



# An estimation of distribution algorithm with multiple intensification strategies for two-stage hybrid flow-shop scheduling problem with sequence-dependent setup time

Huan Liu<sup>1,2</sup> · Fuqing Zhao<sup>1</sup> · Ling Wang<sup>3</sup> · Jie Cao<sup>1</sup> · Jianxin Tang<sup>1</sup> · Jonrinaldi<sup>4</sup>

Accepted: 15 May 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

## Abstract

The estimation of distribution algorithm (EDA) has recently emerged as a promising alternative to the traditional evolutionary algorithms for solving combinatorial optimization problems. In this paper, an estimation of distribution algorithm with multiple intensification strategies (EDA-MIS) is proposed to solve a typical kind of hybrid flow-shop scheduling problem. The two-stage heterogeneous hybrid flow-shop scheduling problem is investigated. The sequence-dependent setup time at the first stage is also considered. In the proposed EDA-MIS, the initial population is constructed through the heuristic method and random strategy. An order matrix is established to estimate the probabilistic model of promising solutions. Then the solutions of the algorithm are evolved through the processes of selection, recombination, sampling, and local search. The obtained results indicate that the EDA-MIS provides good solutions in the aspects of solution quality and computational efficiency.

**Keywords** Hybrid flow-shop scheduling problem · Sequence-dependent setup time · Estimation of distribution algorithm · Local search

## 1 Introduction

Scheduling is one of the most important research issues in steelmaking, textile, logistics, electronic components manufacturing, and other smart manufacturing fields [1–3]. Effective scheduling schemes can significantly reduce the production costs and the production cycle and improve the economic efficiency of enterprises [4]. The hybrid flow-shop scheduling problem (HFSP) is a kind of model that extends from the traditional flow-shop scheduling problem (FSP). In the HFSP, the processing of jobs is divided into several stages, where each stage has several machines to be selected [5]. This

model has some key problems that need to be solved, such as arranging the process sequence of jobs and selecting the right machine at different stages. The objective of scheduling is to reduce the corresponding production indicators, such as production cycle and total flow time [6].

Aiming at the actual workshop conditions of the production process, various studies have introduced several constraints into the basic HFSP [7, 8]. Considering the constraints of stage skipping and adjustable processing time, Long presented a realistic HFSP model that was extracted from the steelmaking-continuous casting production process [9]. Cai and Lei studied a distributed hybrid flow-shop scheduling

---

✉ Fuqing Zhao  
FZhao2000@hotmail.com

Huan Liu  
liuh1204@hotmail.com

Ling Wang  
wangling@tsinghua.edu.cn

Jie Cao  
caoj2976016@qq.com

Jianxin Tang  
582971672@qq.com

Jonrinaldi  
jonrinaldi@eng.unand.ac.id

<sup>1</sup> School of Computer and Communication Technology, Lanzhou University of Technology, Lanzhou, China

<sup>2</sup> College of Information Science and Technology, Gansu Agricultural University, Lanzhou 730070, China

<sup>3</sup> Department of Automation, Tsinghua University, Beijing 10084, China

<sup>4</sup> Department of Industrial Engineering, University Andalas, Padang 25163, Indonesia

problem (DHFSP) with fuzzy processing time, and proposed a cooperated shuffled frog-leaping algorithm (CSFLA) to simultaneously optimize the fuzzy makespan, total agreement index, and fuzzy total energy consumption [10]. Li et al. studied the distributed heterogeneous hybrid flow-shop scheduling problem (DHHFSP) with unrelated parallel machines (UPM) and the sequence-dependent setup time (SDST) [11]. Meng et al. provided some mixed integer-linear programming (MILP) models of HFSP-UPM [12]. The energy consumption of HFSP-UPM has been analyzed in [13]. Zhong and Shi addressed the two-stage no-wait hybrid flow-shop scheduling problem with inter-stage flexibility in [14]. In [15], the HFSP with blocking and due window constraints has been tackled. Chamnanlor et al. studied a reentrant HFSP with time window constraints [16]. Several studies developed algorithms by adding the constraint of sequence-dependent setup time into the HFSP to solve the problem [17–20]. With the development of technology, distributed manufacturing has become the main mode of production in the manufacturing industry. In some studies, the HFSP has been extended to distributed manufacturing environments [7, 21].

The above-mentioned studies have proposed constraints on the relevant production models based on the actual production conditions and business needs. The constraints of the production models vary significantly in different types of machining plants [22–24]. Some studies proposed the relevant model constraints based on the production plants of glass, textile, and food processing to fit the actual production conditions. However, such production workshops differ from the rare earth production workshops in terms of production structure and production constraints. In this paper, an HFSP model with sequence-dependent setup time is constructed from the actual production shop of rare earth.

The two-stage HFSP with makespan minimization is an NP-hard problem. Recently, in the field of combinatorial optimization research, the HFSP has been regarded as an important research direction and an innovative theoretical method to optimize the problem. The main methods for HFSP can be divided into several categories: exact solution method [25], heuristic construction method [26], bio-inspired intelligent optimization method [27], multi-objective optimization algorithm [28–30] and hybrid method [21]. Zhang et al. introduced the constraint of lot streaming into the HFSP and developed a collaborative variable neighborhood descent (CVND) algorithm [31]. H. Öztöp et al. presented four variants of iterated greedy algorithm and a variable block insertion heuristic for the HFSP with total flowtime minimization [26]. Luo et al. proposed hybrid branch and bound algorithms to minimize the makespan for the two-stage assembly scheduling problem with separated setup time [32]. Zhao et al. proposed an optimal block knowledge-driven backtracking search algorithm (BKBSA) to solve the distributed assembly no-wait flow-shop scheduling problem (DANWFSP) [33]. In the above-

mentioned methods, the improvements of the algorithm effectiveness are still substantially conducted based on the balance of intensification and diversification. The characteristics of the problem and the execution process of the algorithm are not organically integrated.

The estimation of distribution algorithm (EDA) has been widely used in scheduling problems as a meta-heuristic method. The EDA algorithm comes with a natural learning mechanism based on the evolution of probabilistic models. The process of the EDA algorithm is divided into sampling, selection, recombination, and updating of the probability model. The update of the probability model is based on the selected individuals of the population. The probability model directly affects the quality of the algorithm candidate solution and determines the optimization efficiency of the algorithm. In [34], the path relinking enhanced estimation of distribution algorithm has been proposed for the direct acyclic graph task scheduling problem. Sun and Gu proposed an effective hybrid estimation of distribution algorithm to solve the HFSP with an unrelated parallel machine [35]. The EDA structure and the teaching learning-based optimization (TLBO) strategy were used for global and local searches, respectively. Du et al. introduced a hybrid EDA for distributed flexible job shop scheduling with crane transportations [36]. An identification rule of four crane conditions was designed to make fitness calculation feasible. In the EDA component, the parameters in probability matrices are set to be self-adaptive for stable convergence to obtain better output. In the variable neighborhood search (VNS) component, five problem-specific neighborhood structures including global and local strategies are employed to enhance exploitation ability.

This paper optimizes the production process of the rare earth. The mathematical model, a two-stage heterogeneous hybrid flow-shop scheduling problem with sequence-dependent setup time constraints (TSHHFSP with SDST), is extracted from the process of ion exchange in the immersion pond during the production of rare earth oxides. The objective of this optimization problem is the maximum completion time of jobs. According to the no free lunch theorem for optimization, the existing approaches may not be effective for solving the TSHHFSP with SDST. Therefore, it is a challenge to develop effective and efficient algorithms for TSHHFSP with SDST, especially for large-sized problems. In this paper, an estimation of the distribution algorithm with multiple intensification strategies (EDA-MIS) is proposed to solve the TSHHFSP model with SDST. The proposed algorithm is based on the framework of the EDA algorithm. The model studied in this paper incorporates a constraint of the sequence-dependent setup time. A constructive heuristic strategy is designed based on this constraint in the initialization method. The initial solutions of the algorithm are generated through the heuristic and random construction method. The probability model of EDA-MIS is constructed based on the order of

jobs in the outstanding individuals of the parent population. The neighborhood search strategy is added to the algorithm to further optimize the performance of the solutions. The fundamental contributions of this paper are summarized as follows.

- A new scheduling model named TSHHFSP with SDST is developed based on the production process of rare earth. In this model, the production capacities of all machines in a certain stage are heterogeneous. The sequence-dependent setup time of the first stage is considered in the new scheduling model.
- An estimation of distribution algorithm with multiple intensification strategies is proposed to solve the TSHHFSP with SDST. In the initialization phase of the proposed algorithm, some constructive heuristics based on the knowledge of the production data are presented. The EDA and two strategies of neighborhood search are used to further intensify the candidate solutions.

The remainder of this paper is organized as follows. Section 2 presents the modeling and application of the Two-Stage Heterogeneous Hybrid Flow-shop Scheduling Problem (TSHHFSP) with Sequence-Dependent Setup Time (SDST). Section 3 provides the description and detailed decoding process of the TSHHFSP with SDST. Section 3 also describes the proposed EDA-MIS algorithm. The experimental analysis of the proposed algorithm and comparison experiments are presented in Section 4. Finally, conclusions are drawn in Section 5.

## 2 Two-stage heterogeneous hybrid flow-shop scheduling problem (TSHHFSP) with sequence-dependent setup time (SDST)

The HFSP is widely used in practical production processes, such as iron, and steel casting, glass processing, and paper production. In this paper, a TSHHFSP with the constraint of SDST is constructed. This model is derived from the actual production process of rare earth (RE) metals in the northwest region of China. The entire generating process can be divided into two stages: ion exchange in the leaching pool and electrolysis of RE metals. The specific production process is shown in Fig. 1. In the ion exchange process, the electrolyte solution is configured as a leaching agent according to the proportion requirements of RE metals. The RE ore containing an ionic phase is filtered and extracted in the leaching tank. The active ions in the solution exchange with RE ions and the ionic phase of RE are exchanged from the ore carrier to become a new state of RE. The leaching agents need to be reconfigured when different RE is produced in turn. Therefore, in this paper, the TSHHFSP model with SDST is abstracted from the actual production process of RE.

The HFSP can be regarded as a combination of two kinds of scheduling problems that are flow-shop scheduling and parallel machine scheduling. The two subproblems should usually be considered, namely, the sorting of jobs and the distribution of machines in each stage. In the TSHHFSP problem with SDST, the production process is divided into two stages. Several machines need to be selected in each stage, and the machines in the same stage are heterogeneous. There are  $n$  jobs to be processed in the initial time. All jobs are processed from Stage 1 to Stage 2. The process of job  $j \in J$  in machine  $i$  of stage  $k$  cannot be interrupted. The processing time of job  $j \in J$  in machine  $i$  of stage  $k$ , named as  $p_{jik}$ , is positive. At the same time, the job and the machine have a one by one relationship. In the first process stage of the job  $j \in J$ , the setup times of the jobs are sequence-dependent, as  $st_{kjq}$ . All machines and jobs are available to be processed at time zero and job preemption is not allowed. The capacity of intermediate buffers between stages is unlimited. The travel time between consecutive stages is included in the processing times of jobs at the corresponding stages. The problem data is assumed to be deterministic and known in advance. The notations for the problem are listed in Table 1.

The TSHHFSP model with SDST aims to find a schedule that optimizes the maximum completion time (makespan). The problem can be denoted as  $FH2, \left( RM^{(k)} \middle| \begin{smallmatrix} 2 \\ k=1 \end{smallmatrix} \right) \middle| S_{sd}^{(1)} \middle| C_{max}$  according to the notation proposed by Vignier et al. [37], which follows the three-fields notation of [38]. The problem is NP-hard since the standard two-stage HFSP is known to be NP-hard [39].

The Mixed Integer Linear Planning (MIP) model of the TSHHFSP with SDST with the maximum completion time objective is constructed as follows.

Objective:

$$\text{Minimize}(\max C_j) \quad (1)$$

Subject to:

$$C_j = s_{2j} + \sum_{i \in I_2} x_{ji2} p_{ji2}, \quad \forall j \in J \quad (2)$$

$$\sum_{i \in I_k} x_{jik} = 1, \quad \forall j \in J, k \in M \quad (3)$$

$$s_{kj} + \sum_{i \in I_k} x_{jik} p_{jik} + st_{kjq} \leq s_{k+1,j} \quad \forall j, q \in J, (k, k+1) \in M \quad (4)$$

$$\begin{aligned} & s_{kj} - \left( s_{kr} + \sum_{i \in I_k} x_{rik} p_{rik} + st_{krq} \right) \\ & + Q(2 + y_{kjr} - x_{jik} - x_{rik}) \geq 0, \quad \forall j, r, q \in J \\ & : j < r, k \in M, i \in I_k \end{aligned} \quad (5)$$

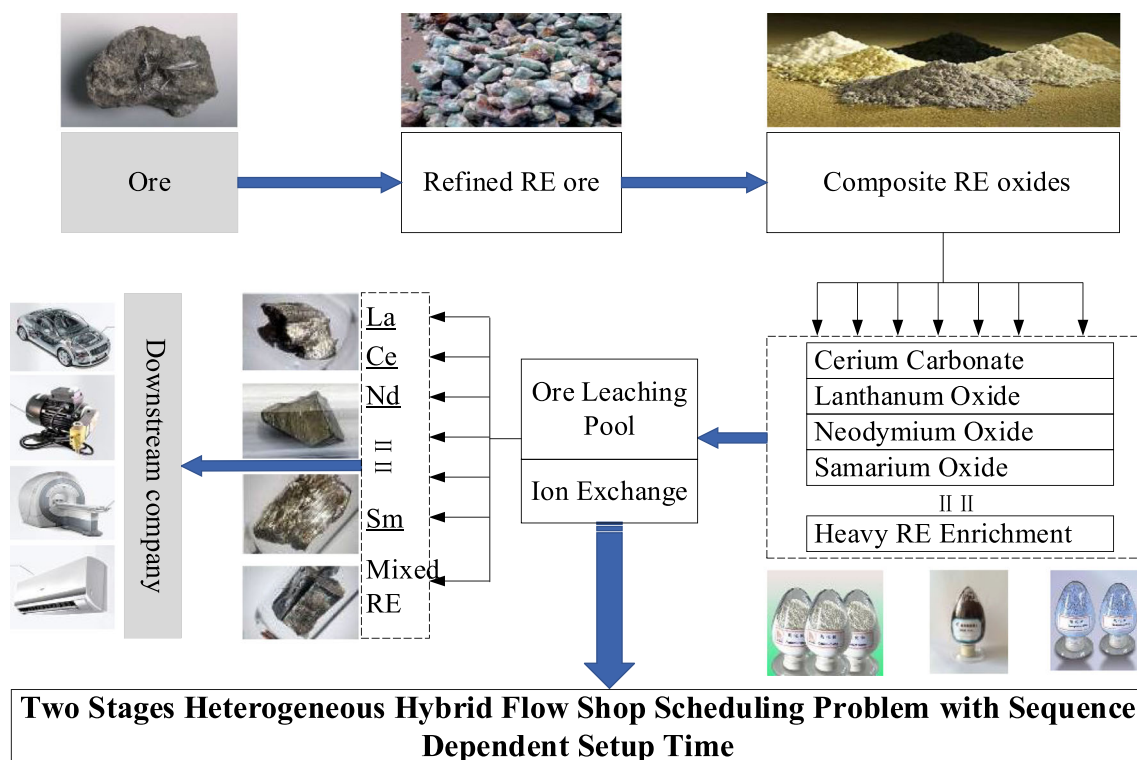


Fig. 1 The production process of rare earth metal

$$\begin{aligned}
 & s_{kr} - \left( s_{kj} + \sum_{i \in I_k} x_{jik} p_{jik} + st_{kjq} \right) \\
 & + Q \left( 3 - y_{kjr} - x_{jik} - x_{rik} \right) \geq 0 \quad \forall j, r, q \in J \\
 & : j < r, k \in M, i \in I_k \quad (6) \\
 & s_{kj} > 0 \quad \forall k \in M, j \in J \quad (7) \\
 & y_{kjr} \in \{0, 1\} \quad \forall k \in M, j, r \in J \quad (8) \\
 & x_{jik} \in \{0, 1\} \quad \forall j \in J, i \in I_k, k \in M \quad (9) \\
 & \text{if } k = 2, st_{kjq} = 0, \quad \forall j, q \in J, k \in M \quad (10)
 \end{aligned}$$

Table 1 Problem notations

Sets	
$M$	Set of stages $\{1, 2\}$
$J$	Set of jobs
$I_k$	Set of machines at stage $k \in M$
Parameters	
$p_{jik}$	Processing time of job $j$ at machine $i$ of stage $k$
$st_{kjq}$	Setup time of job $j$ to job $q$ at stage $k$
$Q$	A very large number
Decision variables	
$s_{kj}$	Starting time of job $j$ at stage $k$
$x_{jik}$	1 if job $j$ is processed at machine $i$ at stage $k$ , 0 otherwise
$y_{kjr}$	1 if job $j$ precedes job $r$ at stage $k$ , 0 otherwise
$C_{max}$	Maximum completion time (makespan)

The objective function (1) minimizes the maximum completion time. Constraint set (2) calculates the completion time of each job. Constraint set (3) guarantees that each job passes through all stages and is processed by exactly one machine at each stage. Constraint set (4) ensures that the next operation of a job starts only after its previous operation is completed. Constraint sets (5) and (6) prevent the overlapping of any two jobs on the same machine and define the sequence of the jobs. For any two jobs assigned to the same machine, the next job can start after the previous one is completed. Constraint set (7)–(10) defines the domains of decision variables.

### 3 The encoding strategy and numerical illustration

The effectiveness of a metaheuristic algorithm relies on the performance of encoding and decoding methods. Job permutation-based encoding method is the most widely used encoding method. The encoding and decoding methods of HFSP are different from the traditional permutation FSP. In the permutation FSP, the coding process is executed based on the sequence of processing jobs. In the decoding process, the sequence of the jobs is determined according to the coding sequence. Then the maximum completion time is calculated based on the processing time and job sequence.

The certain solution of the HFSP can be transformed into a specific code through the encoding method. The encoding method can be understood as a representation of the modern heuristic. A representation assigns the genotypes of the HFSP solution to corresponding phenotypes. The job permutation-based encoding method can effectively clarify the processing details. In the mutation phase of EDA, the probabilistic model is efficiently constructed by the job permutation-based encoding method. The locality of a representation describes how well the neighboring genotypes correspond to the round neighboring phenotypes. A representation with high locality means that the neighboring genotypes correspond neighboring phenotypes. The job permutation-based encoding method is a representation with a high locality. The job permutation-based encoding method is illustrated in Fig. 2.

An example of TSHHFSP with SDST is given as follows. In this instance, there are 10 jobs to be processed. The numbers of machines in the two stages are 2 and 3. The instance is labeled as  $\{10, (2, 3)\}$ . The matrix of process time is generated randomly as shown in Table 2.

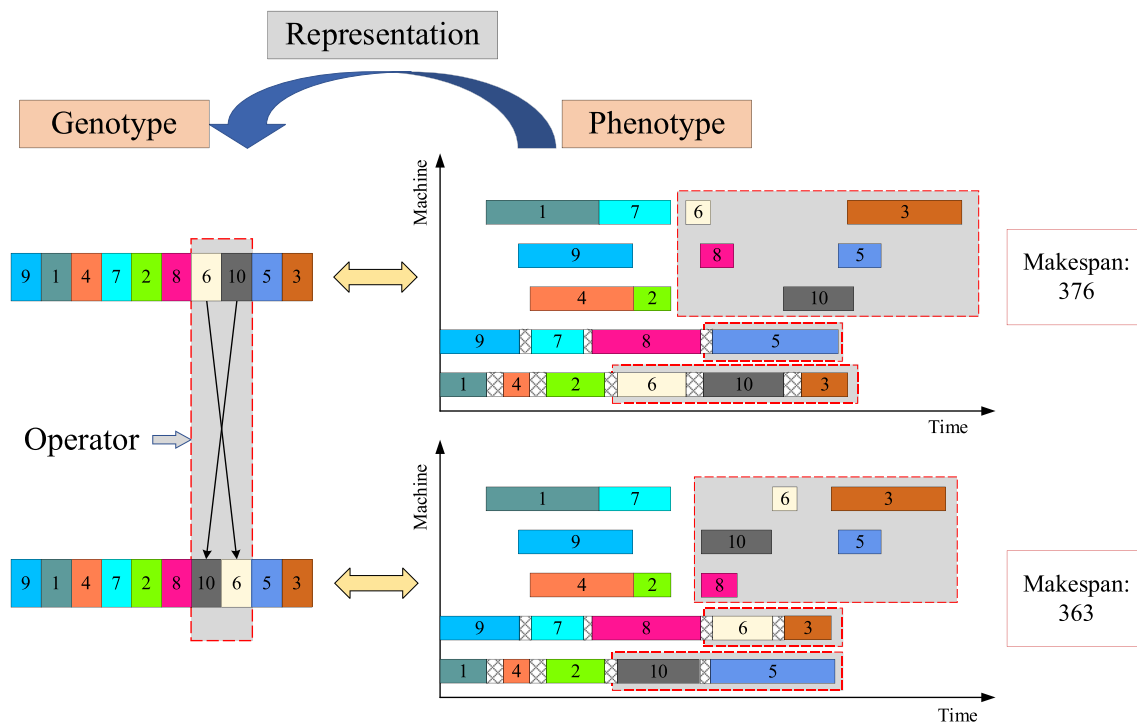
Assuming that job  $q$  is the successor of job  $j$ , the matrix of the sequence-dependent setup time  $st_{kjq}$  is randomly generated as Table 3.

The scheduling solution and its makespan can be generated based on the production data. Fig. 3 shows the Gantt diagram of a scheduling solution. The maximum competition time of the 10 jobs is the competition time of Job 2 and the makespan is 384.

**Table 2** The matrix of process time  $p_{jik}$

$p_{jik}$	Stage 1		Stage 2		
	Machine 1	Machine 2	Machine 3	Machine 4	Machine 5
Job 1	24	29	28	35	30
Job 2	47	50	19	23	19
Job 3	88	83	48	39	46
Job 4	80	76	30	34	34
Job 5	44	46	68	65	61
Job 6	85	78	50	52	50
Job 7	106	101	67	54	64
Job 8	100	86	44	49	51
Job 9	78	87	62	62	69
Job 10	30	25	102	97	97

In the HFSP, the machine allocation of stages should be considered in the process of coding and decoding. The coding rule used in this paper is still based on the job sequence. In the decoding process, the machine with the earliest completion time is selected for processing. In the second processing stage, the job completed early in the first stage is processed preferentially. For example, the sequence of operation in Fig. 3 is (8,5,6,4,9,10,7,3,2,1). Job 9 precedes Job 10 in the first processing stage. The processing time of Job 9 is much longer than that of Job 10. Thus, Job 10 prepares early than Job 9 in the second processing stage. Then Job 10 precedes Job 9 in the



**Fig. 2** The job permutation-based representation of the TSHHFSP with SDST solution



**Table 3** The matrix of sequence-dependent setup time  $st_{kjq}$ 

q \ j	1	2	3	4	5	6	7	8	9	10
1	0	10	6	6	7	10	13	13	10	12
2	8	0	5	11	7	7	7	14	6	7
3	11	12	0	8	5	14	14	11	11	7
4	11	10	12	0	11	5	10	9	10	6
5	5	13	14	5	0	8	5	6	14	8
6	12	11	12	10	12	0	14	7	9	8
7	9	5	5	7	12	10	0	13	5	7
8	11	7	12	5	14	13	9	0	11	5
9	9	13	7	7	11	5	12	7	0	10
10	13	14	14	5	14	6	9	9	13	0

second processing stage. Under this coding rule, the job sequence in Fig. 3 can be coded as  $\pi = \{8, 5, 6, 4, 9, 10, 7, 3, 2, 1\}$ . In the decoding process, the process details can be obtained through the analysis of this code and are shown in Table 4.

## 4 The proposed EDA-MIS algorithm

### 4.1 The framework of the EDA-MIS algorithm

In the proposed EDA-MIS algorithm, the initial solutions are generated by the heuristic method and the stochastic method. In the evolutionary process, the candidate solutions of the algorithm are optimized by the EDA algorithm. Then the individuals in the population evolve ulteriorly by the strategy of neighborhood search. The specific process of the proposed EDA-MIS is shown in Algorithm 1.

### Algorithm 1: EDA-MIS Algorithm

**Input:** Produce Data  
**Output:** Optimal Makespan  
 Initialization of algorithm parameters;  
 Initialization of population;  
**while** The termination conditions are not met **do**  
   Selection;  
   Update of the probability model;  
   Sampling the new population;  
   Neighborhood search;  
**end**

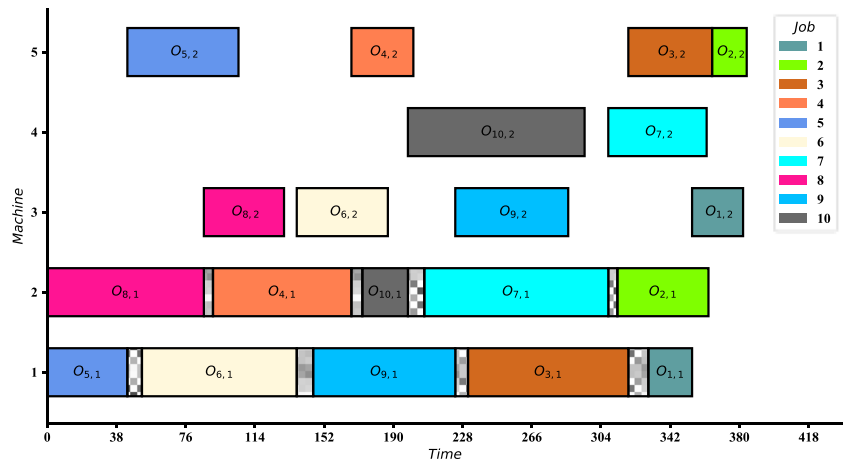
The EDA-MIS algorithm can learn the implicit knowledge of the HFSP in the evolutionary process and that knowledge can be used to decide the job sequence in the next generation. Figure 4 illustrates the framework of the proposed EDA-MIS algorithm.

The probability distribution model is constructed according to the excellent individuals of the candidate solution in the evolutionary process of EDA. The individuals of the next generation are obtained by sampling based on the probability distribution model. Then, the individuals are optimized by performing selection, model mutation, resampling, and other operations, until a certain stopping criterion is met. In this paper, an order matrix  $\phi$  is constructed to count the order of jobs in the excellent individuals of the population.

$$\phi = \begin{bmatrix} \varphi_{1,1} & \varphi_{1,2} & \varphi_{1,3} & \cdots & \varphi_{1,n} \\ \varphi_{2,1} & \varphi_{2,2} & \varphi_{2,3} & \cdots & \varphi_{2,n} \\ \varphi_{3,1} & \varphi_{3,2} & \varphi_{3,3} & \cdots & \varphi_{3,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \varphi_{n,1} & \varphi_{n,2} & \varphi_{n,3} & \cdots & \varphi_{n,n} \end{bmatrix} \quad (11)$$

Where  $\varphi_{j,i}$  is used to count the number of times job  $j$  appears in and before the position  $i$ . The order matrix  $\phi$  describes the order of jobs in the processing sequence. A new probability distribution model  $P$  is obtained by normalizing

**Fig. 3** A feasible scheduling solution of {10, (2, 3)} instance



**Table 4** The details of the decoding processing

Job	Stage 1					Stage 2			
	Machine	Start time	Process time	Setup time	End time	Machine	Start time	Process time	End time
1	1	319	24	11	354	3	354	28	382
2	2	308	50	5	363	5	365	19	384
3	1	224	88	7	319	5	319	46	365
4	2	86	76	5	167	4	167	34	201
5	1	0	44	0	44	5	44	61	105
6	1	44	85	8	137	3	137	50	187
7	2	198	101	9	308	4	308	54	362
8	2	0	86	0	86	3	86	44	130
9	1	137	78	9	224	4	224	62	286
10	2	167	25	6	198	5	198	97	295

the order matrix and learning from the distribution model of the previous generation.

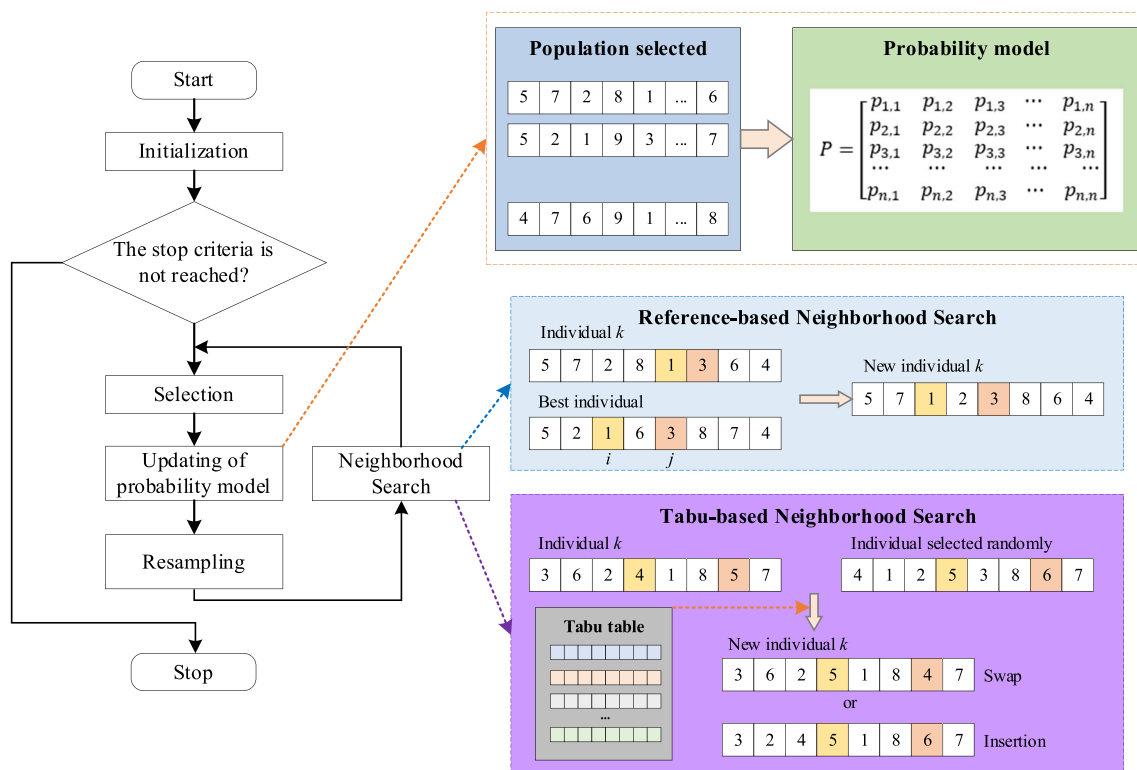
$$P = \begin{bmatrix} p_{1,1} & p_{1,2} & p_{1,3} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & p_{2,3} & \cdots & p_{2,n} \\ p_{3,1} & p_{3,2} & p_{3,3} & \cdots & p_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{n,1} & p_{n,2} & p_{n,3} & \cdots & p_{n,n} \end{bmatrix} \quad (12)$$

The probability statistical model  $P$  describes the probability distribution of the location of each job in the processing

sequence. The individuals of the next generation are sampled from model  $P$ . The probability model is obtained by the previous probability model and the order matrix of the current generation. The specific update formula is as follows:

$$P^{(g+1)} = \alpha P^{(g)} + (1-\alpha) \text{normalise}\{\phi\} \quad (13)$$

where  $\alpha$  is the learning factor and  $\text{normalise}\{\phi\}$  represent the normalization results of the order matrix  $\phi$ .


**Fig. 4** The framework of the proposed EDA-MIS algorithm

## 4.2 Heuristic rules

The HFSP is the coupling of many scheduling problems, including the combination optimization of job sequence and the choice of the processing machine. This paper studies the TSHHFSP with the constraint of SDST. Thus, compared with the traditional HFSP model, the reduction of the total sequence-dependent setup time should be considered when scheduling the job sequences. In this paper, several heuristic construction rules are proposed for the above research problems, which provide the initial solutions for the meta-heuristic algorithm.

### 1. Heuristic rule 1

The constraint of SDST is considered in the study model. In the model of this paper, there is no setup time before the first machining job of each machine. Therefore, the jobs with the largest setup time are operated preferentially in the construction heuristic method to avoid the maximum value of the SDST. The maximum value of SDST is found to decide which job operates first. Such as the maximum setup time in Table 3 is setup time from Job 2 to Job 8, i.e.  $st_{k28}$ . Then the successor job of this setup time, which is Job 8, is chosen as the first machining job. Assuming that there are  $k$  machines in the first stage. In this example,  $k$  is 2. The  $k$ th maximum value is found in turn from the remaining SDST, the  $st_{k36}$  of Table 3 is picked up. The successor jobs of these setup times, Jobs 6, are chosen as the first machining job in the remaining machines of the first stage. According to the above process, the first machining jobs in the machines of the first stage are determined. The job with the lowest setup time is identified as the subsequent machining job. This operation is repeated until all jobs are scheduled for processing in the first stage. The final processing sequence is obtained, the result of Table 3 is {8, 6, 4, 10, 2, 7, 3, 9, 5, 1}.

---

#### Algorithm 2: Heuristic Rule 1

---

**Input:** Process parameters  $st_{kjq}$   
**Output:** Process sequence  $\pi$  at stage 1  
 Process parameters initialization;  
 $\max(st_{kjq}) = st_{k,a,b}$ ;  
 Select job  $b$  as the first process job;  
 Add  $b$  to process sequence  $\pi$ ;  
 $pre\_job = b$ ;  
 $job\_npset = 1, 2, 3, \dots, b-1, b+1, \dots, n$ ;  
**while** *distribution of jobs is not end* **do**  
    $\min_{q \in job\_npset}(k, st_{pre\_job,q}) = st_{k,pre\_job,x}$ ;  
   select job  $x$  as the next process job;  
   Add  $x$  to process sequence  $\pi$ ;  
    $job\_npset = job\_npset - x$ ;  
    $pre\_job = x$ ;  
**end**

---

### 2. Heuristic rule 2

In this paper, the proposed TSHHFSP problem with SDST consists of two stages. The start time of the second stage directly determines the final makespan. The jobs are processed as the planned sequence, in which the start time of each machine in the second stage is early as possible. In this heuristic construction method, the jobs, as well as the processed machines, with the shortest processing time in the first stage are identified in turn. Such as the processing time of Job 1 on machine 1 and Job 10 on machine 2 in Table 2 are selected. Then the jobs with the earliest completion time on each machine in the first stage are processed as successors.

---

#### Algorithm 3: Heuristic Rule 2

---

**Input:** Process parameters  $p_{ij}$   
**Output:** Process sequence  $\pi$  at stage 1  
 Process parameters initialization;  
 $job\_npset = J$ ;  
**while**  *$i$  in the machine set of stage 1,  $I_1$*  **do**  
    $\min_{j \in J}(p_{ij}) = p_{i,x_i}$ ;  
    $x_i$  are not equal to the selected jobs;  
   Add  $x_i$  to process sequence  $\pi$ ;  
    $job\_npset = J - x_i$ ;  
**end**  
**while** *distribution of jobs is not end* **do**  
   Update the earliest completion time of  $I_1$ ;  
    $\min(c_i) = c_y, i \in I_1$ ;  
    $\min_{j \in J}(p_{yj}) = p_{y,a}$ ;  
   select job  $a$  as the next process job;  
   Add  $a$  to process sequence  $\pi$ ;  
    $job\_npset = job\_npset - a$ ;  
**end**

---

## 4.3 Neighborhood search

In the proposed algorithm, two neighborhood search strategies are used to further optimize the performance of the population. These two strategies are reference-based neighborhood search and tabu-based neighborhood search.

### 1. Reference-based neighborhood search

In this method, the optimal individual in the population is recommended as the reference solution. The process of neighborhood search is performed based on the reference solution. Two locations  $i$  and  $j$ , where  $(i < j)$ , are selected during the search process. The jobs at positions  $i$  and  $j$  in the reference solution are inserted into  $i$  and  $j$  positions of the population individuals, respectively. Then, the neighborhood search solution is obtained. The new solution is compared to the original population individual. If the performance of the new



solution is better than that of the original population individual, the original population individual is replaced by the corresponding neighborhood search solution.

---

**Algorithm 4:** Reference-based Neighborhood Search
 

---

**Input:** Population of solution  
**Output:** New solutions  
 Select the best individual as the reference solution;  
**while** *indiv* in the population of solution **do**  
   random *i, j*;  
   if (*i* > *j*): swap *i, j*;  
   Select the jobs of *i, j* locations in the reference solution;  
   Delete the two jobs from *indiv*;  
   Insert the two jobs to the *i, j* locations of *indiv*;  
   Mark the new solution as the *new\_indiv*;  
   Compare the objective of *new\_indiv* and *indiv*;  
   Replace *indiv* if *new\_indiv* is better than *indiv*;  
**end**

---

## 2. Tabu-based neighborhood search

In the method, the truncated individuals of the population in the previous generation are added to the taboo table. The candidate solution is compared with the taboo table. If the candidate solution is included in the taboo table, the neighborhood search is not carried out. Otherwise, the neighborhood search is carried out. One strategy is randomly selected between swap and insertion operators for neighborhood search.

---

**Algorithm 5:** Tabu-based Neighborhood Search
 

---

**Input:** Population of solution, Tabu table  
**Output:** New solutions  
**while** *indiv* in the population of solution **do**  
   **if** *indiv* is included in the Tabu table **then**  
     Not excute the process of neighborhood search;  
   **else**  
     Select one between insertion and swap;  
     Excute the neighborhood search;  
   **end**  
**end**

---

$$\begin{aligned}
 O(n, I, T, Np) &= O(T) \times (O(Np \times n^2) + O(Np \times n \times (I_1 + I_2)) + O(Np \times n + n^2) + O(Np \times 2n + Np \times n^2)) \\
 &= O(T) \times O(Np \times n^2 + Np \times n \times (I_1 + I_2) + Np \times n + n^2 + Np \times 2n + Np \times n^2) \\
 &= O(T) \times O(2 \times Np \times n^2 + Np \times n \times (I_1 + I_2) + 3 \times Np \times n + n^2) \\
 &\approx O(T) \times O(2 \times Np \times n^2 + Np \times n \times (I_1 + I_2) + n^2)
 \end{aligned}$$


---

## 5 Experiment and analysis

To verify the performance of the proposed EDA-MIS algorithm in solving the TSHHFSP, 140 test instances were randomly generated. The size of processed jobs is divided into 4 types, 5, 10, 20, and 50. For each size of the job, there are

## 4.4 Computational complexity of the EDA-MIS

In the proposed TSHHFSP model with SDST, the approximate optimal solution can be obtained by the EDA-MIS algorithm. The specific model is extracted from the actual production process. The computational time is an important indicator to measure the quality of the proposed algorithm. The consumed time is also an important indicator in the actual production process. In this section, the time complexity of the proposed EDA-MIS is analyzed.

Suppose that the number of the processing jobs is  $n$ . The numbers of machines in the two processing stages are  $I_1$  and  $I_2$ . The population size of the proposed EDA-MIS is set as  $Np$ . The number of iterations is supposed as  $T$ . The EDA-MIS algorithm is composed of five phases, which are the initialization of the population and the other four operators for the further evolution of the solutions. In the initialization phase, the individuals are generated based on the two constructive heuristics and stochastic methods with the complexity  $O(Np \times n^2)$ . For the sampling operation, each sequence is generated with the roulette strategy by sampling based on the probability matrix  $P$ . The time complexity of the sampling operator is also  $O(Np \times n^2)$ . In the selection process, the job sequence is decoded into the production details and then the objective value is obtained. The computational complexity of the selection operator is  $O(Np \times n \times (I_1 + I_2))$ . For model updating, the probability matrix  $P$  is updated based on the selected solutions. The computational complexity of the model updating process is  $O(Np \times n + n^2)$ . Finally, the neighborhood search operator is executed for further evolution. The computational of this operator is  $O(Np \times 2n + Np \times n^2)$ . Thus, the time complexity of the proposed EDA-MIS is calculated as follows.

various species of machine numbers. The specific test parameter combinations are shown in Table 5. Each combination of test problems contains 10 test instances. The specific processing time is a randomly generated integer between 1 to 99. The SDST in the first stage is also a randomly generated integer between 1 to 10.

**Table 5** The list of parameters of the production process

Stage number(m)	Job number(n)	Machine number of each stage
2	5	(1,2)
2	5	(2,2)
2	10	(1,3)
2	10	(2,3)
2	10	(3,3)
2	20	(2,3)
2	20	(2,4)
2	20	(3,5)
2	20	(4,4)
2	50	(3,5)
2	50	(3,6)
2	50	(4,6)
2	50	(5,5)
2	50	(5,6)

During the experiment, each instance was solved 51 times by the algorithms [40]. The termination conditions of the experiment were set as follows: the number of iterations reaches  $1000 \times n$  or the maximum stagnation iteration reaches 50. All experiments were programmed in Python language and executed on a Windows 7 system. The integrated development environment of the experiment was Anaconda 3. The experimental computer processor was configured as a 2.3 GHz Intel Core i5. According to the experimental results, the average relative deviation percentage (ARPD) indicator value is selected as the evaluation indicator in this paper. The calculation formula for ARPD value is:

$$ARPD = \text{Average}_{ins} \left( \frac{1}{R} \sum_{i=1}^R \frac{C_i - C_{opt}}{C_{opt}} \times 100\% \right) \quad (14)$$

where  $C_i$  represents the makespan of the optimal solution obtained by an algorithm in the run  $i$ ,  $C_{opt}$  indicates the makespan of the optimal solution found,  $R$  is the number of times the algorithm runs on a single instance and  $ins$  represents the instance label for each problem,  $ins = 1, 2, 3, \dots, 10$ .

In this paper, the estimation of distribution algorithm (EDA) [41], estimation of distribution algorithm with tabu search (EDA-TS) [42], estimation of distribution algorithm with iterated local search (EDA-iLS) [43], discrete artificial bee colony (DABC) [44], shuffled frog-leaping algorithm (SPLA) [10], and the genetic algorithm with tabu search (GA-TS) [45] are selected as the comparison algorithms. The population sizes of the comparison algorithms were uniformly set to  $10 \times n$ , where  $n$  was the number of jobs. In the EDA and its variant algorithms, the selection ratio of the population was set to 0.3. The comparison of ARPD can be obtained according to the running results of the algorithm. The comparison results are shown in Table 6. The best

performances of the comparison algorithms in the test problems are marked as bold in the Table 6. The experimental results show that the ARPD value of the EDA-MIS algorithm is significantly better than that of the EDA, EDA-TS, EDA-iLS, DABC and SFLA algorithms on most problems. It can be found that the performance of the GA-TS algorithm is better than that of the EDA-MIS algorithm in some problems of small sizes, such as  $n = 5, 10$ . However, the EDA-MIS algorithm is better than the GA-TS algorithm in the problems where the number of jobs is high, such as  $n = 20$  and 50. In view of the makespan indicator, the EDA-MIS algorithm is also significantly better than the GA-TS algorithm.

In the actual production process, the number of jobs is generally very large. The weakness of the EDA-MIS algorithm in the low-level problems can be ignored. On the other hand, the TSHHFSP model with SDST in this paper is extracted from the actual production factory. In this model, the limitation of maximum stagnation iteration is added to the termination conditions. In some low-level problems, the termination condition of maximum stagnation iteration leads to the early ending of the evolution process in the EDA-MIS algorithm. Therefore, the performance of the proposed EDA-MIS algorithm in these instances is poor due to the iterations being less than that in other comparative algorithms. For example, the performances of the five algorithms are shown in Fig. 5. The number of jobs is 10. The numbers of machines in Stages 1 and 2 are 1 and 3, respectively. As Fig. 5 shows, the EDA-MIS algorithm stops evolution on the 70th iteration. While the GA-TS algorithm evolves to the iteration of 175. When the iteration of evolution is 70, the EDA-MIS algorithm outperforms other algorithms.

In addition, another set of experiments were conducted for this problem. In the case where the number of jobs was 10, the stop criteria of the GA-TS and EDA-MIS algorithms were uniformly set to a fixed iterations number of 1000 times. The performances of these two algorithms are compared in Table 7. It can be seen from the table that the EDA-MIS shows better performance than the GA-TS in the case of the same number of iterations. These results further validate the effectiveness of the proposed EDA-MIS.

The results of 51 runs of the EDA-MIS algorithm and each comparison algorithm in all test instances are statistically analyzed. The mean and the standard deviation of makespan solved by EDA-MIS and each comparison algorithm are obtained. Finally, the average values of the statistical results on 10 test instances of each problem are obtained. The results are shown in Table 8. The best performances of the comparison algorithms in the test problems are marked as bold in the Table 8.

It can be seen that the performance of the EDA-MIS algorithm is better than that of other comparison algorithms in solving large-scale problems. When the numbers of jobs are 5 and 10, the performance of the EDA-MIS is slightly imperfect compared with the GA-TS. However, the EDA-MIS performs more effectively than the other five algorithms. As the

**Table 6** Results of different algorithms in TSHHFSP problem with SDST

Instance	EDA	EDA-TS	EDA-iLS	DABC	SFLA	GA-TS	EDA-MIS
n5m1-2	0.3154	0.0259	0.2748	0.3092	0.0031	<b>0.0000</b>	<b>0.0000</b>
n5m2-2	1.2195	0.1293	0.5761	1.0358	0.0153	<b>0.0000</b>	<b>0.0000</b>
n10m1-3	1.2503	0.7641	0.7986	0.8399	0.7431	<b>0.2780</b>	0.3555
n10m2-3	2.4535	1.7455	1.9827	2.1416	1.8836	<b>0.6378</b>	1.4296
n10m3-3	3.5679	2.4594	3.0437	3.2579	2.3903	<b>0.7725</b>	2.1357
n20m2-3	2.1771	2.0954	<b>1.6276</b>	1.7295	2.1403	1.9405	1.8474
n20m2-4	2.3095	1.8313	1.7189	1.8323	1.9895	1.6760	<b>1.5112</b>
n20m3-5	3.2089	2.7174	2.5415	2.5211	2.9017	2.4500	<b>2.4217</b>
n20m4-4	4.0233	3.4177	4.0018	<b>2.8003</b>	4.0031	3.1239	3.4437
n50m3-5	2.2085	1.7045	2.2314	1.9701	1.8616	1.6018	<b>1.5712</b>
n50m3-6	2.1755	2.0523	2.0623	1.9598	1.9210	1.6766	<b>1.5314</b>
n50m4-6	2.4473	1.9664	2.5955	1.9751	2.3101	1.9814	<b>1.9444</b>
n50m5-5	2.5087	2.5137	3.1406	2.1976	2.5750	<b>1.9179</b>	2.4046
n50m5-6	2.5235	2.3134	3.0864	<b>1.8565</b>	2.6726	2.4361	2.2802

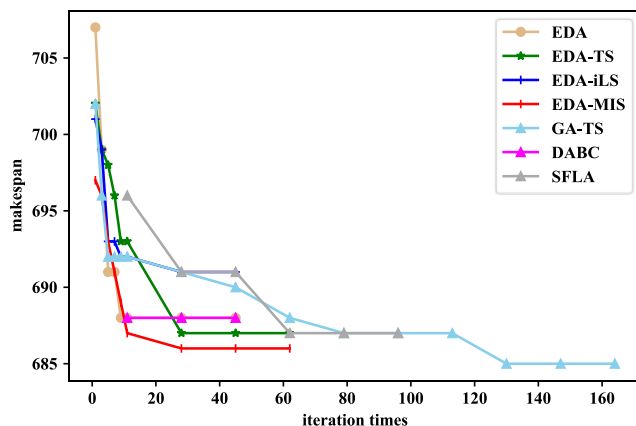
scale of the problem grows, the advantage of the solution performance of the proposed EDA-MIS algorithm becomes more apparent.

In the proposed EDA-MIS algorithm, the EDA algorithm is responsible for performing an exploration search on the decision space. The solution space is effectively explored by performing the operations such as selection, model updating and sampling. The reference-based neighborhood search strategy is responsible for the neighborhood search of the selected optimal reference solution. This strategy mainly performs the exploitation search. In the tabu-based neighborhood search strategy, some solutions with poor performance are liberated into the tabu list during the search process. If the neighborhood solution is included in the tabu list, the neighborhood search is abandoned to avoid an invalid search. An ablation experiment is conducted to verify the effectiveness of the two searches strategies in the proposed EDA-MIS algorithm. The results of the ablation experiment demonstrate that the combination of the two neighborhood search strategies can

lead to a performance boost. The results are shown in Table 9, where “Without RNS” and “Without TNS” represent the EDA-MIS algorithm without the reference-based and the tabu-based neighborhood search strategies, respectively. The best performances of the comparison algorithms in the test problems are marked as bold in the Table 9.

Figures 6 and 7 are the boxplots of the performance comparison between the EDA-MIS algorithm and the EDA, EDA-TS, EDA-iLS, DABC, SFLA and GA-TS algorithms in the TSHHFSP problems with SDST. The sizes of jobs in these problems are 20 and 50. The experimental results show that the solving performance of the EDA-MIS algorithm in 20-jobs and 50-jobs problems are better than that of the other four algorithms in terms of accuracy and stability.

The convergence plots of the compared algorithms are shown in Figs. 8 and 9 on TSHHFSP problems with SDST in which the numbers of jobs are 20 and 50. It is observed that the convergence rate of the EDA-MIS is better than the other algorithms in these problems. In the proposed EDA-MIS algorithm, the EDA algorithm can learn the characteristic of the problems through the updating of the probability model. The performance of the EDA-MIS algorithm can be evolved further by the neighborhood search strategy. The modified EDA algorithm and the neighborhood search operations play a

**Fig. 5** The convergence performance of the five comparison algorithms**Table 7** The performance of the comparison algorithms with the fixed iterations number

Algorithm	Instance	n10m1-3	n10m2-3	n10m3-3
EDA-MIS	ARPD	7.63E-4	6.04E-4	7.59E-4
	Mean	610.02	310.15	228.05
GA-TS	ARPD	1.45E-3	6.51E-3	6.09E-3
	Mean	610.69	311.92	229.16

**Table 8** The statistical results of the comparison experiment

Problem	Criterion	EDA	EDA-TS	EDA-ILS	EDA-MIS	GA-TS	DABC	SFLA
n5m1-2	Mean	326.39	325.48	326.22	325.42	<b>325.40</b>	326.33	325.49
	Std.Dev.	1.6115	0.3488	0.8306	0.0844	0	1.0248	0.4031
n5m2-2	Mean	201.84	199.66	200.59	199.47	<b>199.40</b>	201.54	199.57
	Std.Dev.	3.2510	1.0050	2.3116	0.0963	0	2.6207	0.9115
n10m1-3	Mean	619.08	614.05	615.10	612.32	<b>611.25</b>	616.30	613.56
	Std.Dev.	3.7027	2.6072	2.4584	1.5709	1.4658	2.8829	2.3337
n10m2-3	Mean	319.14	315.62	317.34	314.81	<b>312.02</b>	318.26	313.36
	Std.Dev.	4.0200	3.1188	3.1697	2.5749	1.6571	3.4307	2.3613
n10m3-3	Mean	237.76	233.59	235.93	232.67	<b>229.63</b>	237.12	231.12
	Std.Dev.	4.5425	3.7854	3.5270	3.0718	1.7784	3.7411	2.6984
n20m2-3	Mean	611.36	606.46	603.18	<b>599.40</b>	601.73	608.77	605.16
	Std.Dev.	9.0924	7.8103	10.9150	7.1972	6.9380	5.5720	4.5216
n20m2-4	Mean	641.75	636.62	632.75	<b>629.07</b>	632.49	637.19	635.84
	Std.Dev.	9.7288	8.6828	11.3699	7.2622	7.4547	5.9421	4.9482
n20m3-5	Mean	435.91	430.96	428.62	<b>424.98</b>	426.91	434.74	429.33
	Std.Dev.	8.3572	7.3511	9.7944	7.1809	6.2858	5.3906	4.4965
n20m4-4	Mean	322.55	317.65	315.09	<b>311.49</b>	311.61	322.06	314.24
	Std.Dev.	7.0401	6.9144	9.0792	7.1649	5.9269	4.6968	4.4151
n50m3-5	Mean	1004.09	998.12	990.49	<b>990.38</b>	994.83	1000.43	999.66
	Std.Dev.	7.0608	6.4117	8.8627	6.4324	5.4368	8.4734	6.1972
n50m3-6	Mean	1048.46	1040.65	1033.45	<b>1033.49</b>	1037.75	1044.18	1044.05
	Std.Dev.	6.3314	5.6344	5.2612	5.5790	4.6788	8.3654	6.1212
n50m4-6	Mean	822.06	816.79	810.32	<b>809.76</b>	812.04	821.04	816.98
	Std.Dev.	6.6344	5.7856	5.3191	4.7662	4.4043	6.9792	5.7650
n50m5-5	Mean	654.08	648.18	641.27	<b>638.38</b>	642.15	652.21	646.09
	Std.Dev.	6.2899	5.3953	5.5453	4.8141	4.2412	5.8143	5.6036
n50m5-6	Mean	639.17	634.22	626.98	<b>624.89</b>	628.34	637.29	632.37
	Std.Dev.	5.9281	5.1355	5.8302	5.3193	4.3204	6.0471	5.2294

**Table 9** The results of the ablation experiment

Problem	Without RNS		Without TNS		EDA-MIS	
	Mean	Std.Dev.	Mean	Std.Dev.	Mean	Std.Dev.
n5m1-2	325.48	0.3488	326.22	0.8306	<b>325.42</b>	0.0844
n5m2-2	199.66	1.005	200.59	2.3116	<b>199.47</b>	0.0963
n10m1-3	614.05	2.6072	615.1	2.4584	<b>612.32</b>	1.5709
n10m2-3	315.62	3.1188	317.34	3.1697	<b>314.81</b>	2.5749
n10m3-3	233.59	3.7854	235.93	3.527	<b>232.67</b>	3.0718
n20m2-3	606.46	7.8103	603.18	10.915	<b>599.4</b>	7.1972
n20m2-4	636.62	8.6828	632.75	11.3699	<b>629.07</b>	7.2622
n20m3-5	430.96	7.3511	428.62	9.7944	<b>424.98</b>	7.1809
n20m4-4	317.65	6.9144	315.09	9.0792	<b>311.49</b>	7.1649
n50m3-5	998.12	6.4117	990.49	8.8627	<b>990.38</b>	6.4324
n50m3-6	1040.65	5.6344	1033.45	5.2612	<b>1033.49</b>	5.579
n50m4-6	816.79	5.7856	810.32	5.3191	<b>809.76</b>	4.7662
n50m5-5	648.18	5.3953	641.27	5.5453	<b>638.38</b>	4.8141
n50m5-6	634.22	5.1355	626.98	5.8302	<b>624.89</b>	5.3193

desirable role to balance global exploration and local exploitation. The convergence rate and the convergence speed can be improved by the modified EDA algorithm and the neighborhood search operations. However, according to the no-free lunch theorem (NFL), no algorithm is better than the linear enumeration of the search space or the pure random search algorithm.

Wilcoxon rank tests are performed based on the above experimental results to further test the statistical performance of the proposed EDA-MIS algorithm. The algorithms are compared in pairs to check the significant difference between the EDA-MIS algorithm and the comparison algorithms. Tables 10, 11, 12 and 13 show the statistical results of the Wilcoxon test between EDA-MIS and the comparison algorithm at different scale problems, with the EDA-MIS algorithm as the control algorithm. In Tables 10, 11, 12 and 13, R+ is the sum of the rank that EDA-MIS outperforms other algorithms in the current row, and R- is the sum of the levels that other algorithms in the current row outperform the EDA-

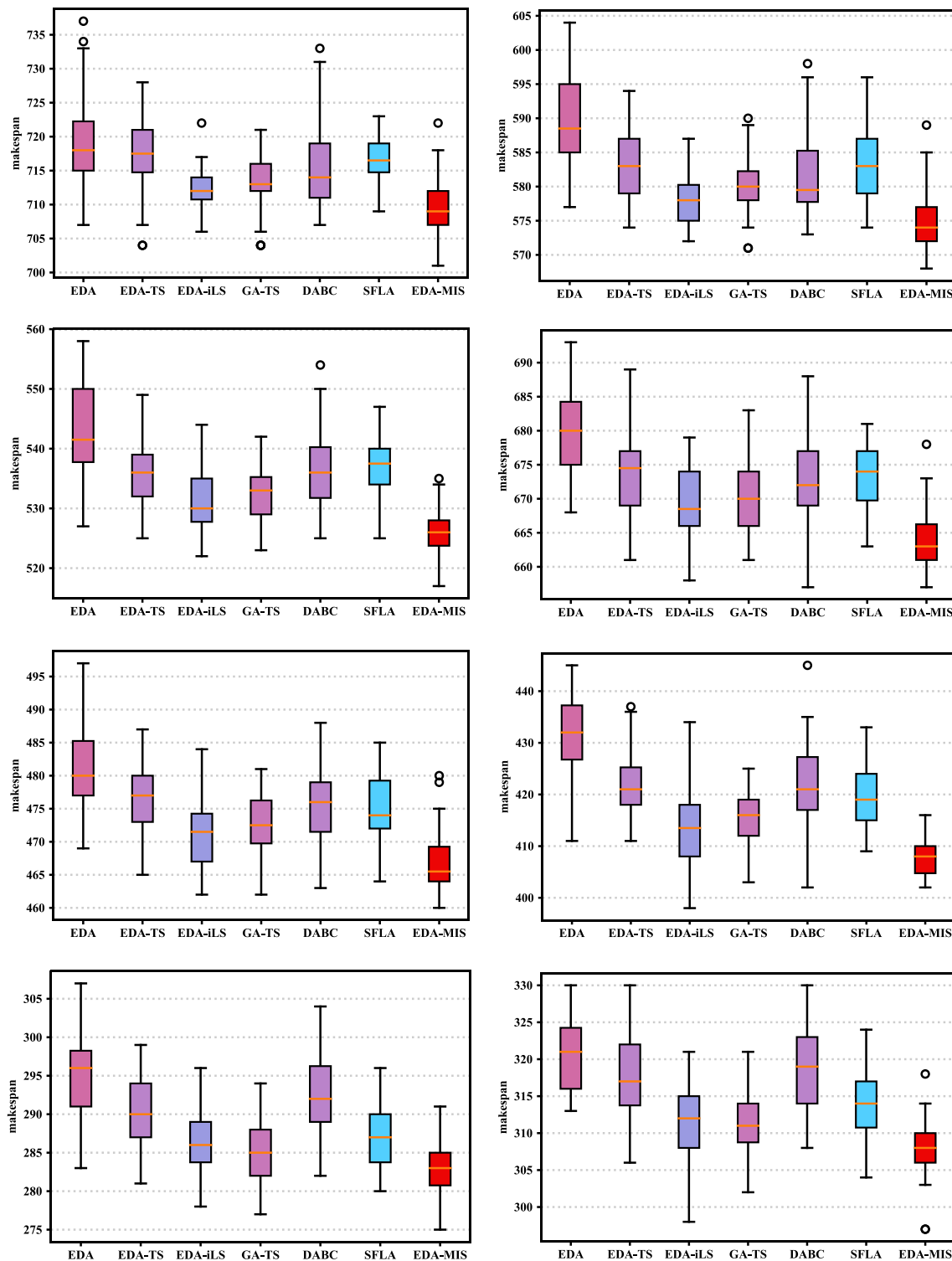


Fig. 6 The boxplots of different algorithms for TSHHFSP with SDST of 20 jobs

MIS. A yes means that EDA-MIS is significantly superior to other algorithms in the current row and a no means the opposite. The “yes” are marked as bold in the Tables 10, 11, 12 and 13. The table is bolded when the EDA-MIS significantly differs from the comparison algorithm and is superior to the comparison algorithm within the confidence interval. It

turns out that the proposed EDA-MIS algorithm has better results than the EDA, EDA-iLS, EDA-TS, DABC and SFLA algorithms in TSHHFSP problems with 5 and 10 jobs. The performance of the EDA-MIS is slightly worse compared to that of the GA-TS algorithm. The reasons that lead to this phenomenon are mentioned above in this paper. The stop



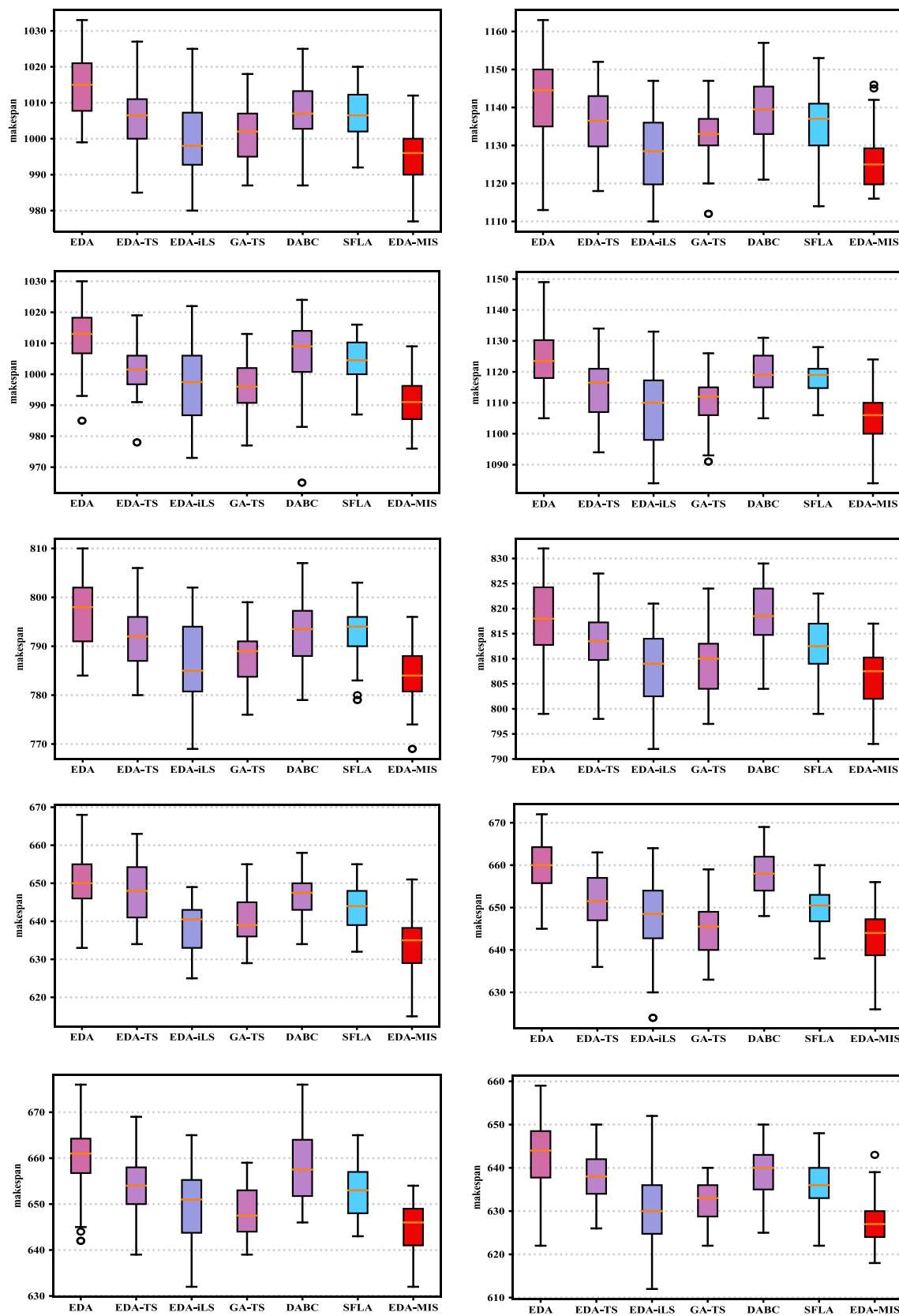


Fig. 7 The boxplots of different algorithms for TSHHFSP with SDST of 50 jobs

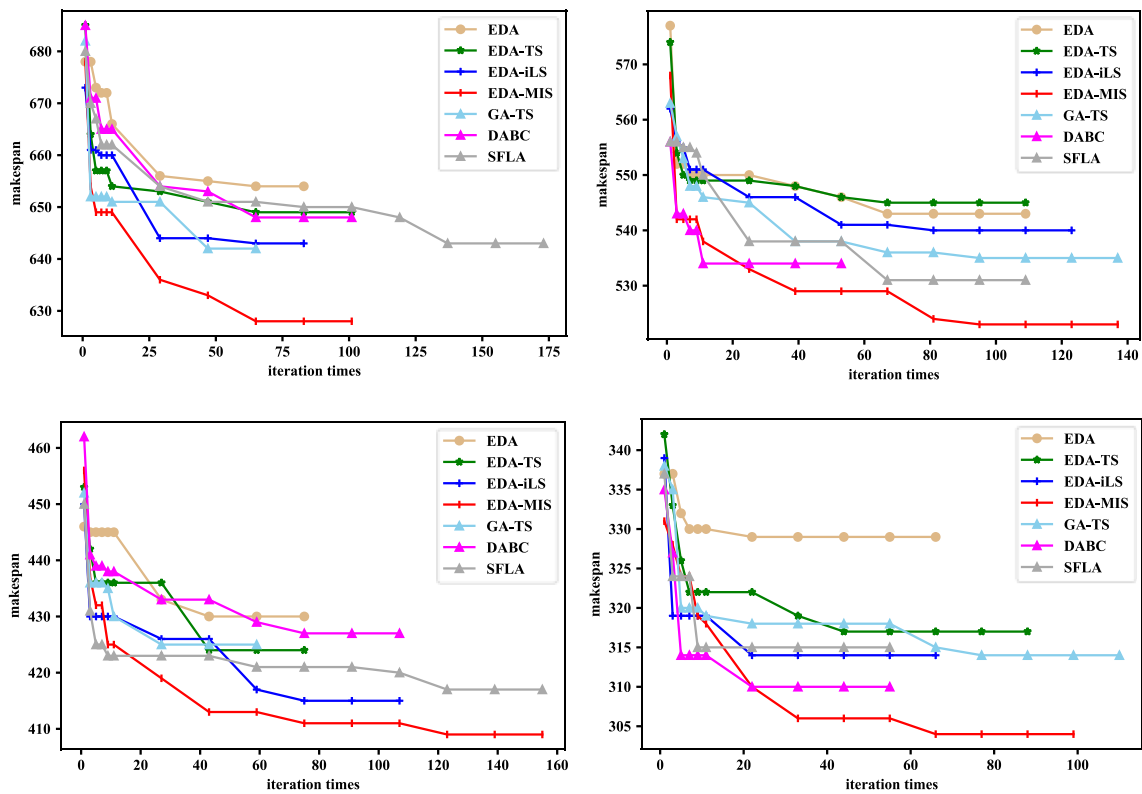


Fig. 8 The convergence plots of different algorithms for TSHHFSP with SDST of 20 jobs

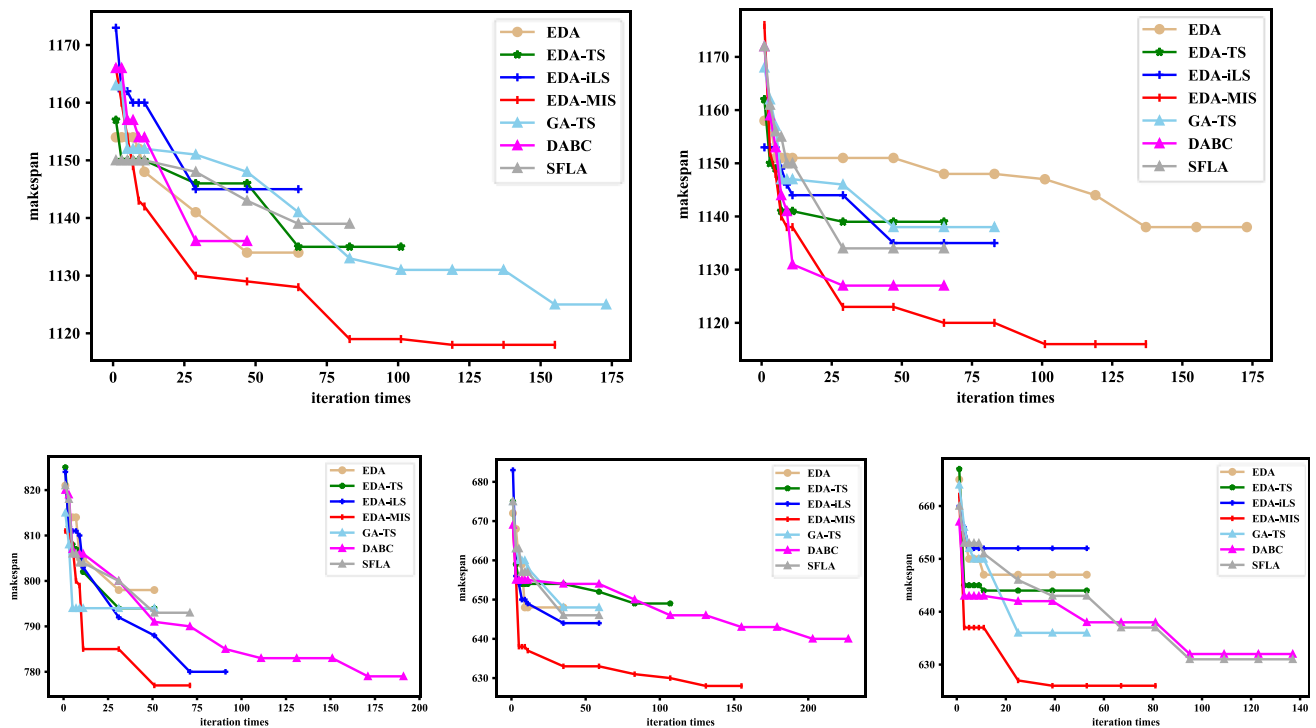


Fig. 9 The convergence plots of different algorithms for TSHHFSP with SDST of 50 jobs

**Table 10** The Wilcoxon's rank-sum test results for 5 jobs

EDA-MIS vs	R+	R-	Z	p Value	$\alpha=0.05$	$\alpha=0.1$
EDA	210	0	-3.920	8.80E-5	yes	yes
EDA-TS	115	21	-2.432	1.50E-2	yes	yes
EDA-iLS	171	0	-3.725	1.95E-4	yes	yes
GA-TS	0	6	1.604	1.09E-1	no	no
DABC	171	0	-3.724	1.96E-4	yes	yes
SFLA	117.5	18.5	-2.563	1.04E-2	yes	yes

**Table 11** The Wilcoxon's rank-sum test results for 10 jobs

EDA-MIS vs	R+	R-	Z	p Value	$\alpha=0.05$	$\alpha=0.1$
EDA	465	0	-4.782	2.00E-6	yes	yes
EDA-TS	410	55	-3.651	2.61E-4	yes	yes
EDA-iLS	435	0	-4.703	3.00E-6	yes	yes
GA-TS	11.5	423.5	4.455	8.00E-6	no	no
DABC	435	0	-4.703	3.00E-6	yes	yes
SFLA	174.5	290.5	-1.193	2.33E-1	no	no

**Table 12** The Wilcoxon's rank-sum test results for 20 jobs

EDA-MIS vs	R+	R-	Z	p Value	$\alpha=0.05$	$\alpha=0.1$
EDA	820	0	-5.511	3.57E-8	yes	yes
EDA-TS	820	0	-5.511	3.57E-8	yes	yes
EDA-iLS	820	0	-5.511	3.57E-8	yes	yes
GA-TS	674.5	145.5	-3.555	3.78E-4	yes	yes
DABC	820	0	-5.511	3.57E-8	yes	yes
SFLA	809	11	-5.363	8.18E-8	yes	yes

**Table 13** The Wilcoxon's rank-sum test results for 50 jobs

EDA-MIS vs	R+	R-	Z	p Value	$\alpha=0.05$	$\alpha=0.1$
EDA	1275	0	-6.154	7.56E-10	yes	yes
EDA-TS	1275	0	-6.154	7.55E-10	yes	yes
EDA-iLS	926.5	298.5	-3.123	1.78E-3	yes	yes
GA-TS	1263	12	-6.038	1.56E-9	yes	yes
DABC	1275	0	-6.154	7.56E-10	yes	yes
SFLA	1275	0	-6.154	7.55E-10	yes	yes

criterion of the maximum stagnation iterations in the actual problem is considered.

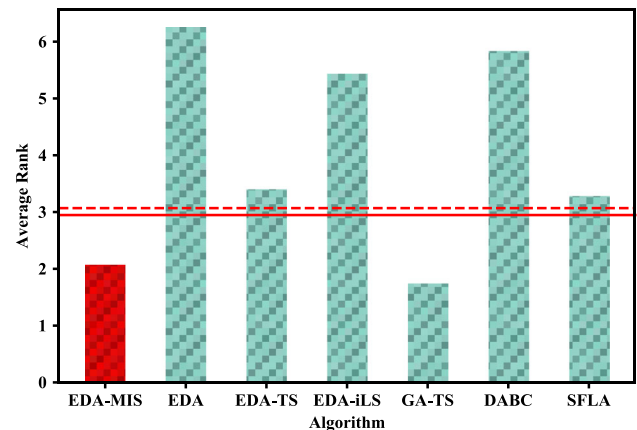
To evaluate the significance level of all algorithms, an additional Bonferroni–Dunn's procedure is applied as a post hoc procedure to calculate the critical difference for comparing their differences with  $\alpha = 0.05$  and  $\alpha = 0.1$ .

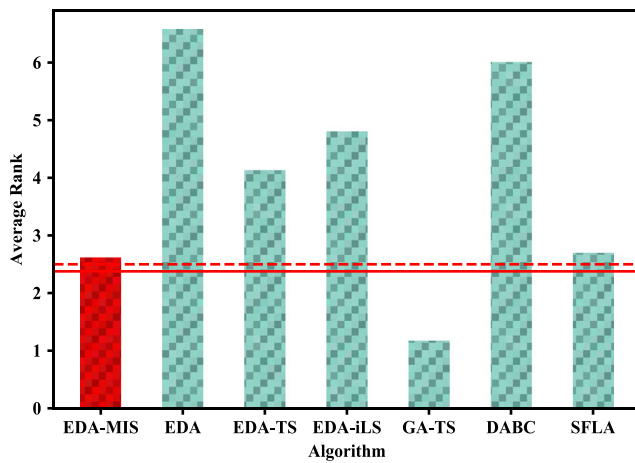
$$CD = q_{\alpha} \sqrt{\frac{k(k+1)}{6N}} \quad (15)$$

Where, the parameters  $k$  and  $N$  are the number of algorithms to be compared and the number of benchmarks, respectively. In the experimental evaluation,  $k = 5$  and  $N = 14$ . When  $\alpha = 0.05$ ,  $q_{\alpha}$  is 2.498 and  $\alpha = 0.1$ ,  $q_{\alpha}$  is 2.241 from Table B.16 (two-tailed  $\alpha(2)$ ) of [46]. Figures 10, 11, 12 and 13 illustrate the results of Bonferroni–Dunn's test that considered the EDA-MIS as a control algorithm. The numerical results of the Friedman's test are shown in Tables 14, 15, 16 and 17.

When the numbers of jobs are 5 and 10, there is no significant difference in the performance of the EDA-MIS and the GA-TS algorithms. The EDA-MIS algorithm is slightly worse than the GA-TS algorithm. However, the performance of EDA-MIS on these small-scale problems has a clear advantage over the EDA, EDA-iLS, EDA-TS, DABC and SFLA algorithms in  $\alpha = 0.05$  and  $\alpha = 0.1$ . In the case of larger-scale problems with 20 and 50 jobs, the proposed EDA-MIS algorithm has a significant advantage over the comparison algorithms. When the number of jobs is 20, the performance of EDA-MIS is better than that of GA-TS. The proposed EDA-MIS algorithm has significant advantages over EDA, EDA-iLS, EDA-TS, DABC and SFLA algorithms. When the number of jobs is 50, the proposed EDA-MIS performs better than the EDA-iLS and offers significant performance advantages compared to GA-TS, EDA, EDA-TS, DABC and SFLA algorithms.

All in all, the proposed EDA-MIS is an effective and robust algorithm for solving the TSHHFSP with SDST. The superiority of the proposed algorithm owes to the following aspects.

**Fig. 10** Results of Friedman's test of different algorithms for 5 jobs



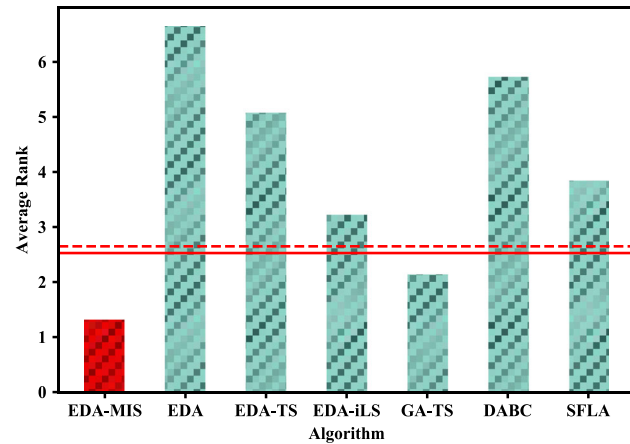
**Fig. 11** Results of Friedman's test of different algorithms for 10 jobs

**Table 14** Ranking of Friedman's test for 5 jobs

Algorithm	Mean Rank
EDA-MIS	2.08
EDA	6.25
EDA-TS	3.40
EDA-iLS	5.43
GA-TS	1.75
DABC	5.83
SFLA	3.28
CD ( $\alpha=0.05$ )	2.638
CD ( $\alpha=0.10$ )	2.394

**Table 15** Ranking of Friedman's test for 10 jobs

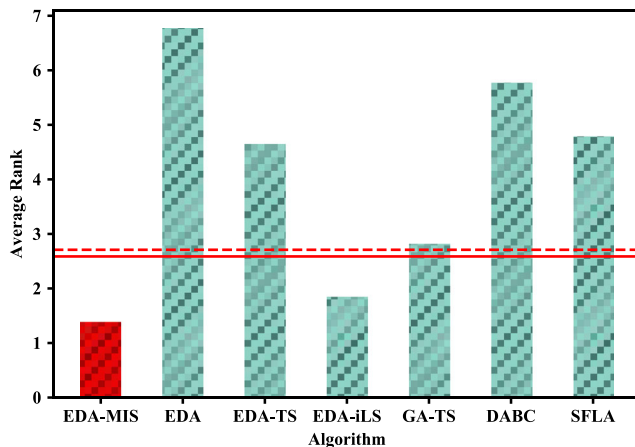
Algorithm	Mean Rank
EDA-MIS	2.62
EDA	6.57
EDA-TS	4.13
EDA-iLS	4.80
GA-TS	1.18
DABC	6.00
SFLA	2.70
CD ( $\alpha=0.05$ )	2.638
CD ( $\alpha=0.10$ )	2.394



**Fig. 12** Results of Friedman's test of different algorithms for 20 jobs

**Table 16** Ranking of Friedman's test for 20 jobs

Algorithm	Mean Rank
EDA-MIS	1.33
EDA	6.65
EDA-TS	5.08
EDA-iLS	3.23
GA-TS	2.15
DABC	5.73
SFLA	3.85
CD ( $\alpha=0.05$ )	2.638
CD ( $\alpha=0.10$ )	2.394



**Fig. 13** Results of Friedman's test of different algorithms for 50 jobs

**Table 17** Ranking of Friedman's test for 50 jobs

Algorithm	Mean Rank
EDA-MIS	1.39
EDA	6.76
EDA-TS	4.64
EDA-iLS	1.85
GA-TS	2.82
DABC	5.76
SFLA	4.78
CD ( $\alpha=0.05$ )	2.638
CD ( $\alpha=0.10$ )	2.394

(1) The self-learning probability model and the suitable updating mechanism are helpful to effectively explore the search space. (2) The multiple intensification mechanisms are helpful to reasonably intensify the qualities of solutions. (3) The suitable parameters setting is helpful to perform the EDA-MIS search.

## 6 Conclusion

This paper studies the actual production process of rare earth (RE) metals in the northwest region of China. The two-stage heterogeneous hybrid flow-shop scheduling problem (TSHHFSP) with sequence-dependent setup time (SDST) is constructed based on the analysis of the actual production process. The makespan criterion is selected as the optimized objective. The proposed model considers three sub-problems: the machining sequence of the jobs, the machine allocation of the jobs, and the constraints of the sequence-dependent setup time. In this paper, an EDA-MIS algorithm is proposed to solve the TSHHFSP problem with SDST. The performance of the proposed algorithm is compared with that of other algorithms in multiple test instances. The obtained results demonstrate that the performance of the proposed EDA-MIS algorithm is better than that of most of the comparison algorithms in small-scale problems. However, the advantages of the EDA-MIS algorithm are not significant compared with a few comparison algorithms. The performance of the EDA-MIS algorithm is superior in the larger-scale problems and has a significant advantage over all the comparison algorithms. A series of statistical analysis experiments are conducted to further demonstrate and validate the performance advantages of the proposed EDA-MIS algorithm compared with other comparison algorithms. In the evolution of the EDA-MIS algorithm, the characteristics of the TSHHFSP problem with SDST are learned by constructing the probability distribution model to optimize the solution. The combination of self-learning ability and neighborhood search improves the efficiency and accuracy of the EDA-MIS algorithm. The HFSP problem is a research direction with high research value and extensive application prospects. The HFSP problems with specific constraints deserve more research attention.

In future research, further consideration will be given to the manufacturing environment of multi-factory collaborative processing. Especially, the energy consumption and collaborative optimization of the production supply chain will be the focal concerns.

**Acknowledgments** This work was financially supported by the National Natural Science Foundation of China under grant 62063021. It was also supported by the Key talent project of Gansu Province (ZZ2021G50700016), the Key Research Programs of Science and Technology Commission Foundation of Gansu Province (21YF5WA086), Lanzhou Science Bureau project (2018-rc-98), and Project of Gansu Natural Science Foundation (21JR7RA204).

## References

1. Shao Z, Shao W, Pi D (2020) Effective constructive heuristic and metaheuristic for the distributed assembly blocking flow-shop scheduling problem. *Appl Intell* 50:4647–4669. <https://doi.org/10.1007/s10489-020-01809-x>
2. Zhao F, Xue F, Zhang Y, Ma W, Zhang C, Song H (2019) A discrete gravitational search algorithm for the blocking flow shop problem with total flow time minimization. *Appl Intell* 49:3362–3382. <https://doi.org/10.1007/s10489-019-01457-w>
3. Li H, He F, Chen Y, Pan Y (2021) MLFS-CCDE: multi-objective large-scale feature selection by cooperative coevolutionary differential evolution. *Memetic Computing* 13:1–18. <https://doi.org/10.1007/s12293-021-00328-7>
4. Zhao F, He X, Zhang Y, Lei W, Ma W, Zhang C, Song H (2020) A jigsaw puzzle inspired algorithm for solving large-scale no-wait flow shop scheduling problems. *Appl Intell* 50:87–100
5. Ruiz R, Vazquez-Rodriguez JA (2010) The hybrid flow shop scheduling problem. *Eur J Oper Res* 205:1–18. <https://doi.org/10.1016/j.ejor.2009.09.024>
6. Lin S-W, Cheng C-Y, Pourhejazy P, Ying KC, Lee CH (2021) New benchmark algorithm for hybrid flowshop scheduling with identical machines. *Expert Syst Appl* 183:1–10. <https://doi.org/10.1016/j.eswa.2021.115422>
7. Cai J, Lei D (2020) Distributed two-stage hybrid flow shop scheduling with setup times. *Comput Integr Manuf Syst* 26:2170–2179. <https://doi.org/10.13196/j.cims.2020.08.017>
8. Li J, Li W, Tao X et al (2020) A survey on time constrained hybrid flow shop scheduling problems. *Control Theory Appl* 37:2273–2290
9. Long J, Zheng Z, Gao X, Pardalos PM (2018) Scheduling a realistic hybrid flow shop with stage skipping and adjustable processing time in steel plants. *Appl Soft Comput* 64:536–549
10. Cai J, Lei D (2021) A cooperated shuffled frog-leaping algorithm for distributed energy-efficient hybrid flow shop scheduling with fuzzy processing time. *Complex Intell Syst* 7:2235–2253. <https://doi.org/10.1007/s40747-021-00400-2>
11. Li Y, Li X, Gao L, Meng L (2020) An improved artificial bee colony algorithm for distributed heterogeneous hybrid flowshop scheduling problem with sequence-dependent setup times. *Comput Ind Eng* 147: 106638. <https://doi.org/10.1016/j.cie.2020.106638>
12. Meng L, Zhang C, Shao X, Zhang B, Ren Y, Lin W (2020) More MILP models for hybrid flow shop scheduling problem and its extended problems. *Int J Prod Res* 58:3905–3930. <https://doi.org/10.1080/00207543.2019.1636324>
13. Meng L, Zhang C, Shao X, Ren Y, Ren C (2019) Mathematical modelling and optimisation of energy-conscious hybrid flow shop scheduling problem with unrelated parallel machines. *Int J Prod Res* 57:1119–1145. <https://doi.org/10.1080/00207543.2018.1501166>
14. Zhong W, Shi Y (2018) Two-stage no-wait hybrid flowshop scheduling with inter-stage flexibility. *J Comb Optim* 35:108–125. <https://doi.org/10.1007/s10878-017-0155-8>
15. Missaoui A, Boujelbene Y (2021) An effective iterated greedy algorithm for blocking hybrid flow shop problem with due date window. *Oper Res* 55:1603–1616. <https://doi.org/10.1051/ro/2021076>
16. Chamnanlor C, Sethanan K, Gen M, Chien C-F (2017) Embedding ant system in genetic algorithm for re-entrant hybrid flow shop scheduling problems with time window constraints. *J Intell Manuf* 28:1915–1931. <https://doi.org/10.1007/s10845-015-1078-9>
17. Wang S, Wang X, Yu L (2020) Two-stage no-wait hybrid flow-shop scheduling with sequence-dependent setup times. *Int J Syst Sci: Operations and Logistics* 7:291–307. <https://doi.org/10.1080/23302674.2019.1575997>
18. Wang S, Kurz M, Mason SJ, Rashidi E (2019) Two-stage hybrid flow shop batching and lot streaming with variable sublots and



- sequence-dependent setups. *Int J Prod Res* 57:6893–6907. <https://doi.org/10.1080/00207543.2019.1571251>
19. Shao W, Pi D, Shao Z (2019) A Pareto-based estimation of distribution algorithm for solving multiobjective distributed no-wait flow-shop scheduling problem with sequence-dependent setup time. *IEEE Trans Autom Sci Eng* 16:1344–1360. <https://doi.org/10.1109/TASE.2018.2886303>
  20. Li Y, Li X, Gao L, Zhang B, Pan QK, Tasgetiren MF, Meng L (2021) A discrete artificial bee colony algorithm for distributed hybrid flowshop scheduling problem with sequence-dependent setup times. *Int J Prod Res* 59:3880–3899. <https://doi.org/10.1080/00207543.2020.1753897>
  21. Shao W, Shao Z, Pi D (2020) Modeling and multi-neighborhood iterated greedy algorithm for distributed hybrid flow shop scheduling problem. *Knowl-Based Syst* 194:1–17. <https://doi.org/10.1016/j.knosys.2020.105527>
  22. Chen Y, He F, Li H, Zhang D, Wu Y (2020) A full migration BBO algorithm with enhanced population quality bounds for multimodal biomedical image registration. *Appl Soft Comput J* 93:106335
  23. Liang Y, He F, Zeng X, Luo J (2022) An improved loop subdivision to coordinate the smoothness and the number of faces via multi-objective optimization. *Integr Comput Aided Eng* 29:23–41. <https://doi.org/10.3233/ICA-210661>
  24. Liang Y, He F, Zeng X (2020) 3D mesh simplification with feature preservation based on whale optimization algorithm and differential evolution. *Integr Comput Aided Eng* 27:417–435
  25. Wang S, Liu M, Chu C (2015) A branch-and-bound algorithm for two-stage no-wait hybrid flow-shop scheduling. *Int J Prod Res* 53:1143–1167. <https://doi.org/10.1080/00207543.2014.949363>
  26. Öztöp H, Fatih Tasgetiren M, Eliyi DT, Pan Q-K (2019) Metaheuristic algorithms for the hybrid flowshop scheduling problem. *Comput Oper Res* 111:177–196. <https://doi.org/10.1016/j.cor.2019.06.009>
  27. Marichelvam MK, Geetha M, Tosun Ö (2020) An improved particle swarm optimization algorithm to solve hybrid flowshop scheduling problems with the effect of human factors – a case study. *Comput Oper Res* 114:1–9. <https://doi.org/10.1016/j.cor.2019.104812>
  28. Zhang B, Pan Q, Gao L et al (2020) A three-stage multiobjective approach based on decomposition for an energy-efficient hybrid flow shop scheduling problem. *IEEE Trans Syst Man Cybern, Syst* 50:4984–4999. <https://doi.org/10.1109/TSMC.2019.2916088>
  29. Lu C, Gao L, Pan Q, Li X, Zheng J (2019) A multi-objective cellular grey wolf optimizer for hybrid flowshop scheduling problem considering noise pollution. *Appl Soft Comput* 75:728–749. <https://doi.org/10.1016/j.asoc.2018.11.043>
  30. Shao W, Shao Z, Pi D (2021) Multi-objective evolutionary algorithm based on multiple neighborhoods local search for multi-objective distributed hybrid flow shop scheduling problem. *Expert Syst Appl* 183:1–17. <https://doi.org/10.1016/j.eswa.2021.115453>
  31. Zhang B, Pan Q-K, Meng L-L, Zhang XL, Ren YP, Li JQ, Jiang XC (2021) A collaborative variable neighborhood descent algorithm for the hybrid flowshop scheduling problem with consistent sublots. *Appl Soft Comput* 106:1–20. <https://doi.org/10.1016/j.asoc.2021.107305>
  32. Luo J, Liu Z, Xing K (2019) Hybrid branch and bound algorithms for the two-stage assembly scheduling problem with separated setup times. *Int J Prod Res* 57:1398–1412. <https://doi.org/10.1080/00207543.2018.1489156>
  33. Zhao F, Zhao J, Wang L, Tang J (2021) An optimal block knowledge driven backtracking search algorithm for distributed assembly No-wait flow shop scheduling problem. *Appl Soft Comput* 112:107750. <https://doi.org/10.1016/j.asoc.2021.107750>
  34. Wu C, Wang L, Wang J (2021) A path relinking enhanced estimation of distribution algorithm for direct acyclic graph task scheduling problem. *Knowl-Based Syst* 228:107255. <https://doi.org/10.1016/j.knosys.2021.107255>
  35. Sun Z, Gu X (2017) A novel hybrid estimation of distribution algorithm for solving hybrid flowshop scheduling problem with unrelated parallel machine. *J Cent S Univ Technol* 24:1779–1788
  36. Du Y, Li J, Luo C, Meng L (2021) A hybrid estimation of distribution algorithm for distributed flexible job shop scheduling with crane transportations. *Swarm Evol Comput* 62:1–24. <https://doi.org/10.1016/j.swevo.2021.100861>
  37. Vignier A, Billaut JC, Proust C (1999) Hybrid flowshop scheduling problems: state of the art
  38. Graham RL, Lawler EL, Lenstra JK, Kan A (1979) Optimization and approximation in deterministic sequencing and scheduling: a survey - *ScienceDirect*. *Anna Discrete Math* 5:287–326
  39. Gupta JND (1988) Two-stage, hybrid Flowshop scheduling problem. *Oper Res* 39:359–364
  40. Liang JJ, Qu BY, Suganthan PN, Hernández-Díaz AG (2013) Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization. Technical Report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore
  41. Wang S, Wang L, Liu M, Xu Y (2015) An order-based estimation of distribution algorithm for stochastic hybrid flow-shop scheduling problem. *Int J Comput Integr Manuf* 28:307–320. <https://doi.org/10.1080/0951192x.2014.880803>
  42. Wang L, Wang S, Liu M (2013) A Pareto-based estimation of distribution algorithm for the multi-objective flexible job-shop scheduling problem. *Int J Prod Res* 51:3574–3592. <https://doi.org/10.1080/00207543.2012.752588>
  43. Zhou S, Li X, Chen H, Guo C (2016) Minimizing makespan in a no-wait flowshop with two batch processing machines using estimation of distribution algorithm. *Int J Prod Res* 54:4919–4937. <https://doi.org/10.1080/00207543.2016.1140920>
  44. Cui Z, Gu X (2015) An improved discrete artificial bee colony algorithm to minimize the makespan on hybrid flow shop problems. *Neurocomputing* 148:248–259. <https://doi.org/10.1016/j.neucom.2013.07.056>
  45. Umam MS, Mustafid M, Suryono S (2021) A hybrid genetic algorithm and tabu search for minimizing makespan in flow shop scheduling problem. *J King Saud Univ Comput Inf Sci*. <https://doi.org/10.1016/j.jksuci.2021.08.025>
  46. Feinleib M, Zar J (1975) Biostatistical analysis. *J Am Stat Assoc* 70:257. <https://doi.org/10.2307/2285423>

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Huan Liu** received B.Eng. degree from Anhui University and M.Eng. degree from Lanzhou University of Technology, in 2011 and 2014, respectively. Since 2015, he has been with the College of Information Science and Technology, Gansu Agricultural University, Lanzhou, China. Now he is a Ph.D. student at Lanzhou University of Technology. His current research interests include intelligent optimization and scheduling.

**Fuqing Zhao** received the B.Sc. and Ph.D. degrees from the Lanzhou University of Technology, Lanzhou, China, in 1994 and 2006, respectively. Since 1998, he has been with the School of Computer Science Department, Lanzhou University of Technology, Lanzhou, China, where he became a Full Professor in 2012. He has been as the post Doctor with the State Key Laboratory of Manufacturing System Engineering, Xi'an Jiaotong University, Xi'an, China in 2009. He has authored two academic book and over 50 refereed papers. His current research interests include intelligent optimization and scheduling.