IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 29, 2020, accepted December 15, 2020, date of publication December 21, 2020,
date of current version January 4, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3046002

Fine-Grained Ensemble of Evolutionary Operators for Objective Space Partition Based Multi-Objective Optimization

XUEFENG HONG[1], MINGFANG JIANG[2], AND JINGLIN YU[1]
[1]School of Internet Finance and Information Engineering, Guangdong University of Finance, Guangzhou 510521, China
[2]School of Information Science and Engineering, Hunan First Normal University, Changsha 410004, China

Corresponding author: Xuefeng Hong (happyvbvb@126.com)

This work was supported by the Guangzhou Social Science Planning Department under Grant 2020GZGJ160.

ABSTRACT Decomposition-based multi-objective optimization algorithms have been widely accepted as a competitive technique in solving complex multi-objective optimization problems (MOPs). Motivated by the facts that evolutionary operators are sensitive to the properties of problems, and even different search stages of an evolutionary operator often pose distinct properties when solving a problem. So far, numerous ensemble approaches have been designed to adaptively choose evolutionary operators for evolving population during different optimization stages. Then, during one stage, all the subproblems/subspaces in these existing ensemble approaches use the same evolutionary operator. But, for a complex MOP, the properties of its subproblems/subspaces are different. Based on the fact that existing ensemble approaches ignore this point, this article develops a fine-grained ensemble approach, namely FGEA, to choose suitable evolutionary operators for different subspaces during one generation. To be specific, the local and global contributions for each evolutionary operator in each subproblem/subspace are first defined. Then, an adaptive strategy is designed to encourage evolutionary operators making more contributions to obtain more opportunities to generate more offspring solutions. When choosing an evolutionary operator for a subspace, the proposed adaptive strategy considers both the local and global contributions of the evolutionary operators. Finally, based on 35 complex MOPs, we evaluate the effectiveness of the proposed FGEA by comparing it with five baseline algorithms. The experimental results reveal the competitive performance of the FGEA, which achieves the lowest inverted generational distance (IGD) values and the highest hypervolume values on 20 and 19 MOPs, respectively.

INDEX TERMS Evolutionary computation, multi-objective optimization, fine-grained ensemble, complex Pareto set.

I. INTRODUCTION

Real-world problems in various fields, such as hybrid electric vehicle control problem [1], optimizing wireless networks [2], [3], feature selection for classification [4], service composition in clouds [5], resource allocation in radar networks [6], [7], intelligent traffic management [8], [9], and resource investment project scheduling [10], usually involve simultaneously optimizing at least two conflicting objectives. These problems are known as multi-objective optimization problems (MOPs) [11]. Their mathematical models can be summarized as:

$$\begin{cases} \text{Minimize} & F(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_m(\boldsymbol{x})], \\ \text{S.t.} & \boldsymbol{x} \in \Omega, \end{cases} \tag{1}$$

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ denotes a decision vector, Ω is the decision space, n and m refer to the number of decision variables and optimization objectives. Besides, the $f_j(\boldsymbol{x})$, $1 \leq j \leq m$ represents the j-th optimization objective. For the m optimization objectives, there are at least two objectives that conflict with each other [12], [13].

Because of the conflicting nature among different optimization objectives, when minimizing one objective of a solution on the Pareto front, the values on other objectives

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

tend to deteriorate [14]. Thus, there exists a set of trade-off solutions, rather than one single solution optimizing all the objectives at the same time. A solution $\boldsymbol{x}_1$ is termed to dominate another one $\boldsymbol{x}_2$ (denoted as $\boldsymbol{x}_1 \prec \boldsymbol{x}_2$) if and only if all the objectives of solution $\boldsymbol{x}_1$ are not worse than that of solution $\boldsymbol{x}_2$ and solution $\boldsymbol{x}_1$ is superior to solution $\boldsymbol{x}_2$ in at least one objective. A solution is considered as Pareto-optimal if and only if no solution can dominate it. All the Pareto-optimal solutions are called Pareto set (PS) in decision space and Pareto front (PF) in objective space, respectively.

Population-based evolutionary algorithms are capable of searching multiple solutions concurrently during each run, and they are good at getting a set of compromise solutions for MOPs. Thus, multi-objective evolutionary algorithms (MOEAs) have been widely employed to approximate the PFs of MOPs. The popularity of evolutionary algorithms in solving MOPs results in a prolific research direction, particularly, a plethora of MOEAs have been developed until now [15]–[18].

Based on the environmental selection mechanisms, most MOEAs can be roughly divided into three categories, i.e., dominance-based, indicator-based, and decomposition-based MOEAs [15]. Compared with the previous two categories, the decomposition-based MOEAs have been well accepted as a competitive technique in solving multi-objective optimization problems with complex PSs [19]–[21]. For decomposition-based MOEAs, one branch is to employ a set of weight vectors to transform an MOP into a series of single-objective sub-problems, and then solve these sub-problems in a cooperative fashion [18], [22]. The other branch is the objective space partition-based MOEAs [20], [22], [23]. The algorithms in this branch, such as MOEA/D-M2M [20], OPE-MOEA [23], and RVEA [24], partition the objective space into a series of sub-spaces using a set of reference vectors, and then search solutions for each sub-spaces.

Different evolutionary operators have distinct capabilities of exploitation and exploration, which leads to different advantages in solving different MOPs [25], [26]. For instance, differential evolution [27], [28] and simulated binary crossover [29] exhibit qiute different search characteristics for exploration, while polynomial-based mutation [30] owns strong local search capacity (exploitation). An MOEA equipped with one single evolutionary operator often has good performance on some certain MOPs, but may performs poorly in other MOPs [25]. Even for the same MOP, an evolutionary operator performs quite differently during different optimization stages. Thus, it is difficult for one single operator to solve various types of MOPs. An intuitive idea is to integrate multiple evolutionary operators to play their respective advantages [31].

So far, there exist numerous ensemble approaches adaptively choosing evolutionary operators during different optimization stages, such improving the capability of decomposition-based MOEAs in coping with MOPs. For instance, Hitomi *et al.* recognized the difficulty in defining a suitable credit assignment for adaptive operator selection, and experimentally compared nine credit assignment mechanisms on benchmark functions [31]. Li *et al.* employed the sliding window method to store the fitness improvement rates of each evolutionary operators, and designed an adaptive operator selection mechanism to select one of four differential evolution variants for decomposition-based MOEA during each generation [32]. Wang *et al.* designed an ensemble framework to combine multiple evolutionary operators and selection criteria on the basis of multiple populations, also employed competition and cooperation strategies to respectively adjust evolutionary operators for different populations and select offspring solutions [25]. Zhao *et al.* employed a learning automaton technique to select evolutionary operators for decomposition-based MOEAs based on feedback from the optimization process, also developed an adaptive strategy to adjust the reference vectors using the obtained solutions in the external archive [33]. Santiago *et al.* suggested fuzzy logic to dynamically choose evolutionary operators during each generation, and proposed a new density estimator with polynomial complexity to decide which solutions should be retained [34]. To enhance the performance of decomposition-based MOEAs, Lin *et al.* designed an adaptive mechanism for dynamically choosing evolutionary operators during the optimization process [35]. Wu *et al.* reported a bicriteria-based adaptive operator selection approach to improve the performance of decomposition-based MOEAs [36]. However, most ensemble approaches are limited to selecting an evolutionary operator during one stage, that is, all the sub-problems or sub-spaces apply the same evolutionary operator.

The MOPs coming from various real-world applications [5], [37], [38] have complex relationships among decision variables, e.g., linear linkage, non-linear linkage, rotation, epistasis, and alike. Then, different sub-problems or sub-spaces of an MOP have distinct properties [31], and the requirements of evolutionary operators should vary from one sub-space to another. Motivated by this, this article proposes a fine-grained ensemble approach, namely FGEA, to choose suitable evolutionary operators for different sub-spaces during different optimization stages. In the proposed FGEA, even for the same generation, different sub-spaces may employ different evolutionary operators to generate new solutions. The key contributions of this article can be summarized as follows.

(1) We define the local and global contributions of evolutionary operators to measure their fitness to subspaces.
(2) A novel adaptive strategy is designed to choose suitable evolutionary operators for each sub-space. When choosing an evolutionary operator for a sub-space, both local and global contributions of evolutionary operators are fully considered.

This article is organized as follows. Section II describes the proposed FGEA, followed by extensive comparison experiments to assess the its performance in Section III. Finally, Section IV concludes this article with two future directions.

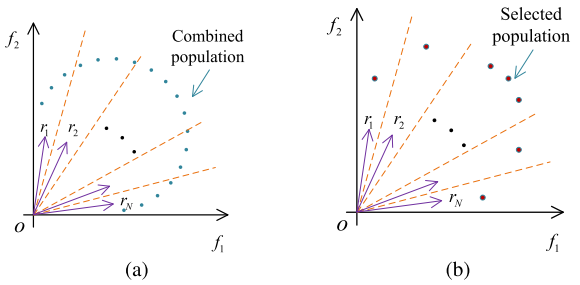

FIGURE 1. Examples of objective space partition.

II. ALGORITHM DESCRIPTION

In this section, the main process of the proposed FGEA is first introduced. Then, the core idea of fine-grained ensemble approach and the adaptive strategy are detailed. Besides, the time complexity of the FGEA is given.

A. MAIN PROCESS

The objective space partition-based multi-objective evolutionary optimization [20], [23], [39], [40] is one competitive branch of decomposition-based approaches [41]. The algorithms in this branch employ a set of reference vectors to partition the objective space of an MOP into a series of connected and disjoint subspaces, and a reference vector corresponds to a subspace. During environmental selection process, each solution is associated to a reference vector with the minimum acute angle, and then a solution with the best fitness in each subspace is selected to survive to the next generation. To visually detail the concepts of objective space partition strategy and solution fitness, two intuitive diagrams are given in Fig. 1.

Suppose that the population size is N and the set of reference vectors is $R = \{r_1, r_2, \cdots, r_N\}$, these N reference vectors divide a two-dimensional objective space into N subspaces, as illustrated in Fig. 1(a). The boundaries between subspaces are represented by dashed orange lines. For a combined population in Fig. 1(a), when each subspace can preserve a solution, the selected population in Fig. 1(b) of course have good diversity. But, how to strengthen the population convergence for various MOPs is still a challenging task.

In this article, the framework of objective space partition-based multi-objective evolutionary optimization [20] is extended to support a fine-grained ensemble strategy to adaptively select efficient evolutionary operators for each subspace during each generation on the basis of evolutionary operators' local and global contributions. For the main process of the FGEA, its Pseudo-code is illustrated in Algorithm 1.

As shown in Algorithm 1, before running the proposed FGEA, the user needs to input the multi-objective optimization problem (MOP) to be solved, population size N and maximum function evaluations (MFE). Once the proposed FGEA completes the optimization search, it will output a population, denoted as P, with well convergence and diversity.

Algorithm 1: Main Process of *FGEA*

Input: An MOP; population size N; maximum function evaluations (MFE);
Output: The final population P;

1 $R \leftarrow$ Obtain a set of uniformly distributed reference vectors, similar to works [24], [42];
2 $P \leftarrow$ Initialize a population in a random way;
3 $UFE \leftarrow |P|$;
4 Initialize the number of evolutionary operators and memory length as K and L;
5 $[M^1, M^2, \cdots, M^N] \leftarrow 0_{K\times L}$;
6 **while** $UFE < MFE$ **do**
7 $\quad Q \leftarrow$ GenerateOffspring$(P, M^1, M^2, \cdots, M^N)$;
8 $\quad UFE \leftarrow UFE + |Q|$;
9 $\quad$ Update ideal and nadir points as Z_{min} and Z_{max};
10 $\quad Fit \leftarrow$ Calculate the fitness of each subspace with population P;
11 $\quad P \leftarrow$ EnvironmentalSelection$(R, P \bigcup Q)$;
12 $\quad NewFit \leftarrow$ Calculate the fitness of each subspace with new population P;
13 $\quad [M^1, M^2, \cdots, M^N] \leftarrow$ UpdateContributionMatrix$(Fit, NewFit)$;

Before entering the main optimization search, the FGEA performs the following five initialization operations. Similar to other decomposition-based algorithms [24], [39], [42], the method in work [42] is used to generate a set of uniformly distributed reference vectors (line 1). Also, a population with N solutions is initialized in a random way (line 2), and the used function evaluations (UFE) is initialized (line 3). Then, the FGEA initializes the number of available evolutionary operators K and the memory length L (line 4). After that, one $K \times L$ memory matrix M is initialized for a subspace, and each element in each matrix is initialized as zero (line 5). The matrix M^i is created for the i-th subspace. Assume l denotes the index of generation, the element $M^i(k, l)$ stores the contribution of the k-th evolutionary operator in the i-th subspace to the population in the previous $(L + 1 - l)$-th generation.

After the above initialization, the FGEA continues to perform the following four main operations: 1) generate a new offspring population; 2) update ideal and nadir points; 3) update the population using environmental selection; 4) update the memory matrix for each subspace. Since how to integrate multiple evolutionary operators to improve the population convergence is our main focus, the first and fourth operations are detailed in Algorithm 2 and Algorithm 3, respectively.

In FGEA, all the non-dominated solutions in the combined population $P \cup Q$ are employed to update the ideal and nadir points (line 9), where P and Q are the parent and offspring population, respectively. Assuming that the subpopulation $D \subseteq P \cup Q$ contains all the

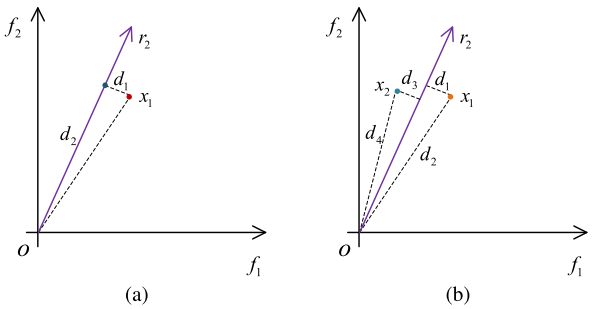

FIGURE 2. Examples of fitness calculation.

non-dominated solutions in the combined population, then the ideal point Z_{min} and nadir point Z_{max} are updated as $Z_{min} = [\min_{p\in D} f_1(p), \min_{p\in D} f_2(p), \cdots, \min_{p\in D} f_m(p)]$ and $Z_{max} = [\max_{p\in D} f_1(p), \max_{p\in D} f_2(p), \cdots, \max_{p\in D} f_m(p)]$, respectively.

For a subspace, it may contain multiple associated solutions, then the penalty-based boundary intersection method [41] is employ to sort these solutions. Then, the fitness $Fit(x_j)$ of a solution x_j, which is associated to the i-th subspace, is defined as:

$$Fit(x_j) = \parallel F'(x_j) \parallel \times(\cos(F'(x_j), r_i) + \sin(F'(x_j), r_i)), \quad (2)$$

where $F'(x_j)$ represents the normalized objective vector of solution x_j, i.e., $F'(x_j) = (F(x_j) - Z_{min})/(Z_{max} - Z_{min})$; the r_i denotes the reference vecotors corresponding to the i-th subspace.

In order to visually illustrate how to calculate the fitness for a solution, Fig. 2(a) gives an example. Assuming that solution x_1 is associated to the second subspace, the normalized objective vector of solution x_1 is represented as a red dot, and its projection on the reference vector r_2 is shown as a blue dot, then the fitness value of the solution is $Fit(x_1) = d_1 + d_2$, where $d_1 = \parallel F'(x_j) \parallel \times \sin(F'(x_j), r_2)$ and $d_2 = \parallel F'(x_j) \parallel \times \cos(F'(x_j), r_2)$.

For a subspace, its fitness value corresponds to the fitness value of its associated solution. The symbol *Fit* in line 10 is a $1 \times N$ array recording the fitness values of all the N subspaces before environmental selection, and the $Fit(i)$ corresponds to fitness of the i-th subspace. Similarly, the array *NewFit* in line 12 represents the fitness for all the subspaces after environmental selection.

For the environmeantal selection approach (line 11), it is not the main contribution of this paper, and we employ the classical objective space partition-based environmeantal selection approach [20]. All the solutions in the combined population are first associated to different subspaces, and one solution with the minimal fitness is selected from each subspace.

B. ADAPTIVE STRATEGY

Assuming that the solution $\boldsymbol{x}$ is generated by the k-th evolutionary operator in the i-th subspace, the contribution $C_{i,k}$ of the k-th evolutionary operator in the i-th subspace is defined as the fitness improvement of the subspace associated with solution $\boldsymbol{x}$.

Fig. 2(b) gives an example to illustrate how to calculate the contribution. Assume that the solution $\boldsymbol{x}_1$ is previously associated with the second subspace defined by reference vector $\boldsymbol{r}_2$. Since the solutions associated with a subspace are not necessarily generated in this subspace, we assume that the solution $\boldsymbol{x}_2$ is generated by the first evolutionary operator in the sixth subspace and is also associated to the second subspace. Then, the contribution $C(6, 1)$ of the first evolutionary operator in the sixth subspace can be calculated as $C(6, 1) = Fit(\boldsymbol{x}_1) - Fit(\boldsymbol{x}_2) = (d_1 + d_2) - (d_3 + d_4)$.

When selecting an evolutionary operator for a subspace, both the local and global contributions of this evolutionary operator are considered, and the aggregation of local and global conributions is called overall conribution. The local contribution of an evolutionary operator refers to the sum of its contribution values in this space during the past L generations, while its global contribution is defined as its contributions in all the subspaces during this generation. Assume that k and i are respectively indexes of evolutionary operator and subspace, and then the overall conribution $OC_{i,k}$ of k-th evolutionary operator in i-th subspace is defined as follows.

$$OC_{i,k} = LC_{i,k} + GC_k, \quad (3)$$

where $LC_{i,k}$ and GC_k denotes the local and global contribution of the k-th evolutionary operator, respectively. They can be calculated as:

$$LC_{i,k} = \sum_{l=1}^{L} M^i(k, l), \quad (4)$$

$$GC_k = \sum_{i=1}^{N} M^i(k, L), \quad (5)$$

where M^i represents the memory matrix for storing the contributions of different evolutionary operators in the i-th subspace.

With the overall contribution of each evolutionary operator in each subspace, the selection probability $p_{i,k}$ of the k-th evolutionary operator for the i-th subspace is defined as:

$$p_{i,k} = \frac{C_{i,k} + \triangle/K}{\sum_{k=1}^{K} C_{i,k} + \triangle}, \quad (6)$$

where $\triangle$ is a very small positive number, e.g., $1 \times e^{-6}$, to avoid the case that the denominator is 0. This definition can guarantee that the sum of selection probabilities of all the K evolutionary operators is 1 for a subspace. That is,

$$\begin{aligned} \sum_{k=1}^{K} p_{i,k} &= \sum_{k=1}^{K} \frac{C_{i,k} + \triangle/K}{\sum_{k=1}^{K} C_{i,k} + \triangle} \\ &= \frac{\sum_{k=1}^{K} C_{i,k} + \sum_{k=1}^{K} \triangle/K}{\sum_{k=1}^{K} C_{i,k} + \triangle} \\ &= \frac{\sum_{k=1}^{K} C_{i,k} + \triangle}{\sum_{k=1}^{K} C_{i,k} + \triangle} = 1. \end{aligned} \quad (7)$$

On the basis of the selection probability of each evolutionary operator for each subspace, we design an algorithm

Algorithm 2: GenerateOffspring()

Input: Current population P; memory matrix for each subspace, i.e., $M^1, M^2, \cdots, M^N$;
Output: An offspring population Q;

1 $Q \leftarrow \emptyset$;
2 **for** $i = 1 \rightarrow N$ **do**
3 $r \leftarrow$ Generate a random number between 0 and 1;
4 $t \leftarrow 1$;
5 **for** $k = 1 \rightarrow K$ **do**
6 **if** $r \leq \sum_{k=1}^{t} p_{i,k}$ **then**
7 $t \leftarrow k$;
8 BREAK;
9 $q \leftarrow$ Generate a new solution using the t-th operator on solutions in i-th subspace and its neighbourhood;
10 $Q \leftarrow Q \bigcup \{q\}$;

Algorithm 3: UpdateContributionMatrix()

Input: Fit and $NewFit$; memory matrix for each subspace, i.e., $M^1, M^2, \cdots, M^N$;
Output: Updated memory matrix for each subspace, i.e., $M^1, M^2, \cdots, M^N$;

1 **for** $i = 1 \rightarrow N$ **do**
2 **for** $k = 1 \rightarrow K$ **do**
3 **for** $l = 1 \rightarrow L - 1$ **do**
4 $M^i(k, l) \leftarrow M^i(k, l + 1)$;
5 $M^i(k, L) \leftarrow 0$;
6 $meanF \leftarrow \frac{1}{N} \sum_{i=1}^{N} Fit(i)$;
7 **for** $i = 1 \rightarrow N$ **do**
8 **if** $NewFit(i) > 0$ **then**
9 **if** $Fit(i) > 0$ **then**
10 $Con \leftarrow Fit(i) - NewFit(i)$;
11 **else**
12 $Con \leftarrow meanF$;
13 $I_s \leftarrow$ Get the index of source space of x_i;
14 $I_o \leftarrow$ Get the index of operator of x_i;
15 $M^{I_s}(I_o, L) \leftarrow Con$;

to generate an offspring population, which is described in Algorithm 2.

As shown in Algorithm 2, the inputs of function GenerateOffspring() are the current population and the N memory matrices. Since this function is used to generate an offspring population based on the contributions of different evolutionary operators, its output is naturally this offspring population, denoted as Q. First of all, the population Q is initialized to an empty set (line 1). Then, all the subspaces are checked to generate new solutions (lines 2-10). When checking a subspace, this function makes use of roulette-wheel approach [43] to select one evolutionary operator based on the selection probabilities (lines 3-8). The symbol r represents a random number, and t denotes the index of evolutionary operators. The process from line 3 to 8 seems to be complex, an example is given to detail it. Suppose there are three evolutionary operators, their selection probabilities are respectively $p_{i,1} = 0.1$, $p_{i,2} = 0.6$, and $p_{i,3} = 0.3$. When the random number r satisfying $0.1 < r \leq 0.7$, the second evolutionary operator will be selected since $r > p_{i,1}$ and $r \leq p_{i,1} + p_{i,2}$. Then, the selected evolutionary operator will be applied to the solutions coming from the subspace and its neighborhoods to generate a new solution (line 9).

In each generation, according to the fitness value of each subspace before and after environment selection, the proposed FGEA updates memory matrices storing contribution of each evolutionary operator in each subspace, which is shown in Algorithm 3.

As shown in Algorithm 3, the main inputs of Function UpdateContributionMatrix() are the fitness of each subspace before and after environmental selection, and memory matrix for each subspace.

The Function UpdateContributionMatrix() consists of two steps. The first step checks each subspace to retain the contributions of each evolutionary operator from the first to the $(L - 1)$-th generation (lines 2-4), and initialize the contribution of each evolutionary operator in the current generationto 0 (line 5). Then, on the basis of the fitness improvement of each subspace, the second step updates the contributions of corresponding evolutionary operators in corresponding subspaces (lines 7-15). When a subspace has an associated solution, its fitness value will be larger than zero (line 8). In this case, the contribution generated by this new associated solution is calculated (lines 9-12). Then, the indexes of the subspace and evolutionary operator are found (lines 13-14). After that, the contribution of the corresponding evolutionary operator in the corresponding subspace is updated (line 15).

C. TIME COMPLEXITY ANALYSIS

From Algorithm 1, we can observe that the time complexity of the proposed FGEA mainly comes from three parts: 1) generate offspring population (line 7, Algorithm 1); 2) environmental selection operator (line 11, Algorithm 1); 3) update contribution matrix (line 13, Algorithm 1). Next, we analyze these three parts respectively.

The main process of the first part is shown in Algorithm 2. The time complexity of choosing an evolutionary operator is $O(K)$ (lines 5-8, Algorithm 2), where K is the number of evolutionary operators. Selecting offspring solutions for generating a new solution takes $O(N)$, where N is the number of sub-spaces. Generating a new solution using an evolutionary operator takes $O(n)$, where n is the number of decision variables. Then, the time complexity of generating a new solution is $O(K+N+n)$, which results in $O(N \cdot K+N^2+N \cdot n)$ for Algorithm 2 to generate an offspring population.

TABLE 1. Main parameters of the five comparison MOEAs.

MOEAs	Evolutionary Operators	Main Parameters
FRRMAB	DE variants + PM	$CR = 1, F = 0.5, p_m = 1/n, \eta_m = T = 20, \delta = 0.9, n_r = 2$
MOEA/D-M2M	SBC + PM	$p_c = 1, \eta_c = 30, p_m = 1/n, \eta_m = 20, K = S = 10$
IM-MOEA	GP-IM + PM	$K = 10, L = 3, p_m = 1/n, \eta_m = 20$
NSGA-III	SBC + PM	$p_c = 1, \eta_c = 30, p_m = 1/n, \eta_m = 20$
MOEA/D-AM2M	SBC	$p_c = 1, \eta_c = 30, p_m = 1/n, \eta_m = 20, K = 10, G = 100$

For the environmental selection operator [20], the time complexity of associating a solution to a reference vector is $O(N \cdot m)$, where m is the number of objective. Then, associating all the N solutions takes $O(N^2 \cdot m)$. It takes $O(N)$ to select one solution for a sub-space, which results in $O(N^2)$ to update all the N sub-spaces. Therefore, the time complexity of the environmental selection operator is $O(N^2 \cdot m + N^2) = O(N^2 \cdot m)$.

As shown in Algorithm 3, the time complexity of moving the memory values in the N memory matrices is $O(N \cdot K \cdot L)$ (lines 1-5, Algorithm 3). Then, it takes $O(N)$ to record the new contributions of the N new solutions (lines 7-15, Algorithm 3). Therefore, the time complexity of updating the memory matrix for each sub-space is $O(N \cdot K \cdot L)$.

III. EXPERIMENTAL STUDIES

In this section, to validate the performance of our proposal, we compre it with five state-of-the-art MOEAs on widely-used benchmarks in multi-objective optimization community. The five comparison MOEAs are FRRMAB [32], MOEA/D-M2M [20], IM-MOEA [44], NSGA-III [42], MOEA/D-AM2M [39].

A. EXPERIMENT DESIGN

Benchmarks: 35 MOPs with complex Pareto sets from four test suites, including MOP1-MOP7 [20], BT1-BT9 [21], IF1-IF10 [44], LF1-LF9 [19], are employed to compare the proposal with the five peer algorithms. These MOPs challenge the MOEAs with complex linkages among decision variables, which make them harder than the other MOPs.

Stop condition: As is frequently done, the stop condition in comparison experiments is set as maximal function evaluations (MFE). The MFE for MOPs from test suites MOP [20] and BT [21] are set as 4×10^5, and that for MOPs from test suites IF [44] and LF [19] are set as 1×10^5.

Population size: Referring to existing works [39], the settings of population size N only consider the number of optimization objectives. For the 2- and 3-objective MOP, we set the population size as 200 and 300, respectively.

Comparison indicator: The indicators inverted generational distance (IGD) [45] and hypervolume (HV) [46] can simultaneously measure both convergence and diversity of the output population. They have been frequently employed to examine the performance of MOEAs. Based on this, we also employ these two indicators to compare the performance of the proposed FGEA and the five peer MOEAs.

1) *IGD*: For an output population P, calculation formula of indicator IGD is as follows:

$$IGD(P) = \frac{\sum_{q \in P^*} \min_{p \in P} D(F(q), F(p))}{|P^*|}, \tag{8}$$

where P^* denotes a set of uniformly distributed solutions on the true PF, $D(F(q), F(p))$ represents the Euclidean distance between objective vectors of the two solutions q and p. From the definition of indicator IGD in (8), it can be derived that a smaller IGD value of an output population means the corresponding MOEAs having better performance in terms of both convergence and diversity. In the experiments, the number of solutions in the set P^* is set to about 8000 for each MOP.

2) *HV*: Suppose $\boldsymbol{r} = \{r_1, r_2, \ldots, r_m\}$ is a reference point. For an output population P, the indicator HV corresponds to the volume of space consisting of the objective vectors of all the solutions in P and the reference point, which is defined as follows.

$$HV(P) = L(\bigcup_{p \in P} [f_1(p), r_1] \times \cdots \times [f_m(p), r_m]), \tag{9}$$

where $L(\triangle)$ denotes the Lebesgue measure. For each MOP, the reference point is set as 1.2 times of the nadir point of its PF. From the definition of indicator HV in (9), a larger HV value of an output population means better performance of the corresponding MOEA with respect to both the convergence and diversity.

Evolutionary operator pool for FGEA: The simulated binary crossover (SBC) [29] and differential evolution (DE) [47] have distinct search patterns and have been widely employed in multi-objective optimization. In the proposed FGEA, the simulated binary crossover [29] and two classical variants of differential evolution (i.e., DE/rand/1 and DE/rand/2) are choosed to construct evolutionary operator pool.

The main parameters for the evolutionary operators are set as follows. The simulated binary crossover employs a distribution index as 20. For the DE/rand/1 and DE/rand/2, the crossover constant and weighting factor are set as $CR = 1.0$ and $F = 0.5$. Similar to the work [32], after applying these evolutionary operators, each new generated solution also performs the polynomial mutation operator [30]. For the polynomial mutation (PM) operator, we set $\eta = 20$ and $p_m = 1/n$.

The main parameters of the five comparison MOEAs are summarized in Table 1. For the DE operator, T represents the

TABLE 2. In terms of IGD, comparison results of the six algorithms on various MOPs.

Problem Names	m	FRRMAB	MOEA/D-M2M	IM-MOEA	NSGA-III	MOEA/D-AM2M	FGEA
MOP1	2	2.3411e-1 (1.22e-1) −	1.4217e-2 (4.81e-4) −	5.5614e-2 (2.85e-3) −	3.4225e-1 (5.11e-3) −	1.2021e-1 (3.71e-2) −	1.1224e-2 (8.32e-4)
MOP2	2	2.0436e-1 (4.80e-2) −	9.8194e-3 (6.84e-3) −	8.7942e-2 (1.07e-2) −	3.5420e-1 (4.06e-3) −	2.1268e-1 (6.99e-2) −	5.8792e-3 (1.77e-3)
MOP3	2	5.2811e-1 (1.35e-1) −	1.2189e-2 (4.14e-3) +	1.5123e-1 (2.98e-2) −	4.9151e-1 (5.06e-2) −	3.4248e-1 (1.04e-1) −	1.7751e-2 (3.84e-2)
MOP4	2	2.3023e-1 (4.51e-2) −	4.0338e-3 (6.11e-4) −	3.1115e-2 (1.08e-2) −	2.9418e-1 (1.23e-2) −	1.2925e-1 (5.00e-2) −	3.4416e-3 (1.38e-3)
MOP5	2	3.0626e-1 (2.31e-2) −	2.0770e-2 (1.61e-3) −	1.1257e-1 (1.40e-2) −	2.1847e-1 (2.07e-2) −	2.3666e-1 (1.01e-1) −	1.7593e-2 (1.22e-3)
MOP6	3	2.6720e-1 (3.20e-2) −	8.5352e-2 (5.62e-3) −	1.6854e-1 (5.35e-3) −	3.0939e-1 (1.98e-5) −	2.3687e-1 (4.52e-2) −	8.0658e-2 (4.92e-3)
MOP7	3	3.0169e-1 (3.63e-2) −	1.5341e-1 (1.66e-2) +	2.7175e-1 (6.29e-3) −	3.5716e-1 (3.54e-5) −	3.3920e-1 (3.71e-2) −	2.0190e-1 (1.49e-2)
BT1	2	2.2737e-1 (1.31e-1) −	2.1930e+0 (1.59e-1) −	2.4620e+0 (9.06e-2) −	5.5198e-3 (1.71e-3) ≈	1.7460e-2 (7.14e-3) −	3.8329e-3 (9.98e-4)
BT2	2	1.8291e-1 (9.21e-3) −	3.3508e-1 (2.58e-2) −	9.4175e-1 (2.66e-2) −	3.2412e-2 (1.93e-2) ≈	1.2638e-1 (1.17e-2) −	3.3471e-2 (8.32e-3)
BT3	2	4.8208e-2 (1.28e-2) −	1.2859e+0 (3.25e-1) −	1.9291e+0 (9.57e-2) −	1.2440e-2 (3.32e-3) −	1.2469e-1 (3.56e-2) −	6.3008e-3 (1.63e-3)
BT4	2	5.4277e-2 (2.86e-2) −	1.1938e+0 (2.17e-1) −	1.9190e+0 (8.08e-2) −	1.2305e-2 (8.60e-4) −	2.0153e-2 (3.33e-3) −	5.8827e-3 (7.71e-4)
BT5	2	3.9839e-1 (1.71e-1) −	2.4055e+0 (8.37e-2) −	2.5426e+0 (6.91e-2) −	3.5487e-3 (8.86e-4) ≈	4.2127e-2 (2.40e-2) −	5.1444e-3 (2.82e-3)
BT6	2	3.7835e-1 (1.86e-1) ≈	2.6999e-1 (7.77e-2) −	2.9351e-1 (2.34e-2) −	3.1544e-1 (2.42e-2) −	3.2437e-1 (7.06e-3) −	2.5138e-1 (1.74e-1)
BT7	2	2.9892e-1 (1.18e-1) −	4.4426e-2 (3.46e-2) +	5.1075e-2 (3.31e-3) +	1.2217e-1 (5.34e-2) ≈	1.7400e-1 (9.53e-2) −	1.0048e-1 (3.38e-2)
BT8	2	1.7998e-1 (1.30e-1) −	4.0656e-1 (1.64e-1) −	1.2227e+0 (1.60e-1) −	3.2192e-1 (2.19e-2) −	3.1891e-1 (1.51e-2) −	1.6312e-1 (2.98e-2)
BT9	3	2.3173e+0 (5.71e-2) −	3.0974e+0 (1.96e-1) −	2.5378e+0 (8.55e-2) −	3.2189e-2 (1.50e-3) +	6.2509e-2 (4.77e-3) +	2.6972e-1 (6.65e-2)
IF1	2	2.0932e-3 (3.89e-5) +	3.6215e-3 (1.73e-4) ≈	3.2278e-3 (2.65e-4) +	1.9405e-1 (7.72e-2) −	9.6772e-2 (4.59e-2) −	3.9864e-3 (3.82e-4)
IF2	2	2.0753e-3 (3.25e-5) +	3.7811e-3 (1.32e-4) +	3.7942e-3 (1.02e-4) +	3.8882e-1 (6.00e-2) −	2.2765e-1 (1.27e-1) −	5.4218e-3 (7.66e-4)
IF3	2	1.9617e-3 (1.28e-4) −	3.8481e-3 (7.44e-5) −	2.7380e-3 (5.69e-5) −	3.4493e-3 (1.14e-3) −	6.0622e-3 (7.41e-4) −	1.7757e-3 (3.35e-4)
IF4	3	4.0937e-2 (1.05e-3) +	1.4833e-1 (1.17e-2) −	9.4109e-2 (1.57e-3) −	3.6362e-1 (8.03e-2) −	2.4602e-1 (3.37e-2) −	6.4742e-2 (5.19e-3)
IF5	2	2.3499e-3 (6.65e-5) +	4.0894e-3 (6.91e-5) ≈	3.0180e-3 (6.70e-5) +	3.6566e-2 (9.08e-3) −	4.9578e-2 (1.71e-2) −	3.3577e-3 (4.34e-4)
IF6	2	4.9512e-3 (8.43e-4) −	6.2171e-3 (3.50e-4) −	4.8682e-3 (4.59e-5) −	5.4150e-2 (1.51e-2) −	5.9350e-2 (2.26e-2) −	4.3155e-3 (4.47e-4)
IF7	2	2.9566e-3 (5.45e-4) −	4.3855e-3 (2.07e-4) −	4.9671e-3 (1.78e-4) −	2.2492e-2 (8.90e-3) −	5.9136e-3 (1.16e-3) −	2.6378e-3 (2.66e-4)
IF8	3	3.5719e-1 (3.18e-5) −	1.3604e-1 (1.34e-2) −	9.2728e-2 (3.91e-3) −	1.9291e-1 (2.93e-2) −	1.9745e-1 (3.40e-2) −	6.9778e-2 (5.32e-3)
IF9	2	1.7583e-2 (2.36e-2) −	2.5747e-2 (3.05e-2) −	4.0031e-3 (1.49e-3) +	8.2916e-3 (1.64e-3) ≈	1.2037e-2 (2.36e-3) −	9.3234e-3 (5.78e-3)
IF10	2	2.8369e+0 (4.53e+0) +	1.2795e+0 (1.14e+0) +	3.8672e+0 (2.12e+0) +	1.2276e+1 (1.66e+0) −	1.9550e+1 (6.06e+0) −	1.0211e+1 (7.18e+0)
LF1	2	1.9939e-3 (4.33e-5) −	3.1546e-3 (8.79e-5) −	3.6024e-3 (9.84e-5) −	1.4455e-2 (3.36e-3) −	1.5844e-2 (4.64e-3) −	1.6873e-3 (1.02e-4)
LF2	2	4.2864e-3 (1.16e-3) −	1.8827e-2 (7.81e-3) −	5.7797e-2 (9.49e-3) −	9.2062e-2 (1.77e-2) −	7.4252e-2 (1.42e-2) −	3.8408e-3 (1.32e-2)
LF3	2	3.6867e-3 (3.95e-4) +	3.3506e-3 (6.04e-4) +	3.4309e-2 (1.86e-2) ≈	3.6822e-2 (6.06e-3) ≈	4.2470e-2 (1.51e-2) ≈	4.3649e-2 (2.84e-2)
LF4	2	3.5689e-3 (5.53e-4) −	1.1171e-2 (1.77e-3) −	1.7751e-2 (4.31e-3) −	4.6925e-2 (1.42e-2) −	3.4113e-2 (7.12e-3) −	3.3787e-3 (3.17e-2)
LF5	2	8.7000e-3 (1.82e-3) +	9.8546e-3 (7.82e-4) +	2.6004e-2 (5.97e-3) ≈	2.7419e-2 (5.23e-3) ≈	3.4956e-2 (5.88e-3) −	2.9288e-2 (1.40e-2)
LF6	3	5.1675e-2 (9.61e-3) −	1.2475e-1 (1.01e-2) −	9.1102e-2 (2.83e-3) −	8.8035e-2 (8.12e-2) −	1.4762e-1 (1.64e-2) −	5.1048e-2 (1.02e-2)
LF7	2	4.0701e-3 (5.46e-5) −	4.0587e-3 (1.39e-4) −	8.6579e-2 (2.15e-2) −	1.3373e-1 (3.29e-2) −	1.8451e-1 (5.78e-2) −	3.4139e-3 (1.23e-3)
LF8	2	2.1227e-3 (6.87e-5) +	8.3892e-3 (5.49e-3) +	1.2656e-1 (1.34e-2) −	1.5431e-1 (3.31e-2) −	1.6339e-1 (3.58e-2) −	1.0953e-1 (7.62e-2)
LF9	2	5.2102e-3 (2.87e-3) −	2.4536e-2 (9.21e-3) −	5.3422e-2 (9.82e-3) −	1.0307e-1 (3.25e-2) −	9.5081e-2 (2.87e-2) −	4.3516e-3 (6.36e-2)
+/ − / ≈		8 / 26 / 1	8 / 25 / 2	6 / 27 / 2	1 / 27 / 7	1 / 33 / 1	

neighborhood size of reference vectors, δ corresponds to the probability to choose parent solutions from T neighbors, and n_r denotes the maximal number of solutions can be replaced by a new solution. For MOEA/D-M2M, IM-MOEA, and MOEA/D-AM2M, K denotes the number of subpopulations, and S represents the number of solutions in a subpopulation. The symbol L in IM-MOEA denotes the model group size. The parameter G in MOEA/D-AM2M means the number of generations for adjusting subregions.

Unless otherwise specified, other parameters of MOPs and peer algorithms use default values in the experimental platform PlatEMO.[1]

[1]https://github.com/BIMK/PlatEMO

B. COMPARISON RESULTS AND ANALYSIS

For each MOP, the proposed FGEA and the five comparison MOEAs run 30 times independently. For indicators IGD and HV, we summarize their average values and standard deviations in Tables 2 and 3, respectively.

To ensure statistical significance, the two-sided Wilcoxon's rank sum test with significance degree as 0.05 is conducted to compare the results of the FGEA with each peer MOEA. The marks +, −, and ≈ in Tables 2 and 3 respectively represent that the peer MOEA performs better than, worse than, and similar to the FGEA. Regarding each MOP, the best average value among the six MOEAs is emphasized using the gray background.

From the results with gray backgrounds in Tables 2 and 3, we can notice that the proposed FGEA ranks

TABLE 3. In terms of HV, comparison results of the six algorithms on various MOPs.

Problem Names	m	FRRMAB	MOEA/D-M2M	IM-MOEA	NSGA-III	MOEA/D-AM2M	FGEA
MOP1	2	4.2718e-1 (1.69e-1) −	7.0675e-1 (5.14e-4) −	6.5267e-1 (3.09e-3) −	2.8388e-1 (9.86e-3) −	5.7688e-1 (4.94e-2) −	7.1086e-1 (8.53e-4)
MOP2	2	2.5968e-1 (3.99e-2) −	4.3465e-1 (9.85e-3) −	3.3070e-1 (1.50e-2) −	1.7371e-1 (8.74e-4) −	2.4872e-1 (4.04e-2) −	4.4070e-1 (2.95e-3)
MOP3	2	1.0019e-1 (4.01e-2) −	3.3931e-1 (3.09e-3) +	2.0849e-1 (2.41e-2) −	9.2012e-2 (5.05e-3) −	1.4876e-1 (4.24e-2) −	3.3461e-1 (3.66e-2)
MOP4	2	3.4075e-1 (5.21e-2) −	5.9723e-1 (8.58e-4) ≈	5.4994e-1 (1.87e-2) −	2.9317e-1 (9.29e-3) −	4.2050e-1 (5.16e-2) −	5.9701e-1 (1.71e-3)
MOP5	2	4.0183e-1 (6.14e-3) −	6.9813e-1 (1.82e-3) −	5.9199e-1 (5.50e-3) −	4.4521e-1 (1.77e-2) −	4.6160e-1 (1.19e-1) −	7.0261e-1 (1.25e-3)
MOP6	3	6.6588e-1 (3.64e-2) −	7.8657e-1 (6.57e-3) −	7.3037e-1 (3.01e-3) −	6.2064e-1 (2.84e-4) −	6.8050e-1 (4.51e-2) −	8.0423e-1 (3.43e-3)
MOP7	3	4.3651e-1 (2.33e-2) −	5.0985e-1 (1.62e-2) +	4.4835e-1 (2.07e-3) −	4.0599e-1 (4.82e-4) −	3.9918e-1 (2.04e-2) −	5.0172e-1 (4.95e-3)
BT1	2	4.1381e-1 (1.51e-1) −	0.0000e+0 (0.00e+0) −	0.0000e+0 (0.00e+0) −	7.1573e-1 (2.40e-3) ≈	7.0018e-1 (6.87e-3) −	7.1849e-1 (1.67e-3)
BT2	2	4.9155e-1 (9.84e-3) −	3.2855e-1 (2.55e-2) −	9.6429e-4 (1.39e-3) −	6.8003e-1 (2.52e-2) ≈	5.5660e-1 (1.47e-2) −	6.7906e-1 (1.07e-2)
BT3	2	6.4320e-1 (2.30e-2) −	1.8095e-4 (4.43e-4) −	0.0000e+0 (0.00e+0) −	6.9937e-1 (6.08e-3) −	5.1426e-1 (4.22e-2) −	7.0985e-1 (5.15e-3)
BT4	2	6.5185e-1 (3.81e-2) −	6.6374e-5 (1.63e-4) −	0.0000e+0 (0.00e+0) −	7.1048e-1 (9.44e-4) −	7.0138e-1 (2.97e-3) −	7.1700e-1 (1.01e-3)
BT5	2	2.4288e-1 (1.62e-1) −	0.0000e+0 (0.00e+0) −	0.0000e+0 (0.00e+0) −	6.5952e-1 (1.75e-3) −	6.3739e-1 (2.00e-2) −	6.6439e-1 (2.61e-3)
BT6	2	2.5988e-1 (1.49e-1) −	3.6559e-1 (3.61e-2) −	3.1871e-1 (2.66e-2) −	3.8416e-1 (2.12e-2) ≈	3.9022e-1 (2.48e-2) +	3.8017e-1 (1.65e-1)
BT7	2	3.2931e-1 (1.38e-1) −	6.4867e-1 (5.46e-2) ≈	6.3377e-1 (9.75e-3) +	5.5592e-1 (7.05e-2) ≈	5.2698e-1 (7.30e-2) ≈	5.8447e-1 (3.56e-2)
BT8	2	5.0764e-1 (1.70e-1) −	2.8339e-1 (1.34e-1) −	7.6320e-5 (1.87e-4) −	3.6730e-1 (3.60e-2) −	3.8317e-1 (2.58e-2) −	5.1654e-1 (1.74e-2)
BT9	3	0.0000e+0 (0.00e+0) −	0.0000e+0 (0.00e+0) −	0.0000e+0 (0.00e+0) −	5.7448e-1 (1.98e-3) +	5.5494e-1 (5.99e-3) +	1.3678e-1 (4.79e-2)
IF1	2	7.2173e-1 (9.77e-5) +	7.1916e-1 (4.07e-4) ≈	7.1968e-1 (7.32e-4) +	5.8522e-1 (4.43e-2) −	6.4048e-1 (2.50e-2) −	7.1847e-1 (6.87e-4)
IF2	2	4.4615e-1 (8.94e-5) +	4.4290e-1 (2.49e-4) +	4.4786e-1 (1.81e-4) +	1.4071e-1 (2.59e-2) −	2.3388e-1 (8.42e-2) −	4.3802e-1 (1.83e-3)
IF3	2	3.8935e-1 (2.21e-4) −	3.8707e-1 (1.46e-4) −	3.8817e-1 (1.81e-4) −	3.8600e-1 (1.57e-3) −	3.8230e-1 (9.82e-4) −	3.9815e-1 (4.50e-4)
IF4	3	5.6096e-1 (7.37e-4) +	3.9305e-1 (1.10e-2) −	4.6009e-1 (3.40e-3) −	3.4642e-1 (3.51e-3) −	3.4933e-1 (4.37e-2) −	5.2265e-1 (4.80e-3)
IF5	2	7.2130e-1 (8.04e-5) +	7.1859e-1 (2.30e-4) −	7.1860e-1 (2.51e-4) −	6.8558e-1 (4.69e-3) −	6.7035e-1 (1.63e-2) −	7.1966e-1 (6.25e-4)
IF6	2	4.4522e-1 (2.18e-4) −	4.4189e-1 (2.41e-4) −	4.4241e-1 (7.50e-5) ≈	3.7019e-1 (1.63e-2) −	3.6274e-1 (2.36e-2) −	4.5303e-1 (5.24e-4)
IF7	2	3.8800e-1 (5.16e-4) −	3.8620e-1 (3.42e-4) −	3.8547e-1 (1.63e-4) −	3.6573e-1 (9.39e-3) −	3.8353e-1 (1.06e-3) −	3.8997e-1 (3.40e-4)
IF8	3	4.0551e-1 (5.40e-6) −	4.5568e-1 (1.41e-2) −	4.5102e-1 (4.64e-3) −	4.0995e-1 (5.56e-3) −	3.7537e-1 (3.49e-2) −	5.3346e-1 (9.86e-3)
IF9	2	7.0219e-1 (2.66e-2) −	6.9318e-1 (3.51e-2) −	7.1839e-1 (2.38e-3) −	7.1128e-1 (2.64e-3) −	7.0735e-1 (3.44e-3) −	7.2097e-1 (8.06e-3)
IF10	2	7.6972e-2 (5.96e-2) +	9.8661e-2 (4.84e-2) +	0.0000e+0 (0.00e+0) −	0.0000e+0 (0.00e+0) −	0.0000e+0 (0.00e+0) −	2.3265e-2 (5.20e-2)
LF1	2	7.2211e-1 (8.65e-5) −	7.2043e-1 (1.60e-4) −	7.1951e-1 (1.39e-4) −	7.0170e-1 (3.35e-3) −	7.0180e-1 (4.79e-3) −	7.2908e-1 (1.96e-4)
LF2	2	7.1732e-1 (1.73e-3) −	6.9650e-1 (1.30e-2) −	6.3754e-1 (1.60e-2) −	6.0677e-1 (2.50e-2) −	6.2069e-1 (1.90e-2) −	7.7405e-1 (1.51e-2)
LF3	2	7.1888e-1 (4.78e-4) +	7.1414e-1 (1.20e-3) +	6.8955e-1 (1.04e-2) ≈	6.7924e-1 (6.29e-3) −	6.7841e-1 (8.40e-3) −	6.8645e-1 (1.55e-2)
LF4	2	7.0861e-1 (1.12e-3) +	7.0923e-1 (2.14e-3) +	7.0407e-1 (4.57e-3) +	6.7375e-1 (9.02e-3) −	6.8702e-1 (5.16e-3) −	6.9431e-1 (1.79e-2)
LF5	2	7.1285e-1 (2.13e-3) +	7.1120e-1 (1.37e-3) +	6.9504e-1 (4.98e-3) ≈	6.8910e-1 (5.49e-3) −	6.8500e-1 (5.71e-3) −	6.9439e-1 (7.01e-3)
LF6	3	5.3193e-1 (1.20e-2) ≈	3.9794e-1 (2.10e-2) −	4.5658e-1 (5.97e-3) −	4.8230e-1 (2.90e-2) −	4.6638e-1 (1.54e-2) −	5.3282e-1 (2.31e-2)
LF7	2	7.2191e-1 (1.14e-4) −	7.1880e-1 (2.10e-4) −	5.9921e-1 (3.63e-2) −	5.5004e-1 (3.41e-2) −	5.0869e-1 (5.24e-2) −	7.3989e-1 (1.55e-3)
LF8	2	7.2167e-1 (1.53e-4) +	7.1257e-1 (7.55e-3) +	5.4172e-1 (1.93e-2) −	5.0547e-1 (4.05e-2) −	4.8830e-1 (4.06e-2) −	5.7659e-1 (1.10e-1)
LF9	2	4.4188e-1 (3.16e-3) −	4.1703e-1 (1.39e-2) −	3.6954e-1 (1.53e-2) −	3.3839e-1 (2.32e-2) −	3.2306e-1 (3.78e-2) −	4.5948e-1 (5.89e-2)
+/ − / ≈		10 / 24 / 1	8 / 24 / 3	4 / 28 / 3	1 / 30 / 4	2 / 32 / 1	

first in more than half of the MOPs. In terms of indicator IGD, as shown in Table 2, the proposed FGEA significantly outperforms all the five comparison algorithms (i.e., FRRMAB, MOEA/D-M2M, IM-MOEA, NSGA-III, and MOEA/D-AM2M) in 18 out of the 35 MOPs. Similarly, as shown in Table 3, the proposed FGEA has better performance than all the five comparison algorithms in 19 out of the 35 MOPs with respect to the indicator HV. As summarized in the last row of Table 2, among the 35 MOPs, the FGEA generates significantly lower IGD values than FRRMAB, MOEA/D-M2M, IM-MOEA, NSGA-III, and MOEA/D-AM2M on 24, 24, 28, 30, and 32 MOPs, respectively. As summarized in the last row of Table 3, compared with the five comparation algorithms, the algorithm FGEA as before has obvious advantages in obtaining output populations with higher HV values. To sum up, the proposed FGEA has competitive overall performance with respect to both IGD and HV indicators.

The excellent performance of the proposed FGEA is twofold. Firstly, the proposed FGEA employs the objective space partition strategy to maintain population diversity. Although the comparison algorithm FRRMAB also includes an ensemble strategy to integrate multiple evolutionary operators, its performance on the functions MOP1-MOP7 is far worse than the proposed FGEA. As the analysis in the work [23] that the populations of these functions are outwardly divergent, except for the objective space partition strategy, it is difficult for other strategies to maintain good diversity. Secondly, the FGEA considers both local and global contributions

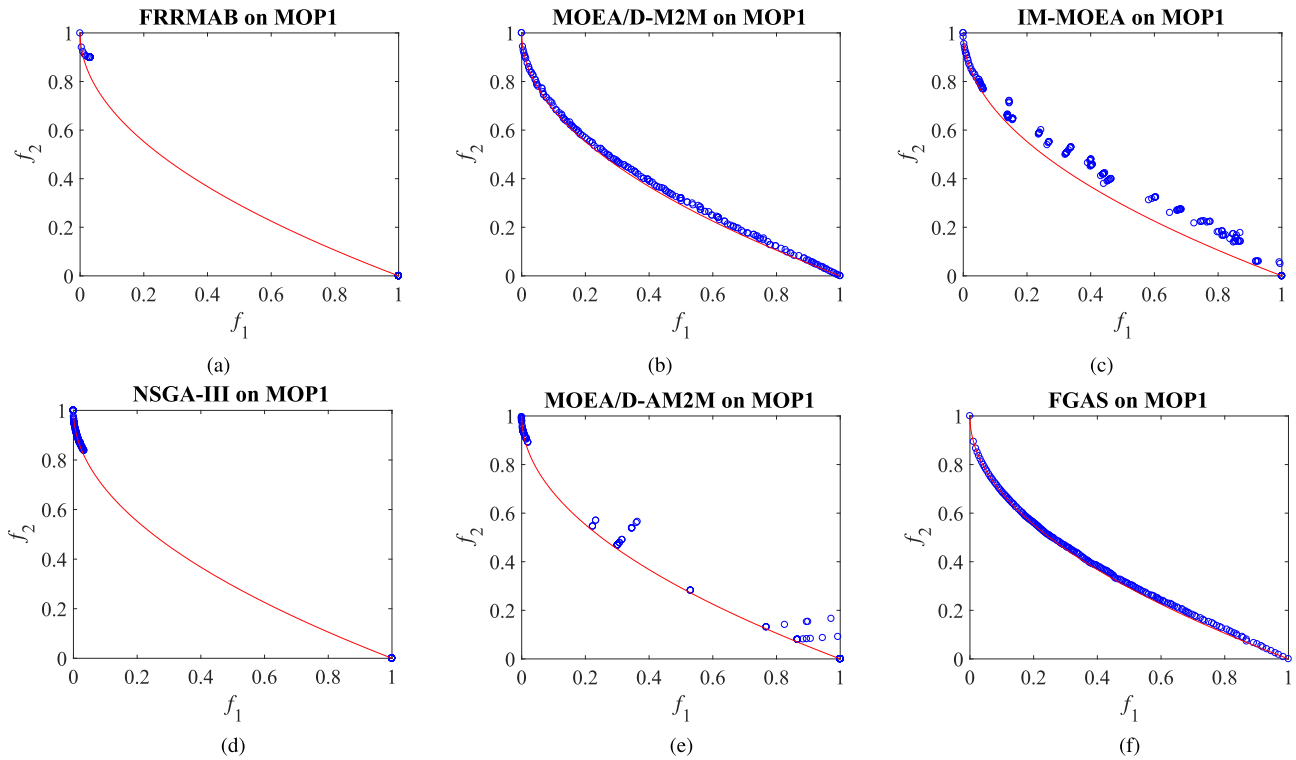

FIGURE 3. Pareto front approximations of the six algorithms on MOP1.

of evolutionary operators when selecting evolutionary operators for each subspace. The comparison algorithms MOEA/D-M2M and MOEA/D-AM2M also employes an objective space partition strategy, but their performance is far behind the algorithm proposed in this article. The main difference between the proposed FGEA and these two comparison algorithms is that the FGEA includes a fine-grained ensemble strategy. By comparing the proposed FGEA and these two comparison algorithms, we can contribute to the advantages of the FGEA to its fine-grained ensemble strategy.

Although both IGD and HV indicators can simultaneously measure the convergence and diversity of populations, the comparison results in Tabele 2 and Tabele 3 are not always consistent according to these two indicators. Taking MOP BT5 as an example, algorithm NSGA-III ranks first based on the indicator IGD, while the algorithm ranked first by HV is our proposed FGEA. The reason lies in the way these two indicators are calculated. The IGD is computed based on the distance between output population and true PF, while the HV is calculated using a predefined reference vector and the non-dominated solutions in output populations.

It is easy to notice that in some MOPs, the HV values of some algorithms are 0, for instance, the HV value of algorithm MOEA/D-M2M in MOP BT1. This is because the convergence of corresponding algorithm is not good enough, and there is no solution in its output population dominating the reference vector.

To visualize the convergence and diversity of the six algorithms, i.e., FRRMAB, MOEA/D-M2M, IM-MOEA, NSGA-III, MOEA/D-AM2M, and FGEA, their Pareto-front approximations for MOP1 and MOP5 with the smallest IGD values among 30 runs are given in Figs. 3 and 4. In these figures, the PFs of MOPs are represented by red curves. For a solution in the output populations, its objective vector is denoted as a blue circle.

As illustrated in Figs. 3(a), (d), and (e), the output populations of the FRRMAB, NSGA-III, and MOEA/D-AM2M mainly concentrates near the two extreme solutions, and cannot well approximate the Pareto-front of MOP1, showing poor diversity. Compared with the above three algorithms, the comparison algorithm IM-MOEA has relatively good diversity, as shown in Fig. 3(c). But, many solutions distributed in the middle region are still far away from the Pareto-front of MOP1, and the convergence of IM-MOEA can be further improved. Among the five comparison algorithms, the MOEA/D-M2M shows the best convergence and diversity, as shown in Fig. 3(b). Compared the output populations in Fig. 3(b) and Fig. 3(f), we can observe that the proposed FGEA and MOEA/D-M2M have similar diversity, but the FGEA has better convergence.

Fig. 4 depicts the distributions of output populations for the six algorithms. It is obvious that the diversity of the comparison algorithms FRRMAB, IM-MOEA, NSGA-III, and MOEA/D-AM2M is very poor. Compared to the results in Figs. 4(b) and (f), we can see that the proposed FGEA

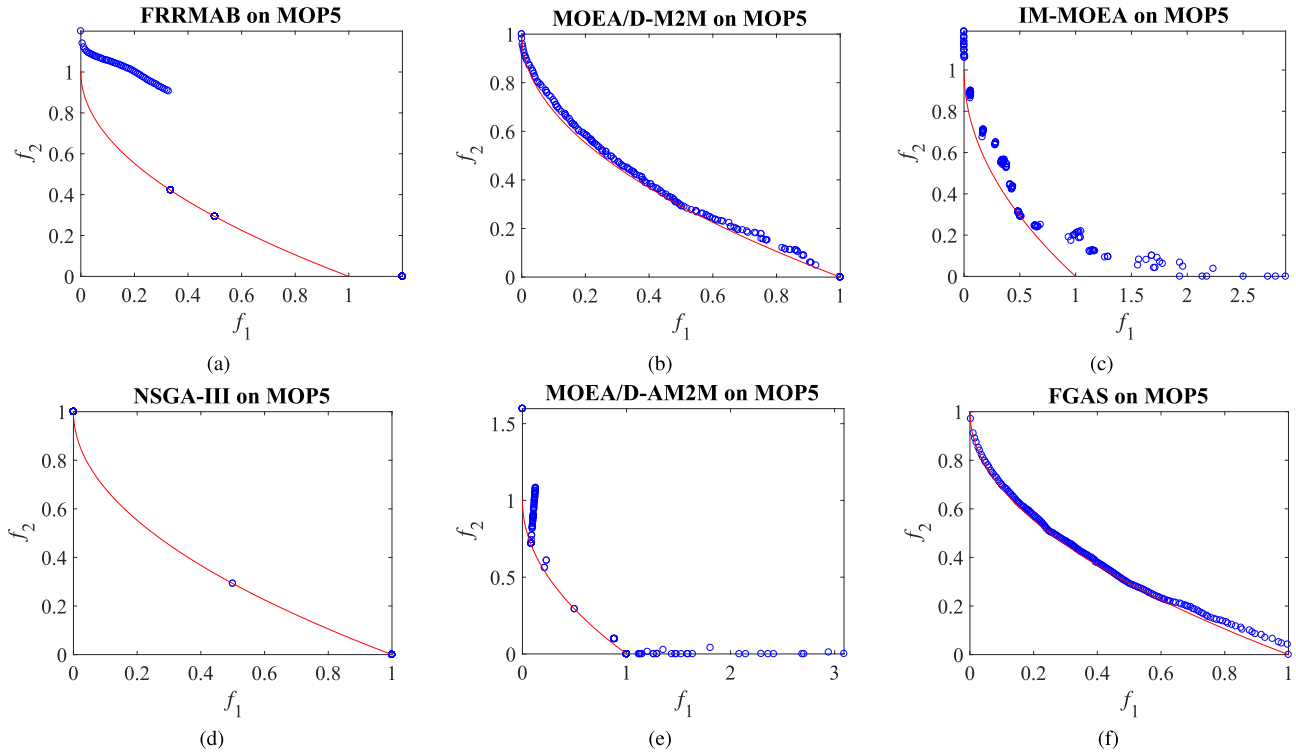

FIGURE 4. Pareto front approximations of the six algorithms on MOP5.

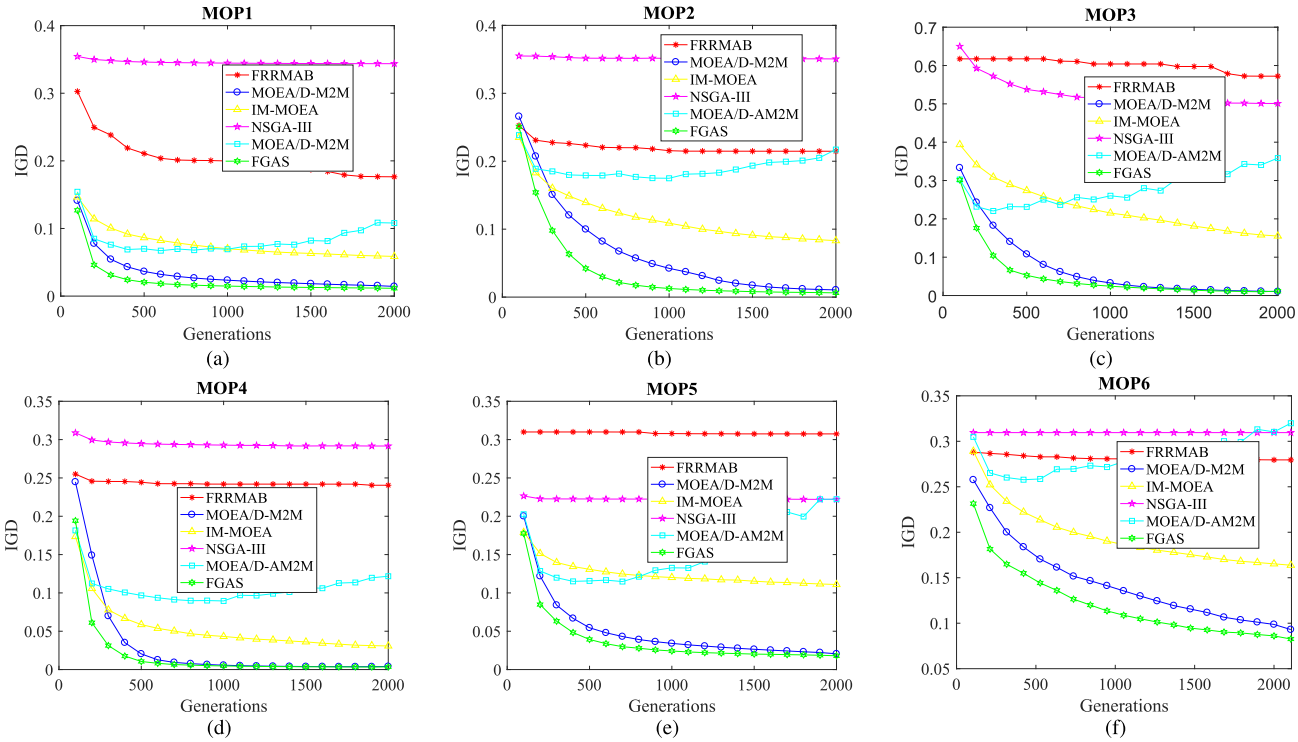

FIGURE 5. The decreasing trend of IGD metric of six algorithms in MOP1-MOP6.

is capable of acquiring a better approximation for the PF of MOP1.

To further compare the behaviors of the six algorithms, the downtrends of their IGD values in solving MOP1-MOP6 are provided in Fig. 5. The horizontal axis represents the number of iteration generations, and the vertical axis represents the IGD values of the output populations from the six algorithms. On each MOP, each algorithm run 31 times, and the change of its average IGD values with the advance of iterations is plotted.

As can be observed from Fig. 5, as the optimization progresses, the IGD values of the algorithms FRRMAB, MOEA/D-M2M, IM-MOEA, NSGA-III, and FGEA decrease gradually. This is easy to understand. With the progress of optimization search, the populations of these algorithms keep approaching the Pareto-fronts, resulting in lower IGD values. For the comparison algorithm MOEA/D-AM2M, it has an abnormal situation, first falling and then rising. This is because the algorithm MOEA/D-AM2M adjusts the reference vector according to the obtained population. When the obtained population can not well cover the Pareto-fronts, the adjusted reference vector can not cover the Pareto-fronts, resulting in poor diversity.

Comparing with the five comparison algorithms, the proposed FGEA appears to contribute significantly to the outstanding convergence speed. Especially before 1000 generations, the convergence speed of the FGEA has an overwhelming advantage. These comparison results visually demonstrate that the fine-grained ensemble mechanism in FGEA is capable of effectively improving the convergence speed.

IV. CONCLUSION AND FUTURE WORK

This article focuses on the properties of complex multi-objective problems that vary throughout both the decision and objective spaces, and develops a fine-grained ensemble approach to choose a suitable evolutionary operator for a subspace during each generation. When choosing an evolutionary operator for a subspace, we design an adaptive strategy to consider both the local and global contributions of the evolutionary operators. To assess the effectiveness of the proposed ensemble approach, extensive experiments on 35 complex MOPs are carried out to compare the FGEA with five baseline algorithms. The numeral experiments demonstrate that the proposed FGEA performs the best overall performance in terms of both convergence and diversity.

Applying the proposed fine-grained ensemble approach to solving real-world MOPs coming from various fields, such as internet of things [48]–[50] and collaborative robots [51], worths future direction. Also, the MOPs from real-world applications often have a large number of decision variables, terms as large-scale MOPs [52]–[56]. It is interesting and challenging to research new techniques for large-scale MOPs.

REFERENCES

[1] R. Cheng, T. Rodemann, M. Fischer, M. Olhofer, and Y. Jin, "Evolutionary many-objective optimization of hybrid electric vehicle control: From general optimization to preference articulation," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 1, no. 2, pp. 97–111, Apr. 2017.

[2] B. Cao, J. Zhao, P. Yang, Y. Gu, K. Muhammad, J. J. P. C. Rodrigues, and V. H. C. de Albuquerque, "Multiobjective 3-D topology optimization of next-generation wireless data center network," *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3597–3605, May 2020.

[3] B. Cao, J. Zhao, Y. Gu, S. Fan, and P. Yang, "Security-aware industrial wireless sensor network deployment optimization," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5309–5316, Aug. 2020.

[4] Y. Zhang, D.-W. Gong, and J. Cheng, "Multi-objective particle swarm optimization approach for cost-based feature selection in classification," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 14, no. 1, pp. 64–75, Jan. 2017.

[5] J. Zhou, X. Yao, Y. Lin, F. T. S. Chan, and Y. Li, "An adaptive multi-population differential artificial bee colony algorithm for many-objective service composition in cloud manufacturing," *Inf. Sci.*, vol. 456, pp. 50–82, Aug. 2018.

[6] J. Yan, W. Pu, S. Zhou, H. Liu, and Z. Bao, "Collaborative detection and power allocation framework for target tracking in multiple radar system," *Inf. Fusion*, vol. 55, pp. 173–183, Mar. 2020.

[7] J. Yan, W. Pu, S. Zhou, H. Liu, and M. S. Greco, "Optimal resource allocation for asynchronous multiple targets tracking in heterogeneous radar networks," *IEEE Trans. Signal Process.*, vol. 68, pp. 4055–4068, 2020.

[8] Y. Liu, C. Yang, and Q. Sun, "Thresholds based image extraction schemes in big data environment in intelligent traffic management," *IEEE Trans. Intell. Transp. Syst.*, early access, Jun. 5, 2020, doi: 10.1109/TITS.2020.2994386.

[9] Q. Zhu, "Research on road traffic situation awareness system based on image big data," *IEEE Intell. Syst.*, vol. 35, no. 1, pp. 18–26, Jan. 2020.

[10] J. Xiong, J. Liu, Y. Chen, and H. A. Abbass, "A knowledge-based evolutionary multiobjective approach for stochastic extended resource investment project scheduling problems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 5, pp. 742–763, Oct. 2014.

[11] D. Hadka and P. Reed, "Borg: An auto-adaptive many-objective evolutionary computing framework," *Evol. Comput.*, vol. 21, no. 2, pp. 231–259, May 2013.

[12] H. Chen, R. Cheng, W. Pedrycz, and Y. Jin, "Solving many-objective optimization problems via multistage evolutionary search," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Aug. 7, 2019, doi: 10.1109/TSMC.2019.2930737.

[13] B. Cao, W. Dong, Z. Lv, Y. Gu, S. Singh, and P. Kumar, "Hybrid microgrid many-objective sizing optimization with fuzzy decision," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 11, pp. 2702–2710, Nov. 2020.

[14] R. Wang, W. Ma, M. Tan, G. Wu, L. Wang, D. Gong, and J. Xiong, "Preference-inspired coevolutionary algorithm with active diversity strategy for multi-objective multi-modal optimization," *Inf. Sci.*, vol. 546, pp. 1148–1165, Feb. 2021.

[15] 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, Mar. 2011.

[16] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Comput. Surv.*, vol. 48, no. 1, pp. 1–35, 2015.

[17] J. G. Falcón-Cardona and C. A. C. Coello, "Indicator-based multi-objective evolutionary algorithms: A comprehensive survey," *ACM Comput. Surv.*, vol. 53, no. 2, pp. 1–35, Jul. 2020.

[18] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, "A survey of multiobjective evolutionary algorithms based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 440–462, Jun. 2017.

[19] 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.

[20] H.-L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 450–455, Jun. 2014.

[21] H. Li, Q. Zhang, and J. Deng, "Biased multiobjective optimization and decomposition algorithm," *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 52–66, Jan. 2017.

[22] Y. Zhang, D.-W. Gong, J.-Y. Sun, and B.-Y. Qu, "A decomposition-based archiving approach for multi-objective evolutionary optimization," *Inf. Sci.*, vols. 430–431, pp. 397–413, Mar. 2018.

[23] H. Chen, G. Wu, W. Pedrycz, P. N. Suganthan, L. Xing, and X. Zhu, "An adaptive resource allocation strategy for objective space partition-based multiobjective optimization," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Mar. 12, 2019, doi: 10.1109/TSMC.2019.2898456.

[24] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, Oct. 2016.

[25] W. Wang, S. Yang, Q. Lin, Q. Zhang, K.-C. Wong, C. A. C. Coello, and J. Chen, "An effective ensemble framework for multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 645–659, Aug. 2019.

[26] K. C. Tan, S. C. Chiam, A. A. Mamun, and C. K. Goh, "Balancing exploration and exploitation with adaptive variation for evolutionary multi-objective optimization," *Eur. J. Oper. Res.*, vol. 197, no. 2, pp. 701–713, Sep. 2009.

[27] K. R. Opara and J. Arabas, "Differential evolution: A survey of theoretical analyses," *Swarm Evol. Comput.*, vol. 44, pp. 546–558, Feb. 2019.

[28] G. Wu, X. Shen, H. Li, H. Chen, A. Lin, and P. N. Suganthan, "Ensemble of differential evolution variants," *Inf. Sci.*, vol. 423, pp. 172–186, Jan. 2018.
[29] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, no. 2, pp. 115–148, 1995.
[30] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Comput. Sci. Inf.*, vol. 26, pp. 30–45, Aug. 1996.
[31] N. Hitomi and D. Selva, "A classification and comparison of credit assignment strategies in multiobjective adaptive operator selection," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 294–314, Apr. 2017.
[32] K. Li, A. Fialho, S. Kwong, and Q. Zhang, "Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 114–130, Feb. 2014.
[33] H. Zhao and C. Zhang, "An online-learning-based evolutionary many-objective algorithm," *Inf. Sci.*, vol. 509, pp. 1–21, Jan. 2020.
[34] A. Santiago, B. Dorronsoro, A. J. Nebro, J. J. Durillo, O. Castillo, and H. J. Fraire, "A novel multi-objective evolutionary algorithm with fuzzy logic based adaptive selection of operators: FAME," *Inf. Sci.*, vol. 471, pp. 233–251, Jan. 2019.
[35] Q. Lin, Z. Liu, Q. Yan, Z. Du, C. A. C. Coello, Z. Liang, W. Wang, and J. Chen, "Adaptive composite operator selection and parameter control for multiobjective evolutionary algorithm," *Inf. Sci.*, vol. 339, pp. 332–352, Apr. 2016.
[36] W. Lin, Q. Lin, J. Ji, Z. Zhu, C. A. C. Coello, and K.-C. Wong, "Decomposition-based multiobjective optimization with bicriteria assisted adaptive operator selection," *Swarm Evol. Comput.*, vol. 60, Feb. 2021, Art. no. 100790.
[37] J. Xiong, C. Zhang, G. Kou, R. Wang, H. Ishibuchi, and F. E. Alsaadi, "Optimizing long-term bank financial products portfolio problems with a multiobjective evolutionary approach," *Complexity*, vol. 2020, pp. 1–18, Apr. 2020.
[38] J. Xiong, R. Wang, and J. Jiang, "Weapon selection and planning problems using MOEA/D with distance-based divided neighborhoods," *Complexity*, vol. 2019, pp. 1–18, Nov. 2019.
[39] H.-L. Liu, L. Chen, Q. Zhang, and K. Deb, "Adaptively allocating search effort in challenging many-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 22, no. 3, pp. 433–448, Jun. 2018.
[40] H. Bai, J. Zheng, G. Yu, S. Yang, and J. Zou, "A Pareto-based many-objective evolutionary algorithm using space partitioning selection and angle-based truncation," *Inf. Sci.*, vol. 478, pp. 186–207, Apr. 2019.
[41] 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.
[42] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.
[43] A. Lipowski and D. Lipowska, "Roulette-wheel selection via stochastic acceptance," *Phys. A, Stat. Mech. Appl.*, vol. 391, no. 6, pp. 2193–2196, Mar. 2012.
[44] R. Cheng, Y. Jin, K. Narukawa, and B. Sendhoff, "A multiobjective evolutionary algorithm using Gaussian process-based inverse modeling," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 838–856, Dec. 2015.
[45] P. A. N. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 174–188, Apr. 2003.
[46] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.
[47] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
[48] Z. Lv and W. Xiu, "Interaction of edge-cloud computing based on SDN and NFV for next generation IoT," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5706–5712, Jul. 2020.
[49] B. Cao, X. Wang, W. Zhang, H. Song, and Z. Lv, "A many-objective optimization model of industrial Internet of Things based on private blockchain," *IEEE Netw.*, vol. 34, no. 5, pp. 78–83, Sep. 2020.
[50] Z. Lv and H. Song, "Mobile Internet of Things under data physical fusion technology," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4616–4624, May 2020.
[51] Z. Lv and L. Qiao, "Deep belief network and linear perceptron based cognitive computing for collaborative robots," *Appl. Soft Comput.*, vol. 92, Jul. 2020, Art. no. 106300.
[52] H. Chen, R. Cheng, J. Wen, H. Li, and J. Weng, "Solving large-scale many-objective optimization problems by covariance matrix adaptation evolution strategy with scalable small subpopulations," *Inf. Sci.*, vol. 509, pp. 457–469, Jan. 2020.
[53] C. He, R. Cheng, C. Zhang, Y. Tian, Q. Chen, and X. Yao, "Evolutionary large-scale multiobjective optimization for ratio error estimation of voltage transformers," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 868–881, Oct. 2020.
[54] B. Cao, S. Fan, J. Zhao, P. Yang, K. Muhammad, and M. Tanveer, "Quantum-enhanced multiobjective large-scale optimization via parallelism," *Swarm Evol. Comput.*, vol. 57, Sep. 2020, Art. no. 100697.
[55] B. Cao, J. Zhao, Y. Gu, Y. Ling, and X. Ma, "Applying graph-based differential grouping for multiobjective large-scale optimization," *Swarm Evol. Comput.*, vol. 53, Mar. 2020, Art. no. 100626.
[56] H. Zille, H. Ishibuchi, S. Mostaghim, and Y. Nojima, "A framework for large-scale multiobjective optimization based on problem transformation," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 260–275, Apr. 2018.

XUEFENG HONG graduated in software engineering from Hunan University. He is currently a Lecturer with the School of Internet Finance and Information Engineering, Guangdong University of Finance. He has presided over or participated in five national, provincial, and ministerial projects. He has published more than ten papers in top conferences and journals. His research interests include multi-objective optimization, data mining, education, and teaching management.

MINGFANG JIANG graduated in computer technology from the School of Information Science and Engineering, Hunan University. She is currently an Associate Research Librarian with Hunan First Normal University. She has presided over one national project and five provincial and ministerial projects. She has also participated in many projects of national and provincial projects as the main researcher. She has published more than 20 academic articles. Her research interests include digital resource management, information security, multimedia intelligence, and evolutionary optimization.

JINGLIN YU graduated in software engineering from East China Normal University. She is currently a Teacher with the Internet Finance and Information Engineering College, Guangdong University of Finance. She has presided over or participated in two provincial projects. She has published several academic articles. Her research interests include computer application and multimedia technology.