A Two-Stage Cooperative Evolutionary Algorithm With Problem-Specific Knowledge for Energy-Efficient Scheduling of No-Wait Flow-Shop Problem

Fuqing Zhao, Xuan He, and Ling Wang

***Abstract*—Green scheduling in the manufacturing industry has attracted increasing attention in academic research and industrial applications with a focus on energy saving. As a typical scheduling problem, the no-wait flow-shop scheduling has been extensively studied due to its wide industrial applications. However, energy consumption is usually ignored in the study of typical scheduling problems. In this article, a two-stage cooperative evolutionary algorithm with problem-specific knowledge called TS-CEA is proposed to address energy-efficient scheduling of the no-wait flow-shop problem (EENWFSP) with the criteria of minimizing both makespan and total energy consumption. In TS-CEA, two constructive heuristics are designed to generate a desirable initial solution after analyzing the properties of the problem. In the first stage of TS-CEA, an iterative local search strategy (ILS) is employed to explore potential extreme solutions. Moreover, a hybrid neighborhood structure is designed to improve the quality of the solution. In the second stage of TS-CEA, a mutation strategy based on critical path knowledge is proposed to extend the extreme solutions to the Pareto front. Moreover, a co-evolutionary closed-loop system is generated with ILS and mutation strategies in the iteration process. Numerical results demonstrate the effectiveness and efficiency of TS-CEA in solving the EENWFSP.**

***Index Terms*—Cooperative algorithm, energy efficient, knowledge, no-wait flow-shop scheduling, total energy consumption (TEC).**

Manuscript received April 21, 2020; revised July 19, 2020; accepted September 18, 2020. Date of publication October 23, 2020; date of current version November 9, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 62063021 and Grant 61873328; in part by the Key Research Programs of Science and Technology Commission Foundation of Gansu Province under Grant 2017GS10817; in part by the Lanzhou Science Bureau Project under Grant 2018-rc-98; in part by the Public Welfare Project of Zhejiang Natural Science Foundation under Grant LGJ19E050001; and in part by the Wenzhou Public Welfare Science and Technology Project under Grant G20170016. This article was recommended by Associate Editor L. Tang. *(Corresponding authors: Fuqing Zhao; Ling Wang.)*

Fuqing Zhao and Xuan He are with the School of Computer and Communication Technology, Lanzhou University of Technology, Lanzhou 730050, China (e-mail: fzhao2000@hotmail.com; 2369084655@qq.com).

Ling Wang is with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: wangling@tsinghua.edu.cn).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2020.3025662.

Digital Object Identifier 10.1109/TCYB.2020.3025662

I. Introduction

A. Background and Motivation

SCHEDULING plays a pivotal role in manufacturing systems. The flow-shop scheduling problem (FSP) is a commonly used model in manufacturing systems and has been a hot research topic. The massive real-world applications can be modeled as flow-shop scheduling problems [1]. According to different constraints, the flow-shop problems can be divided into blocking flow shop [2], no-wait flow shop [3], no-idle flow shop [4], distributed flow-shop [5], [6], etc. The assumptions widely used in the FSP are as follows.

1) The processing path of each job is the same and not allowed to change.
2) At one time, each machine only processes one operation, and the operation is not allowed to be interrupted.
3) A job is not allowed to be processed on different machines at one time.
4) The setup time of the operation is negligible or included in the processing time.

The no-wait flow-shop scheduling problem (NWFSP) widely exists in various industrial systems [7], such as steelmaking, food processing, chemical industry, and pharmaceuticals. In the NWFSP, once a job starts to be processed, the operations of the job are not allowed to be interrupted. The objective of classical NWFSP is to minimize completion time (makespan) by arranging all the jobs reasonably. As shown in Fig. 1, in the steel making-continuous casting production process of iron and steel enterprises, the temperature of molten steel is not allowed to decrease quickly to conduct the hot-feeding and hot-packing processes. So, the steel making continuous casting production process of iron and steel is a no-wait scheduling process with the corresponding operation modes under different power modes. The processing procedure may vary with the type of production. Those operations that do not go through the assembly line are regarded as passing without any operation, that is, the processing time of operations is zero.

The NWFSPs with the objectives of production efficiency have been extensively studied. However, objectives related to energy efficiency are rarely considered in NWFSP [8]. As we know, the energy crisis has become one of the most serious and urgent issues in the world [6]. The huge consumption of nonrenewable resources will lead to energy depletion and

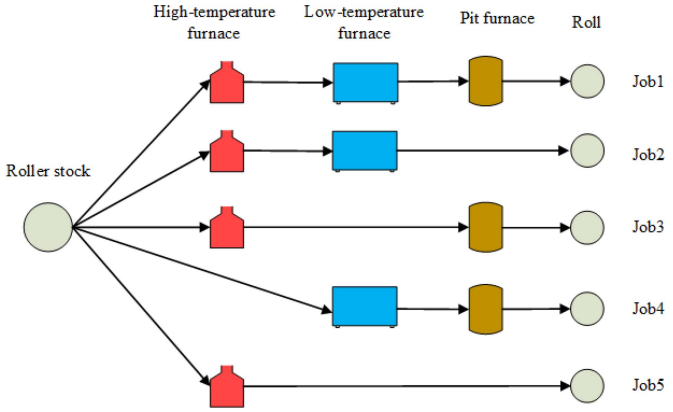

Fig. 1. Roll stock heat treatment process.

the greenhouse effect. Energy consumption in industrial manufacturing systems accounts for more than half of the total energy consumption (TEC) in the world. In many situations, machines in factories may operate with different processing modes, and the energy consumption of machines is different and the impact on the manufacturing process is different as well. Under the pressure of public awareness of sustainable development and the competition of the market, manufacturing enterprises are driven to pay great attention to reduce energy consumption [9]–[12]. So it is extraordinarily important for manufacturing enterprises to develop effective energy-efficient measures and technologies including scheduling approaches [13], [14].

The classical NWFSP has been proved to be NP-hard [15] according to computational complexity. The size of the solution space of NWFSP is $n!$, where n is the number of jobs. The search efficiency of the mathematical programming methods that implicitly enumerate the entire space is rather low. Although such methods can obtain an optimal solution theoretically, it is difficult to solve the large-scale scheduling problems in a reasonable time. For the energy-efficient scheduling of the no-wait flow-shop problem (EENWFSP), both the economic and environmental criteria should be considered simultaneously, which is a large-scale multiobjective optimization problem (MOP). The size of the solution space of EENWFSP is $n!\cdot x^{n\cdot m}$, where x is the number of speed levels of the machines, n is the number of jobs, and m is the number of machines. Clearly, it is more difficult to solve EENWFSP than classical NWFSP.

B. Literature Review

In [16], heuristics for NWFSP were roughly divided into two categories: 1) constructive heuristics and 2) metaheuristics. In [17], an average departure time (ADT) heuristic was presented to minimize makespan for NWFSP. The experimental results showed that the ADT heuristic outperformed three existing state-of-the-art heuristics with the same computational effort. In [18], an average idle time heuristic called AIT was presented to minimize makespan for NWFSP. First, the current waiting time and future waiting time of machines were utilized to construct an initial scheduling sequence. Then, the insert and swap operations were employed to enhance the local search of AIT. In [3], a modified iterated greedy algorithm was presented for the mixed no-wait flow-shop problems. The experimental results showed that the algorithm yielded the best performance among all the methods for comparison. In [16], the NWFSP was converted to an asymmetric traveling salesman problem (ATSP), and then two metaheuristics were proposed to solve the ATSP. The optimal solutions of the small-scale NWFSP were obtained by the Gurobi optimizer, and good results were obtained by the two metaheuristics for the large-scale problems with makespan criterion.

For metaheuristics, various research works were implemented with different search frameworks to solve the NWFSP [19], [20]. Inspired by the process of teaching–learning, an extended framework of metaheuristic was presented to solve the NWFSP [21]. An effective new hybrid ant colony algorithm called HACO based on crossover and mutation was proposed for NWFSP with the makespan criterion [22]. According to the theory of shallow water wave, two discrete water wave optimization algorithms were proposed to solve the NWFSP [23], [24]. The local search algorithms have also gained much attention. The search process of a local search method depends on neighborhood structures and environment selection rules [25]. For the greedy local search, the current solution is updated by the best neighbor solution. The search process conducts until no better neighbor solution can be found. Some techniques are used to avoid being trapped in local optima. For instance, restarting the search process (iterated local search) [26], jumping with a certain probability (simulated annealing) [27], and using history information to avoid revisit (tabu search) [21].

In recent years, the research of green scheduling in manufacturing has gained increasing attention [6], [28]–[30]. A multiobjective method based on decomposition was implemented by Jiang and Wang [14] to solve the energy-efficient permutation flow-shop scheduling problem (EEPFSP). A multilevel algorithm for energy-efficient flexible flow-shop scheduling was proposed by Yan *et al.* [31] to investigate the potential energy saving in the management of shop floor. A hybrid backtracking search algorithm called HMOBSA was proposed by Lu *et al.* [32] to address EEPFSP from a real-world manufacturing enterprise. In [29], an ant colony system-based approach was presented to achieve the goals of the placement of virtual machine and energy efficiency in cloud computing. An adaptive multiobjective variable neighborhood search (AM-VNS) algorithm was proposed to solve the energy-efficient no-wait permutation flow-shop problem [33] by adopting an adaptive mechanism to dynamically determine which structure was selected to handle the current solution. Chen *et al.* [6] proposed a collaborative optimization algorithm (COA) for solving the energy-efficient scheduling of distributed no-idle permutation flow-shop problem, in which multiple search strategies collaborated in a competitive way to improve the capability of search.

Considering the constraint that TEC cannot exceed a given threshold, Lei *et al.* [34] proposed a two-phase metaheuristic to solve the flexible job shop scheduling problem with consideration of the makespan and total tardiness. In [35],

an energy-aware multiobjective optimization algorithm was proposed for solving the energy-efficient scheduling of hybrid flow-shop scheduling problem. Recently, Zheng *et al.* [28] proposed a hybrid ant colony optimization algorithm to solve the blocking permutation flow-shop scheduling problem with the objectives of minimizing makespan and total energy costs. A large amount of literature about energy-efficient scheduling for intelligent production systems has been published [30], [36]–[39]. Since the EENWFSP considers both the economic and environmental criteria simultaneously, it is a complex MOP. For most realistic problems, they are large scale with huge search space. Moreover, the considered multiple objectives are conflicting. So, it is full of challenge to solve the large-scale MOPs [40]–[42].

C. Contribution

Inspired by [43] and [44], a two-stage cooperative evolutionary algorithm with problem-specific knowledge (TS-CEA) is proposed in this article to solve the EENWFSP. Considering a speed scaling strategy, machines may operate at three speed levels (1: fast, 2: normal, and 3: slow speed levels). According to [45], different constructive heuristic algorithms are suitable for scheduling problems of different scales. So, both extended energy-efficient Nawaz–Enscore–Ham (NEH) heuristic and Jigsaw puzzle inspired algorithm (JPA) are used to produce initial solutions.

Moreover, the evolution process of the co-evolutionary algorithm is divided into two stages. In the first stage, the processing speed of machines keeps constant, and the local search strategy (ILS) is used to obtain a potential job scheduling sequence. At this stage, the smaller makespan of the solution is, the smaller the TEC of the solution is. Therefore, EENWFSP is converted to a single-objective optimization problem. In the second stage, the scheduling sequence of the jobs remains unchanged. A mutation strategy based on critical path knowledge is used to change the processing speed of the machine to obtain potential nondominated solutions. At this stage, the EENWFSP is a MOP. A scheduling sequence of jobs is provided by ILS for the mutation strategy, and the processing speed of the machines is provided by the mutation strategy for ILS. Therefore, a co-evolutionary closed-loop system is formed by the operators of stage 1 and stage 2 of TS-CEA.

The main contributions of this article are summarized as follows.

1) Formulate the considered EENWFSP as a mixed-integer linear programming (MILP) model.
2) A two-stage co-evolutionary algorithm with problem-specific knowledge that can reduce the solution space is proposed to solve EENWFSP.
3) According to the problem characteristics, some properties of EENWFSP are derived to obtain the knowledge to guide the design of the mutation strategy of machine processing speed.
4) The EENWFSP is converted to a special case of the $(n+1)$-city ATSP, where the distance between cities is variable. The effectiveness of the proposed model and algorithm is verified.

The remainder of this article is organized as follows. The formulation of EENWFSP is described in Section II. The framework of TS-CEA and its detailed implementation is presented in Section III. The verification of the proposed model is presented in Section IV. The computational results of TS-CEA are provided in Section V. The comparisons between TS-CEA and the state-of-the-art algorithms are given in Section VI. Finally, the conclusions and future work are summarized in Section VII.

II. Problem Description and Mathematical Model

In this article, the notation is described as follows.

N	Set of jobs $N = \{1, 2, \ldots, n\}$.
M	Set of machines $M = \{1, 2, \ldots, m\}$.
$p_{j,i}$	Standard processing time of job j on machine i.
v_l	Speed factor of speed level $l \in L$, $L = \{1, 2, 3\}$.
EC	Energy consumption.
φ_i	EC of the machine i at standby mode per unit time $i \in M$ (kW).
$\tau_{i,l}$	EC per unit time of machine i running at speed l, $l \in L$, $i \in M$ (kW).
D	Very large number.
$y_{j,i,l}$	1 if job j is processed at speed l on machine i; 0, otherwise.
$x_{j,k}$	1 if job j precedes job k; 0 otherwise.
$C_{j,i}$	Completion time of job j on machine i.
θ_i	Idle time on machine i.
C_{max}	Maximum completion time (makespan).
TEC	TEC (kWh).

The EENWFSP is considered to obtain a feasible schedule that minimizes the two conflicting objectives, that is, makespan and TEC. Unlike the classical NWFSP, a discrete set of l different processing speed levels is provided for each machine. Therefore, the processing time of jobs is different according to the chosen speed level. The proposed MILP is described as follows:

$$\text{Minimize}(C_{\max}) = \min\left(C_{n,m}\right) \tag{1}$$

$$\text{Minimize TEC} \tag{2}$$

$$C_{j,1} \geq \sum_{l\in L} \frac{p_{j,1}y_{j,1,l}}{v_l} \tag{3}$$

$$C_{j,i} - C_{j,i-1} \geq \sum_{l\in L} \frac{p_{j,i}y_{j,i,l}}{v_l} \quad \forall\, j \in N \quad \forall\, i \in M : i \geq 2 \tag{4}$$

$$C_{j,i} - C_{k,i} + Dx_{j,k} \geq \sum_{l\in L} \frac{p_{j,i}y_{j,i,l}}{v_l} \quad \forall j, k \in N : k \neq j \;\forall\; i \in M \tag{5}$$

$$C_{j,i} - C_{k,i} + Dx_{j,k} \leq D - \sum_{l\in L} \frac{p_{j,i}y_{j,i,l}}{v_l} \times \forall\, j, k \in N : k \neq j \;\forall\; i \in M \tag{6}$$

$$C_{\max} \geq C_{j,m} \quad \forall\, j \in N \tag{7}$$

$$C_{j,i} - C_{j,i-1} \leq \sum_{l\in L} \frac{p_{j,i}y_{j,i,l}}{v_l} \quad \forall\, j \in N \;\forall\; i \in M,\; i \geq 2 \tag{8}$$

$$\sum_{l\in L} y_{j,i,l} = 1; \quad \forall\, j \in N \;\forall\; i \in M \tag{9}$$

TABLE I
ENERGY-RELATED PARAMETERS

l	v_l	$\tau_{j,l}$	θ_j
1	1.2	1.5	
2	1	1	0.05
3	0.8	0.6	

$$\text{TEC} = \sum_{j\in n}\sum_{i\in m}\sum_{l\in L}\frac{p_{j,i}\tau_{i,l}}{v_l}y_{j,i,l} + \sum_{i\in M}\varphi_i\theta_i \tag{10}$$

$$\theta_i = C_{\max} - \sum_{j\in n}\sum_{l\in L}\frac{p_{j,i}y_{j,i,l}}{v_l} \quad \forall\, i \in M \tag{11}$$

$$y_{j,i,l} \in \{0,1\}, C_{j,i} \geq 0 \quad \forall\, j \in N\ \forall\, i \in M\ \forall\, l \in L$$
$$x_{j,k} \in \{0,1\} \quad \forall\, j,k \in N, k \neq j. \tag{12}$$

Objectives are defined as (1) and (2). Constraint (3) implies that the completion time of the jobs on machine 1 is not less than the processing time of the jobs on that machine 1. Constraint set (4) means that the job j is started to process on the machine i only after its preceding operation on previous machine $i-1$ is finished. Constraint sets (5) and (6) guarantee that the relative position of the job j and job k in the scheduling sequence is unique. The makespan is calculated by (7). Constraint sets (8) and (9) ensure that the job j is processed on the machine i, and only one speed level is selected, and all the operations of the job j on all machines are not allow to be interrupted. The TEC is calculated by (10). The idle time on each machine is calculated by (11). All the decision variables are defined as (12).

An example is presented as follows to illustrate the EENWFSP. The energy-related parameters used in the calculation of TEC are adopted from [46] as shown in Table I.

Suppose the speed level matrix SLM and the standard processing time matrix P are as follows:

$$P = \begin{pmatrix} 2.4 & 3 & 4 \\ 3 & 4.8 & 6 \\ 6 & 6 & 4 \end{pmatrix} \quad SLM = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 1 & 1 & 2 \end{pmatrix}.$$

Then, the speed matrix v and unit time energy matrix τ are as follows:

$$v = \begin{pmatrix} 1.2 & 1 & 0.8 \\ 1 & 0.8 & 1.2 \\ 1.2 & 1.2 & 1 \end{pmatrix} \tau = \begin{pmatrix} 1.5 & 1 & 0.6 \\ 1 & 0.6 & 1.5 \\ 1.5 & 1.5 & 1 \end{pmatrix}.$$

The actual processing time matrix T is calculated as follows:

$$T = P/v = \begin{pmatrix} 2 & 3 & 5 \\ 3 & 6 & 5 \\ 5 & 5 & 4 \end{pmatrix}.$$

Consider a job processing sequence $\pi = \{1,2,3\}$. Thus, $C_{\max}$ is 20. As shown in Fig. 2, the processing time energy consumption of machines is calculated as follows:

$$\text{TEC}_{\text{processing}} = \text{sum}(T \cdot \tau) = 42.1.$$

The standby time energy consumption of machines is calculated as follows:

$$\text{TEC}_{\text{standby}} = (C_{\max} \cdot m - \text{sum}(T)) \cdot \varphi_j = 1.1.$$

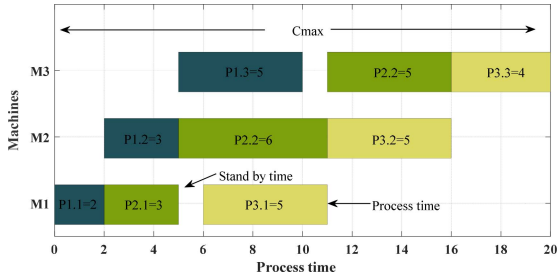

Fig. 2. Gantt chart with $n = 3$ and $m = 3$.

Algorithm 1 Procedure of NN + MNEH

1 **Inputs:** The standard processing time matrix P, the speed level matrix SLM, the population size NP
2 **Initialize:** The best solution $\pi_{\min} = \{\}$, the actual processing time matrix T.
3 Pick up NP job pairs with minimized makespan from a set $s_0 = \{J_1, J_2, \cdots J_n\}$ and construct a sequence $JF_s.//JF_s$ is the set of the first and second jobs.
4 **For** $k = 1{:}NP$ **do**
5 $s_1 = s_0 - JF_k$
6 $\pi_k = JF_k + MNEH(s_1)$
7 **End for**
8 **Output:** the best solution $\pi_{\min}$ with the smallest makespan.

Algorithm 2 Procedure of JPA

1 **Inputs:** The standard processing time matrix P;
The speed level matrix SLM.
2 **Initialize:** The best block $\pi_{\min} = \{\}$, the actual processing time matrix T.
3 Calculate the stand-by time of machine (WT) between two jobs;
4 Jobs is sorted according to the value of WT from small to large;
5 **For** $i = 1{:}n$ **do**
6 Taking the job J_i as the head job;
7 According to the value of WT, the unscheduled jobs are sequentially linked behind the job J_i. The scheduling sequence π is obtained.
8 Calculate $C_{\max}(\pi)$;
9 **End for**
10 **Output:** the best solution π_{min} with the smallest makespan.

Therefore, the TEC of machines is calculated as follows:

$$\text{TEC} = \text{TEC}_{\text{processing}} + \text{TEC}_{\text{standby}} = 43.2.$$

III. DESCRIPTION OF TS-CEA

A. Initialization

Since the initial v is generated randomly and remained unchanged, the smaller standby time of the machine is, the smaller the energy consumption of the machine is. Moreover, the smaller makespan of the solution is, the smaller the TEC of the solution is. So, different indicators may be used to design different constructive initialization methods.

Here, two heuristics are proposed for TS-CEA to generate initial solutions, focusing on different subproblems ($C_{\max}$ and TEC). As we know, NEH is one of the most popular constructive heuristics for NWFSP with the makespan criterion. The extended NEH by [23] (NN + MNEH) is more effective than NEH by considering the arrangement and combination of the first and second jobs. Inspired by such an idea, the NN + MNEH is used to generate certain initial solutions for solving EENWFSP. The description of NN + MNEH is shown in Algorithm 1.

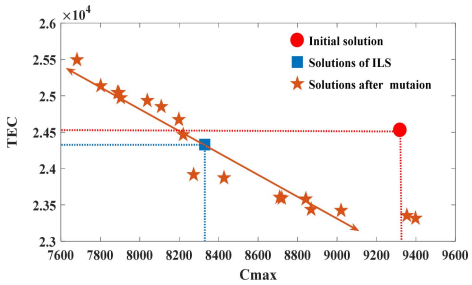

Fig. 3. Operating mechanism of TS-CEA.

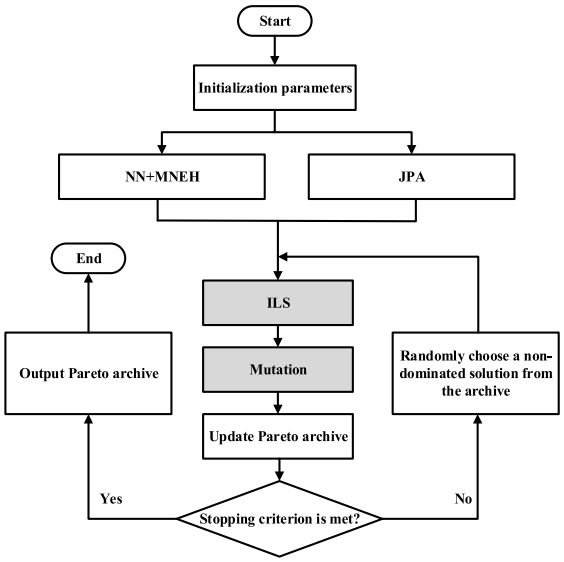

Fig. 4. Flowchart of TS-CEA.

Aiming at reducing TEC, JPA is used to generate an initial solution. As shown in Algorithm 2, the standby time of machines is used as an indicator to construct the initial solution. The core idea of JPA is to find the best matching job for each job and then construct a complete scheduling solution in an iteratively incremental manner. The JPA constructs a solution with fast speed and the result of each run is unique.

B. Cooperative Evolutionary Algorithm With Problem-Specific Knowledge

According to the principle of a single variable, the evolution process of the proposed cooperative evolution algorithm is divided into two stages.

As shown in Fig. 3, in the first stage, the processing speed of machines remains unchanged. The smaller the makespan is, the smaller the TEC is. Therefore, the EENWFSP is a single-objective optimization problem. An iterative ILS is used to explore the feasible promising solutions while optimizing makespan and TEC (the movement from the red circle to the blue square).

In the second stage, the scheduling sequence of the jobs remains unchanged. The mutation strategy is employed to change the processing speed of the machine to obtain the potential nondominated solution of EENWFSP (the diffusion process from the blue square to brown five-pointed star). The flowchart of the proposed TS-CEA algorithm is shown in Fig. 4.

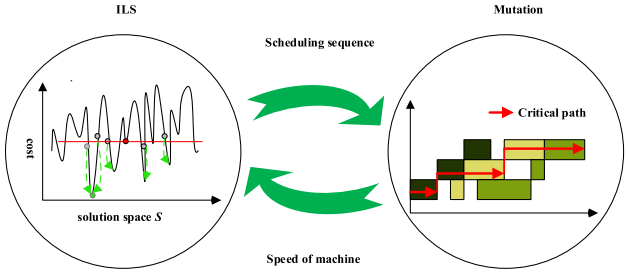

Fig. 5. Module interaction diagram.

Algorithm 3 Procedure of ILS

```
Input: π0=constructive heuristic NN + MNEH and JPA
While stop condition not met do
// The perturbation procedure
π = perturbation (π0)
// The local search procedure
π1 = insert (π)
π2 = block neighborhood search (π1)
// The acceptance criterion
   If Cmax(π2) < Cmax(π0)
      π0 = π2
   End if
End while
Output: π0//best solution so far
```

As shown in Fig. 5, a scheduling sequence of jobs is provided by ILS for the mutation strategy, and the mutation strategy provides the processing speed of the machines for ILS. Therefore, a co-evolutionary closed-loop system is generated with ILS and mutation strategies in the iteration process.

C. Iterative Local Search

As a simple and powerful search framework, the iterative local search (ILS) is a trajectory-based metaheuristic. Only one solution is manipulated in the iterative local search, which consists of initialization, local search, perturbation, and acceptance criterion. The better solution of the NEH and JPA, which has the smallest makespan and TEC, was chosen as the initial solution for ILS.

The local search operator is assisted by the perturbation operator to escape from the current local optimum. The perturbation operator generates a perturbation solution of the current local optimum and continues to explore from the perturbation solution. Here, a perturbed solution is generated by performing *SP* times random insertion operation to the current local optimum. A simple variable neighborhood search mechanism, including insertion and block-shift operations (the length of block len $\in [1, n/2]$) [27], is used in the local search operator. Since the processing speed of the machine has certain randomness, the greedy preferential selection strategy is used as the acceptance criterion. The procedure of the proposed ILS is given in Algorithm 3.

D. Critical Path Knowledge-Based Mutation Strategy

In the second stage of TS-CEA, the scheduling sequence of the jobs is kept unchanged, and the processing speed of the machines is changed to detect potential nondominated solutions. Here, certain attributes of the EENWFSP are extracted

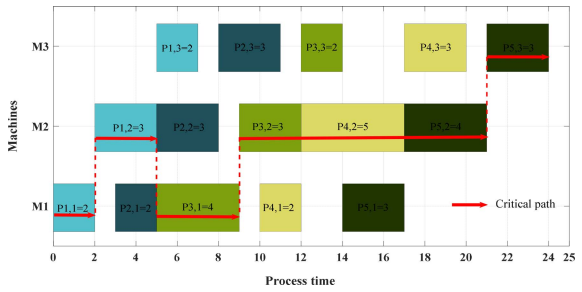

Fig. 6. Illustration of critical path.

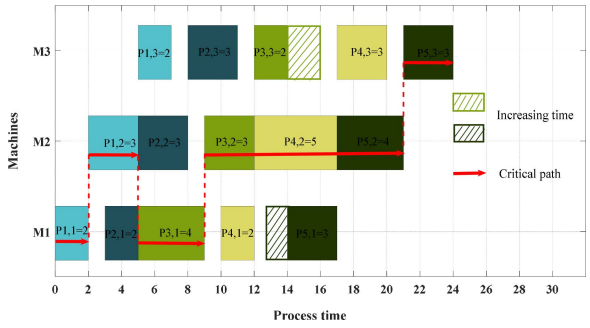

Fig. 7. Reduce the processing speed of jobs on noncritical paths.

and used as prior knowledge to guide the design of a mutation operator.

Definition: For NWFSP, the length of the critical path is equal to the makespan of a solution. The critical path is the continuous job path from the beginning of the entire process to the completion of the last job with no standby time between any two jobs, as shown in Fig. 6.

Property 1: If two schedules $\boldsymbol{a} = (\boldsymbol{\pi}_1, \boldsymbol{v}_1)$ and $\boldsymbol{b} = (\boldsymbol{\pi}_2, \boldsymbol{v}_2)$ satisfy $C_{\max}(\boldsymbol{a}) = C_{\max}(\boldsymbol{b})$ and $\mathrm{TEC}(\boldsymbol{a}) < \mathrm{TEC}(\boldsymbol{b})$, then $\boldsymbol{a} \succ \boldsymbol{b}$.

Property 2: Given a scheduling solution, increasing the processing speed of the jobs on the noncritical path without affecting the critical path, the makespan of the solution is unchanged, and energy consumption of the solution is increased.

In general, the makespan is reduced and the TEC is increased if the processing speed of the jobs on the critical path is increased. The makespan is increased and the TEC is reduced if the processing speed of the jobs on the critical path is reduced. According to Properties 1 and 2, given a scheduling solution $\boldsymbol{a} = (\boldsymbol{\pi}_1, \boldsymbol{v}_1)$, decreasing the speeds of the noncritical operations without deteriorating the makespan of the solution, TEC of the solution can be reduced. That is, a better scheduling solution $\boldsymbol{b} = (\boldsymbol{\pi}_1, \boldsymbol{v}_2)$ is obtained and $\boldsymbol{b} \succ \boldsymbol{a}$, as shown in Fig. 7. A mutation strategy is proposed to detect potential nondominated solutions as follows. An operation is chosen randomly. If the operation is on a critical path, the processing speed of the operation is increased or reduced randomly. Otherwise, the processing speed of the operation is reduced.

IV. Model Verification

In this article, the constraints of energy efficiency are considered on the classical no-wait flow-shop problem. The EENWFSP is mathematically modeled by using the MILP. The CPLEX optimizer is used to obtain the optimal solution

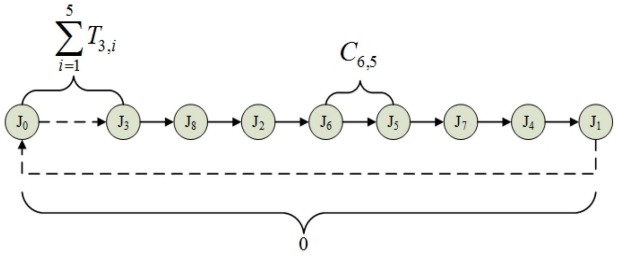

Fig. 8. Directed Hamiltonian cycle of a scheduling sequence.

for the model. The model is correct if the CPLEX solution is the same as heuristic solution for small-scale instances. First, the EENWFSP is transformed into a special case of the $(n+1)$-city ATSP, where the distance between cities is variable. The distance between any two cities/jobs is calculated as follows:

$$T_{j,i} = P_{j,i}/v_{j,i} \tag{13}$$

$$C_{j,j'} = \begin{cases} \sum_{i=1}^{m} T_{j',i}, & \text{if } j = 0 \\ 0, & \text{if } j' = 0 \\ C_{j'} - C_j \\ \quad = \max\limits_{1 \le i \le m} \left\{0, \sum_{i'=i}^{m} \left(T_{j',i'} - T_{j,i'}\right) + T_{j,i'}\right\}, & \text{otherwise} \end{cases} \tag{14}$$

where J_0 is a dummy job with zero-processing time on all machines. Let $\pi = \{J_3, J_8, J_2, J_6, J_5, J_7, J_4, J_1\}$ be a scheduling sequence. The corresponding directed Hamiltonian cycle of the ATSP is as follows.

Furthermore, the transformed ATSP is mathematically modeled by using the binary integer programming (BIP) as follows:

$$\text{Minimize} \quad C_{\max} = \sum_{j=0}^{n} \sum_{j'=0}^{n} C_{j,j'} x_{jj'} \tag{15}$$

$$\text{Minimize TEC} \tag{16}$$

$$\sum_{\substack{j=0 \\ j \neq j'}}^{n} x_{jj'} = 1, j' = 0, 1, \ldots, n \tag{17}$$

$$\sum_{\substack{j'=0 \\ j' \neq j}}^{n} x_{jj'} = 1, j = 0, 1, \ldots, n \tag{18}$$

$$X = \{x_{jj'}\} \in S \tag{19}$$

$$S = \left\{ x_{jj'} \Big| \sum_{j \in Q} \sum_{j' \notin Q}^{n} x_{jj'} = 1 \right. \\ \left. Q \text{ is a nonempty proper subset of } n \text{ jobs} \right\} \tag{20}$$

$$\theta_i = C_{max} - \sum_{j \in n} T_{j,i} \tag{21}$$

$$\mathrm{TEC} = \sum_{j \in n} \sum_{i \in m} \sum_{l \in L} T_{j,i} \cdot \tau_{i,l} \cdot y_{j,i,l} + \sum_{i \in M} \varphi_i \theta_i \tag{22}$$

$$x_{jj'} \in \{0, 1\} \quad \forall j, j' = 0, 1, \ldots, n, \text{ and } j \neq j', y_{j,i,l} \in \{0, 1\} \\ \forall\, l \in L. \tag{23}$$

TABLE II
PROCESSING TIMES OF A 8 × 5 INSTANCE

	M_1	M_2	M_3	M_4	M_5
J_1	54	79	16	66	58
J_2	83	3	89	58	56
J_3	15	11	49	31	20
J_4	71	99	15	68	85
J_5	77	56	89	78	53
J_6	36	70	45	91	35
J_7	53	99	60	13	53
J_8	38	60	23	59	41

The makespan and TEC are defined as (15) and (16), respectively. Constraint set (17) ensures that only one city/job is directly visited before city/job $j'(j' = 0, 1, \ldots, \mathrm{n})$. Constraint set (18) ensures that only one city/job is directly visited after city/job $j(j = 0, 1, \ldots, n)$. Constraints (19) and (20) prevent from subtours. The TEC is calculated by (21) and (22). Constraint set (23) defines the decision variable $x_{jj'}$ and $y_{j,i,l}$ as binary.

Then, the branch and bound algorithm (BB) is used to solve the above BIP model. BB is implemented by using the CPLEX-12.5 optimizer. BB is a type of mathematical programming method that implicitly enumerates the entire space. So, its search efficiency is low. Although it can theoretically provide the optimal solution to the ATSP, it is difficult to solve large-scale scheduling problems in a reasonable time.

To illustrate the correctness of the EENWFSP model, an 8-job 5-machine instance from the Taillard benchmark problem set is provided in Table II. The Taillard benchmark set comes from real-life flow-shop scheduling problems.

Such an 8-job 5-machine instance in the EENWFSP, the number of solutions is $(8!)*3^{8*5} = 4.9020E + 23$. Just the number of speed matrices is $3^{8*5} = 1.2158E + 19$. Therefore, it is impossible to enumerate all solutions. Here, 10 000 speed matrices are randomly sampled. Then, BB is used to obtain the approximate Pareto front of EENWFSP. The calculation results of BB and TS-CEA are shown in Fig. 9.

As shown in Fig. 9, when Cmax(TS-CEA) = Cmax(BB), TEC(TS-CEA)<TEC(BB). It just illustrates the correctness of the proposed model and algorithm. Because the knowledge of the critical path is not used in the BB algorithm, the solution obtained by TS-CEA dominates the solution obtained by BB. However, the knowledge based on the critical path is added to the BB algorithm (BB-V1), the result obtained by BB-V1 and TS-CEA is the same, as shown in Fig. 10. So, the proposed EENWFSP model is correct and the corresponding algorithm is effective for small-scale problems.

V. NUMERICAL RESULTS OF TS-CEA

In this section, 120 benchmark instances, where $n = \{20, 50, 100, 200, 500\}$, $m = \{5, 10, 20\}$, are used to test the performance of TS-CEA in solving EENWFSP [47]. The benchmark set consists of 12 groups of questions of different sizes, and each group contains ten instances. The relevant parameters of energy consumption are shown in Table I. We

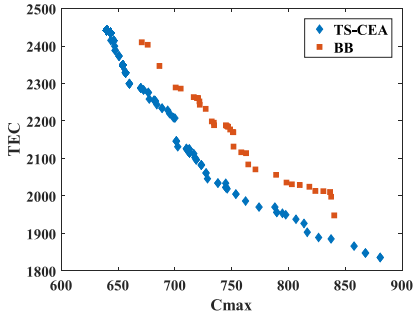

Fig. 9. Pareto fronts obtained by TS-CEA and BB.

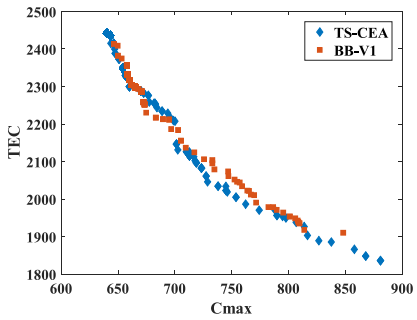

Fig. 10. Pareto fronts obtained by TS-CEA and BB-v1.

code all the algorithms in MATLAB and run them on a computer with an Intel Core i5-3230M CPU @ 2.60-GHz and 8 GB of RAM for comparison. Every algorithm is run independently ten times on each instance. The maximum running time of each algorithm is $n \times m \times \rho/100$ s, where $\rho = \{5, 10\}$.

Since the EENWFSP is an MOP, the following metrics are used to evaluate the quality of the obtained Pareto set.

1) *Overall Nondominated Vector Generation (ONVG):* The number of the nondominated solutions in the obtained Pareto archive E is counted (denoted as $|E|$).
2) *C Metric:* The dominance relationship between the solutions in two Pareto archives E_1 and E_2 is measured, which reflects the percentage of solutions in E_2 that are dominated by or the same as the solutions in E_1. $C(E_1, E_2)$ is calculated as follows:

$$C(\boldsymbol{E}_1, \boldsymbol{E}_2) = |\{b \in \boldsymbol{E}_2 | \exists a \in \boldsymbol{E}_1, a \succ b \text{ or } a = b\}|/|\boldsymbol{E}_2|.$$

3) *Spacing Metric (TS):* The metric measures how evenly the nondominated solutions distribute, which is calculated as follows:

$$\mathrm{TS} = \sqrt{\frac{1}{|\boldsymbol{E}|}\sum_{a \in \boldsymbol{E}} \left(d_a - \overline{d}\right)^2}/\overline{d}$$

where d_a is the Euclid distance in the objective space between solution $\boldsymbol{a}$ and its nearest solution, and $\overline{d} = \sum_{a \in \boldsymbol{E}} d_a/|\boldsymbol{E}|$ is the average distance. The smaller TS is, the more uniformly the solutions distribute.

A. Parameter Setting

There are three key parameters in TS-CEA: 1) the number of iterations of ILS (K); 2) the strength of perturbation (SP); and 3) the population size of mutation (PS). The famous design of

TABLE III
PARAMETERS FOR DIFFERENT LEVELS

Parameters	Factor level		
	1	2	3
K	5	10	15
SP	5	10	15
PS	50	100	200

TABLE IV
ANOVA RESULTS FOR PARAMETER SETTINGS OF TS-CEA

Source	Sum of squares	Degrees of freedom	Mean Square	F-ratio	p-value
K	102.01	2	51.005	22.6	**0.0005**
SP	22.088	2	11.044	4.89	0.0409
PS	294.975	2	147.488	65.36	**0**
$K * SP$	12.847	4	3.212	1.42	0.3102
$K * PS$	71.978	4	17.994	7.97	**0.0068**
$SP * PS$	18.835	4	4.709	2.09	0.1745
Residual	18.051	8	2.256		
Total	540.784	26			

experiment (DOE) is used to analyze the effect of parameters on the performance of TS-CEA. The levels of each parameter are listed in Table III.

A total of 60 instances from different scales are randomly selected for investigation. For every instance, the TS-CEA with each parameter combination is run independently ten times to obtain the nondominated solution set $E_i(i = 1, 2, \ldots, 27)$. The rigid nondominated solutions among $E_1 - E_{27}$ consist of the final set FE. To evaluate the contribution of each parameter combination, an evaluation metric is defined as $\mathrm{CON}(i) = |E_i'|/|FE|$, where $E_i' = E_i \cap FE$. After all instances are tested, the average contribution of each parameter combination is calculated as the response value (RV). Following [2], the experimental results are analyzed by the multivariate analysis of variance (ANOVA). Whether the interaction between parameters is significant or not is expressed by variance analysis. The results of the ANOVA are shown in Table IV.

According to the results from Table IV, the p-values of parameters K and PS are less than the confidence level ($\alpha = 0.05$), which implies these parameters are more influential than other parameters in TS-CEA. Meanwhile, the parameter PS corresponds to the greatest F-ratio, which suggests that the parameter PS is of the greatest effect on the average performance of the TS-CEA among all factors. According to Fig. 11, the best parameters are $K = 10$, $SP = 5$, and $PS = 50$. Furthermore, if the p-value between the two parameters is less than 0.05, the main effect plot is meaningless [48]. As shown in Table IV, the p-value of parameter $K * PS$ is 0.0068, which is less than 0.05. The interactions between the parameters K and PS are significantly demonstrated. From the interaction plot between K and PS in Fig. 12, the selection of $K = 10$ and $PS = 50$ contributes to the best performing TS-CEA. So, the following parameters are used in the TS-CEA: $K = 10$, $SP = 5$, and $PS = 50$. Besides, since NN + MNEH and JPA are both constructive heuristics, their performance is not sensitive to the value of the parameters [45]. Therefore, the population size of NN + MNEH is set as $NP = 50$.

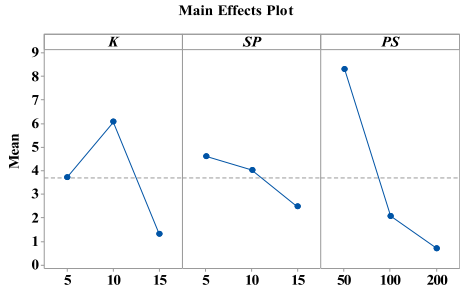

Fig. 11. Main effects plot of parameters.

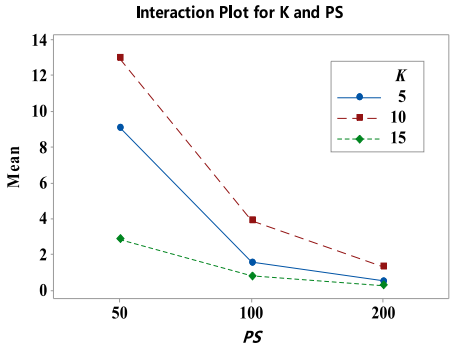

Fig. 12. Interaction plot of parameters.

TABLE V
MAKESPAN OBTAINED BY NN + MNEH AND JPA

Instance	NN+MNEH	JPA
20×5	**1.54E+03**	1.59E+03
20×10	**2.06E+03**	2.12E+03
20×20	**3.08E+03**	3.18E+03
50×5	**3.48E+03**	3.48E+03
50×10	**4.50E+03**	4.57E+03
50×20	**6.16E+03**	6.35E+03
100×5	6.68E+03	**6.51E+03**
100×10	8.51E+03	**8.43E+03**
100×20	1.13E+04	**1.13E+04**
200×10	1.63E+04	**1.59E+04**
200×20	2.11E+04	**2.09E+04**
500×20	4.97E+04	**4.83E+04**

B. Effectiveness of Initialization Methods

Both NN + MNEH and JPA are constructive heuristics. The speeds at which NN + MNEH and JPA construct a solution are fast. So, the result of each run is unique. In the initial stage of the TS-CEA, both NN + MNEH and JPA use the same speed matrix v. Therefore, the smaller the makespan is, the smaller the TEC is. 120 benchmark instances [47] are used to test the performance of NN + MNEH and JPA.

As shown in Table V, NN + MNEH and JPA are suitable for scheduling problems of different scales. The former is effective for constructing initial solutions of the small-scale scheduling problems, and the latter is effective for constructing initial solutions of large-scale scheduling problems.

C. Effect of Combined Neighborhood Structures

According to [25], there are a lot of local optima and a big valley structure in the fitness landscape of NWFSP. Therefore, the hybridization with local search procedures enabled the

TABLE VI
COMPARISON OF THREE COMBINATION OPERATORS ON MAKESPAN METRIC

Instance	insert and swap	swap and block shift	insert and block shift
20×5	**1.48E+03**	1.49E+03	**1.48E+03**
20×10	**1.98E+03**	1.99E+03	**1.98E+03**
20×20	**2.97E+03**	**2.97E+03**	**2.97E+03**
50×5	3.31E+03	3.36E+03	**3.28E+03**
50×10	4.31E+03	4.40E+03	**4.28E+03**
50×20	5.94E+03	6.06E+03	**5.91E+03**
100×5	6.33E+03	6.44E+03	**6.25E+03**
100×10	8.15E+03	8.33E+03	**8.05E+03**
100×20	1.09E+04	1.11E+04	**1.07E+04**
200×10	1.56E+04	1.58E+04	**1.53E+04**
200×20	2.04E+04	2.08E+04	**2.01E+04**
500×20	4.78E+04	4.83E+04	**4.74E+04**
Average	1.08E+04	1.09E+04	**1.06E+04**

TABLE VII
COMPARISON OF THREE COMBINATION OPERATORS ON RPI METRIC

Instance	Insert and swap		Swap and block shift		Insert and block shift	
	RPI	SD	RPI	SD	RPI	SD
20×5	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00
20×10	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00
20×20	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00
50×5	0.01	0.00	0.02	0.00	**0.00**	0.00
50×10	0.01	0.00	0.03	0.00	**0.00**	0.00
50×20	0.01	0.00	0.03	0.00	**0.00**	0.00
100×5	0.01	0.00	0.03	0.00	**0.00**	0.00
100×10	0.01	0.00	0.04	0.00	**0.00**	0.00
100×20	0.01	0.00	0.03	0.00	**0.00**	0.00
200×10	0.02	0.00	0.03	0.00	**0.00**	0.00
200×20	0.02	0.00	0.04	0.00	**0.00**	0.00
500×20	0.01	0.00	0.02	0.00	**0.00**	0.00
Average	0.01	0.00	0.02	0.00	**0.00**	0.00

algorithm to find high-quality solutions. Variable neighborhood search is an effective strategy to solve combinatorial optimization problems [49]–[51]. So, in the first stage of TS-CEA, the processing speed of machines remains unchanged. The smaller the makespan is, the smaller the TEC is. Thus, EENWFSP is formulated as a single-objective optimization problem. The three local search operators, insert and swap, swap and block shift, insert and block shift, are tested in the section. The same stop criterion is used by the three local search operators. The relative percentage increase (RPI) is adopted to evaluate the effect of the three local search operators as $RPI = (C_r - C_r^*)/C_r^* \times 100$, where C_r is the makespan of the rth solution produced by the corresponding algorithm, and C_r^* is the makespan of the best solution among all the algorithms for comparison. The values of *RPI* reflect the performance of the algorithms. The average makespan, average RPI, and the standard deviation (SD) for each group with different size are listed in Tables VI and VII, respectively.

First, from Tables VI and VII, it can be seen that the solutions generated by insert and block shift are better than those obtained by insert and swap as well as swap and block shift on almost all the test cases. As shown in Fig. 13, it is worth mentioning that insert and block shift obtain significantly better results. Second, a neighborhood structure corresponds to a fitness landscape in the solution space of NWFSP. The transformation between different neighborhood structures helps local search algorithms escape from local optima. Finally, given a solution, the number of neighborhoods generated by the insertion operator is $(n-1)^2$, and the number of neighborhoods generated by the swap operator is $n(n-1)/2$. The search range of the insertion operator is larger than that of the swap operator. Furthermore, block structure is an intrinsic property of the NWFSP. Therefore, the insert and block shift neighborhood structures are used as local search operators for ILS.

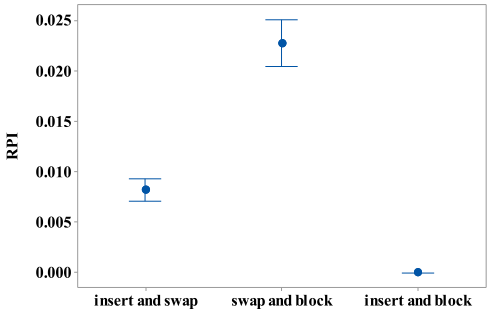

Fig. 13. Means plot and 95% Tukey's honest significant difference intervals for local search.

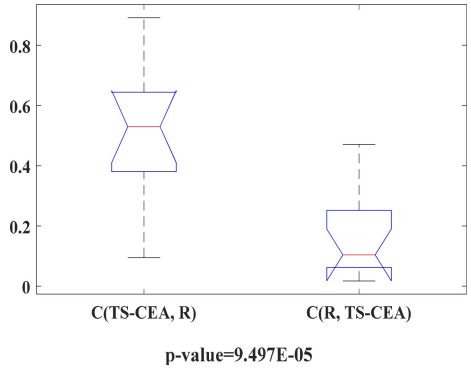

Fig. 14. Mean and 95% LSD interval for the TS-CEA.

D. Effect of Mutation Strategy

In the second stage of TS-CEA, the scheduling sequence of the jobs keeps unchanged. A mutation strategy based on critical path knowledge is proposed to detect potential non-dominated solutions. A random mutation experiment (denoted as ***R***) is carried out to verify the effect of the mutation strategy. The random mutation experiment means that an operation is randomly selected and then randomly changes the processing speed of the operation, regardless of whether the operation is on the critical path.

For each scale, the average C metric of ten instances is presented in Table VIII. The mean and 95% Fisher's least-significant difference (LSD) interval for TS-CEA are illustrated in Fig. 14. It is concluded that the TS-CEA significantly outperforms ***R*** that does not consider the critical path knowledge. From Table VIII, the values of C(TS-CEA, ***R***) are

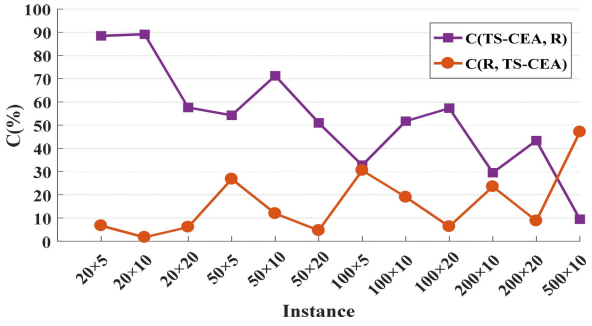

Fig. 15. Trend of C metric between TS-CEA and $\boldsymbol{R}$.

TABLE VIII
COMPARISON OF C BETWEEN TS-CEA AND $\boldsymbol{R}$

Instance	C(TS-CEA, $\boldsymbol{R}$)	C($\boldsymbol{R}$, TS-CEA)
20×5	**0.88**	0.07
20×10	**0.89**	0.02
20×20	**0.58**	0.06
50×5	**0.54**	0.27
50×10	**0.71**	0.12
50×20	**0.51**	0.05
100×5	**0.33**	0.31
100×10	**0.52**	0.19
100×20	**0.57**	0.06
200×10	**0.30**	0.24
200×20	**0.43**	0.09
500×20	0.09	**0.47**
Average	**0.53**	0.16

TABLE IX
COMPARISONS OF ONVG BETWEEN TS-CEA AND $\boldsymbol{R}$

Instance	TS-CEA	$\boldsymbol{R}$
20×5	75.1	**120.9**
20×10	78.7	**94.6**
20×20	67.4	**85.6**
50×5	97.1	**107.6**
50×10	82.0	**90.3**
50×20	**66.8**	54.8
100×5	**125.8**	88.9
100×10	**79.3**	60.9
100×20	**55.6**	44.0
200×10	**61.0**	52.0
200×20	**46.3**	30.7
500×20	**42.4**	25.2
Average	73.1	71.3

larger than those of $C(\boldsymbol{R}$, TS-CEA) on almost all instances, except instance 500 × 20. This implies that the most nondominated solutions obtained by $\boldsymbol{R}$ are dominated by those obtained by TS-CEA.

From Fig. 15, it can be seen that as the size of the problem increases the value of C(TS-CEA, $\boldsymbol{R}$) shows a decreasing trend.

Furthermore, Wilcoxon's test [52] is executed to check the behaviors of the two algorithms. From Table IX-XI, TS-CEA is significantly better than $\boldsymbol{R}$ with $\alpha = 0.05$ in terms of C metric. $\boldsymbol{R}$ is significantly better than TS-CEA with $\alpha = 0.05$ in terms of TS metric. Although the significant differences

TABLE X
COMPARISONS OF TS BETWEEN TS-CEA AND $\boldsymbol{R}$

Instance	TS-CEA	$\boldsymbol{R}$
20×5	2.90	**2.54**
20×10	3.52	**3.07**
20×20	4.29	**2.35**
50×5	3.46	**2.29**
50×10	3.57	**2.39**
50×20	5.81	**2.64**
100×5	3.18	**2.26**
100×10	4.80	**2.87**
100×20	7.84	**3.39**
200×10	3.59	**2.98**
200×20	5.48	**2.76**
500×20	6.81	10.33
Average	4.60	**3.32**

TABLE XI
WILCOXON'S TEST OF C METRIC BETWEEN TS-CEA AND $\boldsymbol{R}\alpha = 0.05$ SIGNIFICANCE LEVEL

TS-CEA*vs*	Metrics	R+	R-	Z	p-value	$\alpha = 0.05$
	C	6235.50	1024.50	-6.824	0.000	**Yes**
$\boldsymbol{R}$	ONVG	3806.00	2980.00	-1.138	0.255	No
	TS	919.50	6340.50	-7.098	0.000	**Yes**

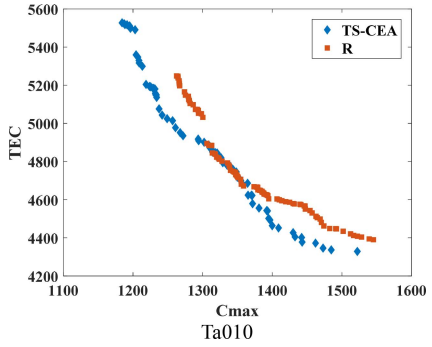

Fig. 16. Pareto front of TS-CEA and $\boldsymbol{R}$.

are not observed between the TS-CEA and $\boldsymbol{R}$ in terms of the ONVG metric, the value of R+ is better than the value of R-. That is, the number of nondominated solutions obtained by the TS-CEA is larger than that of $\boldsymbol{R}$. From Fig. 16, it can be seen that the performance of TS-CEA is better than that of $\boldsymbol{R}$. Therefore, the mutation strategy based on critical path knowledge is more effective than the random mutation strategy in solving EENWFSP.

VI. COMPARISONS TO THE STATE-OF-THE-ART ALGORITHMS

A. Numerical Result

Next, we compare the TS-CEA with the state-of-the-art algorithms, including NSGA-II [53], IG_ALL [8], and KCA [13], to further test the performance of TS-CEA. The control parameter of each algorithm is set as the value recommended in the original paper. Although most of the comparative algorithms are proposed for the permutation flow-shop scheduling problem, they can be generalized without changing their framework, idea, encoding, initialization, and local search. NSGA-II is a nondominated sorting-based

TABLE XII
ONVG OF THE ALL ALGORITHMS

Instance	TS-CEA	NSGA-II	IG_ALL	KCA
20×5	**75.10**	33.00	14.30	20.20
20×10	**78.70**	27.00	8.60	14.50
20×20	**67.40**	14.30	4.30	9.80
50×5	**97.10**	29.40	10.40	15.40
50×10	**82.00**	17.90	4.20	10.70
50×20	**66.80**	7.10	3.00	6.10
100×5	**125.80**	22.90	11.20	13.50
100×10	**79.30**	13.30	3.00	8.00
100×20	**55.60**	7.30	3.00	7.60
200×10	**61.00**	16.40	3.00	8.40
200×20	**46.30**	7.90	3.00	5.50
500×20	**42.40**	5.90	3.00	7.80
Average	**73.13**	16.87	5.92	10.63

TABLE XIII
TS OF THE ALGORITHMS

Instance	TS-CEA	NSGA-II	IG_ALL	KCA
20×5	**2.90**	3.08	16.80	4.78
20×10	3.52	**2.25**	28.83	8.77
20×20	4.29	**4.05**	32.81	15.11
50×5	**3.46**	3.57	39.61	7.41
50×10	**3.57**	4.05	42.14	15.84
50×20	5.81	6.35	**1.07**	24.95
100×5	**3.18**	7.50	60.23	19.78
100×10	4.80	11.59	**0.17**	23.56
100×20	7.84	11.33	**1.32**	35.92
200×10	3.59	10.95	**0.21**	38.45
200×20	5.48	11.06	**1.62**	82.37
500×20	6.81	17.63	**2.30**	44.82
Average	**4.60**	7.78	18.93	26.81

TABLE XIV
COMPARISONS OF C METRIC DURING THE ALGORITHMS
($A = C$(TS-CEA, NSGA-II), $A' = C$(NSGA-II, TS-CEA), $B = C$(TS-CEA,IG_ALL), $B' = C$(IG_ALL, TS-CEA), $C = C$(TS-CEA, KCA), $C' = C$(KCA, TS-CEA))

Instance	A	A'	B	B'	C	C'
20×5	**0.95**	0.00	**0.82**	0.00	**0.32**	0.00
20×10	**1.00**	0.00	**0.75**	0.00	**0.29**	0.00
20×20	**1.00**	0.00	**0.51**	0.00	**0.35**	0.00
50×5	**0.96**	0.00	**0.80**	0.00	**0.36**	0.00
50×10	**1.00**	0.00	**0.52**	0.00	**0.56**	0.00
50×20	**1.00**	0.00	**0.33**	0.00	**0.55**	0.00
100×5	**0.99**	0.00	**0.82**	0.00	**0.54**	0.00
100×10	**1.00**	0.00	**0.33**	0.00	**0.66**	0.00
100×20	**1.00**	0.00	**0.33**	0.00	**0.62**	0.00
200×10	**1.00**	0.00	**0.33**	0.00	**0.63**	0.00
200×20	**1.00**	0.00	**0.33**	0.00	**0.61**	0.00
500×20	**1.00**	0.00	**0.27**	0.03	**0.67**	0.00
Average	**0.99**	0.00	**0.51**	0.00	**0.51**	0.00

multiobjective evolutionary algorithm using the fast nondominated sorting approach. A local search operator is applied to the partial solutions after a complete solution is destructed in the IG_ALL and a speed-scaling strategy [8] is employed. The core idea of KCA is that different solutions search along different search directions. According to the location of the solution in the target space, the corresponding collaborative search operator is used to obtain potential nondominant solutions.

Same as the experiment setting in Section V, 120 testing cases are used to evaluate the performance of all algorithms. Every algorithm is run independently ten times on each instance and executed in the same operating environment. The maximum running time of each algorithm is $n \times m \times \rho/100$ s, where $\rho = \{5, 10\}$. The average *ONVG*, *TS*, and *C* metrics in solving all the instances are listed in Tables XII–XIV grouped by the scale of the testing instances.

TABLE XV
WILCOXON'S TEST OF TS METRIC OF ALL ALGORITHMS $\alpha = 0.05$ SIGNIFICANCE LEVEL

TS-CEA *vs*	R+	R-	Z	p-value	$\alpha = 0.05$
NSGA-II	1942.00	5318.00	-4.421	0.000	**Yes**
IG_ALL	1983.00	5277.00	-4.313	0.000	**Yes**
KCA	391.50	6868.50	-8.481	0.000	**Yes**

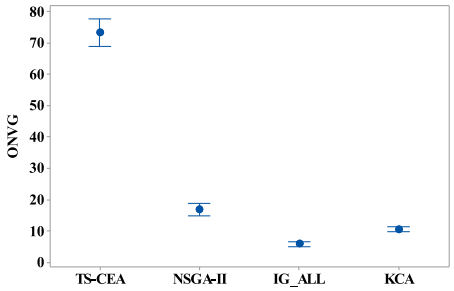

Fig. 17. Means plot and 95% Tukey's honest significant difference intervals for ONVG.

B. Analysis and Discussion

First, as shown in Fig. 17, ONVG of the TS-CEA is larger than those of the NSGA-II, IG_ALL, and KCA, which implies that TS-CEA can obtain more nondominated solutions than NSGA-II, IG_ALL, and KCA. Second, from Table XV, TS-CEA is significantly better than the other algorithms with $\alpha = 0.05$ in terms of TS metric. That is, the nondominated solutions obtained by the TS-CEA distribute more uniformly than those obtained by other algorithms. Third, from Table XIV, almost on all instances the values of C(TS-CEA, $NSGA-II$) are almost equal to 1 while the values of $C(NSGA-II$, TS-CEA) are almost equal to 0, which implies that almost all the nondominated solutions obtained by NSGA-II are dominated by those obtained by TS-CEA. Similarly, comparing TS-CEA to IG_ALL and KCA, it can be seen that the values of C(TS-CEA, *IG_ALL*) are larger than the values of C(*IG_ALL*, TS-CEA) and the values of C(TS-CEA, *KCA*) are larger than the values of C(*KCA*, TS-CEA) on all instances, which implies that the Pareto front obtained by TS-CEA is better than those obtained by the IG_ALL and KCA in terms of convergence. Furthermore, from Figs. 18 and 19, for solving Ta005 instance with different ρ, the performance of TS-CEA outperforms the other three algorithms in terms of convergence and distribution of the nondominated solutions. So, it can be concluded that TS-CEA is more effective than state-of-the-art algorithms in solving EENWFSP.

For TS-CEA, its first stage focuses on finding extreme solutions. In the stage, the processing speed of machines remains unchanged. Therefore, the size of the solution space of EENWFSP is $n!$. In its second stage, the explored extreme solutions extend to approximate the entire Pareto front. At this time, the scheduling sequence of the jobs remains

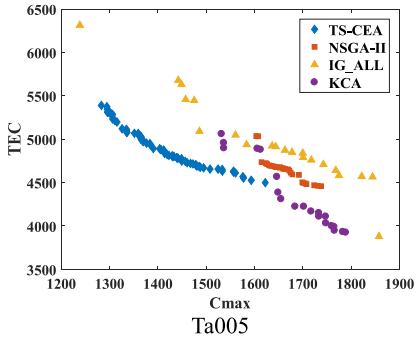

Fig. 18. Pareto fronts obtained by all algorithms ($\rho = 5$).

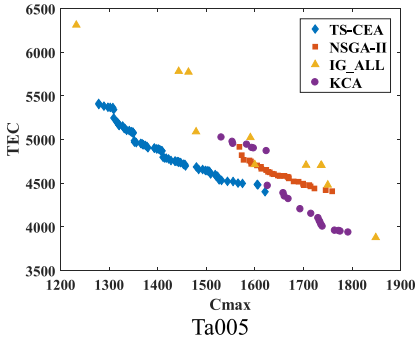

Fig. 19. Pareto fronts obtained by all algorithms ($\rho = 10$).

unchanged. The size of the solution space of EENWFSP is $3^{n \cdot m}$. Without the above two-stage search, the solution space of the EENWFSP will be $n! \cdot 3^{n \cdot m}$. So, the two-stage search mechanism of TS-CEA is effective in solving the EENWFSP.

VII. Conclusion

This article presented a TS-CEA to solve the energy-efficient no-wait FSP with minimizing makespan and TEC simultaneously. According to the control variable method, a reasonable co-evolutionary search framework is designed to reduce the solution space of EENWFSP. By reasonably designing the two-stage cooperative evolutionary algorithm, better results can be achieved than the existing algorithms. Moreover, extensive numerical comparisons show that our ideas are effective in designing the initialization method, combined neighborhood structures, and mutation strategy.

Although the performance of TS-CEA outperforms other algorithms for comparison, its performance could be further improved by using some adaptive mechanisms and learning strategy when designing the algorithm. We will extend our research to the scheduling problems with blocking, no idle, and distributed manufacturing scenarios. Moreover, some practical applications about energy efficiency is worth the investigation. It is also interesting to explore domain knowledge and employ them in designing powerful algorithms for solving other energy-efficient scheduling problems.

References

[1] Y. Y. Han, D. W. Gong, Y. C. Jin, and Q. K. Pan, "Evolutionary multiobjective blocking lot-streaming flow shop scheduling with machine breakdowns," *IEEE Trans. Cybern.*, vol. 49, no. 1, pp. 184–197, Jan. 2019.

[2] Z. S. Shao, D. C. Pi, W. S. Shao, and P. S. Yuan, "An efficient discrete invasive weed optimization for blocking flow-shop scheduling problem," *Eng. Appl. Artif. Intell.*, vol. 78, pp. 124–141, Feb. 2019.

[3] Y. M. Wang, X. P. Li, R. Ruiz, and S. C. Sui, "An iterated greedy heuristic for mixed no-wait flowshop problems," *IEEE Trans. Cybern.*, vol. 48, no. 5, pp. 1553–1566, May 2018.

[4] W. S. Shao, D. C. Pi, and Z. S. Shao, "A hybrid discrete teaching-learning based meta-heuristic for solving no-idle flow shop scheduling problem with total tardiness criterion," *Comput. Oper. Res.*, vol. 94, pp. 89–105, Jun. 2018.

[5] Q.-K. Pan, L. Gao, L. Wang, J. Liang, and X.-Y. Li, "Effective heuristics and metaheuristics to minimize total flowtime for the distributed permutation flowshop problem," *Expert Syst. Appl.*, vol. 124, pp. 309–324, Jun. 2019.

[6] J.-F. Chen, L. Wang, and Z.-P. Peng, "A collaborative optimization algorithm for energy-efficient multi-objective distributed no-idle flow-shop scheduling," *Swarm Evol. Comput.*, vol. 50, Nov. 2019, Art. no. 100557.

[7] N. G. Hall and C. Sriskandarajah, "A survey of machine scheduling problems with blocking and no-wait in process," *Oper. Res.*, vol. 44, no. 3, pp. 510–525, 1996.

[8] H. Oztop, M. F. Tasgetiren, D. T. Eliiyi, and Q.-K. Pan, *Green Permutation Flowshop Scheduling: A Trade- Off- Between Energy Consumption and Total Flow Time* (Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)). Cham, Switzerland: Springer, 2018, pp. 753–759.

[9] D. M. Lei, L. Gao, and Y. Zheng, "A novel teaching-learning-based optimization algorithm for energy-efficient scheduling in hybrid flow shop," *IEEE Trans. Eng. Manag.*, vol. 65, no. 2, pp. 330–340, May 2018.

[10] X. Y. Li, C. Lu, L. Gao, S. Q. Xiao, and L. Wen, "An effective multiobjective algorithm for energy-efficient scheduling in a real-life welding shop," *IEEE Trans. Ind. Informat.*, vol. 14, no. 12, pp. 5400–5409, Dec. 2018.

[11] B. Zhang, Q.-K. Pan, L. Gao, L.-L. Meng, X. Li, and K.-K. Peng, "A three-stage multiobjective approach based on decomposition for an energy-efficient hybrid flow shop scheduling problem," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, May 29, 2019, doi: 10.1109/TSMC.2019.2916088.

[12] K. Z. Gao, Y. Huang, A. Sadollah, and L. Wang, "A review of energy-efficient scheduling in intelligent production systems," *Complex Intell. Syst.*, vol. 6, no. 2, pp. 237–249, 2020.

[13] J. J. Wang and L. Wang, "A knowledge-based cooperative algorithm for energy-efficient scheduling of distributed flow-shop," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 5, pp. 1805–1819, May 2020.

[14] E.-D. Jiang and L. Wang, "An improved multi-objective evolutionary algorithm based on decomposition for energy-efficient permutation flow shop scheduling problem with sequence-dependent setup time," *Int. J. Prod. Res.*, vol. 57, no. 6, pp. 1756–1771, 2019.

[15] M. R. Garey and D. S. Johnson, *A Guide to the Theory of NP-Completeness*, WH Freemann, New York, NY, USA, 1979.

[16] S.-W. Lin and K.-C. Ying, "Optimization of makespan for no-wait flow-shop scheduling problems using efficient matheuristics," *Omega*, vol. 64, pp. 115–125, Oct. 2016.

[17] H. H. Ye, W. Li, and E. M. Miao, "An effective heuristic for no-wait flow shop production to minimize makespan," *J. Manuf. Syst.*, vol. 40, pp. 2–7, Jul. 2016.

[18] H. H. Ye, W. Li, and A. Abedini, "An improved heuristic for no-wait flow shop to minimize makespan," *J. Manuf. Syst.*, vol. 44, pp. 273–279, Jul. 2017.

[19] F. Q. Zhao, S. Qin, G. Q. Yang, W. M. Ma, C. Zhang, and H. B. Song, "A factorial based particle swarm optimization with a population adaptation mechanism for the no-wait flow shop scheduling problem with the makespan objective," *Expert Syst. Appl.*, vol. 126, pp. 41–53, Jul. 2019.

[20] F. Q. Zhao, S. Qin, Y. Zhang, W. M. Ma, C. Zhang, and H. B. Song, "A hybrid biogeography-based optimization with variable neighborhood search mechanism for no-wait flow shop scheduling problem," *Expert Syst. Appl.*, vol. 126, pp. 321–339, Jul. 2019.

[21] W. S. Shao, D. C. Pi, and Z. S. Shao, "An extended teaching-learning based optimization algorithm for solving no-wait flow shop scheduling problem," *Appl. Soft Comput. J.*, vol. 61, pp. 193–210, Dec. 2017.

[22] O. Engin and A. Guclu, "A new hybrid ant colony optimization algorithm for solving the no-wait flow shop scheduling problems," *Appl. Soft Comput. J.*, vol. 72, pp. 166–176, Nov. 2018.

[23] F. Q. Zhao, H. Liu, Y. Zhang, W. M. Ma, and C. Zhang, "A discrete water wave optimization algorithm for no-wait flow shop scheduling problem," *Expert Syst. Appl.*, vol. 91, pp. 347–363, Jan. 2018.

[24] F. Q. Zhao *et al.*, "An improved water wave optimization algorithm with the single wave mechanism for the no-wait flow-shop scheduling problem," *Eng. Optim.*, vol. 51, no. 10, pp. 1727–1742, 2019.
[25] J. Czogalla and A. Fink, "Fitness landscape analysis for the no-wait flow-shop scheduling problem," *J. Heuristics*, vol. 18, no. 1, pp. 25–51, 2012.
[26] Q.-K. Pan, R. Ruiz, and P. Alfaro-Fernandez, "Iterated search methods for earliness and tardiness minimization in hybrid flowshops with due windows," *Comput. Oper. Res.*, vol. 80, pp. 50–60, Apr. 2017.
[27] J.-Y. Ding, S. J. Song, R. Zhang, S. W. Zhou, and C. Wu, "A novel block-shifting simulated annealing algorithm for the no-wait flowshop scheduling problem," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Sendai, Japan, 2015, pp. 2768–2774.
[28] X. Zheng, S. C. Zhou, R. Xu, and H. P. Chen, "Energy-efficient scheduling for multi-objective two-stage flow shop using a hybrid ant colony optimisation algorithm," *Int. J. Prod. Res.*, vol 58, no. 13, pp. 4103–4120, 2020.
[29] X.-F. Liu, Z.-H. Zhan, J. D. Deng, Y. Li, T. L. Gu, and J. Zhang, "An energy efficient ant colony system for virtual machine placement in cloud computing," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 113–128, Feb. 2018.
[30] X. L. Wu and Y. J. Sun, "A green scheduling algorithm for flexible job shop with energy-saving measures," *J. Clean. Prod.*, vol. 172, pp. 3249–3264, Jan. 2018.
[31] J. H. Yan, L. Li, F. Zhao, F. Y. Zhang, and Q. L. Zhao, "A multi-level optimization approach for energy-efficient flexible flow shop scheduling," *J. Clean. Prod.*, vol. 137, pp. 1543–1552, Nov. 2016.
[32] C. Lu, L. Gao, X. Y. Li, Q.-K. Pan, and Q. Wang, "Energy-efficient permutation flow shop scheduling problem using a hybrid multi-objective backtracking search algorithm," *J. Clean. Prod.*, vol. 144, pp. 228–238, Feb. 2017.
[33] X. Q. Wu and A. Che, "Energy-efficient no-wait permutation flow shop scheduling by adaptive multi-objective variable neighborhood search," *Omega*, vol. 94, pp. 102–117, Jul. 2020.
[34] D. M. Lei, M. Li, and L. Wang, "A two-phase meta-heuristic for multiobjective flexible job shop scheduling problem with total energy consumption threshold," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 1097–1109, Mar. 2019.
[35] J.-Q. Li, H.-Y. Sang, Y.-Y. Han, C.-G. Wang, and K.-Z. Gao, "Efficient multi-objective optimization algorithm for hybrid flow shop scheduling problems with setup energy consumptions," *J. Clean. Prod.*, vol. 181, pp. 584–598, Apr. 2018.
[36] M. Aghelinejad, Y. Ouazene, and A. Yalaoui, "Production scheduling optimisation with machine state and time-dependent energy costs," *Int. J. Prod. Res.*, vol. 56, no. 16, pp. 5558–5575, 2018.
[37] G. R. Chen, L. Zhang, J. Arinez, and S. Biller, "Energy-efficient production systems through schedule-based operations," *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 1, pp. 27–37, Jan. 2013.
[38] X. Gong, Y. Liu, N. Lohse, T. D. Pessemier, L. Martens, and W. Joseph, "Energy- and labor-aware production scheduling for industrial demand response using adaptive multiobjective memetic algorithm," *IEEE Trans. Ind. Informat.*, vol. 15, no. 2, pp. 942–953, Feb. 2019.
[39] N. Sundstrom, O. Wigstrom, and B. Lennartson, "Conflict between energy, stability, and robustness in production schedules," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 2, pp. 658–668, Apr. 2017.
[40] W.-N. Chen, Y.-H. Jia, F. Zhao, X.-N. Luo, X.-D. Jia, and J. Zhang, "A cooperative co-evolutionary approach to large-scale multisource water distribution network optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 842–857, Oct. 2019.
[41] D. W. Gong, B. Xu, Y. Zhang, Y. N. Guo, and S. X. Yang, "A similarity-based cooperative co-evolutionary algorithm for dynamic interval multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 142–156, Feb. 2020.
[42] Y. Wang and Z. X. Cai, "Combining multiobjective optimization with differential evolution to solve constrained optimization problems," *IEEE Trans. Evol. Comput.*, vol. 16, no. 1, pp. 117–134, Feb. 2012.
[43] J. H. Wang, T. Y. Weng, and Q. F. Zhang, "A two-stage multiobjective evolutionary algorithm for multiobjective multidepot vehicle routing problem with time windows," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2467–2478, Jul. 2019.
[44] Y. Wang, B. Xu, G. Y. Sun, and S. X. Yang, "A two-phase differential evolution for uniform designs in constrained experimental domains," *IEEE Trans. Evol. Comput.*, vol. 21, no. 5, pp. 665–680, Oct. 2017.
[45] F. Q. Zhao *et al.*, "A jigsaw puzzle inspired algorithm for solving large-scale no-wait flow shop scheduling problems," *Appl. Intell.*, vol. 50, no. 1, pp. 87–100, 2020.
[46] S. A. Mansouri, E. Aktas, and U. Besikci, "Green scheduling of a two-machine flowshop: Trade-off between makespan and energy consumption," *Eur. J. Oper. Res.*, vol. 248, no. 3, pp. 772–788, 2016.
[47] E. Taillard, "Benchmarks for basic scheduling problems," *Eur. J. Oper. Res.*, vol. 64, no. 2, pp. 278–285, 1993.
[48] W. S. Shao, D. C. Pi, and Z. S. Shao, "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.*, vol. 16, no. 3, pp. 1344–1360, Jul. 2019.
[49] A. Herran, J. M. Colmenar, and A. Duarte, "A variable neighborhood search approach for the vertex bisection problem," *Inf. Sci.*, vol. 476, pp. 1–18, Feb. 2019.
[50] Y. Z. Qiu, L. Wang, X. L. Xu, X. J. Fang, and P. M. Pardalos, "A variable neighborhood search heuristic algorithm for production routing problems," *Appl. Soft Comput. J.*, vol. 66, pp. 311–318, May 2018.
[51] B. W. Thomas and E. Manni, "Scheduled penalty variable neighborhood search," *Comput. Oper. Res.*, vol. 52, pp. 170–180, Dec. 2014.
[52] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 special session on real parameter optimization," *J. Heuristics*, vol. 15, no. 6, p. 617, 2009.
[53] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

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, where he became a Full Professor in 2012. He was the Postdoctoral Fellow with the State Key Laboratory of Manufacturing System Engineering, Xi'an Jiaotong University, Xi'an, China, since 2009. He was a Visiting Scholar with Exeter Manufacturing Enterprise Center, Exeter University, Exeter, U.K., from 2008 to 2019, and the Georgia Tech Manufacturing Institute, Georgia Institute of Technology, Atlanta, GA, USA, from 2014 to 2015. He has authored two academic book and over 50 refereed papers. His current research interests include intelligent optimization and scheduling.

Xuan He received the B.S. and M.S. degrees in computer and communication technology from Lanzhou University of Technology, Lanzhou, China, in 2017 and 2020, respectively. He is currently pursuing the Ph.D. degree with the School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, China.

His current research interests include intelligent optimization and scheduling algorithms.

Ling Wang received the B.Sc. degree in automation and the Ph.D. degree in control theory and control engineering from Tsinghua University, Beijing, China, in 1995 and 1999, respectively.

Since 1999, he has been with the Department of Automation, Tsinghua University, where he became a Full Professor in 2008. He has authored five academic books and more than 300 refereed papers. His current research interests include intelligent optimization and production scheduling.

Prof. Wang was a recipient of the National Natural Science Fund for Distinguished Young Scholars of China, the National Natural Science Award (Second Place) in 2014, the Science and Technology Award of Beijing City in 2008, and the Natural Science Award (First Place in 2003, and Second Place in 2007) nominated by the Ministry of Education of China. He is currently an Editor-in-Chief of *International Journal of Automation and Control* and an Associate Editor of IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and *Swarm and Evolutionary Computation*.