IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received January 21, 2019, accepted March 5, 2019, date of publication March 13, 2019, date of current version April 1, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2904848

Energy-Oriented Scheduling for Hybrid Flow Shop With Limited Buffers Through Efficient Multi-Objective Optimization

SHENG-LONG JIANG[1], (Member, IEEE), AND LONG ZHANG[2,3]
[1]College of Materials Science and Engineering, Chongqing University, Chongqing 400044, China
[2]School of Electronic Engineering and Automation, Guilin University of Electronic Technology, Guilin 541004, China
[3]Guangxi Key Laboratory of Automatic Detection Technology and Instrument, Guilin 541004, China

Corresponding author: Sheng-Long Jiang (sh.l.jiang.jx@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 6187020702, in part by the Fundamental Research Funds for the Central Universities under Grant 2018CDXYCL0018, in part by the Promotion Project of Basic Ability of Young and Middle-aged Teachers in Universities of Guangxi under Grant ky2016YB152, and in part by the Director Fund of the Guangxi Key Laboratory of Automatic Detection Technology and Instrument under Grant YQ16109.

ABSTRACT Efficient scheduling benefits productivity promotion, cost savings, and customer satisfaction. In recent years, with a growing concern about the energy price and environmental impact, energy-oriented scheduling is going to be a key issue for sustainable manufacturing. In this paper, we investigate an energy-oriented scheduling problem deriving from the hybrid flow shop with limited buffers. First, we formulate the scheduling problem with a mixed integer linear programming (MILP) model, which considers two objectives including minimizing the total weighted tardiness (TWT) and non-processing energy (NPE). To solve the NP-hard problem in the strong sense, we develop an efficient multi-objective optimization algorithm under the framework of the multi-objective objective evolutionary algorithm based on decomposition (MOEA/D). We devise a job-permutation vector to represent the scheduling solution and cover its search space. Since NPE is a non-regular function, we develop a two-pass decoding procedure composed of a discrete-event system (DES) simulation procedure and a greedily post-shift procedure. Besides, we apply an external archive population (EAP) to guide the algorithm to converge on a Pareto frontier and a local search procedure to enhance the diversity of the population. Finally, we conduct extensive computational experiments to verify the effectiveness of the proposed energy-oriented multi-objective optimization (EOMO) algorithm. The results presented in this paper may be useful for future research on energy-oriented scheduling problems in realistic production systems.

INDEX TERMS Hybrid flow shop, limited buffer, energy-oriented scheduling, multi-objective optimization, discrete event system.

I. INTRODUCTION

Production scheduling is a typical decision process for manufacturing companies, where jobs are appropriately assigned and sequenced on machines or production units to achieve one or multiple optimal objectives, such as the maximum completion time (makespan), tardiness, lateness, etc. The hybrid flow shop (HFS) scheduling problem (as shown **Fig. 1**) is a variant of flow shop scheduling problem stemming from a wide range of production systems including steelmaking [1], semi-conductor [2], chemical process [3] and so on. This type of problems have been thoroughly studied from academia to industry, and its surveys and recent advances can be referred to [4] and [5].

The majority of studies of the HFS scheduling problem commonly assume the intermediate buffers between two contiguous stages are infinite. However, in realistic production systems, the intermediate buffer capacities between the stages always are restricted, in which a completed job may stay on the incumbent machine when the buffer is full or wait in the buffer until a machine at the downstream stage is available. The steelmaking and continuous casting (SCC)

The associate editor coordinating the review of this manuscript and approving it for publication was Juan Liu.

2169-3536 © 2019 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

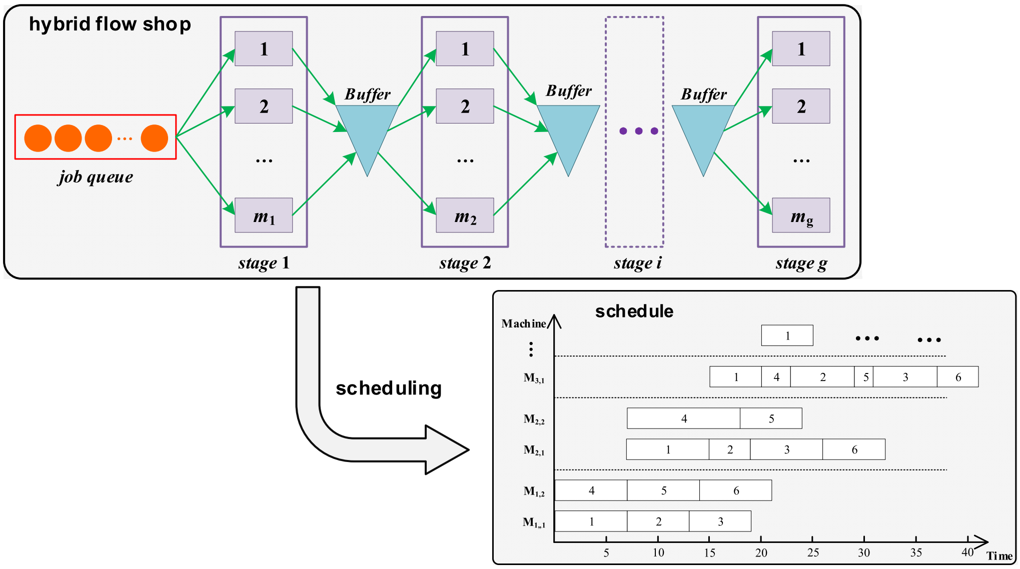

FIGURE 1. Scheduling in hybrid flow shop.

process in the steel industry is a realistic instance of this style. Tang *et al.* [6] illustrated the limited capacity in the SCC process caused by transporters (i.e. trolleys and cranes), and showed that the effective scheduling is meaningful not only for improving productivity but also for reducing energy consumption. In addition to this realistic constraint, another important goal of modern manufacturing companies is to reduce energy consumption due to the increasing concern about environmental impact and energy price [7]. Zhao *et al.* [8] cited an instance from rolling sector, which is an energy-intensive process in the steel industry. They constructed a mathematic model considering varied electricity pricing, and showed that scheduling decisions can take advantage of processing flexibility to make better use of electric power, and reduce the energy cost. With these two motivations, we investigate the special HFS scheduling problem involving limited buffers and energy consumption.

Considering the constraint of limited buffers and the goal of reducing energy consumption, the studied HFS scheduling problem tends to be more heavily constrained and pursues multiple optimal objectives. The remaining sections of the paper is organized as follows. **Section 2** reports the relevant literature review. **Section 3** formulates the new HFS scheduling problem considering limited buffers and energy consumption. **Section 4** develops an energy-oriented multi-objective optimization (EOMO) algorithm under the framework of the multi-objective objective evolutionary algorithm based on decomposition (MOEA/D). The experimental setup and results are analyzed and discussed in **Section 5**. Finally, **Section 6** concludes this study and identifies several topics for future studies.

II. RELATED PREVIOUS WORKS

Although the HFS with limited buffers (called HFS-LB for short) is a very flexible production layout with strong industrial background, only few studies have been reported on its scheduling problem comparing with the classic HFS scheduling problem. Sawik [9] proposed a mixed-integer programming approach for solving the batch scheduling problem in flexible flow lines with machine blocking. To solve the HFS scheduling problem with finite buffer sizes, Tang and Xuan [10] developed a Lagrangian relaxation algorithm, in which buffers were identified as machines with zero processing time. However, when finding the optimal solution of the problems with medium or large scales, the computational time of these exact algorithms is proved to be very high. To accelerate the computational speed on solving the aforementioned scheduling problems, Wardono and Fathi [11] proposed a tabu search (TS) algorithm and employed permutation vectors to limit its search space, Wang and Tang [12] also developed an improved TS heuristic incorporated with a scatter search mechanism. Furthermore, some studies focused on more complex problems in HFS-LB. Yaurima *et al.* [13] presented an improved genetic algorithm (GA) to solve the HFS with unrelated machines and including constraints of limited buffers, sequence-dependent setup times and availability. Abyaneh and Zandieh [14] proposed an improved multi-objective GA to solve the scheduling problems in HFS-LB with setup times, in which the objectives of total tardiness and total completion time were simultaneously optimized. Li and Pan [15] developed a novel hybrid algorithm combining an artificial bee colony (ABC) and a TS algorithm to solve the large-scale scheduling problems.

Nowadays, the increasing energy price and the environmental crisis motivates more energy-intensive manufacturing industries to reduce energy consumption and innovate new energy-saving techniques. Making optimal decisions on production scheduling is an efficient and green way to reduce energy consumption without machine renovation or product development [7], [16]. For instance, Dai *et al.* [17] adopted a genetic-simulated annealing algorithm to make a trade-off between total energy consumption and makespan when scheduling in HFS. Luo *et al.* [18] proposed a multi-objective ant colony optimization (MOACO) to solve the HFS scheduling problem considering makespan and electric power cost (EPC) with the presence of time-of-use (TOU) prices. Tang *et al.* [19] developed an improved particle swarm optimization (PSO) algorithm to solve the dynamic HFS scheduling problem minimizing energy consumption and makespan. Li *et al.* [20] proposed a multi-objective neighbor search algorithm for solving the HFS scheduling problem simultaneously considering setup energy consumption and makespan.

The scheduling in HFS-LB is an optimization problem of high complexity and practical value, however, the up-to-date studies on HFS-LB only consider maximum completion time (makespan), tardiness, flow time and other time-related objectives, not take energy consumption or environmental impacts completely into account. To solve this kind of multi-objective optimization problem (MOP), a large variety of evolutionary algorithms can be employed [21]. The MOEA/D is a well-known multi-objective optimization algorithm proposed by Zhang and Li [22]. Its main idea of is to decompose a multi-objective optimization problem into a number of single-objective subproblems by using a scalar function. Compared with other multi-objective algorithms, it has advantages on finding Pareto optimal solutions and computational time [23], [24]. Recently, MOEA/D also has applied to solve various scheduling problems [25], [26].

III. PROBLEM FORMULATION

A. ASSUMPTION

The problem investigated in this paper can be illustrated as follows. All machines at the same stage are identical and available at time zero. The buffer between consecutive stages can be treated as a stage with zero-processing-time machines [9], [10]. Hence, the HFS-LB scheduling problem can be transformed into the one with blocking. Taking these cues, each job is available at time zero and sequentially flow through both processing stages and buffering stages. A machine is not allowed to process simultaneously more than one job and any job is allowed to be processed by only one machine at the same time. Therefore, the completion time of a job at the buffering stage is equal to its departure time from the upstream stage.

In modern manufacturing companies, the due-date is a key indicator for management decision, because the just-in-time delivery from manufacturer to customer poses severe impacts on cost savings and service reputation. Therefore, this paper adopts the total weighted tardiness as the time-related objective due to its importance to the make-to-order (MTO) manufacturing. The energy consumption objective investigated in this study is referred to the work of Liu *et al.* [27], in which it is supposed that the machines still consume energy when idle. It has been proven that reducing the total non-processing energy (NPE) in a job shop can achieve the same effect as the decrease of the total energy consumption.

B. NOTATIONS

The notations cited in the problem formulation are defined as follows:

Parameters

j	index of jobs, $j \in J$, $J = \{1, 2, \ldots, \|J\|\}$, and 0 represents a dummpy job.
i	index of stages, $i \in I.$, $I = \{1, 2, \ldots, \|I\|\}$.
M_i	set of the parallel machines at stage i, $M_i = \{1, \ldots, k, \ldots, \|M_i\|\}$.
w_j	weight of job j.
d_j	due-date of job j.
$P_{i,j}$	processing time of job j at stage i.
$E_{i,k}$	non-processing energy of $M_{i,k}$
T_j	tardiness of job j.
α	non-processing energy price.

Decisions

x_{i,k,j_1,j_2}	if a pair of jobs (j_1, j_2) are consecutively assigned to machine $M_{i,k}$, then $x_{i,k,j_1,j_2} = 1$, otherwise, $x_{i,k,j_1,j_2} = 0$.
$C_{i,j}$	completion time of job j at stage i,.
$D_{i,j}$	departure time of job j from stage i.

C. MILP FORMULATION

$$(P)\ \min f = \begin{cases} f_1 = \sum_{j \in J} w_j T_j = \sum_{j \in J} w_j \max\left\{C_{|I|,j} - d_j, 0\right\} \\ f_2 = \sum_{i \in I} \sum_{k \in M_i} \alpha E_{i,k} \\ \quad = \sum_{i \in I} \sum_{k \in M_i} \alpha \begin{pmatrix} C_{i,j} x_{i,k,j,0} - C_{i,j} x_{i,k,0,j} \\ - \sum_{j_2 \in J, j_1 \neq j_2} \sum_{j_1 \in J} x_{i,k,j_1,j_2} P_{i,j_2} \end{pmatrix} \end{cases} \tag{1}$$

Subject to:

(1) *Job assignment and sequence constraints*

$$\sum_{j_1 \in J \cup \{0\}, j_1 \neq j_2} \sum_{k \in M_i} x_{i,k,j_1,j_2} = 1, \quad \forall i \in I,\ j_2 \in J \tag{2}$$

$$\sum_{j_1 \in J \cup \{0\}, j_1 \neq j_2} \sum_{k \in M_i} x_{i,k,j_1,j_2} = 1, \quad \forall i \in I,\ j_1 \in J \tag{3}$$

$$\sum_{k \in M_i} x_{i,k,0,j} = 1, \quad \forall i \in I,\ \forall j \in J \tag{4}$$

$$\sum_{k \in M_i} (x_{i,k,j_1,j_2} + x_{i,k,j_2,j_1} = 1, \quad \forall i \in I,\ (j_1, j_2) \in J \cup \{0\} \tag{5}$$

(2) *Job completion constraints*

$$C_{i,j} \geq C_{i-1,j} + P_{i,j}, \quad \forall i \in I \backslash \{1\},\ \forall j \in J \tag{6}$$

(3) *Job departure constraints*

$$C_{i,j} \geq C_{i,j}, \quad \forall i \in I, \ \forall j \in J \tag{7}$$

$$C_{|I|,j} = D_{|I|,j}, \quad j \in J \tag{8}$$

(5) *Buffering constraints*

$$C_{i,j} = D_{i-1,j} + P_{i,j}, \quad i \in I\backslash\{1\}, \ j \in J \tag{9}$$

(6) *Noninterference constraints*

$$D_{i,j_1} + P_{i,j_1} \leq C_{i,j_2} + L\left(1 - x_{i,k,j_1,j_2}\right), \quad \forall i \in I, \ \forall k \in M_i, \ \forall j_1, j_2 \in J \tag{10}$$

(7) *Binary and non-negativity constraints*

$$x_{i,k,j_1,j_2} \in \{0, 1\}, \quad \forall i \in I, \ \forall k \in M_i, \ \forall j_1, j_2 \in \{0\} \cup J \tag{11}$$

$$C_{i,j} \geq 0, \quad \forall i \in I, \ \forall j \in J \tag{12}$$

$$D_{i,j} \geq 0, \quad \forall i \in I, \ \forall j \in J \tag{13}$$

Eq. (1) represents the bi-objective functions: the total weighted tardiness (TWT) and the total price of NPE. Eq. (2)-(5) respectively ensure that each job only has one precedence and successive job assigned on one each machine, each job is assigned to only one machine at each stage, and the sequence of any pair of jobs is unique. Eq. (6) guarantees each job goes through each stage in line. Eq. (7) and (8) indicate that each job can be departed from a stage after it is completed at the current stage. Eq. (9) represents that each job at the current stage starts immediately after this job depart from the previous stage. Eq. (10) expresses that any two jobs are not allowed simultaneously processed on the same machine. Finally, Eq. (11)-(13) impose the binary and nonnegative constraints on decision variables.

IV. PROPOSED ALGORITHM

A. OUTLINE of EOMO

The classic MOEA/D always solve a MOP with G objectives by decomposing it into N single-objective subproblems by choosing N weight vectors $\left(\lambda^1, \ldots, \lambda^N\right)$, where $\lambda^q = \left(\lambda_1^q, \ldots, \lambda_G^q\right)$ and $\sum_{g=1}^{G} \lambda_g^q = 1$. Then, the objective of the q^{th} subproblem can be defined as follows:

$$\min \ F\left(\pi \left|\lambda^q\right.\right) = \sum_{g=1}^{G} \lambda_g^q f_g\left(\pi\right), \quad s.t. \ \pi \in \Pi \tag{14}$$

For each subproblem $q \in \{1, \ldots, N\}$, let $B\left(q\right)$ be the set including the indices of the T closest weight vectors by calculating their Euclidean distances. If $r \in B\left(q\right)$, sub-problem r is called a neighbor of subproblem q. Under the framework of MOEA/D, the proposed EOMO algorithm is improved by adding an external archive population [28] and a local search procedure [29]. The former policy is used to guide the algorithm to converge on the Pareto frontier, and the other one means to maintain a high-diversity internal population. Then we define the following two sets:

- $Q = \left\{\pi^1, \ldots \pi^N\right\}$, where π^q is the best solution found in subproblem q.
- A, which store N solutions selected by non-dominated sorting approach and crowding distance in NSGA-II [32].

The pseudocode of EOMO is detailed in **Algorithm 1**.

Algorithm 1 EOMO

Input:
a) a multi-objective scheduling problem MOP.
b) N: number of subproblems, size of set Q and A.
$\lambda^1, \ldots, \lambda^N$: a set of N weight vectors.
T: size of the neighborhood of each subproblem.
c) Pc, Pm: probability of crossover and mutation of the genetic operator.
d) τ: control parameter of the local search.

Step 1: Initialization:
a) randomly generate an initial population, $Q = \{\pi^1, \ldots, \pi^N\}$, and set $A = Q$.
b) compute the Euclidean distance between any two weight vectors and obtain the T closest weight vectors to each weight vector.

Step 2: New solution generation:
for each $q \in \{1, \ldots, N\}$ **, do**
a) select subproblem r with the probailtiy introduced by Cai et al. [28], and select two indices r_1 and r_2 in $B\left(r\right)$.
b) $\sigma^q \leftarrow GA\left(\pi^{r_1}, \pi^{r_2}, Pc, Pm\right)$, generate a new solution for subproblem r.
end for

Step 3: Population update:
for each $q \in \{1, \ldots, N\}$ **, do**
a) if σ^q is generated from subproblem r, **then**
for each $l \in B\left(r\right)$
if $F\left(z^q \left|\lambda^l\right.\right) \leq F\left(\pi^l \left|\lambda^l\right.\right)$ **then** set $\pi^l = \sigma^q$.
end for
b) Set $E = A \cup Q$, select N best solution for E by NSGA-II approach to replace A.

Step 4: local search:
Apply a local search procedure on the replicated solutions existed in Q.

Step 5: Termination check
If stopping criteria are satisfied, terminate the algorithm and output the non-dominated solutions of A. Otherwise, go to **Step 2**.

B. SOLUTION ENCODING

The full job permutation on each machine can map the completed schedule with known variables x_{i,k,j_1,j_2}, $C_{i,j}$ and $D_{i,j}$, but it is an ineffective way to design neighborhood structures and keep feasibly in when finding a solution for HFS-LB. Refer to the encoding approach stated in [11], [12], and [15], we employ a vector π of the job permutation located in the first stage to represent an completely feasible schedule of HFS-LB. In a vector π, the integer of each bit denotes the job number in J. This encoding scheme can cover all promising solutions as well as the optimal ones, and

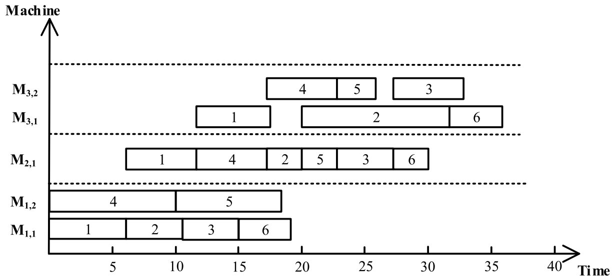

FIGURE 2. Gantt representation by vector [1, 4, 2, 5, 3, 6].

benefit to devise neighborhood operators. According to the given job-permutation vector, we can construct a complete schedule with the first available machine (FAM) rule, as an example shown in Fig.2.

C. SOLUTION DECODING

When decoding a job permutation π to a completely feasible schedule of HFS-LB, there are two issues must be addressed. The first one is that the schedule constructed by FAM rule may suffer from infeasibility due to the limited buffers. The second one is that the NPE is a non-regular objective that also cannot be directly calculated. The TWT is a regular objective [31] that is an increasing function of the completion times in a feasible schedule. As minimizing the TWT, it is appropriate to complete each job with the least delay possible. On the contrary, the NPE objective is not directly related to the completion times in a feasible schedule.

To address above two issues, we apply a two-pass procedure to decode a job permutation vector π into a complete schedule S. In the first pass, we adopt a discrete event system (DES) simulation procedure proposed by Wardono and Fathi [11] to produce a completely feasible schedule $\bar{S}$ for HFS-LB.

Then, all jobs have been assigned and each machine has its own job set $J_{i,k}$. Afterwards, we can reduce the NPE object by two greedy means: (1) delaying completion time $C^H(i,k)$ on machine $M_{i,k}$ of its head job without increasing the completion time $C^T(i,k)$ of its tail job; (2) decreasing $C^T(i,k)$ without increasing $C^H(i,k)$. The second mean is impossible to apply in our method, because all jobs have been scheduled as soon as possible in the first pass. Therefore, we develop a post-shift procedure in the second pass (as shown in **Algorithm 3**) to minimize the NPE objective.

Given a job permutation [1,4,2,5,3,6], the completely feasible schedule decoded by DES procedure is shown **Fig. 3**. Then, the final schedule improved by post-shift procedure is shown in **Fig. 4**.

D. POPULATION INITIALIZATION

The proposed EOMO algorithm begins with an initial population Q has N solutions. A population with a high quality and diversity contributes to the performance of the EOMO. We initialize the population as follows:

Algorithm 2 DES Simulation Procedure

Input: π job permutation at the first stage
Out: a feasible compete schedule $\bar{S}$
Step 1: release all jobs to the shop in the order of π.
Step 2: generate the *arrival* events of all jobs arrived buffer buf_1, and put them into event list Ψ.
Step 3: while $|\Psi| > 0$ **do**
  a) $\psi \leftarrow$ POP (Ψ)
  b) **if** ψ.type $=$ *job_arrival* **then**
    Select the first available machine (FAM) $M_{i,k}$ of ongoing job j at the current stage i.
    if $m_{i,k}$ = null **then**
      Job j waits in buf_i, set ψ.time $\leftarrow \psi$.time + 1, and put it into Ψ.
    Else
      Set $C_{i,j} \leftarrow \psi$.time$+PT_{i,j}$, and generate a departure event ψ', and put it into Ψ.
  c) **if** ψ.type $=$ *job_departure* **then**
    if ψ.stage $< |I|$ **then**
      if buf_{i+1} is full **then**
        Job j stay at stage i, set ψ.time $\leftarrow \psi$.time + 1, and put it into Ψ.
      else
        Job j arrive at buf_{i+1}, set $D_{i,j} \leftarrow \psi$.time, and generate a *arrival* event ψ', and put it into Ψ
      end if
    else
      Set $D_{i,j} \leftarrow \psi$.time
    end if
end while

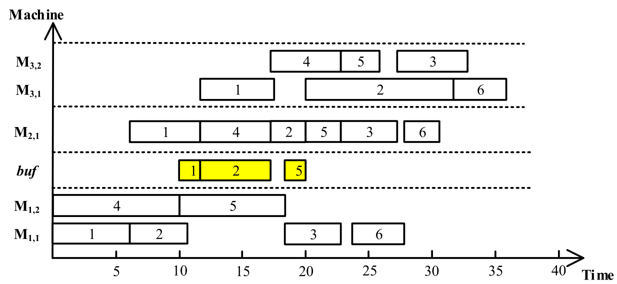

FIGURE 3. Gantt representation decoded by DES procedure.

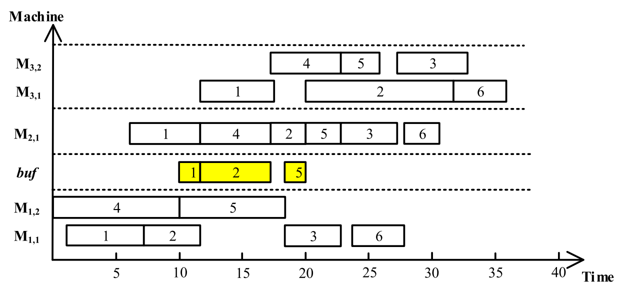

FIGURE 4. A Gantt representation improved by post-shift procedure.

E. GENETIC OPERATOR

Since the job permutation directly affects the performance of the completed solution, genetic operator on the two candidate

Algorithm 3 Post-Shift Procedure

Input: a completely feasible schedule $\bar{S}$
Output: a final schedule S.
Step 1: let $i \leftarrow |I| - 1$.
Step 2: while $i > 0$ **do**
a) **for each** j in J, set $\bar{C}_{i,j} \leftarrow C_{i+1,j} - P_{i+1,j} - W_{i+1,j}$, where $W_{i+1,j}$ is the waiting time in buf_{i+1}.
b) get a set J_i by sorting jobs in J with $\bar{C}_{i,j}$.
c) **while** $|J_i| > 0$ do
$j \leftarrow$ POP (J_i)
if j is the last job on a machine **then**
$C_{i,j} \leftarrow \bar{C}_{i,j}$.
else $C_{i,j} \leftarrow \min\left(\bar{C}_{i,j}, C_{i,j^+} - PT_{i,j^+}\right)$, where j^+ denotes the next job of j on the same machine.
end if
end while
d) $i \leftarrow i - 1$.
end while

Algorithm 4 Population Initialization

Step 1: sequence all jobs with the EWDD (earliest weighted due date) rule, get π_1, and put it into Q.
Step 2: let $Cnt \leftarrow 1$.
Step 3: while $Cnt < N + 1$ **do**
if Cnt mod 2 $= 0$ **then**
Randomly two different position b_1 and b_2 from π_1, and randomly shuffle the job permutation between $[b_1, b_2]$, and get new vector π.
else
Generate a new vector π in a random manner.
end if
put π_1 into Q, and $Cnt \leftarrow Cnt + 1$
end while

solutions must be implemented in an appropriate way. First, the LOX (linear order crossover) operator initially suggested by Falkenauer and Bouffouix [30] are applied to produce new solutions. Then, an interchange is applied for mutation. The pseudocode of genetic operator is detailed in **Algorithm 5**, and an example is illustrated in **Fig. 5**.

F. LOCAL SEARCH PROCEDURE

During the running of the algorithm, EOMO is likely to produce the same solutions. To enlarge the diversity of the population and enhance the deep search ability of EOMO. We apply a local search procedure proposed by Li and Landa-Silva [29]. In this simple procedure, we firstly take a solution π and then apply a single random interchange of jobs, i.e., two jobs are randomly selected, and exchange their positions in the initial sequence. Then the new permutation is decoded again by the DES simulation and post-shift procedure. During the iterations, if the new solution is not dominated by the old one, then output the new one.

Algorithm 5 GA

Input: Crossover probability Pc, mutation probability Pm, two parent solutions π_1 and π_2.
Output: a child solution π.
Step 1: if $rand \le Pc$ do following LOX steps.
a) Select a subsequence of jobs from π_1 at random.
b) Produce a proto-solution π by copying the subsection into the corresponding positions of it.
c) Exclude the jobs which are already in the subsequence from π_2. The resulted sequence of jobs insert the null position of π.
Step 2: if $rand \le Pm$ do following steps.
a) **if** no solution is produced, **then** $\pi \leftarrow \pi_1$.
b) Randomly interchange two different jobs in π.

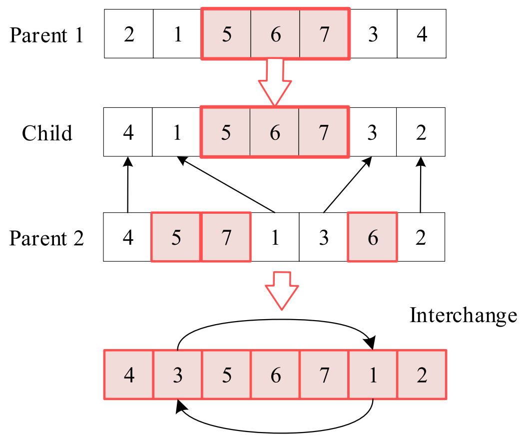

FIGURE 5. An example for genetic operator.

Otherwise, it will be accepted within the probability defined by Eq. (15). The procedure works as the steps stated in **Algorithm 6**.

$$AP\left(\sigma, \sigma', \lambda^q, Te\right) = \min\left(1, \exp\left(-\tau \times \frac{f\left(\sigma', \lambda\right) - f\left(\sigma, \lambda\right)}{Te}\right)\right) \quad (15)$$

Algorithm 6 Local Search

Input: π^q, λ^q, Te
Step 1: $\sigma \leftarrow \pi^q$.
Step 2: repeat
a) $\sigma' \leftarrow$ interchange (σ)
b) **if** $f\left(\sigma'\right)$ is not dominated **by** $f(\sigma)$ **then**
exit the repeat.
end if
c) calculate the acceptance probability $AP\left(\sigma, \sigma', \lambda^q, Te\right)$
d) **if** $AP > rand\,(0, 1)$ **then**.
$\sigma \leftarrow \sigma'$
end if
until Stopping conditions are satisfied.
Output: $\pi^q \leftarrow \sigma$

V. EXPERIMENTS

A. EXPERIMENTAL SETTINGS

In this section, we carried out the computational experiments to validate the performance of the proposed EOMO algorithm, coded all test algorithms by C++ language with Microsoft visual studio 2015, and run on a PC with Intel Core i7 3.4 GHz and 16 GB RAM. To evaluate quantitatively the improvement of the proposed algorithm, we choose following three algorithms reported relevant literature for comparison:

(1) SPGA-II proposed by Abyaneh and Zandieh [14], which simultaneously optimize makespan or total tardiness objectives for HFS-LB scheduling problem.

(2) NSGA-II proposed by Deb *et al.* [32], which is the most popular multi-objective evolutionary algorithm based on non-dominated sorting and elitism strategy.

(3) MOEA/D proposed by Zhang and Li [22], which is the classic version of the proposed algorithm.

The proposed EOMO algorithm and the three compared algorithms adopted the same solution coding and decoding mechanism, and the same stopping criterion. The compared algorithms and the proposed EOMO algorithm differ in following aspects: (1) the first population is randomly initialized; (2) no local search procedure is implemented, (3) the algorithmic parameters are recommended by their original papers.

In the experiments, we produced the test instances adopting the approach given by Pan *et al.* [33]. The processing times were generated with a uniformly distribution $\boldsymbol{U}(1,99)$, and tardiness weights were generated with $\boldsymbol{U}(0.1,1.0)$, the due dates were generated as following equation

$$d_j = \max\left(0, U\left(\lfloor LB\,(1-T-R/2)\rfloor, \lceil LB\,(1-T+R/2)\rceil\right)\right) \quad (16)$$

where LB is the makespan lower bound of a test instance, T is a tardiness factor and R is a due date range. In this paper, we set $T = 0.2$, $R = 0.3$ and the maximum buffer size of each stage set to 1. With respect to the objective function of NPE, we set the energy price as a constant number $\alpha = 1.0$.

B. EVALUATION METRICS

In order to evaluate the performance of the four candidate algorithms on solving MOP, we firstly needed to address following issues [34]: (1) the distance of the obtained Pareto frontier with respect to the true Pareto frontier (assume its location is known), (2) the diversity of object vectors found, (3) the number of Pareto optimal solutions found.

Based on these assumptions, we employed following three metrics to quantitatively evaluate the performance of EOMO.

(1) Modified mean ideal distance (V_{dis}), which is a metric implies the normalized distance from the obtained Pareto frontier to an ideal point [35].

$$V_{dis} = \frac{\sum_{q\in A^*}\sqrt{\left(\left[\left(f_1^q - f_1^{\min}\right)/\Delta f_1\right]^2 + \left[\left(f_2^q - f_2^{\min}\right)/\Delta f_2\right]^2\right)}}{|A^*|} \quad (17)$$

where $\Delta f_i = f_i^{\max} - f_i^{\min}$, A^* is a set of objective vectors located in the obtained Pareto frontier.

(2) Spacing (V_{sp}), which evaluates the diversity of objective vectors throughout the obtained Pareto frontier [36].

$$V_{sp} = \sqrt{\frac{1}{|A^*|-1}\sum_{q\in A^*}\left(\bar{d} - d_q\right)^2} \quad (18)$$

where $d_q = \min_{r\in A^*}\left(\left|f_1^q(\pi) - f_1^r(\pi)\right| + \left|f_2^q(\pi) - f_2^r(\pi)\right|\right)$, $\bar{d}_q$ is the average value of d_q. The metric indicates the smaller value of spacing is, the obtained objective vectors distribute in Pareto frontier more uniformly.

(3) Rate of the Pareto optimal solutions (V_{np}), which provides the information on total number of non-dominated solutions and the size of external archive population.

$$V_{np} = \frac{|A^*|}{|A|} \times 100\% \quad (19)$$

C. PARAMETER SELECTIONS

The algorithmic parameters have significant effects on the quality of the proposed EOMO. To observe the parameter impacts, we focused on the metric V_{dis} since it is most relevant to the optimization quality. In this study, we considered five parameters (N, T, Pc, Pm, τ) on three levels as shown in **Table 1**. Then, we applied the design of experiments (DOE) approach with a Taguchi design $L_{27}\left(3^5\right)$, which only 48 parameter combinations were tested. To calculate various response under different level, we tested the representative instance that scheduling 20 jobs in the HFS with 3 stage and 3 machines per stage. Since the smaller V_{dis} indicated the better solutions set, we analyzed the algorithm quality with a signal-to-noise (S/N) ratio $S/N = -\log_{10}\left(f^2\right)$ which runs in smaller-the-better manner.

TABLE 1. Parameter levels of EOMO.

Parameters	Levels
Population size (ρ)	n*3, **n*5**, n*8,
Number of Neighbors (T)	**10**, 15 20
Crossover rates (Pc)	0.8 0.9 **1.0**
Mutation rates (Pm)	0.1 **0.2** 0.3
Local search rates τ	lg0.25 **lg0.5** lg0.75

The main results of the DOE were shown in **Fig. 6** (plotted by Minitab® 18). As the figure suggests, we tuned the best combination for the level of each parameter: $\rho = 5$, $T = 10$, $Pc = 1.0$, $Pm = 0.2$ and $\tau = \lg 0.5$, because they had the biggest contribution on improving the algorithm performance. Additionally, **Fig 7**. plotted the convergence curve of EOMO when applying the tuned parameters.

TABLE 2. Computational results for test instances (best values are in bold).

No.	$\|J\|\times\|M_i\|\times\|I\|$	EOMO			SPGAII			NSGAII			MOEA/D		
		V_{dis}	V_{sp}	V_{np}	V_{dis}	V_{sp}	V_{np}	V_{dis}	V_{sp}	V_{np}	V_{dis}	V_{sp}	V_{np}
1.	10×2×2	0.65	**11.90**	**6.00**	0.66	24.19	**6.00**	**0.64**	23.83	**6.00**	0.65	23.40	**6.00**
2.	10×3×2	0.63	22.31	**8.00**	0.69	24.32	**8.00**	**0.60**	**19.10**	6.00	0.67	23.58	**8.00**
3.	10×4×2	**0.52**	**18.72**	**8.00**	0.58	19.90	**8.00**	0.79	29.92	6.00	0.57	29.99	**8.00**
4.	10×2×3	**0.44**	**10.14**	**10.00**	0.61	28.40	6.00	0.77	26.40	8.00	0.64	22.68	8.00
5.	10×3×3	0.62	**11.49**	8.00	0.71	39.67	8.00	0.63	19.22	8.00	**0.61**	26.42	**10.00**
6.	10×4×3	**0.55**	25.16	**18.00**	0.60	**19.02**	10.00	0.65	29.37	6.00	0.56	23.35	8.00
7.	15×2×2	**0.56**	**20.43**	**10.67**	0.73	24.53	9.33	0.79	23.54	6.67	0.58	31.13	9.33
8.	15×3×2	0.68	**22.82**	**13.33**	0.62	34.21	**13.33**	**0.55**	38.31	10.67	0.61	23.55	12.00
9.	15×4×2	0.61	21.14	**10.67**	0.62	22.06	6.67	**0.55**	**19.49**	9.33	0.71	40.04	8.00
10.	15×2×3	**0.54**	32.22	**9.33**	0.65	**18.27**	**9.33**	0.71	29.88	**9.33**	0.72	29.52	**9.33**
11.	15×3×3	**0.45**	**16.95**	**13.33**	0.51	32.62	8.00	0.69	20.12	6.67	0.71	24.46	8.00
12.	15×4×3	0.70	**27.39**	**16.00**	**0.60**	39.10	5.33	0.65	27.82	6.67	0.65	28.43	6.67
13.	20×2×2	**0.43**	**16.87**	**12.00**	0.67	24.64	6.00	0.78	23.13	7.00	0.56	29.56	**12.00**
14.	20×3×2	**0.66**	**15.05**	8.00	0.68	30.77	**10.00**	0.69	27.03	8.00	0.70	21.89	9.00
15.	20×4×2	**0.51**	35.38	**10.00**	0.67	37.48	**10.00**	0.67	**20.37**	**10.00**	0.64	28.80	**10.00**
16.	20×2×3	0.62	**19.42**	**14.00**	0.74	27.27	11.00	0.58	25.55	10.00	**0.51**	24.02	11.00
17.	20×3×3	**0.52**	**20.52**	**12.00**	0.57	27.25	8.00	0.56	29.34	8.00	0.54	20.69	8.00
18.	20×4×3	0.64	**13.25**	**11.00**	0.69	31.39	6.00	**0.56**	24.80	9.00	0.71	20.33	6.00
19.	25×2×2	**0.48**	28.38	8.80	0.52	28.27	**11.20**	0.56	**25.33**	**11.20**	0.64	28.88	10.40
20.	25×3×2	**0.53**	21.47	9.60	0.55	28.89	9.60	0.72	22.91	**10.40**	0.57	**18.41**	9.60
21.	25×4×2	**0.63**	**19.68**	8.00	0.72	34.81	**9.60**	0.73	21.67	**9.60**	0.64	24.95	**9.60**
22.	25×2×3	0.66	26.77	9.60	**0.57**	23.46	**12.00**	0.72	**21.72**	**12.00**	0.70	28.16	11.20
23.	25×3×3	**0.39**	**24.21**	**12.00**	0.60	38.37	7.20	0.67	24.50	8.00	0.51	24.39	8.00
24.	25×4×3	**0.56**	24.44	**10.40**	0.65	**22.72**	8.00	0.69	29.11	8.80	0.58	25.61	8.80
Mean		**0.57**	**21.09**	**10.70**	0.63	28.40	8.61	0.66	25.10	8.39	0.62	25.93	8.96

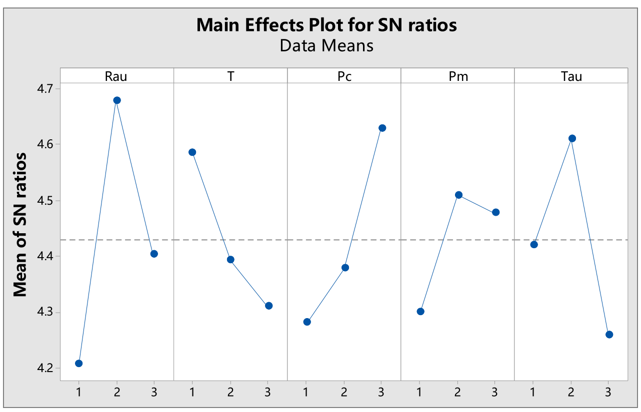

FIGURE 6. The main effects plot corresponding to the control parameters.

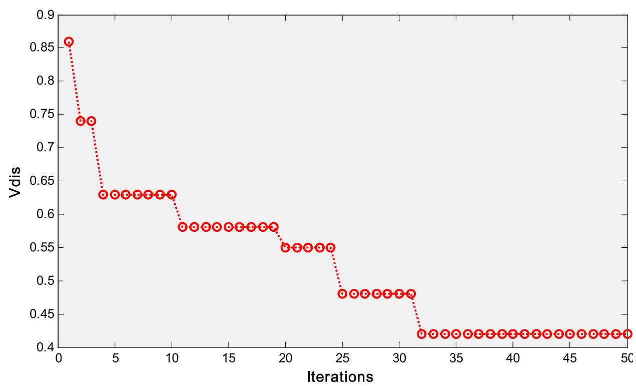

FIGURE 7. The convergence curve of the tuning EOMO.

D. RESULTS AND DISCUSSION

After fabricating the appropriate parameters of EOMO, we conducted the comparison experiments on various test instances. In following experiments, we considered the job size with four levels $|J| \in \{10, 15, 20, 25\}$, two stage numbers $|I| \in \{2, 3\}$ and three machine numbers per stage $|M_i| \in \{2, 3, 4\}$. Then $4 \times 2 \times 3 = 24$ test instances were generated, and all algorithms were terminated when $20 \times N$ solutions had evaluated.

The experimental results in terms of V_{dis}, V_{sp} and V_{np} were reported in **Table 2** and the mean values were listed in its bottom. To analyze how different is the performance between EOMO and other algorithms, we also applied a one-way ANOVA on the adopted metrics, where the four compared algorithms were considered at different levels. Then we calculated the source of variation in performance (Source), the degrees of freedom (DF), the sum of squares (SS), the mean squares (MS), the F statistic (F), and the probability that the F statistic is greater than the critical F (P-value). Similar to the T-tests, we set the

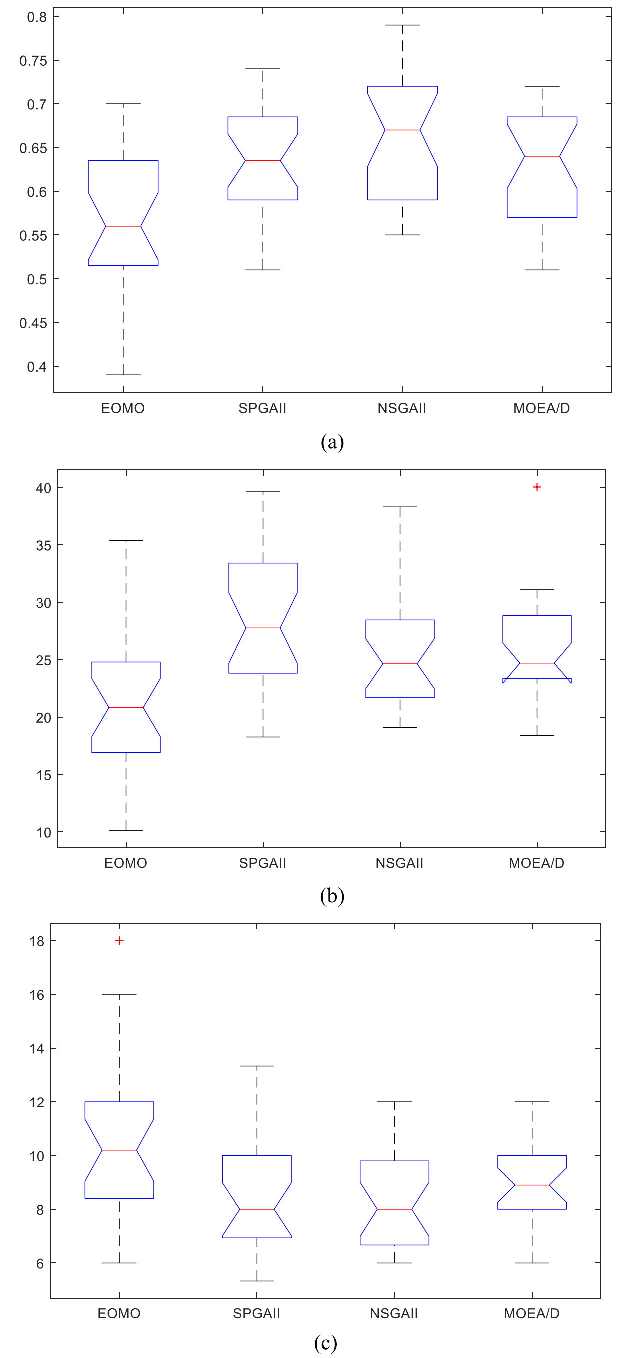

FIGURE 8. Box plot of the compared algorithms. (a) V_{dis}. (b) V_{sp}. (c) V_{np}.

significance level to 0.01. The ANOVA results were listed in **Table 3**.

From the comparison results, we observed the following results: (1) In comparisons of the modified mean ideal distance, the proposed EOMO algorithm had 15 best out of the given 24 test instances, which was significantly better than other compared algorithms. According to the mean value and the statistic of ANOVA, EOMO obtained the minimum value of 0.57 and significantly different with others on V_{dis}, since P-value=0.0002<0.01. Hence, EOMO had the best convergence ability. (2) In comparisons of the spacing, the proposed EOMO algorithms had 15 best out of the given 24 test instances, which was significantly better than other compared algorithms. According to the mean value and the statistic of ANOVA, EOMO obtained minimum value of 21.09 and substantially different with others on V_{sp}, P-value=0.0002<0.01. It means that EOMO had the best diversity. (3) In comparisons of the ratio of non-dominated solutions, the proposed EOMO algorithm obtained 18 best out of the given 24 test instances, which was considerably better than other compared algorithms. According to the mean value and the statistic of ANOVA, EOMO obtained minimum value of 10.70 and significantly different with others on V_{np}, since P-value = 0.0011<0.01. It means that EOMO was able to produce more non-dominated solutions.

TABLE 3. ANOVA table for multi-objective metrics.

Metric	Source	DF	SS	MS	F	P-value
	Group	3	0.12266	0.04089	7.35	0.0002
V_{dis}	Error	92	0.51153	0.00556		
	Total	95	0.63418			
	Group	3	664.07	221.356	7.18	0.0002
V_{sp}	Error	92	2836.86	30.835		
	Total	95	3500.93			
	Group	3	79.282	26.427	5.84	0.0011
V_{np}	Error	92	416.455	4.5267		
	Total	95	495.737			

VI. CONCLUSION

In this paper, we have investigated the scheduling problem for HFS-LB considering the TWT and the NPC objectives. To solve this bi-objective scheduling problem effectively, we devised an efficient multi-objective optimization algorithm based on decomposition. In the outline of the EOMO, we devised a job-permutation vector to represent a completed schedule, which can easily cover the solution space and can easily construct a feasible completed solution. Then, we developed a two-pass decoding procedure, where the first one calculated the TWT objective by DES simulation and another calculated the NPE objective by the greedily post-shift procedure. Besides, we applied two enhancing strategies including an external archive population to guide the algorithm to converge on the Pareto frontier, and a local search procedure to enhance the diversity of the internal population. Based on the randomly synthetic experimental comparisons, we can conclude that the proposed algorithm exhibits high capabilities in handling the multi-objective scheduling problem in HFS-LB.

The proposed scheduling model and algorithm are potentially applicable to a wide range of manufacturing systems since the HFS-LB is a typical production layout with strong industrial background. We believe that scheduling HFS-LB with energy considerations is an important research topic, which is a very meaningful tool to achieve the sustainable manufacturing goal of the energy-intensive companies. Our future research topics including following two aspects:

(1) Extending the proposed HFS-LB scheduling model to dynamic or uncertain environments. Because a large variety of unforeseen events frequently occur, that all parameters are known is an ideal assumption. Therefore, how to keep the model feasible and stable under dynamic or uncertain environments is a very challengeable issue.

(2) Further enhancing search efficiency by exploiting the structure characteristics. Because critical path, job block, bottleneck and other structure characteristics has important effects on optimal objectives. Therefore, how to use them to speed up the optimization algorithm is another vital task for improving our algorithm.

REFERENCES

[1] J. Li, P. Duan, H. Sang, S. Wang, Z. Liu, and P. Duan, "An efficient optimization algorithm for resource-constrained steelmaking scheduling problems," *IEEE Access*, vol. 6, pp. 33883–33894, 2018.

[2] J. T. Lin and C.-M. Chen, "Simulation optimization approach for hybrid flow shop scheduling problem in semiconductor back-end manufacturing," *Simul. Model. Pract. Theory*, vol. 51, pp. 100–114, Feb. 2015.

[3] C. Gicquel, L. Hege, M. Minoux, and W. Van Canneyt, "A discrete time exact solution approach for a complex hybrid flow-shop scheduling problem with limited-wait constraints," *Comput. Oper. Res.*, vol. 39, no. 3, pp. 629–636, 2012.

[4] R. Ruiz and J. A. Vázquez-Rodríguez, "The hybrid flow shop scheduling problem," *Eur. J. Oper. Res.*, vol. 205, no. 1, pp. 1–18, 2010.

[5] I. Ribas, R. Leisten, and J. M. Framiñan, "Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective," *Comput. Oper. Res.*, vol. 37, no. 8, pp. 1439–1454, 2010.

[6] L. Tang, J. Guan, and G. Hu, "Steelmaking and refining coordinated scheduling problem with waiting time and transportation consideration," *Comput. Ind. Eng.*, vol. 58, no. 2, pp. 239–248, 2010.

[7] C. Gahm, F. Denz, M. Dirr, and A. Tuma, "Energy-efficient scheduling in manufacturing companies: A review and research framework," *Eur. J. Oper. Res.*, vol. 248, no. 3, pp. 744–757, 2016.

[8] S. Zhao, I. E. Grossmann, and L. Tang, "Integrated scheduling of rolling sector in steel production with consideration of energy consumption under time-of-use electricity prices," *Comput. Chem. Eng.*, vol. 111, pp. 55–65, Mar. 2018.

[9] T. Sawik, "An exact approach for batch scheduling in flexible flow lines with limited intermediate buffers," *Math. Comput. Model.*, vol. 36, nos. 4–5, pp. 461–471, 2002.

[10] L. Tang and H. Xuan, "Lagrangian relaxation algorithms for real-time hybrid flowshop scheduling with finite intermediate buffers," *J. Oper. Res. Soc.*, vol. 57, no. 3, pp. 316–324, 2006.

[11] B. Wardono and Y. Fathi, "A tabu search algorithm for the multi-stage parallel machine problem with limited buffer capacities," *Eur. J. Oper. Res.*, vol. 155, no. 2, pp. 380–401, 2004.

[12] X. Wang and L. Tang, "A tabu search heuristic for the hybrid flowshop scheduling with finite intermediate buffers," *Comput. Oper. Res.*, vol. 36, no. 3, pp. 907–918, 2009.

[13] V. Yaurima, L. Burtseva, and A. Tchernykh, "Hybrid flowshop with unrelated machines, sequence-dependent setup time, availability constraints and limited buffers," *Comput. Ind. Eng.*, vol. 56, no. 4, pp. 1452–1463, 2009.

[14] S. H. Abyaneh and M. Zandieh, "Bi-objective hybrid flow shop scheduling with sequence-dependent setup times and limited buffers," *Int. J. Adv. Manuf. Technol.*, vol. 58, nos. 1–4, pp. 309–325, 2012.

[15] J. Li and Q.-K. Pan, "Solving the large-scale hybrid flow shop scheduling problem with limited buffers by a hybrid artificial bee colony algorithm," *Inf. Sci.*, vol. 316, pp. 487–502, Sep. 2015.

[16] M. Rager, C. Gahm, and F. Denz, "Energy-oriented scheduling based on evolutionary algorithms," *Comput. Oper. Res.*, vol. 54, pp. 218–231, Feb. 2015.

[17] M. Dai, D. B. Tang, A. Giret, M. A. Salido, and W. D. Li, "Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm," *Robot. Comput.-Integr. Manuf.*, vol. 29, no. 5, pp. 418–429, 2013.

[18] H. Luo, B. Du, G. Q. Huang, H. Chen, and X. Li, "Hybrid flow shop scheduling considering machine electricity consumption cost," *Int. J. Prod. Econ.*, vol. 146, no. 2, pp. 423–439, 2013.

[19] D. Tang, M. Dai, M. A. Salido, and A. Giret, "Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization," *Comput. Ind.*, vol. 81, pp. 82–95, Sep. 2016.

[20] J.-Q. Li, H.-Y. Sang, Y.-Y. Han, C.-Q. 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.

[21] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 32–49, 2011.

[22] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[23] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, May 2009, pp. 203–208.

[24] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, Apr. 2009.

[25] A. Alhindi and Q. Zhang, "MOEA/D with Tabu search for multiobjective permutation flow shop scheduling problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2014, pp. 1155–1164.

[26] X. Li and M. Li, "Multiobjective local search algorithm-based decomposition for multiobjective permutation flow shop scheduling problem," *IEEE Trans. Eng. Manag.*, vol. 62, no. 4, pp. 544–557, Nov. 2015.

[27] Y. Liu, H. B. Dong, N. Lohse, S. Petrovic, and N. Gindy, "An investigation into minimising total energy consumption and total weighted tardiness in job shops," *J. Cleaner Prod.*, vol. 65, pp. 87–96, Feb. 2014.

[28] X. Cai, Y. Li, Z. Fan, and Q. Zhang, "An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 508–523, Aug. 2015.

[29] H. Li and D. Landa-Silva, "An adaptive evolutionary multi-objective approach based on simulated annealing," *Evol. Comput.*, vol. 19, no. 4, pp. 561–595, 2011.

[30] E. Falkenauer and S. Bouffouix, "A genetic algorithm for job shop," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, Apr. 1991, pp. 824–829.

[31] Y. Mati, S. Dauzère-Pérès, and C. Lahlou, "A general approach for optimizing regular criteria in the job-shop scheduling problem," *Eur. J. Oper. Res.*, vol. 212, no. 1, pp. 33–42, 2011.

[32] 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.

[33] Q.-K. Pan, R. Ruiz, and P. Alfaro-Fernández, "Iterated search methods for earliness and tardiness minimization in hybrid flowshops with due windows," *Comput. Oper. Res.*, vol. 80, pp. 50–60, Apr. 2017.

[34] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 256–279, Jun. 2004.

[35] E. Ahmadi, M. Zandieh, M. Farrokh, and S. M. Emami, "A multi objective optimization approach for flexible job shop scheduling problem under random machine breakdown by evolutionary algorithms," *Comput. Oper. Res.*, vol. 73, pp. 56–66, Sep. 2016.

[36] J. R. Schott, "Fault tolerant design using single and multicriteria genetic algorithm optimization," M.S. thesis, Dept. Aeronaut. Astronaut., Massachusetts Inst. Technol., Cambridge, MA, USA, May 1995.

SHENG-LONG JIANG (M'16) received the B.S. and M.S. degrees from the College of Materials Science and Engineering, Chongqing University, Chongqing, China, in 2004 and 2007, respectively, and the Ph.D. degree from the Department of Automation, Tsinghua University, Beijing, China, in 2016.

His current research interests mainly include modeling, simulation, and optimization of manufacturing systems.

LONG ZHANG received the M.S. degree from the Guilin University of Electronic Technology and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences.

His research interests mainly include modeling, intelligent scheduling, and optimization of complex manufacturing systems.