–
	550	<i>Pend</i>	–	–	–	–	–	[0.6, 0.8, 0.7]	[0.7, 0.6, 0.7]	[0.8, 0.6, 0.8]	–
	200	<i>T</i>	[5, 6, 4]	[10, 9, 12, 8]	[13, 14]	[4, 5, 4]	[9, 8, 7]	[7, 6, 5]	[9, 6, 5]	[7, 6, 4]	[4, 5]
J ₃	1	<i>L</i>	–	–	–	–	–	0.75	0.8	0.9	–
	300	<i>Pst</i>	–	–	–	–	–	[0.8, 0.9, 0.8]	[0.7, 0.7, 0.8]	[0.9, 0.9, 0.7]	–
	300	<i>Pend</i>	–	–	–	–	–	[0.6, 0.8, 0.7]	[0.6, 0.7, 0.8]	[0.8, 0.7, 0.6]	–
	100	<i>T</i>	[6, 6, 5]	[12, 9, 8, 11]	[14, 15]	[3, 5, 3]	[6, 7, 8]	[6, 4, 5]	[8, 7, 6]	[5, 7, 6]	[5, 5]
J ₄	1	<i>L</i>	–	–	–	–	–	0.75	0.8	0.9	–
	600	<i>Pst</i>	–	–	–	–	–	[0.9, 0.8, 0.8]	[0.7, 0.8, 0.9]	[0.7, 0.9, 0.7]	–
	600	<i>Pend</i>	–	–	–	–	–	[0.8, 0.7, 0.6]	[0.6, 0.8, 0.7]	[0.6, 0.8, 0.7]	–
	250	<i>T</i>	[4, 5, 6]	[8, 11, 9, 10]	[13, 13]	[2, 3, 4]	[6, 6, 7]	[4, 7, 5]	[7, 6, 7]	[6, 5, 7]	[3, 4]
J ₅	2	<i>L</i>	–	–	–	–	–	0.75	0.8	0.9	–
	350	<i>Pst</i>	–	–	–	–	–	[0.9, 0.7, 0.7]	[0.8, 0.9, 0.8]	[0.7, 0.8, 0.9]	–
	350	<i>Pend</i>	–	–	–	–	–	[0.8, 0.6, 0.6]	[0.8, 0.7, 0.7]	[0.6, 0.7, 0.8]	–
	150	<i>T</i>	[6, 5, 5]	[9, 9, 13, 10]	[14, 15]	[5, 4, 3]	[7, 6, 5]	[5, 4, 5]	[5, 6, 7]	[9, 6, 7]	[4, 3]
J ₆	2	<i>L</i>	–	–	–	–	–	0.75	0.8	0.9	–
	550	<i>Pst</i>	–	–	–	–	–	[0.9, 0.7, 0.7]	[0.7, 0.8, 0.8]	[0.9, 0.8, 0.7]	–
	550	<i>Pend</i>	–	–	–	–	–	[0.8, 0.6, 0.7]	[0.6, 0.7, 0.8]	[0.8, 0.7, 0.6]	–
	200	<i>T</i>	[6, 4, 5]	[8, 9, 10, 11]	[15, 13]	[3, 4, 3]	[6, 6, 5]	[6, 4, 5]	[8, 6, 7]	[5, 6, 7]	[5, 3]
J ₇	2	<i>L</i>	–	–	–	–	–	0.75	0.8	0.9	–
	300	<i>Pst</i>	–	–	–	–	–	[0.7, 0.7, 0.8]	[0.9, 0.8, 0.9]	[0.8, 0.9, 0.7]	–
	300	<i>Pend</i>	–	–	–	–	–	[0.6, 0.7, 0.7]	[0.8, 0.6, 0.8]	[0.7, 0.8, 0.6]	–
	100	<i>T</i>	[4, 5, 4]	[13, 11, 9, 9]	[14, 15]	[5, 4, 5]	[6, 7, 5]	[5, 6, 7]	[5, 7, 6]	[8, 7, 6]	[4, 5]
J ₈	2	<i>L</i>	–	–	–	–	–	0.75	0.8	0.9	–
	600	<i>Pst</i>	–	–	–	–	–	[0.9, 0.8, 0.7]	[0.8, 0.7, 0.8]	[0.9, 0.9, 0.7]	–
	600	<i>Pend</i>	–	–	–	–	–	[0.8, 0.6, 0.7]	[0.7, 0.6, 0.7]	[0.8, 0.8, 0.6]	–
	250	<i>T</i>	[6, 5, 5]	[12, 9, 9, 10]	[14, 13]	[2, 4, 3]	[5, 6, 5]	[4, 4, 5]	[6, 6, 7]	[5, 6, 7]	[4, 3]
J ₉	3	<i>L</i>	–	–	–	–	–	0.75	0.8	0.9	–
	350	<i>Pst</i>	–	–	–	–	–	[0.8, 0.7, 0.8]	[0.9, 0.8, 0.7]	[0.9, 0.7, 0.9]	–
	350	<i>Pend</i>	–	–	–	–	–	[0.7, 0.6, 0.7]	[0.8, 0.6, 0.7]	[0.8, 0.6, 0.8]	–
	150	<i>T</i>	[5, 4, 6]	[9, 11, 8, 10]	[12, 14]	[4, 3, 2]	[8, 9, 7]	[5, 4, 7]	[7, 7, 6]	[8, 9, 7]	[]
J ₁₀	3	<i>L</i>	–	–	–	–	–	0.75	0.8	0.9	–
	550	<i>Pst</i>	–	–	–	–	–	[0.7, 0.7, 0.9]	[0.8, 0.9, 0.7]	[0.9, 0.8, 0.9]	–
	550	<i>Pend</i>	–	–	–	–	–	[0.6, 0.7, 0.8]	[0.7, 0.8, 0.6]	[0.8, 0.6, 0.7]	–
	200	<i>T</i>	[6, 4, 5]	[9, 12, 13, 9]	[13, 12]	[4, 5, 2]	[6, 7, 5]	[4, 4, 7]	[5, 6, 7]	[5, 7, 6]	[]
J ₁₁	3	<i>L</i>	–	–	–	–	–	0.75	0.8	0.9	–
	300	<i>Pst</i>	–	–	–	–	–	[0.8, 0.9, 0.7]	[0.9, 0.8, 0.9]	[0.7, 0.8, 0.9]	–
	300	<i>Pend</i>	–	–	–	–	–	[0.7, 0.8, 0.7]	[0.8, 0.7, 0.6]	[0.6, 0.7, 0.8]	–
	100	<i>T</i>	[5, 4, 4]	[9, 12, 10, 9]	[15, 14]	[2, 3, 2]	[8, 9, 7]	[5, 4, 6]	[5, 7, 6]	[8, 9, 6]	[]
J ₁₂	3	<i>L</i>	–	–	–	–	–	0.75	0.8	0.9	–

Table 5 continued

Job	Process										
	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9		
J ₁₃	600	<i>Pst</i>	–	–	–	–	–	[0.9, 0.8, 0.8]	[0.7, 0.8, 0.9]	[0.7, 0.9, 0.7]	–
	600	<i>Pend</i>	–	–	–	–	–	[0.8, 0.7, 0.6]	[0.6, 0.8, 0.7]	[0.6, 0.8, 0.7]	–
	250	<i>T</i>	[6, 4, 5]	[8, 11, 9, 13]	[15, 13]	[4, 3, 2]	[6, 7, 8]	[5, 6, 7]	[9, 7, 6]	[9, 7, 5]	[]
	4	<i>L</i>	–	–	–	–	–	0.75	0.8	0.9	–
	350	<i>Pst</i>	–	–	–	–	–	[0.7, 0.9, 0.7]	[0.8, 0.7, 0.7]	[0.9, 0.8, 0.9]	–
	350	<i>Pend</i>	–	–	–	–	–	[0.6, 0.8, 0.7]	[0.7, 0.6, 0.7]	[0.8, 0.6, 0.8]	–
J ₁₄	150	<i>T</i>	[5, 4, 4]	[9, 10, 12, 8]	[13, 14]	[4, 4, 3]	[9, 8, 7]	[4, 4, 5]	[8, 6, 9]	[6, 8, 9]	[]
	4	<i>L</i>	–	–	–	–	–	0.75	0.8	0.9	–
	550	<i>Pst</i>	–	–	–	–	–	[0.7, 0.7, 0.8]	[0.9, 0.8, 0.9]	[0.8, 0.9, 0.7]	–
	550	<i>Pend</i>	–	–	–	–	–	[0.6, 0.7, 0.7]	[0.8, 0.6, 0.8]	[0.7, 0.8, 0.6]	–
J ₁₅	200	<i>T</i>	[6, 4, 6]	[8, 11, 8, 13]	[13, 12]	[4, 5, 3]	[6, 8, 7]	[4, 6, 6]	[7, 8, 9]	[7, 7, 8]	[]
	4	<i>L</i>	–	–	–	–	–	0.75	0.8	0.9	–
	300	<i>Pst</i>	–	–	–	–	–	[0.8, 0.9, 0.8]	[0.7, 0.7, 0.8]	[0.9, 0.9, 0.7]	–
	300	<i>Pend</i>	–	–	–	–	–	[0.6, 0.8, 0.7]	[0.6, 0.7, 0.8]	[0.8, 0.7, 0.6]	–
J ₁₆	100	<i>T</i>	[5, 5, 6]	[11, 9, 8, 10]	[12, 14]	[2, 4, 3]	[7, 8, 9]	[6, 6, 5]	[8, 5, 9]	[6, 8, 9]	[]
	4	<i>L</i>	–	–	–	–	–	0.75	0.8	0.9	–
	600	<i>Pst</i>	–	–	–	–	–	[0.9, 0.7, 0.7]	[0.7, 0.8, 0.8]	[0.9, 0.8, 0.7]	–
	600	<i>Pend</i>	–	–	–	–	–	[0.8, 0.6, 0.7]	[0.6, 0.7, 0.8]	[0.8, 0.7, 0.6]	–
	250	<i>T</i>	[4, 6, 6]	[9, 12, 9, 10]	[13, 14]	[4, 4, 3]	[7, 8, 7]	[4, 6, 5]	[8, 9, 7]	[8, 9, 7]	[]

Table 6 Parameters setting of algorithms

	HDMICA	NSGAI	DPSO	MODVOA	PESA2
Population size	100	100	100	100	100
Maximum iteration	150	150	150	150	150
Archive size	–	100	100	100	100
Crossover rate	0.8	0.8	–	0.8	–
Mutation rate	0.3	0.3	–	0.3	–
RPX parameters: pf_{max}	0.9	0.9	0.9	0.9	0.9
RPX parameters: pf_{min}	0.2	0.2	0.2	0.2	0.2
Assimilatory coefficient: β	1.5	–	–	–	–
Revolution rate	0.05	–	–	–	–
Success rate of revolution: μ	0.1	–	–	–	–
Average colonial cost factor: ζ	0.2	–	–	–	–
Neighborhood selection proportion: $p1$	0.3	–	–	–	–
Neighborhood selection proportion: $p2$	0.5	–	–	–	–
Neighborhood selection proportion: $p3$	0.7	–	–	–	–
Initial temperature: T_0	3	–	–	–	–
End temperature: T_{end}	0.1	–	–	–	–
Temperature drop rate: K_b	0.9	–	–	–	–
Markov length: K_m	10	–	–	–	–

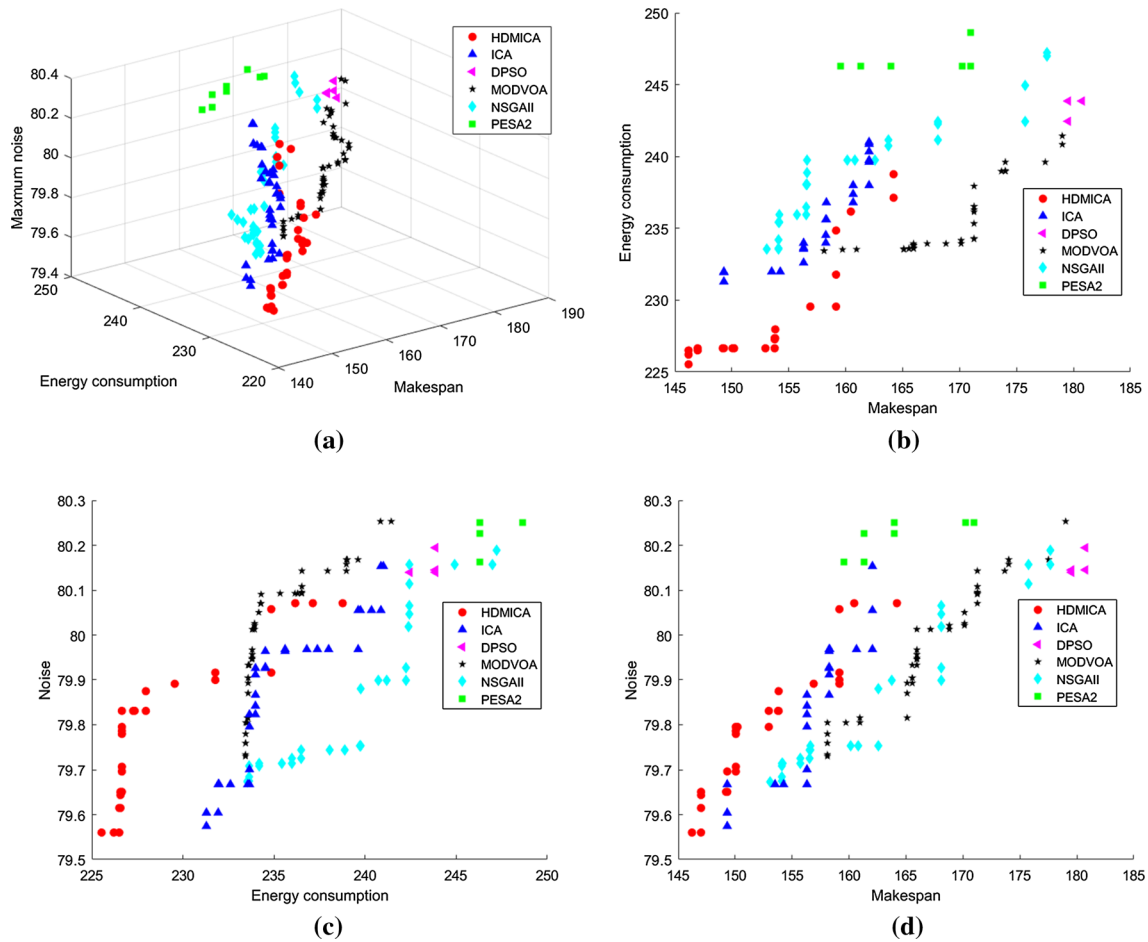


Fig. 12 Pareto fronts obtained by different algorithms under different angles, **a** Value by different algorithms with three criteria, **b** value with makespan and energy consumption, **c** value with noise and energy consumption, **d** value with makspan and noise

Table 7 Comparison of non-dominated solutions of various algorithms

	Δ_{\min}	Δ_{\max}	Δ_{avg}	Δ_{std}	Δ_{\min}	Δ_{\max}	Δ_{avg}	Δ_{std}	Δ_{\min}	Δ_{\max}	Δ_{avg}	Δ_{std}
HDMICA	146.20	160.10	150.32	1.77	225.54	232.65	227.97	0.7600	79.26	79.87	79.52	0.0009
ICA	152.40	176.05	156.96	0.0058	228.95	235.80	233.40	0.0003	79.33	79.94	79.47	0.0002
DPSO	174.40	223.35	200.00	1.7376	243.09	253.41	248.07	0.4755	80.23	80.80	80.47	0.0017
MODVOA	156.30	248.54	197.90	0.2522	236.03	254.40	243.95	0.1201	79.82	80.66	80.23	0.0007
NSGAI	148.05	171.1	156.95	0.1940	229.10	238.50	233.78	0.5699	79.39	79.91	79.56	0.0004
PESA2	156.20	240.07	186.32	11.0626	243.92	264.75	253.77	0.2331	80.13	80.68	80.45	0.0001

$$SNS = \sqrt{\frac{1}{n-1} \cdot \sum_{i=1}^n (MID - c_i)^2} \quad (39)$$

(3) The rate of achievement to three objectives simultaneously (RAS).

$$RAS = \frac{1}{n} \cdot \sum_{i=1}^n \left(\frac{f_{1i} - f_i}{f_i} + \frac{f_{2i} - f_i}{f_i} + \frac{f_{3i} - f_i}{f_i} \right) \quad (40)$$

where $f_i = \min\{f_{1i}, f_{2i}, f_{3i}\}$. The smaller the RAS solution, the better the quality.

Results analysis

In order to show the performance of the proposed algorithm, compared with the well-known algorithms: NSGAI, DPSO, MODVOA, and PESA2. To ensure the validity of the comparison of the results of all algorithms, the operating environment of all experiments is 2.7 GHz CPU, 8G memory,

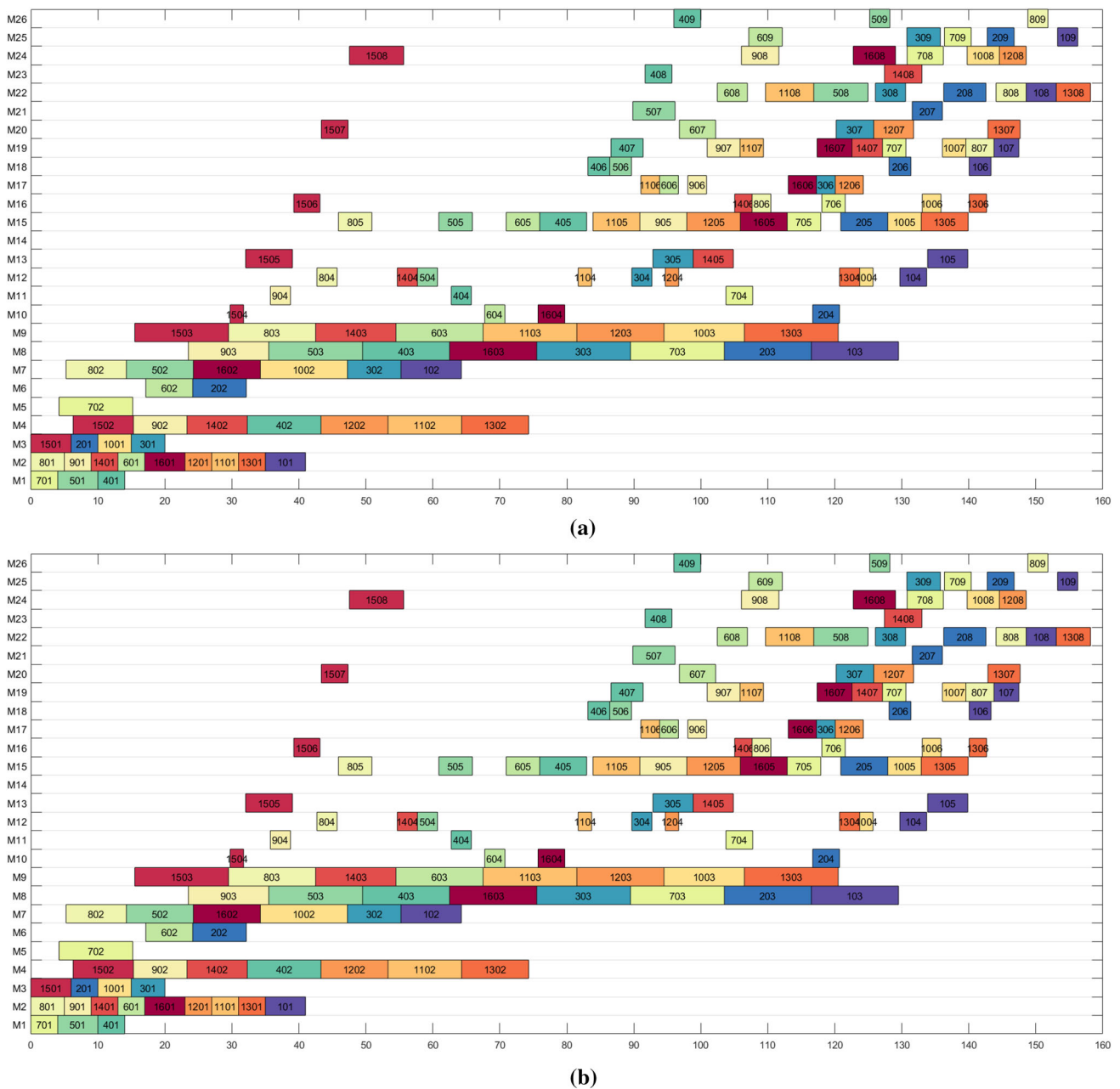


Fig. 13 Gantt chart of the first non-dominated solution obtained by HDMICA

64-bit win7 system computer, and the programming environment is Matlab 2016. Moreover, all algorithms choose the same initialization method, that is, the rule-based initialization method and random initialization method with a 1:1 ratio. In addition, the crossover and mutation methods in all algorithms are the same. The process segment selects IPOX crossover, the machine segment adopts RPX crossover, and the worker selection segment using two-point crossover. The specific parameter settings are shown in Table 6.

To form a population of size 100. The number of independent runs is 150. The comparison of the Pareto first

frontal solutions of each algorithm is shown in Fig. 12, where ICA represents the original algorithm. Above all, from the Pareto non-dominated solutions three-dimensional distribution maps Fig. 12a of different algorithms, it can be clearly seen that the solution set generated by HDMICA is distributed in the corner with the smallest three target values, followed by the distribution of the original ICA solution. While, the target values of other algorithms deviate more from the optimal solution. Then, from the two-dimensional distribution diagrams of Fig. 12b, c, it can be seen that the original ICA is superior to other algorithms, but there is

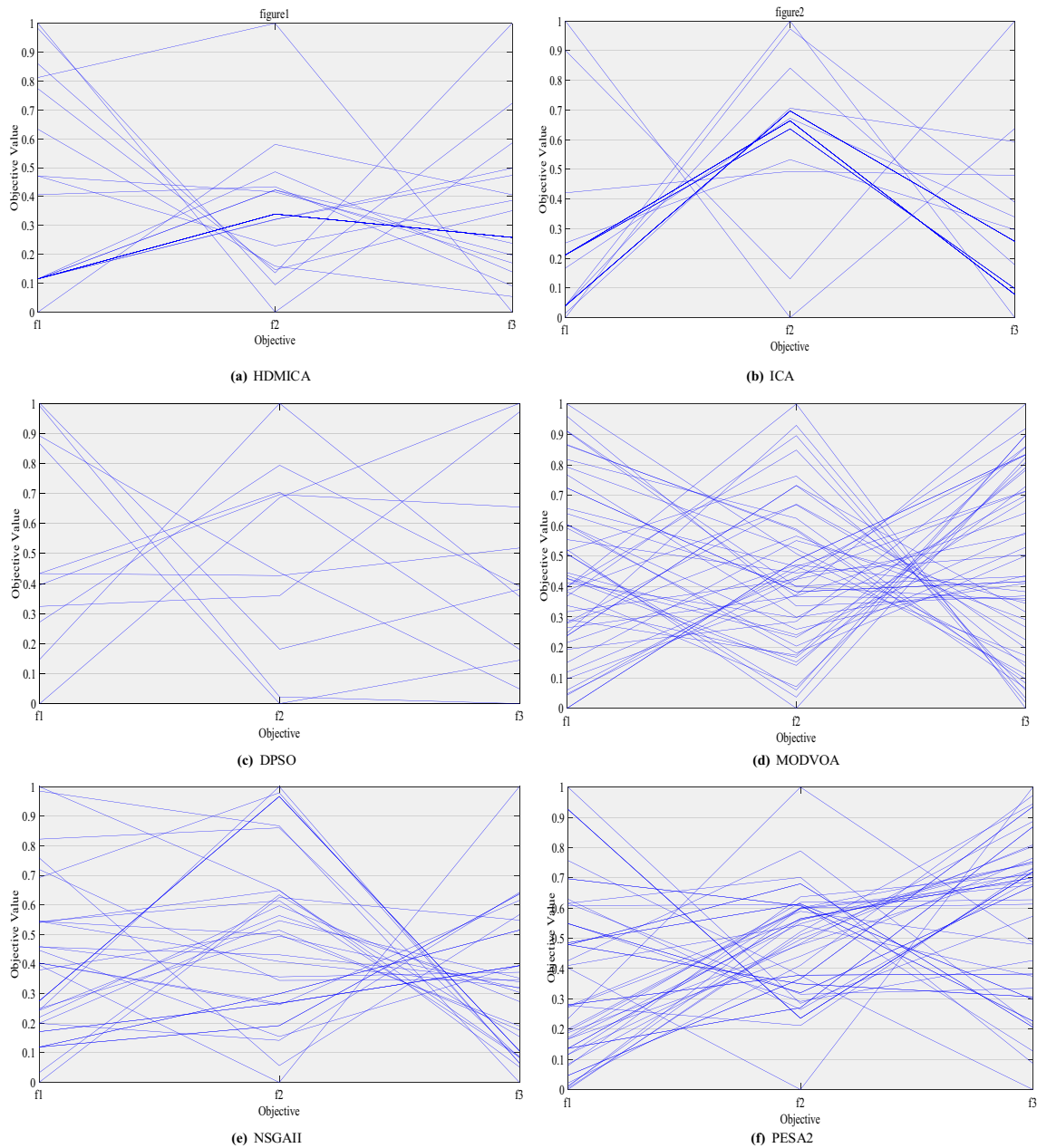


Fig. 14 The parallel coordinates plot for different algorithms

not much difference with the results produced by NSGAI. Finally, Fig. 12d clearly shows the optimization effect of ICA. However, it can be seen that the improved HDMICA obviously optimizes the results produced by the original ICA. In short, HDMICA is superior to the other five algorithms.

Figure 13 is a Gantt chart of a solution with the maximum crowding distance selected from the non-dominated solution set of HAMICA. The abscissa is the processing time (h) and the ordinate is the resource number. Each color of the rectangle represents a job, and the number represents the process of the job. For example, 1503 can be described as the third pro-

cess of job J_{15} . Figure 13a is a Gantt chart of a job processed on a processing machine. Actually, Processes 6 to 8 require human participation. Figure 13b is a Gantt chart of the job processed by the worker. The makespan is 158.1500(h), the energy consumption is 226.5029(kgce), and the noise generated is 80.0702(dB).

Table 7 is the target value of the non-dominated solution set obtained by each algorithm, where Δ_{\min} , Δ_{\max} , Δ_{avg} , and Δ_{std} represent the maximum, minimum, average, and variance of each target value, and the data in bold in the table indicate the current optimal value. It can be seen that the

Table 8 The value of all metrics obtained by different algorithms

	HDMICA	ICA	DPSO	MODVOA	NSGAI	PESA2
MID	284.4928	296.1851	328.4773	332.3760	295.9683	322.5845
SNS	2.2184	2.2748	2.8111	2.1778	2.2978	2.8083
RAS	2.7454	2.9776	3.5522	3.6451	2.9657	3.4201

HDMICA target value is better than other algorithms, but the variance of the target value does not show a clear advantage, indicating that the diversity of non-dominated solutions is general.

So as to compare the quality of the non-dominated solution sets of the six algorithms intuitively, the parallel coordinates plot (Li et al. 2017) is adopted. Since the range of values of the three objective functions differs greatly, in this paper, the target value is normalized to make the results more intuitive. Normalized value (NV) can be formulated as follows:

$$NV = \frac{S_c - S_b}{S_w - S_b} \quad (41)$$

where S_c is the current value of the non-dominated solution set, S_b is the optimal value, and S_w is the worst value. It should be noted that the smaller value for NV is preferred.

Figure 14 is a parallel coordinate diagram of the six algorithms. The horizontal coordinate of the graph represents the three goals of the problem, and the vertical coordinate represents the normal standardized target values of the three goals. It can be seen that HDMICA is superior to other algorithms to some extent. Furthermore, the fluctuation range of the target value of HDMICA is more concentrated, which indicates that the quality and stability of the non-dominated solution obtained by HDMGWO precede the other five algorithms.

From Table 8, It can be seen that HDMICA is significantly better than other algorithms in terms of MID and RAS. Pareto non-dominated solutions are closer to the global optimal solution, but the SNS values of all algorithms are not much different. HDMICA has not significantly improved the non-dominated solutions diversity.

Conclusions and prospects

This article takes an actual foundry as an example. Considering the constraints of man-machine and dual resources, a mathematical model (minimum processing time, energy consumption, and noise) of a three-dimensional target is established under the constraints of the learning effect of workers and the transport time of the job. A hybrid discrete multi-objective imperial competition algorithm is proposed to solve the model. In order to optimize the quality of the solution and preserve the diversity of the population, the population initialization method that combines the shortest processing time rule and random initialization; When select-

ing a colonial country, if the number of non-dominating solutions is less than the number of target empires, a neighborhood search is performed on the non-dominating solutions until the number of non-dominating solutions is greater than the number of empires. Finally, the results obtained by HDMICA are compared with the original ICA, DPSO, MODVOA, NSGAI and PESA2 to verify the effectiveness of the proposed algorithm.

For green sustainable manufacturing, this article mainly researches energy consumption and noise. In fact, green includes many aspects. As casting is a high energy consumption and high pollution industry, a large amount of waste gas and residue will be generated during the production process, which provides a direction for the next step of research. This paper does not consider the processing time limit of workers, so a balance based on wages can be found between the overtime of workers and increasing the number of workers; distributed scheduling and reverse scheduling are also mainstream scheduling methods, and they can be used as the main research direction in the future. In addition, distributed scheduling and inverse scheduling are also the main scheduling methods, which can be taken as the main research direction in the future.

References

- Atashpaz-Gargari, E., & Lucas, C. (2007). Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. In *2007 IEEE congress on evolutionary computation* (pp. 4661–4667). IEEE.
- Azzouz, A., Ennigrou, M., & Said, L. B. (2017). A self-adaptive evolutionary algorithm for solving flexible job-shop problem with sequence dependent setup time and learning effects. In *2017 IEEE congress on evolutionary computation (CEC)* (pp. 1827–1834). IEEE.
- Behnamian, J., & Ghomi, S. M. T. F. (2011). Hybrid flowshop scheduling with machine and resource-dependent processing times. *Applied Mathematical Modelling*, *35*(3), 1107–1123.
- Bewoor, L. A., Prakash, V. C., & Sapkal, S. U. (2018). Production scheduling optimization in foundry using hybrid particle swarm optimization algorithm. *Procedia Manufacturing*, *22*, 57–64.
- Biskup, D. (1999). Single-machine scheduling with learning considerations. *European Journal of Operational Research*, *115*(1), 173–178.
- Boutsinas, B. (2013). Machine-part cell formation using biclustering. *European Journal of Operational Research*, *230*(3), 563–572.
- Cao, S., & Lei, D. (2019). A novel constrained multi-objective imperialist competitive algorithm. *Information and Control*, *48*(04), 437–451. (in Chinese).

- Coca, G., Castrillón, O. D., Ruiz, S., et al. (2019). Sustainable evaluation of environmental and occupational risks scheduling flexible job shop manufacturing systems. *Journal of Cleaner Production*, 209, 146–168.
- Deb, K., Agrawal, S., Pratap, A., et al. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *International conference on parallel problem solving from nature* (pp. 849–858). Berlin: Springer.
- Gen, M., & Lin, L. (2014). Multiobjective evolutionary algorithm for manufacturing scheduling problems: State-of-the-art survey. *Journal of Intelligent Manufacturing*, 25(5), 849–866.
- Gong, G., Deng, Q., Gong, X., et al. (2018). A new double flexible job-shop scheduling problem integrating processing time, green production, and human factor indicators. *Journal of Cleaner Production*, 174, 560–576.
- Hajiaghahi-Keshteli, M., & Fathollahi-Fard, A. M. (2018). A set of efficient heuristics and metaheuristics to solve a two-stage stochastic bi-level decision-making model for the distribution network problem. *Computers & Industrial Engineering*, 123, 378–395.
- Hans, E., & van de Velde, S. (2011). The lot sizing and scheduling of sand casting operations. *International Journal of Production Research*, 49(9), 2481–2499.
- He, Z., & Pan, H. (2015). Germany “Industry 4.0” and “Made in China 2025”. *Journal of Changsha University of Science and Technology*, 30(3), 103–110. (in Chinese).
- Hosseini, S., & Al, Khaled A. (2019). A hybrid ensemble and AHP approach for resilient supplier selection. *Journal of Intelligent Manufacturing*, 30(1), 207–228.
- Ji, Zhou. (2015). Intelligent manufacturing—main direction of “Made in China 2025”. *China Mechanical Engineering*, 26(17), 2273–2284. (in Chinese).
- Karimi, S., Ardalan, Z., Naderi, B., et al. (2017). Scheduling flexible job-shops with transportation times: Mathematical models and a hybrid imperialist competitive algorithm. *Applied Mathematical Modelling*, 41, 667–682.
- Lei, D., Pan, Z., & Zhang, Q. (2018). Researches on multi-objective low carbon parallel machines scheduling. *Journal of Huazhong University of Science and Technology (Science and Technology)*, 46(08), 104–109. (in Chinese).
- Li, M., Zhen, L., & Yao, X. (2017). How to read many-objective solution sets in parallel coordinates [educational forum]. *IEEE Computational Intelligence Magazine*, 12(4), 88–100.
- Liu, Q., Xiong, S., & Zhan, M. (2019). Many-objectives Scheduling Optimization based on PCA-NSGAI with an improved elite strategy. In *Computer integrated manufacturing system* (pp. 1–13). Retrieved July 12, 2019, from <http://kns.cnki.net/kcms/detail/11.5946.tp.20190419.1016.002.html>.
- Lu, C., Gao, L., Li, X., et al. (2018). A multi-objective approach to welding shop scheduling for makespan, noise pollution and energy consumption. *Journal of Cleaner Production*, 196, 773–787.
- Nowicki, E., & Smutnicki, C. (1996). A fast taboo search algorithm for the job shop problem. *Management Science*, 42(6), 797–813.
- Olayinka, O. S., & Abdullahi, S. A. (2009). An overview of industrial employees’ exposure to noise in sundry processing and manufacturing industries in Ilorin metropolis, Nigeria. *Industrial Health*, 47(2), 123–133.
- Shivasankaran, N., Kumar, P. S., & Raja, K. V. (2015). Hybrid sorting immune simulated annealing algorithm for flexible job shop scheduling. *International Journal of Computational Intelligence Systems*, 8(3), 455–466.
- Tandon, N. (2000). Noise-reducing designs of machines and structures. *Sadhana*, 25(3), 331–339.
- Tang, H., Chen, R., Li, Y., et al. (2019). Flexible job-shop scheduling with tolerated time interval and limited starting time interval based on hybrid discrete PSO-SA: An application from a casting workshop. *Applied Soft Computing*, 78, 176–194.
- Wan, J., Yi, M., Li, D., et al. (2016). Mobile services for customization manufacturing systems: An example of industry 4.0. *IEEE Access*, 4, 8977–8986.
- Wang, S., Wan, J., Zhang, D., et al. (2016). Towards smart factory for industry 4.0. *Computer Networks*, 101(101), 158–168.
- Wang, B., Wang, X., Lan, F., et al. (2018). A hybrid local-search algorithm for robust job-shop scheduling under scenarios. *Applied Soft Computing*, 62, 259–271.
- Wirojanagud, P., Gel, E. S., Fowler, J. W., et al. (2007). Modelling inherent worker differences for workforce planning. *International Journal of Production Research*, 45(3), 525–553.
- Wu, M., Song, Z., & Moon, Y. B. (2019). Detecting cyber-physical attacks in CyberManufacturing systems with machine learning methods. *Journal of Intelligent Manufacturing*, 30(3), 1111–1123.
- Wu, X., & Sun, Y. (2018). A green scheduling algorithm for flexible job shop with energy-saving measures. *Journal of Cleaner Production*, 172, 3249–3264.
- Xinchao, Z. (2011). Simulated annealing algorithm with adaptive neighborhood. *Applied Soft Computing*, 11(2), 1827–1836.
- Xu, Y., Ye, C., & Cao, L. (2019). Research on flexible Job-Shop scheduling problem with AGVs constraints and behavioral effects. *Application Research of Computers*, 36(10), 3033–3038. (in Chinese).
- Yin, L., Li, X., Gao, L., et al. (2017). A novel mathematical model and multi-objective method for the low-carbon flexible job shop scheduling problem. *Sustainable Computing: Informatics and Systems*, 13, 15–30.
- Yu, J. (2016). Research of learning curve based similarity and single machine scheduling. *Chongqing University (in Chinese)*
- Zhang, J. (2014). Research of flexible job shop scheduling problem based on hybrid discrete particle swarm optimization algorithm. *Zhejiang University of Technology*. (in Chinese)
- Zhang, J., Ding, G., Zou, Y., et al. (2019). Review of job shop scheduling research and its new perspectives under Industry 4.0. *Journal of Intelligent Manufacturing*, 30(4), 1809–1830.
- Zhao, C., Fang, J., Cheng, T. C. E., et al. (2017). A note on the time complexity of machine scheduling with DeJong’s learning effect. *Computers & Industrial Engineering*, 112, 447–449.

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.