Multiobjective Optimization of Energy-Efficient JOB-Shop Scheduling With Dynamic Reference Point-Based Fuzzy Relative Entropy

Lijun He, Raymond Chiong, *Senior Member, IEEE*, Wenfeng Li, *Senior Member, IEEE*, Sandeep Dhakal, Yulian Cao, *Member, IEEE*, and Yu Zhang

***Abstract*—Energy-efficient production scheduling research has received much attention because of the massive energy consumption of the manufacturing process. In this article, we study an energy-efficient job-shop scheduling problem with sequence-dependent setup time, aiming to minimize the makespan, total tardiness and total energy consumption simultaneously. To effectively evaluate and select solutions for a multiobjective optimization problem of this nature, a novel fitness evaluation mechanism (FEM) based on fuzzy relative entropy (FRE) is developed. FRE coefficients are calculated and used to evaluate the solutions. A multiobjective optimization framework is proposed based on the FEM and an adaptive local search strategy. A hybrid multiobjective genetic algorithm is then incorporated into the proposed framework to solve the problem at hand. Extensive experiments carried out confirm that our algorithm outperforms five other well-known multiobjective algorithms in solving the problem.**

***Index Terms*—Energy-efficient job shop scheduling, fitness evaluation mechanism (FEM), fuzzy relative entropy (FRE), genetic algorithm (GA), multiobjective optimization, sequence-dependent setup time (SDST).**

I. Introduction

ENERGY is vital for the development of human society. The rapid increase in energy consumption, which leads to the release of greenhouse gases, such as CO_2 and NO_2, has a major impact on the global environment. As the basis of the economies of most countries, the manufacturing industry is responsible for almost half of the global energy consumption [1]–[3], and thereby the most greenhouse gas emissions. Therefore, the reduction in the energy consumption of manufacturing companies is of great practical significance.

Research has shown that optimized production scheduling can reduce energy consumption in the manufacturing process with a low financial cost [3]–[6]. In recent years, there has been a growing interest in energy-efficient production scheduling research, with a focus on machine on/off control [4], [7], speed scaling [5], [8], and time-of-use electricity pricing [9], [10]. However, it should be noted that existing studies mainly focus on processing energy consumption of machines, while neglecting the energy consumed during setup.

In this era of Industry 4.0, customers are increasingly interested in personalized customization requirements. The diversified product needs of customers result in significant differences among products in terms of shape, size, color, and processing technology requirements. These differences lead to frequent sequence-dependent setup activities between two adjacent jobs processed on the same machine, such as job loading and unloading, equipment cleaning, and tool replacement [11]–[13]. Frequent setup activities lead to large sequence-dependent setup time (SDST) and energy consumption, prolong the manufacturing cycle, and bring some new constraints to the scheduling problem. Hence, it is of great importance to consider SDST in energy-efficient scheduling.

The job-shop scheduling problem (JSP)is one of the classical production scheduling problems [14], [15], and many real-world production systems adopt the job-shop layout. Job-shop scheduling with energy related objectives has been studied widely in recent years. Zhang and Chiong [5] proposed a multiobjective genetic algorithm (GA) to minimize total weighted tardiness and total energy consumption of the JSP within a speed scaling framework. Liu *et al.* [15] also used a non-dominated GA to solve a biobjective JSP by taking total electricity consumption into consideration. Salido *et al.* [16] studied an energy-efficient JSP with speed scaling and solved it using a GA. However, to the best of our knowledge, no prior studies have considered energy-efficient JSPs with SDST (EJSP-SDST). Given its strongly *NP*-hard nature, effective methods are needed to deal with the EJSP-SDST.

Manuscript received September 30, 2020; revised December 16, 2020 and January 5, 2021; accepted January 15, 2021. Date of publication February 3, 2021; date of current version September 29, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61571336 and Grant 71874132, and in part by Fundamental Research Funds for the Central Universities under Grant 2019-YB-033. Paper no. TII-20-4592. *(Corresponding author: Wenfeng Li.)*

Lijun He, Wenfeng Li, and Yu Zhang are with the School of Logistics Engineering, Wuhan University of Technology, Wuhan 430063, China (e-mail: helj@whut.edu.cn; liwf@whut.edu.cn; sanli@whut.edu.cn).

Raymond Chiong and Sandeep Dhakal are with the School of Electrical and Computing, The University of Newcastle, Callahan, NSW 2308, Australia (e-mail: raymond. chiong@newcastle.edu.au; sandeep. dhakal@newcastle.edu.au).

Yulian Cao is with the School of Aviation, University of New South Wales, Sydney, NSW 2052, Australia (e-mail: yulian.cao@unsw.edu.au).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2021.3056425.

Digital Object Identifier 10.1109/TII.2021.3056425

Nature-inspired population-based intelligent algorithms, such as the GA [5] and particle swarm optimization (PSO) [17]–[19], are widely utilized to address various multiobjective optimization problems (MOPs). A key factor affecting the performance of multiobjective optimization algorithms is the fitness evaluation mechanism (FEM) used [20], which involves dealing with the conflicting relationship among multiple objectives, and the evaluation and selection of solutions. A good FEM contributes to the fast convergence of a multiobjective algorithm.

Mainstream FEMs for MOPs can be divided into three categories. The first is scalar-based FEMs; the weighted sum is the most used scalar aggregation function, and the multiobjective evolutionary algorithm based on decomposition (MOEA/D) [21] is the most representative algorithm based on the weighted-sum approach. However, it is difficult to assign relatively impartial weights to the objectives in this weighted-sum approach. Moreover, it has been identified that specification of weights may impact the distribution of the solution set [22], which is an important metric for evaluating a multiobjective algorithm's performance. The second category is Pareto dominance FEMs, which employ the dominance concept to evaluate and sort solutions. The nondominated sorting GA (NSGA)-II [23] is the most popular Pareto dominance-based multiobjective algorithm. It should be noted that many Pareto-based algorithms are time-consuming for large-scale problems; and when the number of objectives increases, their performance decreases rapidly [20]. Additionally, decision makers often need one preferred solution for a real-world engineering optimization problem, but selecting a desired solution from the final solution set obtained by Pareto-based methods is challenging. The third category is indicator-based FEMs, which utilize performance indicators such as the generational distance and hypervolume (HV) as fitness values to guide the search and sort the solutions [24], [25]. However, the calculation of performance indicators is very complex, resulting in low search efficiency [25].

Given the above, we study the EJSP-SDST, and the main contributions of this article can be summarized as follows.

1) A new mathematical model of EJSP-SDST is formulated for the first time in the literature, in order to simultaneously minimize the makespan, total tardiness, and total energy consumption.
2) A novel FEM based on fuzzy relative entropy (FRE) is developed. To implement this FEM, three kinds of points based on objective values are constructed to obtain the fuzzy sets.
3) An FRE-based multiobjective optimization framework is developed to address the EJSP-SDST by incorporating an adaptive local search (ALS) strategy.
4) A hybrid multiobjective GA (HMOGA) is proposed based on the above optimization framework. The performance of HMOGA is verified by comparing it with five well-known multiobjective algorithms, namely the NSGA-II [23], strength Pareto evolutionary algorithm (SPEA)-II [26], NSGA-III [27], MOEA/D [21], and multiobjective PSO (MOPSO) [28]. Experimental results show that our HMOGA is able to outperform the above algorithms in solving the EJSP-SDST.

The rest of this article is organized as follows. Section II presents the definition and model for the EJSP-SDST. Section III discusses the development of FRE as a new FEM for MOPs. Section IV presents the multiobjective optimization framework based on FRE. The extensive experiments conducted to investigate the performance of HMOGA are discussed in Section V. Finally, Section VI concludes this article.

II. Problem Formulation

A. Problem Description

The EJSP-SDST can be formulated as follows. Let there be a set of n jobs that should be processed on a set of m machines. Each job has its own processing path, i.e., the machines used and their sequence both might be different for every job. Each job may need to be processed only on a fraction of m machines, not all of them. The basic processing time for each operation within a job is known in advance. There is a finite and discrete speed set for each machine to process an operation. A machine cannot be turned off completely unless all its scheduled operations have been completed. There is a certain amount of standby energy consumed during the idle period of each machine. The corresponding setup operation can be started only when the job arrives, which means the SDST in this article is nonanticipatory [11]–[13]. In addition, the setup operation leads to the consumption of a large amount of setup energy. Every job has a given due date.

Current energy-efficient production scheduling research mainly focuses on machine ON/OFF control, speed scaling, and time-of-use electricity pricing. Most of the existing studies on machine ON/OFF control and time-of-use electricity pricing assume that the processing speed of machines is static, which is unrealistic in a modern production environment. Computer numerical control machines are widely used in modern manufacturing, where the processing speeds of machines are continuously adjustable within a range. Moreover, machine ON/OFF control is not applicable for some real-world manufacturing systems because of the additional energy required for frequently restarting the machines and the possibility of damaging them. In this article, in order to facilitate modeling and encoding, we use a speed scaling mechanism, i.e., each machine has a finite and discrete speed set, which is a special case of continuously adjustable speed. Speed scaling has been widely used in recent energy-efficient production scheduling studies (e.g., see [5], [8]).

The aim of EJSP-SDST is, therefore, to find a reasonable schedule consisting of production operations and machine speeds, such that environmental and economic objectives are optimized simultaneously. The following assumptions are made as follows.

1) All the machines and operations are instantaneously available at time zero.
2) Each machine can perform only one operation at a time, and each operation can be processed only once on one machine.
3) There are no precedence constraints among the operations of different jobs.

4) Once an operation has begun on a machine, it cannot be interrupted. Pre-emption is not considered.
5) The speed of a machine cannot be changed once it begins to process an operation.

B. Mathematical Modeling

1) Indices:

i, h, i' Index of jobs.
j, j' Index of operations.
k, k' Index of machines.
z Index of machine speed levels.

2) Parameters:

n Total number of jobs.
m Total number of machines.
l_i Total number of operations of job *i*, $l_i \leq m$.
L Total number of speed levels of each machine.
O_{ij} *j*th operation of job *i*.
t_{ij} Processing time of operation O_{ij}.
v_z *z*th speed of each machine
PM_{kz} Processing power of machine k when its machine speed is set to v_z.
PS_k sTandby power of machine k.
PST_k Setup power of machine k.
ST_{ihk} SDST for processing job h immediately after job i on the same machine *k*.
ST_{0hk} Setup time of job *h* on machine *k* when job *h* is the first job processed on machine *k*.
D_i Due date of job i.
H Large enough integer.

3) Variables:

C_{ij} Completion time of operation O_{ij}.
C_i Completion time of job i.
C^k Completion time of machine k.
$C_{\max}$ Completion time of all jobs.

$$y_{i'ik} = \begin{cases} 1, & \text{if job } i \text{ is processed on machine} \\ & \quad k \text{ immediately following job } i' \\ 0, & \text{otherwise} \end{cases}$$

$$\gamma_{ik'k} = \begin{cases} 1, & \text{if machine } k' \text{ processes job } i \text{ before machine } k \\ 0, & \text{otherwise} \end{cases}$$

$$z_{ijkz} = \begin{cases} 1, & \text{if the } z^{th} \text{ speed of machine} \\ & \quad k \text{ is selected for operation } O_{ij} \quad . \\ 0, & \text{otherwise} \end{cases}$$

4) Objective Functions and Constraints:

$$f_1 = \max\{C_i \mid i = 1,\ 2,\ \ldots, n\} \tag{1}$$

$$f_2 = \sum_{i=1}^{n} \max\{0, C_i - D_i\} \tag{2}$$

$$f_3 = E_1 + E_2 + E_3 \tag{3}$$

$$E_1 = \sum_{i=1}^{n}\sum_{j=1}^{l_i}\sum_{k=1}^{m}\sum_{z=1}^{L} z_{ijkz}\frac{t_{ij}}{v_z}PM_{kz} \tag{4}$$

$$E_2 = \sum_{k=1}^{m}\sum_{i=0,h=1}^{n} y_{ihk}ST_{ihk}PST_k \tag{5}$$

$$E_3 = \sum_{k=1}^{m}[C^kPS_k - \sum_{i=1}^{n}\sum_{j=1}^{l_i}\sum_{z=1}^{L} z_{ijkz}\frac{t_{ij}}{v_z}PS_k - \sum_{i=0,h=1}^{n} y_{ihk}ST_{ihk}PS_k] \tag{6}$$

S.t.

$$C_{i'j'} + \frac{t_{ij}}{v_z}z_{ijkz} + ST_{i'ik}y_{i'ik} \leq C_{ij} + (1 - y_{i'ik})\,H \tag{7}$$

where $i = 12, \ldots, n;\ i' = 01, 2, \ldots, n;\ \ i \neq i';\ j = 12, \ldots, l_i;$ $j' = 12, \ldots, l_{i'}\ z = 12, \ldots, L$

$$C_{i(j-1)} + \frac{t_{ij}}{v_z}z_{ijkz} + ST_{i'ik}y_{i'ik} \leq C_{ij} + (1 - \gamma_{ik'k})\,H \tag{8}$$

where $i = 12, \ldots, n;$ $i' = 01, 2, \ldots, n;$ $i \neq i';$ $k, k' = 12, \ldots, m;\ k \neq k';\ j = 2, \ldots, l_i;\ z = 12, \ldots, L$

$$\frac{t_{ij}}{v_z}z_{ijkz} + ST_{0ik}y_{0ik} \leq C_{ij} + (1 - y_{0ik})\,H \tag{9}$$

where $i = 12, \ldots, n;$ $k = 12, \ldots, m;$ $j = 2, \ldots, l_i;$ $z = 12, \ldots, L$

$$\sum_{i'=0}^{n} y_{i'ik} = 1 \tag{10}$$

where $i = 12, \ldots, n;\ i \neq i';\ k = 12, \ldots, m$

$$\sum_{i=1}^{n} y_{i'ik} = 1 \tag{11}$$

where $i' = 01, \ldots, n;\ i \neq i';\ k = 12, \ldots, m$

$$\sum_{z=1}^{L} z_{ijkz} = 1 \tag{12}$$

where $i = 12, \ldots, n;\ j = 12, \ldots, l_i;\ k = 12, \ldots, m.$

Equation (1) is the makespan, i.e., the maximum completion time for all jobs. Equation (2) is the sum of the tardiness of all jobs. (3) is the total energy consumption of all machines, and is composed of three parts: processing energy; standby energy; and setup energy, represented by (4)–(6), respectively. Constraint (7) ensures that the processing of an operation can be started on a machine only when the previous operation on the machine and relevant setup activity have finished. Constraint (8) states the precedence relationship for operations of each job such that the completion time of an operation must be larger than that of the preceding operation considering processing and setup times. Constraint (9) specifies the precedence relationship for the operations of each job when that job is the first to be processed on the machine. Constraint (10) specifies that a job must follow one and only one predecessor except when it is the first job on the machine. Constraint (11) means that when a job has finished processing on a machine, one and only one different job can be selected for processing next. Constraint (12) implies that only one speed setting of a machine can be selected when an operation is being processed on it, i.e., the machine speed setting cannot be changed during processing of the operation.

III. Proposed FEM

A. Basic FRE

Suppose that there are two *M*-dimensional fuzzy sets $U(A) = \{\mu_1(A), \mu_2(A), \ldots, \mu_M(A)\}$ and $U(B) = \{\mu_1(B), \mu_2(B), \ldots, \mu_M(B)\}$, where *A* is a reference fuzzy set and *B* is a comparison fuzzy set. $\mu_i(A)$ and $\mu_i(B)$ are the *i*th membership values of *A* and *B*, respectively. $0 \le \mu_i(A) \le 1$, $0 \le \mu_i(B) \le 1$. Then, the similarity relationship between $U(A)$ and $U(B)$ is determined as follows [29], [30].

1) Calculate the Information Entropies of $U(A)$ and $U(B)$:

$$E(A) = -K\sum_{i=1}^{M}\{\mu_i(A)\ln\mu_i(A) + [1 - \mu_i(A)]\ln[1 - \mu_i(A)]\} \tag{13}$$

$$E(B) = -K\sum_{i=1}^{M}\{\mu_i(B)\ln\mu_i(B) + [1 - \mu_i(B)]\ln[1 - \mu_i(B)]\} \tag{14}$$

where $K = 1/(M\ln 2)$ represents a normalization factor.

2) Calculate the Partial Entropy of $U(A)$ Involving $U(B)$ ($E_B(A)$) and the Partial Entropy of $U(B)$ Involving $U(A)$ ($E_A(B)$):

$$E_B(A) = -\sum_{i=1}^{M}\{\mu_i(B)\ln\mu_i(A) + [1 - \mu_i(B)]\ln[1 - \mu_i(A)]\} \tag{15}$$

$$E_A(B) = -\sum_{i=1}^{M}\{\mu_i(A)\ln\mu_i(B) + [1 - \mu_i(A)]\ln[1 - \mu_i(B)]\}. \tag{16}$$

3) Calculate the FRE Between $U(A)$ and $U(B)$:

$$E(A,B) = E_B(A) + E_A(B). \tag{17}$$

4) Calculate the FRE Coefficient (Denoted As $C_e(A,B)$) Between $U(A)$ and $U(B)$:

$$C_e(A,B) = \frac{1}{K}\frac{E(A)+E(B)}{E(A,B)} \tag{18}$$

where $0 \le C_e(A,B) = C_e(B,A) \le 1$, $C_e(A,B) = C_e(B,A) = 1$ if and only if $U(A) = U(B)$.

With the above steps, the overall difference between $U(A)$ and $U(B)$ can be evaluated effectively. The C_e value is regarded as a measure of the similarity between $U(A)$ and $U(B)$ [29], [30]. A greater C_e means a higher similarity between $U(A)$ and $U(B)$.

B. Principles of FRE-based FEM

Each solution in an MOP has multiple objective function values that can construct a comparison point (CP). Suppose that there is a high-quality reference point (RP), which is the reference objective value sequence. The RP and CP can be converted into a reference fuzzy set and a comparison fuzzy set, respectively, by a membership function. Then, the similarity between the reference and comparison fuzzy sets can be evaluated by FRE. High similarity signifies that the CP is closer to the RP, which means that the corresponding solution is better. We can, therefore, utilize the similarity value to evaluate solutions for the MOP. However, it is worth noting that there are no fuzzy sets in general MOPs, and how to link an MOP with fuzzy sets is a challenging issue. In the following subsections, we provide a detailed discussion of how to build the two kinds of fuzzy sets, and then propose the FRE-based FEM.

C. Dynamic Reference Point and Maximum Point

For many realistic MOPs, it is very difficult to obtain an ideal *RP*. In this article, we construct a dynamic RP and a maximum point (MP) to obtain fuzzy sets. Detailed construction steps are as follows.

1) For a solution X_j^g in the current population *P*, its objective values are used to construct a *CP*, referred to as $Y(X_j^g) = \{f_1(X_j^g), \ldots, f_i(X_j^g), \ldots, f_M(X_j^g)\}$, where *M* is the number of objectives; $f_i(X_j^g)$ is the *i*th objective value of the *j*th solution; $j = 12, \ldots, N$ (*N* is the number of solutions in *P*); and $g = 12, \ldots, G$ (*G* is the maximum number of generations).
2) All solutions are sorted in ascending order of the *i*th objective's function value. The best and worst function values are obtained and referred to as $f_{i,\min}^g$ and $f_{i,\max}^g$, respectively. Then, dynamic *RP* $Y_R^g = \{f_{1,\min}^g, \ldots, f_{i,\min}^g, \ldots, f_{M,\min}^g\}$ and dynamic *MP* $Y_{\text{Max}}^g = \{f_{1,\max}^g, \ldots, f_{i,\max}^g, \ldots, f_{M,\max}^g\}$ are constructed.
3) For the subsequent $g+1$ generation population, repeat steps 1) and 2) to rank all solutions and achieve the best and worst function values of the *i*th objective, namely $f_{i,\min}^{g+1}$ and $f_{i,\max}^{g+1}$.
4) For $f_{i,\min}^{g+1}$, if $f_{i,\min}^{g+1} > f_{i,\min}^g$, replace $f_{i,\min}^{g+1}$ by $f_{i,\min}^g$; otherwise, retain $f_{i,\min}^{g+1}$. For $f_{i,\max}^{g+1}$, if $f_{i,\max}^{g+1} < f_{i,\max}^g$, replace $f_{i,\max}^{g+1}$ by $f_{i,\max}^g$; otherwise, retain $f_{i,\max}^{g+1}$.

With the above steps, our new dynamic *RP* and *MP* can be generated in each generation of the population.

D. Dynamic Reference Fuzzy Set and Comparison Fuzzy Set

The dynamic RP and CP are expected to be converted into fuzzy sets. Detailed steps are as follows.

1) To realize fuzzification, we construct the following lower and upper bounds for each objective

$$f_{i,lb}^{g} = \alpha f_{i,\min}^{g} \tag{19}$$

$$f_{i,ub}^{g} = \beta f_{i,\max}^{g} \tag{20}$$

where α and β are the lower and upper bound factors, respectively. $0 < \alpha \leq 1$ and $\beta > 1$. $i = 1, \ldots, M$.

2) For the *j*th CP $Y(X_j^g)$, its *i*th objective value $f_i(X_j^g)$ is converted into a membership value $\mu\{f_i(X_j^g)\}$ by a semitrapezoidal membership function

$$\mu\left\{f_i\left(X_j^g\right)\right\} = \begin{cases} 1, & f_i\left(X_j^g\right) \leq f_{i,lb}^{g} \\ \dfrac{f_i\left(X_j^g\right) - f_{i,ub}^{g}}{f_{i,lb}^{g} - f_{i,ub}^{g}}, & f_{i,lb}^{g} < f_i\left(X_j^g\right) < f_{i,ub}^{g}. \\ 0, & f_i\left(X_j^g\right) \geq f_{i,ub}^{g} \end{cases} \tag{21}$$

With (21), each CP $Y(X_j^g)$ is converted into a comparison fuzzy set $\mathrm{FS}(Y(X_j^g))$. Likewise, the dynamic RP Y_R^g is converted into a reference fuzzy set $\mathrm{FS}(Y_R^g)$.

E. Solution Evaluation and Sorting

Afterwards, the C_e value between each comparison fuzzy set and the reference fuzzy set can be extracted with the basic FRE method. A larger C_e value implies higher similarity between the CP and dynamic RP, and vice-versa. A solution corresponding to the CP with larger C_e is deemed to be the better solution. We can, therefore, integrate the FRE-based FEM into a specific algorithm and utilize C_e as the fitness value to sort and select solutions in an MOP. The solution with a larger C_e value ranks higher and is prioritized more for being selected for subsequent operations.

IV. Multiobjective Optimization Framework

In this section, an FRE-based multiobjective optimization framework is developed for the EJSP-SDST. Because most algorithms have the tendency of falling into local optima, achieving a balance between exploration (global search) and exploitation (local search) is a common problem that should be considered in designing an algorithm. Therefore, we adopt an ALS strategy. The proposed multiobjective optimization framework is presented in Fig. 1, where *P*, *G*, *H*, *R*, *O*, and *NP* are populations during the execution process. Solutions in the final external archive are output when the termination condition is satisfied.

A. Population Initialization and Encoding

In contrast to the traditional JSP, a solution X of EJSP-SDST has two parts; more specifically, $X = [\pi, v]$, where π represents the production operation and v the machine speed assignment.

For π, an operation-based encoding method is used. We first generate N candidate solutions of length $Nu = \sum_{i=1}^{n} l_i$. Each dimension of a candidate solution is in the range of (0, 10). We rank all the dimensions of each solution in ascending order. By scanning the permutation from left to right, the *j*th occurrence of a job number *i* refers to the *j*th operation in the technological sequence of this job *i*. Each job number occurs as many times as the number of its operations. When this is the case, the solution is always feasible.

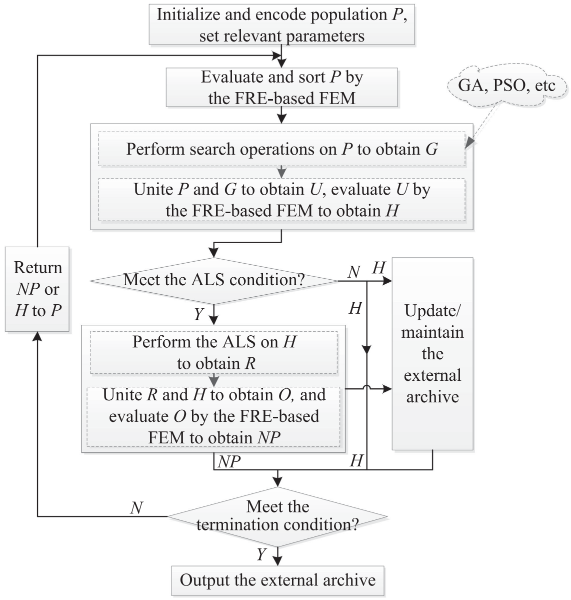

Initialize and encode population *P*, set relevant parameters
Evaluate and sort *P* by the FRE-based FEM
GA, PSO, etc
Perform search operations on *P* to obtain *G*
Unite *P* and *G* to obtain *U*, evaluate *U* by the FRE-based FEM to obtain *H*
Return *NP* or *H* to *P*
Meet the ALS condition?
N
H
Y
H
Update/ maintain the external archive
Perform the ALS on *H* to obtain *R*
Unite *R* and *H* to obtain *O*, and evaluate *O* by the FRE-based FEM to obtain *NP*
NP
H
N
Meet the termination condition?
Y
Output the external archive

Fig. 1. Multiobjective optimization framework.

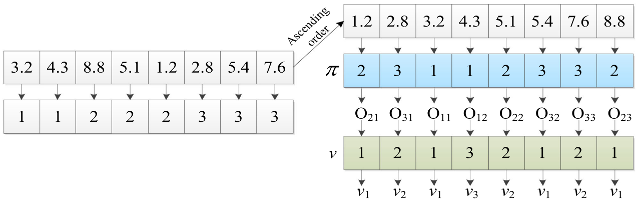


Fig. 2. Encoding example for the EJSP-SDST.

For each operation O_{ij} in π, a corresponding integer z in the range of [1, *L*] is obtained randomly. A machine speed v_z is assigned to operation O_{ij}. These integers construct a feasible v. Then, a machine speed sequence v corresponding to operation π is obtained. A completely encoded solution where the size is $2 \times Nu$ can be obtained by combining π and v. Finally, an initial population *P* containing *N* encoded candidate solutions is generated.

To better understand the encoding process, an example with three jobs, three machines, and three speed levels is presented in Fig. 2. Job 1 has two operations; jobs 2 and 3 consist of three operations and both of them are, therefore, repeated three times. We thus have eight operations [1 1 2 2 2 3 3 3]. After sorting in ascending order, a new sequence with eight operations, i.e., [2 3 1 1 2 3 3 2], is obtained. Next, an integer z in the range of [1], [3] is obtained randomly and a machine speed v_z is assigned to each operation. Finally, a possible machine speed [1 2 1 3 2 1 2 1] is acquired.

B. Fitness Evaluation With FRE

For the current population P^g (g is the iterated generation), the FRE-based FEM is used to evaluate and sort all solutions in P^g. This process is as follows.

1) Each solution in P^g is first decoded and its multiple objective function values for the EJSP-SDST are calculated using (2)–(4). Then, the *N* CPs $Y(X_j^g)$, dynamic *RP* Y_R^g, and *MP* Y_{Max}^g are obtained.
2) The CPs and dynamic RP are converted into fuzzy sets by a semi-trapezoidal membership function. Each solution X_j^g has a C_e value, which is used as the fitness value. Then, all solutions in P^g are evaluated and sorted by their C_e values.

C. Search Operations

Algorithm search operations are performed on the current population to explore the problem's solution space. Many evolutionary search operations, such as those from the GA and PSO, can be incorporated. After a specific algorithm search, a new population G^g is generated. Then, the following steps are conducted to generate further new populations.

1) Unite G^g and P^g to form a combined population, U^g, with the size of 2*N*.
2) Update dynamic RP Y_R^g and *MP* Y_{Max}^g with the method described in Section III-B. Evaluate population U^g by the FRE-based FEM, and choose *N* better solutions to obtain a new population H^g.

D. ALS Strategy

While local search can improve the search performance, it can also increase the computation time. Therefore, it is important to determine when to start the local search. To this end, we adopt quasi-entropy [31] and FRE to propose an ALS strategy. The main concept behind this strategy is that population diversity gradually decreases with the evolution process, and the population with lower diversity is more likely to fall into local optima. Therefore, we can decide when to start local search according to the population diversity. Quasi-entropy is an effective method to measure population diversity. With the FRE-based FEM, quasi-entropy can be used to evaluate the population diversity in MOPs. The steps of ALS are as follows.

1) Evaluate the diversity of the current population by

$$QE^g = -\sum_{j=1}^{N} \rho_j^g log \rho_j^g \tag{22}$$

$$\rho_j^g = \frac{C_e\left(FS\left(Y\left(X_j^g\right)\right), FS\left(Y_R^g\right)\right)}{\sum_{j=1}^{N} C_e\left(FS\left(Y\left(X_j^g\right)\right), FS\left(Y_R^g\right)\right)} \tag{23}$$

where QE^g is the quasi-entropy value of the g^{th} generation's population.

2) Determine whether to start local search by

$$QE^g \leq QE^{(g-1)} \tag{24}$$

where $g = 2, \ldots, G$.

3) If (24) is satisfied, a specified local search method is conducted on the best *r*% solutions selected by the FRE-based FEM in the current population. For each of the best *r*% solutions, the time to execute local search is $T_{\max}$.
4) A population, R^g, containing the new solutions is generated, and R^g is combined with H^g to create a new population, O^g. Then, the dynamic *RP* and *MP* are updated to conduct the FRE-based FEM, and *N* better solutions are chosen to form another new population NP^g.

E. External Archiving

An external archive is used to store the elite solutions found. The construction of this external archive involves two subtasks: constructing the initial external archive and maintaining the external archive thereafter.

1) *Construction of the Initial External Archive:* For the first-generation population, the following steps are conducted to construct the initial external archive:
   a) Create an empty external archive and set $E_{\max}$ as the maximum number of solutions it can accommodate.
   b) Evaluate all the solutions and sort them by their FRE coefficients. If $E_{\max} \geq N$, all solutions are used to obtain the initial external archive. Otherwise, the first $E_{\max}$ solutions with larger C_e values are selected to obtain a full initial external archive.
2) *Maintenance of the External Archive:* In the subsequent generations, the external archive should be maintained to avoid the number of solutions reaching the maximum capacity. FRE apart, the crowded distance [23] is also used to maintain the diversity. Each solution $X_j^g (j = 12, \ldots, N)$ in the current population is compared with every solution E_h^g $(h = 12, \ldots, |E|)$ in the current external archive, whose size is $|E|$, and each solution is judged on whether it can be added to the external archive. The following cases are considered.
   a) If $C_e(FS(Y(X_j^g)), FS(Y_R^g)) \geq C_e(FS(Y(E_h^g)), FS(Y_R^g))$, for any $h = 12, \ldots, E_{\max}$, solution X_j^g is added to the external archive and solution E_h^g is discarded from the external archive.
   b) If $C_e(FS(Y(X_j^g)), FS(Y_R^g)) < C_e(FS(Y(E_h^g), FS(Y_R^g))$, for all $h = 12, \ldots, E_{\max}$, solution X_j^g is not allowed to be added to the external archive.
   c) If the number of solutions in the external archive is larger than $E_{\max}$, the most crowded solutions are removed. The operation is repeated until the number of solutions is equal to $E_{\max}$. Then, a new external archive is created.

F. Computational Complexity

The major computational cost of our multiobjective optimization framework involves the FRE-based FEM, ALS strategy and external archive. For the FRE-based FEM, the construction and updating of dynamic RP and MP require $O(M)$ computations, and the fitness assignment with FRE requires $O(N * M)$ computations. Therefore, the overall complexity of our FRE-based FEM is $O(N * M)$.

For the ALS strategy, two main subtasks are included, i.e., population diversity evaluation and local search execution. The population diversity evaluation based on quasi-entropy and FRE requires $O(N * M)$ computations. The local search requires $O(r\% * N * T_{\max})$ computations. The overall complexity for the ALS is thus $O(\max(M, r\% * T_{\max}) * N)$ in the worst-case scenario.

The maintenance of the external archive involves the FRE-based FEM and crowding distance. The solution evaluation with FRE requires $O(N * |E| * M)$ computations. According to [23], the crowding distance requires $O(M * |E| * \log |E|)$ computations in the worst case. In this article, we have $N > \log |E|$. Therefore, the worst-case complexity for the maintenance of the external archive is $O(N * |E| * M)$.

Other operations have lower complexities. With the above considerations, the overall worst-case complexity of our multiobjective optimization framework is $O(N * |E| * M * G)$, where G is the maximum number of generations.

V. Experimental Results and Analysis

A. Test Instances

We incorporated the well-known GA and inversion local search [32] into our multiobjective optimization framework, resulting in the HMOGA. To test the effectiveness of the HMOGA for the EJSP-SDST, we selected three types of well-known JSP instances from the OR-Library [33]. They include three ORB instances (ORB1∼ORB3), thre ABZ instances (ABZ7∼ABZ9) and 6 LA instances (LA26∼LA28 and LA31∼LA33), with sizes ($n \times m$) ranging from small scale 10 × 10 to large scale 30 × 10. Here, size $n \times m$ in each instance means n jobs and m machines. To better quantify SDST, we generated two types of SDST for each instance. Let "-1" and "-2" represent the two types of SDST, which are 50% and 125% of the processing time of operations, respectively [13]. Then, there are (3 + 3 + 6) × 2 = 24 instances in total. The processing speed of each machine for each operation was selected from the set $v = \{1,\ 1.3,\ 1.55,\ 1.75,\ 2.1\}$ [5]. The processing power of machine k was set as $PM_{kz} = \varepsilon_k v_z^2$ ($z = 1,\ldots,5$), where ε_k submits to uniform distribution $U[5, 10]$. The standby power and setup power were set to $PS_k = \varepsilon_k/4$ and $PST_k = \varepsilon_k/2$, respectively. The due date of each job was set to $D_i = 1.5 \sum_{j=1}^{l_i} t_{ij}$.

B. Parameter Settings

After parameter tuning, the key parameters of the HMOGA were set as $N = 150$, $\alpha = 0.8$, $\beta = 1.4$, $r = 20$, and $T_{\max} = 15$. The simulated binary crossover probability was set as $p_c = 1$, the polynomial mutation probability was set to $p_m = 1/(2 \times n \times m)$, and the distribution indexes for crossover and mutation were set to $\eta_m = 20$ and $\eta_m = 20$, respectively. We also set the maximum number of generations $G = 1000$ and the size of the external archive $E_{\max} = 10$.

In the following experiments, each instance was independently run for 20 times, and all the algorithms were terminated by the same time limit of $2 \times m \times n$ s. Two well-known metrics, namely the HV [13] and spread [2], were adopted to evaluate the performance of the algorithms being compared. The HV is a comprehensive metric for evaluating convergence and diversity, and Spread measures distribution uniformity. A greater HV value means better convergence and diversity, while a smaller Spread value indicates a better distribution.

TABLE I
Processing Times and Due Dates of JOBS

Job	Machine (Processing time)			Due date
1	M1(15)	M2(12)	-	40.5
2	M2(20)	M3(16)	M1(18)	81
3	M1(15)	M3(10)	M2(25)	75

TABLE II
SDST of Jobs on Machines

Job		M1			M2			M3		
i'	Job i	1	2	3	1	2	3	1	2	3
0		9	10	12	8	12	10	10	9	12
1		0	8	6	0	8	9	0	8	10
2		8	0	10	8	0	10	8	0	6
3		6	10	0	9	10	0	10	6	0

TABLE III
Optimal Results Obtained by the HMOGA and CPLEX

Optimal solution	HMOGA			CPLEX		
	f_1	f_2	f_3	f_1	f_2	f_3
[2 3 2 3 1 2 1 3 3 2 3 4 1 1 4 2]	85.6	27.5	1828	85.6	27.5	1828

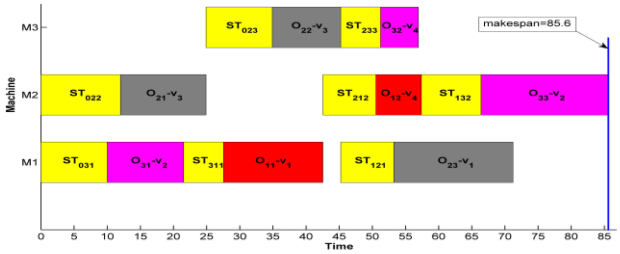


Fig. 3. Scheduling Gantt chart of the example.

C. Validation of the Mathematical Model

In this section, we report on the results of validating the proposed mathematical model by solving a small-size problem instance with three jobs and three machines, as shown in Tables I and II. Other relevant data was generated using the method described in Section V-A.

The problem instance was solved by our mathematical model using the CPLEX software. As given in Table III, the obtained optimal solution X is [2 3 2 3 1 2 1 3 3 2 3 4 1 1 4 2]. The optimal values of the objectives, i.e., makespan, total tardiness and total energy consumption, are 86.5, 27.5, and 1828, respectively. With this optimal solution, the HMOGA was used to test the mathematical model. As given in Table III, the optimal values found by the HMOGA and CPLEX are exactly the same. Hence, the objective functions and constraints in our mathematical model are validated. A Gantt chart for the optimal solution is shown in Fig. 3, where the yellow box indicates the sequence-dependent setup, ST_{ihk} is the SDST of machine k for processing two adjacent jobs i and h, and $O_{ij} - v_z$ refers to the operation O_{ij} with processing speed v_z.

TABLE IV
MEAN HV RESULTS FOR THE SIX MOGAS BASED ON DIFFERENT FEMS

Inst.	size	HV					
		WSMOGA	TMOGA	PBIMOGA	PDMOGA	HVMOGA	HMOGA
ORB1-1	10 × 10	0.5678	1.2023	1.2567	1.1140	1.3062	**1.3245**
ORB1-2		0.6600	0.9875	1.1982	0.9815	1.1333	**1.2897**
ORB2-1		0.8062	1.1529	1.2785	1.2666	**1.3171**	1.3042
ORB2-2		0.7836	1.2022	**1.3377**	1.1678	1.2085	1.3184
ORB3-1		0.7585	1.1866	1.3012	1.2890	1.3211	**1.3678**
ORB3-2		0.6695	1.2060	1.2437	**1.3562**	1.2899	1.3339
LA26-1	20 × 10	0.5784	1.1189	1.2365	1.3017	**1.3330**	1.3171
LA26-2		0.6993	1.1856	**1.3319**	1.1910	1.2891	1.3056
LA27-1		0.7689	1.2020	1.2511	1.2782	**1.3033**	1.2879
LA27-2		0.7800	1.2233	1.2678	1.3012	1.3518	**1.4333**
LA28-1		0.8606	1.2678	1.3012	1.2681	1.3375	**1.3891**
LA28-2		0.8578	1.1899	1.2670	1.2896	1.3016	**1.3800**
ABZ7-1	20 × 15	0.7714	0.9678	1.1980	1.2018	**1.3656**	1.3479
ABZ7-2		0.8237	1.2011	1.2375	**1.3555**	1.2680	1.3488
ABZ8-1		0.9018	1.2512	1.2018	1.3030	1.2879	**1.3567**
ABZ8-2		0.8877	1.1156	1.1992	1.2876	**1.3760**	1.3661
ABZ9-1		0.8370	1.2786	1.3012	1.2987	**1.4132**	1.3898
ABZ9-2		0.7744	1.1011	1.1973	1.2450	1.3255	**1.4156**
LA31-1	30 × 10	0.7467	1.2022	1.2451	1.2783	1.3014	**1.3862**
LA31-2		0.7866	1.2127	1.2392	1.3011	1.2876	**1.3215**
LA32-1		0.9238	1.2189	1.2670	1.2583	1.3670	**1.4008**
LA32-2		0.9333	0.9989	1.1987	1.3067	1.3434	**1.3800**
LA33-1		0.8087	1.0456	1.2136	1.2780	1.3119	**1.3877**
LA33-2		0.9711	1.1677	1.2436	1.2883	1.3571	**1.4217**
Ave.		0.7899	1.1619	1.2506	1.2586	1.3124	**1.3573**

TABLE V
MEAN HV AND SPREAD RESULTS OBTAINED BY THE TWO HMOGAS WITHOUT AND WITH ALS

Inst.	HV		Spread	
	HMOGA1	HMOGA2	HMOGA1	HMOGA2
ORB1-1	0.7890	**0.8345**	**0.8678**	0.8878
ORB1-2	1.0213	**1.2122**	**0.7890**	0.8512
ORB2-1	1.1244	**1.3025**	0.8463	**0.8236**
ORB2-2	0.9457	**1.1560**	**0.7718**	0.8349
ORB3-1	0.8585	**1.1782**	**0.8600**	0.9137
ORB3-2	**1.2012**	1.1980	**0.8247**	0.8444
LA26-1	1.1200	**1.2343**	0.8615	**0.8428**
LA26-2	0.9234	**1.2567**	0.8767	**0.8210**
LA27-1	0.8563	**1.1829**	**0.7910**	0.8369
LA27-2	**1.1678**	1.1530	**0.8019**	0.8788
LA28-1	**1.3023**	1.2845	**0.8340**	0.8915
LA28-2	**1.1360**	1.1048	**0.6882**	0.8237
ABZ7-1	1.0234	**1.3420**	0.8715	**0.8190**
ABZ7-2	0.9072	**1.2122**	**0.7789**	0.8213
ABZ8-1	1.1234	**1.3346**	0.8718	**0.8441**
ABZ8-2	0.8899	**1.2400**	**0.8343**	0.8620
ABZ9-1	**1.2341**	1.2012	0.8712	**0.8199**
ABZ9-2	1.1178	**1.3460**	0.8917	**0.8410**
LA31-1	1.0190	**1.2785**	**0.8330**	0.8567
LA31-2	0.9870	**1.2237**	0.8882	**0.8266**
LA32-1	1.1578	**1.3680**	**0.6922**	0.7895
LA32-2	1.2136	**1.2616**	0.8789	**0.8612**
LA33-1	0.9856	**1.1829**	0.9124	**0.8780**
LA33-2	1.1620	**1.3512**	0.8881	**0.8333**
Ave.	1.0528	**1.2267**	**0.8343**	0.8460

D. Effectiveness of the FRE-Based FEM

We evaluated the effectiveness of our FRE-based FEM by comparing it with five commonly used FEMs for solving MOPs: Weighted sum [21], [34]; Tchebycheff function [21], [34]; penalty boundary intersection (PBI) [21]; Pareto dominance; and HV indicator. Incorporating these FEMs into the GA, we have five multiobjective GAs (MOGAs): weighted sum-based MOGA (WSMOGA); Tchebycheff-based MOGA (TMOGA); PBI-based MOGA (PBIMOGA); Pareto dominance-based MOGA (PDMOGA); and HV indicator-based MOGA (HVMOGA). The only difference in the abovementioned algorithms is their FEM.

Table IV gives the comparison results based on the mean HV. From the table, we see that the HMOGA has the best HV results in 14 out of the 24 instances; while the PBIMOGA, PDMOGA and HVMOGA have the best HV results in 2, 2 and 6 instances, respectively. Based on the average values, our HMOGA performed the best in terms of HV. The results in Table IV clearly indicate that the HMOGA has better convergence and diversity in most cases. These results imply that the FRE-based FEM performs better than the other five FEMs in solving the EJSP-SDST.

E. Effectiveness of the ALS Strategy

To assess the effectiveness of the ALS strategy, we conducted a comparison between HMOGA1 and HMOGA2, which are the proposed HMOGA without and with ALS, respectively. Table V gives the mean HV and Spread results obtained by the two HMOGAs.

We see from Table V that HMOGA2 performed better in 19 out of 24 instances based on the HV. In terms of spread, HMOGA2 performed better in only 11 out of the 24 instances. On average, HMOGA2 performed better than HMOGA1 based on the HV, while it performed only slightly worse than HMOGA1 in terms of Spread. The results in Table V indicate that the convergence and diversity of HMOGA2 are significantly better than those of HMOGA1, whereas both HMOGA1 and HMOGA2 have similar distributions. The reason behind this being that ALS is based on quasi-entropy and FRE. Quasi-entropy is adopted to evaluate population diversity, and FRE is used to select solutions. Convergence and diversity can be guaranteed by quasi-entropy and FRE. However, ALS does not include a distribution mechanism, which renders it unable to effectively improve the distribution.

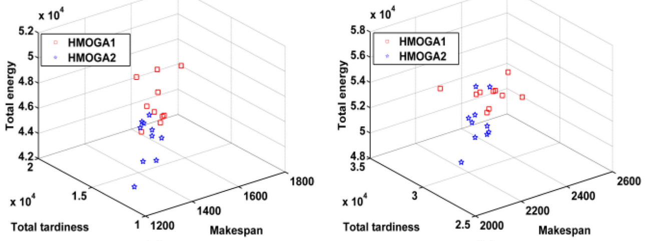
Fig. 4 shows the Pareto fronts of the two HMOGAs on medium-size instances ABZ9-1 and ABZ9-2. We can see that the solution set of HMOGA2 is closer to the origin of the coordinate axis, although it has no obvious advantage in terms of distribution.

F. Performance Comparison with Other Algorithms

In this section, we compare the performance of the HMOGA with five well-known multiobjective algorithms: NSGA-II [23]; SPEA-II [26]; NSGA-III [27]; MOEA/D [21]; and MOPSO [28]. The first four algorithms are based on an evolutionary

TABLE VI
MEAN HV AND SPREAD RESULTS OF THE SIX MULTIOBJECTIVE ALGORITHMS

Inst.	HV						Spread					
	NSGA-II	SPEA-II	NSGA-III	MOEA/D	MOPSO	HMOGA	NSGA-II	SPEA-II	NSGA-III	MOEA/D	MOPSO	HMOGA
ORB1-1	1.2002	1.0134	0.9883	1.3660	1.3545	**1.4441**	**0.7300**	0.7340	0.8021	1.3345	0.8345	0.7662
ORB1-2	1.1456	0.8345	0.9117	1.4012	1.2897	**1.4580**	0.7823	**0.7451**	0.8233	1.2400	0.8034	0.7875
ORB2-1	1.2550	0.9678	0.8818	1.3775	1.4160	**1.5228**	0.8012	**0.7667**	0.8116	1.2006	0.8278	0.8200
ORB2-2	1.1893	1.0874	1.0072	1.3214	1.3773	**1.5545**	0.8340	0.8122	0.8450	1.3045	0.8667	**0.8105**
ORB3-1	1.0175	1.1023	0.9680	1.2879	1.4018	**1.4716**	0.8417	**0.8089**	0.8778	1.3554	0.8300	0.8406
ORB3-2	1.2205	0.9761	0.8589	1.4212	1.3456	**1.4879**	0.8524	0.8310	0.8819	1.4042	**0.8217**	0.8431
LA26-1	1.1118	0.9919	0.8771	1.4444	1.3860	**1.5125**	0.8686	**0.8455**	0.9039	1.3748	0.8620	0.8671
LA26-2	1.2876	1.0134	1.0782	1.3380	1.4145	**1.4840**	0.8888	0.8607	0.9114	1.2897	**0.8526**	0.8774
LA27-1	1.1779	1.1005	0.8256	1.3710	1.2899	**1.4678**	0.9021	0.8765	0.9345	1.3340	**0.8600**	0.8812
LA27-2	1.2011	0.9877	1.0063	1.4029	1.3851	**1.5062**	**0.8787**	0.8913	0.9526	1.4222	0.9811	0.8799
LA28-1	1.1708	0.8557	0.9975	1.3832	1.3712	**1.4753**	0.9134	**0.8771**	0.9863	1.4437	1.0122	0.9003
LA28-2	1.0783	0.9672	0.8870	1.2991	1.3855	**1.4681**	0.9335	**0.8818**	1.0346	1.3829	0.9619	0.9016
ABZ7-1	1.0065	0.8797	0.9165	1.3389	1.4011	**1.5127**	0.9247	**0.8780**	1.1010	1.4245	0.9334	0.9090
ABZ7-2	1.1820	1.1108	0.9678	1.3044	1.3860	**1.5036**	0.9348	0.9116	1.0562	1.3867	0.9531	**0.9100**
ABZ8-1	1.2272	1.1215	0.9017	1.3608	1.3788	**1.4783**	0.9018	**0.8987**	0.9727	1.4238	1.0650	0.9155
ABZ8-2	1.0784	0.9781	0.8898	1.2976	1.3360	**1.4580**	**0.8919**	0.9013	0.9866	1.4344	0.9811	0.9234
ABZ9-1	1.1418	0.9560	1.0578	1.2545	1.2344	**1.5614**	0.9016	0.8885	0.9135	1.4500	1.0342	**0.8866**
ABZ9-2	1.2018	1.1540	0.9346	1.3417	1.3753	**1.4478**	0.9113	**0.9005**	0.9678	1.4078	0.9775	0.9248
LA31-1	1.2316	0.9648	0.9055	1.3636	1.4018	**1.4875**	0.9248	0.9223	0.9871	1.4317	0.9300	**0.9217**
LA31-2	1.1874	1.0045	0.8846	1.3507	1.3449	**1.5152**	0.9019	0.8919	0.9567	1.3997	0.9712	**0.8818**
LA32-1	1.2163	1.1016	0.9666	1.3825	1.4040	**1.4665**	0.9225	0.9344	0.9800	1.4456	0.9888	**0.9019**
LA32-2	1.1960	1.1223	0.9230	1.3677	1.3914	**1.5320**	0.8978	0.9023	0.9519	1.3860	0.9456	**0.8879**
LA33-1	1.1783	1.0004	0.8972	1.3484	1.3752	**1.5421**	0.9129	0.8978	0.9711	1.4078	0.9788	**0.8794**
LA33-2	1.0188	0.9067	0.8680	1.2872	1.3669	**1.4977**	0.9214	0.9189	0.9570	1.4247	0.9795	**0.9006**
Ave.	1.1634	1.0083	0.9334	1.3505	1.3672	**1.4940**	0.8822	**0.8657**	0.9402	1.3796	0.9272	0.8722



(a) (b)

Fig. 4. Pareto fronts obtained by the two MOGAs on ABZ9-1 and ABZ9-2. (a) ABZ9-1 (b) ABZ9-2.

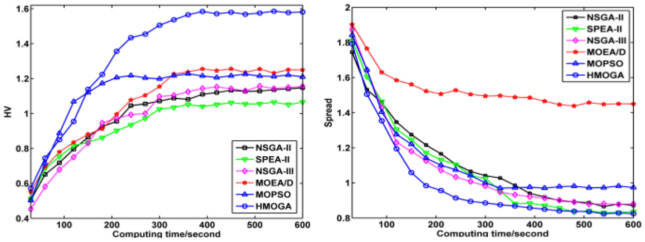


Fig. 5. Evolution trends of the six algorithms on instance ABZ9-1.

framework, while the last one, MOPSO, is a swarm intelligence optimization algorithm. All five algorithms are commonly used as benchmark algorithms to test the performance of newly proposed multiobjective algorithms.

Table VI gives the results obtained by the six multiobjective algorithms. We observe that the HMOGA was able to obtain the best HV results on all 24 instances. For Spread, the NSGA-II, SPEA-II, MOPSO, and HMOGA achieved the best results on 3, 9, 3, and 9 instances, respectively; MOEA/D performed the worst on most instances in terms of spread. For large instances (LA31∼LA33) in particular, our HMOGA performed the best on both the metrics. In terms of average HV, the HMOGA significantly outperformed the other five multiobjective algorithms. For the average value of spread, our HMOGA performed better than the NSGA-II, NSGA-III, MOPSO and MOEA/D, and its performance was similar to that of SPEA-II. Based on these results, we can say that the convergence and diversity of the HMOGA are significantly better than those of the other five algorithms.

Fig. 5 shows the evolution trends of HV and Spread for the six algorithms on a selected instance, ABZ9-1. The HV and Spread curves of our algorithm changed smoothly and maintained a better evolution trend. The evolution trend of MOEA/D's HV was better than that of NSGA-II, SPEA-II, NSGA-III and MOPSO, but it had the worst Spread value during the entire evolution process.

Finally, we carried out the Kruskal-Wallis ANOVA test with a significance level of 0.05 to statistically ascertain the significant dominance of the HMOGA. As can be seen in Fig. 6, the p-values of HV and Spread are less than 0.05, and therefore, we can conclude that the HMOGA has significantly outperformed the other algorithms from a statistical perspective. The significant

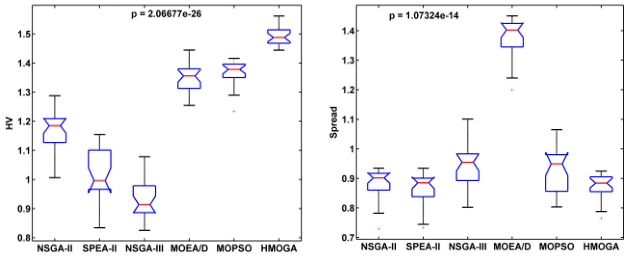
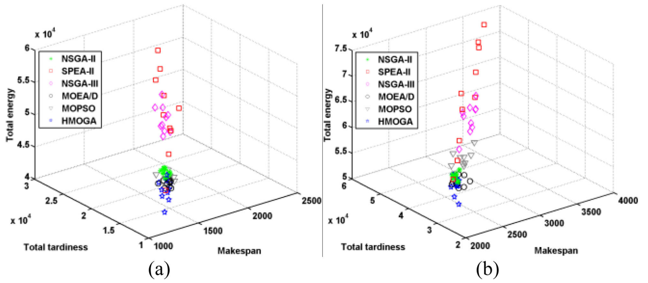


Fig. 6. Kruskal-Wallis ANOVA significance tests for the six algorithms.



(a)

(b)

Fig. 7. Pareto fronts obtained by the six algorithms on ABZ9-1 and ABZ9-2. (a) ABZ9-1 (b) ABZ9-2.

differences among these algorithms may be due to the differences in their major components, such as the FEM and the search operations.

Fig. 7 presents the Pareto fronts of the six algorithms on a selected medium-scale instance (ABZ9-1) and a large-scale instance (ABZ9-2). From Fig. 7, we observe that the solution set of HMOGA is closer to the origin of the coordinate axis, which intuitively illustrates that our HMOGA performs better than the other five algorithms.

Based on the above analysis and discussion, it can be seen that the HMOGA performs the best in terms of the HV but not spread, although its performance is similar to the SPEA-II, which is the best performer in terms of Spread. We believe the following are the main reasons behind this.

1) The FRE-based FEM is used during the entire evolution process of HMOGA. Via this FEM, multiple objectives are aggregated into a simple and intuitive coefficient, i.e., C_e. With a dynamic RP, the population can be guided toward the Pareto front quickly, which greatly improves the convergence of HMOGA.
2) Another novel strategy, i.e., ALS, is incorporated in the HMOGA. As discussed earlier, ALS is executed based on quasi-entropy and FRE, which can guarantee population diversity and convergence, respectively.
3) The HMOGA does not use any strategy that can effectively improve distribution performance. Owing to this, the HMOGA does not perform the best in terms of spread; however, its performance is similar to the best algorithm for spread, SPEA-II.

To summarize, our HMOGA is very competitive in solving the EJSP-SDST. Particularly, our HMOGA is suitable for large-scale engineering optimization problems that require a solution with good convergence, since the HMOGA performs the best on both the HV and Spread for large instances (i.e., LA31$\sim$LA33).

VI. Conclusion

In this article, we studied the EJSP-SDST, aiming to simultaneously minimize the makespan, total tardiness and total energy consumption. A new FRE-based FEM were developed to evaluate and select solutions, with the dynamic RP and MP introduced to construct fuzzy sets. A general multiobjective optimization framework integrating the FRE-based FEM and ALS were presented. A hybrid GA approach—HMOGA—were proposed based on the optimization framework. Extensive experiments showed that the proposed HMOGA can outperform the NSGA-II, SPEA-II, NSGA-III, MOEA/D and MOPSO on most instances of different scales.

The proposed FRE-based FEM was a very promising solution evaluation method for MOPs. We have only utilized the GA and inversion local search as an example of integration with the proposed multiobjective optimization framework; other evolutionary and swarm intelligence algorithms was be used in future studies. Furthermore, we plan to apply this FRE-based optimization framework to other complex real-world MOPs.

References

[1] K. Fang, N. Uhan, F. Zhao, and J. W. Sutherland, “A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction,” *J. Manuf. Syst.*, vol. 37, no. 4, pp. 234–240, Oct. 2011.

[2] C. Lu, L. Gao, X. Li, Q. Pan, and Q. Wang, “Energy-efficient permutation flow shop scheduling problem using a hybrid multiobjective backtracking search algorithm,” *J. Cleaner Prod.*, vol. 144, pp. 228–238, Feb. 2017.

[3] 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.

[4] M. Dai, D. 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, Oct. 2013.

[5] R. Zhang and R. Chiong, “Solving the energy-efficient job shop scheduling problem: A multiobjective genetic algorithm with enhanced local search for minimizing the total weighted tardiness and total energy consumption,” *J. Cleaner Prod.*, vol. 112, pp. 3361–3375, Jan. 2016.

[6] 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. 2018.

[7] G. Mouzon, M. B. Yildirim, and J. Twomey, “Operational methods for minimization of energy consumption of manufacturing equipment,” *Int. J. Prod. Res.*, vol. 45, pp. 4247–4271, May 2007.

[8] K. T. Fang and B. M. Lin, “Parallel-machine scheduling to minimise tardiness penalty and power cost,” *Comput. Ind. Eng.*, vol. 64, no. 1, pp. 224–234, Jan. 2013.

[9] J. Y. Ding, S. Song, R. Zhang, R. Chiong, and C. Wu, “Parallel machine scheduling under time-of-use electricity prices: New models and optimization approaches,” *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 12, pp. 1138–1154, Apr. 2016.

[10] K. Fang, N. A. Uhan, F. Zhao, and J. W. Sutherland, “Scheduling on a single machine under time-of-use electricity tariffs,” *Ann. Oper. Res.*, vol. 238, no. 1/2, pp. 199–227, Sep. 2016.

[11] L. J. He, W. F. Li, Y. Zhang, and Y. L. Cao, “A discrete multi-objective fireworks algorithm for flowshop scheduling with sequence-dependent setup times,” *Swarm Evol. Comput.*, vol. 51, Dec. 2019, Art. no. 100575.

[12] A. Allahverdi, “The third comprehensive survey on scheduling problems with setup times/costs,” *Eur. J. Oper. Res.*, vol. 246, no. 2, pp. 345–378, Oct. 2015.

[13] M. Ciavotta, G. Minella, and R. Ruiz, “Multiobjective sequence dependent setup times permutation flowshop: A new algorithm and a comprehensive study,” *Eur. J. Oper. Res.*, vol. 227, no. 2, pp. 301–313, Jun. 2013.

[14] G. C. Buttazzo, M. Bertogna, and G. Yao, “Limited preemptive scheduling for real-time systems. A survey,” *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 3–15, Feb. 2013.

[15] 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, no. 1, pp. 87–96, Feb. 2014.

[16] M. A. Salido, J. Escamilla, A. Giret, and F. Barbe, “A genetic algorithm for energy-efficiency in job-shop scheduling,” *Int. J. Adv. Manuf. Technol.*, vol. 85, no. 5–8, pp. 1303–1314, Nov. 2016.

[17] H. Gao, S. Kwong, B. J. Fan, and R. Wang, “A hybrid particle-swarm tabu search algorithm for solving job shop scheduling problems,” *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2044–2054, Nov. 2014.

[18] J. H. Zhao, F. S. Wen, Y. D. Zhao, Y. S. Xue, and K. P. Wong, “Optimal dispatch of electric vehicles and wind power using enhanced particle swarm optimization,” *IEEE Trans. Ind. Informat.*, vol. 8, no. 4, pp. 889–899, Nov. 2012.

[19] K. Y. Chan, C. K. F. Yiu, T. S. Dillon, and S. Nordholm, “Enhancement of speech recognitions for control automation using an intelligent particle swarm optimization,” *IEEE Trans. Ind. Informat.*, vol. 8, no. 4, pp. 869–879, Nov. 2012.

[20] G. Zhu, L. He, X. Ju, and W. Zhang, “A fitness assignment strategy based on the grey and entropy parallel analysis and its application to MOEA,” *Eur. J. Oper. Res.*, vol. 265, no. 3, pp. 813–828, Mar. 2018.

[21] 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.

[22] R. Wang, Z. Zhou, H. Ishibuchi, T. Liao, and T. Zhang, “Localized weighted sum method for many-objective optimization,” *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 3–17, Feb. 2018.

[23] 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.

[24] A. Zhou, B. Y. Qu, H. Li, S. 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, Mar. 2011.

[25] S. Jiang, J. Zhang, Y. Ong, A. Zhang, and P. Tan, “A simple and fast hypervolume indicator-based multiobjective evolutionary algorithm,” *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2202–2213, Oct. 2015.

[26] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization,” in *Proc. Evol. Methods Des. Optim. Control Appl. Ind. Problem*, 2002, pp. 95–100.

[27] K. Deb and H. Jain, “An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: Solving problems with box constraints,” *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 557–601, Aug. 2014.

[28] 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.

[29] X. Z. Guo and X. L. Xin, “Partial entropy and relative entropy of fuzzy sets,” *Fuzzy Syst. Math.*, vol. 19, no. 2, pp. 97–102, 2005.

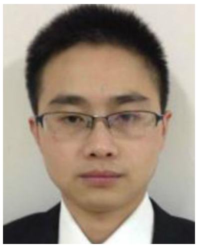
[30] S. Sharma and R. Kumar, “Fuzzy relative entropy based classification scheme for discrimination of odors/gases using a poorly selective sensor array,” in *Proc. IEEE Conf. Fuzzy Syst.*, 2016, pp. 1195–1200.

[31] Y. Cao, H. Zhang, W. Li, M. Zhou, Y. Zhang, and W. Chaovalitwongse, “Comprehensive learning particle swarm optimization algorithm with local search for multimodal functions,” *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 718–731, Aug. 2019.

[32] P. Hansen and N. Mladenović, “Variable neighborhood search: Principles and applications,” *Eur. J. Oper. Res.*, vol. 130, pp. 449–467, May 2001.

[33] J. E. Beasley, “OR-Library: Distributing test problems by electronic mail,” *J. Oper. Res. Soc.*, vol. 41, no. 11, pp. 1069–1072, 1990.

[34] K. Miettinen, *Nonlinear Multiobjective Optimization*. Norwell, MA, USA: Kluwer, 1999.



Lijun He received the B.S. degree in mechanical and electronic engineering from the Nanjing Institute of Technology, Nanjing, China, in 2012, and the M.S. degree in mechanical design and theory from Fuzhou University, Fuzhou, China, in 2016. He is currently working toward the Ph.D. degree in mechanical engineering with the School of Logistics Engineering, Wuhan University of Technology, Wuhan, China.

His research interests include use of evolutionary and swarm intelligence algorithms for multiobjective optimization and production scheduling.

Raymond Chiong (Senior Member, IEEE) received the M.Sc. degree in advanced computer science from the University of Birmingham, Birmingham, U.K., in 2004, and the Ph.D. degree in engineering from the University of Melbourne, Parkville, VIC, Australia, in 2012.

He is currently an Associate Professor with the University of Newcastle, Callaghan, NSW, Australia. He is also a Guest Research Professor with Center for Modern Information Management, Huazhong University of Science and Technology, Wuhan, China, and a Visiting Scholar with the Department of Automation, Tsinghua University, Beijing, China. He has authored or coauthored more than 200 papers in these areas. He is the Editor-in-Chief of the Journal of Systems and Information Technology, and an Editor of Engineering Applications of Artificial Intelligence. His research interests include evolutionary optimization, data analytics, and modeling of complex adaptive systems, with applications in areas such as production scheduling, stock market prediction, energy load forecasting, and social behavior modeling, among others.

Wenfeng Li (Senior Member, IEEE) received the B.S. degree from the Zhengzhou Institute of Light Industry, the M.S. degree from the Huazhong University of Science and Technology, and the Ph.D. degree from the Wuhan University of Technology, Wuhan, China, respectively, all in mechanical engineering.

He is currently a Professor with the School of Logistics Engineering, Wuhan University of Technology. His research Interests include the Internet of Things and its application, robotics, logistics information and automation, and simulation and planning of logistics/supply chains.

Sandeep Dhakal received the B.S. degree from the Swinburne University of Technology, Hawthorn, VIC, Australia, and the M.S. degree from the University of St. Andrews, St Andrews, U.K. He is currently working toward the Ph.D. degree with the University of Newcastle, Callaghan, NSW, Australia.

His research interests include evolutionary game theory and agent-based modeling.

Yulian Cao (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in logistics engineering from the Wuhan University of Technology, Wuhan, China, in 2009, 2012, and 2018, respectively.

From 2012 to 2014, she was a Visiting Scholar with the University of Washington, Seattle, WA, USA. She is currently a Postdoctoral Fellow with the University of New South Wales, Sydney, NSW, Australia. Her research interests include evolutionary computing, swarm intelligence, and logistic system modeling and optimization.

Yu Zhang received the B.S., M.S., and Ph.D. degrees in logistics engineering from the Wuhan University of Technology, Wuhan, China, in 1996, 2001, and 2007, respectively.

He is currently a Professor with the School of Logistics Engineering, Wuhan University of Technology. His research interests include modeling and simulating analysis of logistic systems, fleet assignment and routing, and machine scheduling.