

Research Article

Decomposition-Based Multiobjective Evolutionary Optimization with Adaptive Multiple Gaussian Process Models

Xunfeng Wu, Shiwen Zhang, Zhe Gong, Junkai Ji , Qiuzhen Lin , and Jianyong Chen 

College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China

Correspondence should be addressed to Jianyong Chen; jychen@szu.edu.cn

Received 27 October 2019; Revised 20 December 2019; Accepted 2 January 2020; Published 11 February 2020

Academic Editor: Yong Xu

Copyright © 2020 Xunfeng Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In recent years, a number of recombination operators have been proposed for multiobjective evolutionary algorithms (MOEAs). One kind of recombination operators is designed based on the Gaussian process model. However, this approach only uses one standard Gaussian process model with fixed variance, which may not work well for solving various multiobjective optimization problems (MOPs). To alleviate this problem, this paper introduces a decomposition-based multiobjective evolutionary optimization with adaptive multiple Gaussian process models, aiming to provide a more effective heuristic search for various MOPs. For selecting a more suitable Gaussian process model, an adaptive selection strategy is designed by using the performance enhancements on a number of decomposed subproblems. In this way, our proposed algorithm owns more search patterns and is able to produce more diversified solutions. The performance of our algorithm is validated when solving some well-known F, UF, and WFG test instances, and the experiments confirm that our algorithm shows some superiorities over six competitive MOEAs.

1. Introduction

Multiobjective optimization problems (MOPs) widely exist in the fields of scientific research and engineering applications. Since the first multiobjective evolutionary algorithm (MOEA) was reported in 1985 [1], MOEAs have been studied sufficiently and become one of the hottest research directions in the field of evolutionary computation [2–4]. Internationally, MOEAs represented by NSGA-II [5], SPEA2 [6], PAES [7], HypE [8], MOEA/D [9], etc., have been widely used in many application fields. According to the selection strategy to handle the convergence enhancement and diversity maintenance, most of existing MOEAs can be roughly divided into the following three categories.

1.1. Pareto-Based MOEAs. The basic idea of such algorithms, represented by NSGA-II and SPEA2, is to use Pareto-based ranking scheme for sorting the population into different convergence layers and then calculate the density of individuals in the last layer. In this way, the population can be sorted according to the dominance relationship and density

estimation, and then the relatively superior individuals are selected to the next generation. Crowded distance [5], K -nearest neighbor method [6], ϵ -domination [10, 11], grading [12], and other methods [13–15] are often used to estimate the density of the individuals. As Pareto-based MOEAs have some advantages like simple principle, easy understanding, and fewer parameters, this kind of MOEAs has induced many research and extensive applications. However, their ability to guarantee convergence dramatically degrades when the number of objectives is larger than three, mainly due to the loss of selection pressure.

1.2. Decomposition-Based MOEAs. Decomposition-based MOEAs transform a MOP into a set of subproblems and then solve them simultaneously using a collaborative evolutionary search, such as MOEA/D [9], RVEA [16], NSGA-III [17], RPEA [18], SPEA/R [19], and RdEA [20]. Note that most of these algorithms adopt additional reference information (reference vectors, reference points, or weight vectors) during the environmental selection, which helps to maintain the diversity of the population. Due to the

advantage with a better mathematical explanation, decomposition-based MOEAs have become very popular in recent years [21–24].

1.3. Indicator-Based MOEAs. Indicator-based MOEAs directly employ a performance indicator, like hypervolume (HV), generational distance (GD), and R2, to effectively guide the selection of promising solutions for next generation. IBEA [25], MOMBI-II [26], HyPE [8], GD-MOEA [27], R2-IBEA [28], and DDE [29] are representatives for the indicator-based MOEAs. In these MOEAs, performance indicators are used as selection criterion to rank non-dominated solutions that cannot be distinguished by traditional Pareto dominance. However, for this kind of MOEAs, high computational complexity is usually required for calculating the performance indicator, which is very challenging especially when the number of objectives is large.

These MOEAs usually include two main components, i.e., variation and selection [6]. Selection plays an important role in MOEAs to maintain the promising solutions, as introduced above for classifying MOEAs, while variation is the key factor to determine the generation quality of solutions. In [30], the effects of several variation operators are studied on some test problems with variable linkages, showing that variable linkages may cause some difficulties for MOEAs. Actually, there are a number of real-valued variation approaches having been proposed during the recent decades [31], which can be classified into the three main kinds, i.e., traditional recombination operators, estimation of distribution algorithms, and the inverse model methods for variation.

Traditional recombination operators are generally used in most existing MOEAs due to their simplicity. This kind of variation simulates the binary crossover method to produce the real-valued offspring, such as simulated binary crossover (SBX) [32], Laplace crossover [33], parent central crossover [34], blend crossover α [35], unimodal normal distribution crossover [36], and simple crossover [37]. Moreover, differential evolution (DE) [38] is also used as the recombination operator in many MOEAs, which samples offspring based on the difference vectors of parents. The evolution paths are used in DE [39] to depict the population movement and predict its tendency, which could produce potential solutions to speed up convergence toward the PS. Recently, a number of hybridized recombination operators have been proposed, trying to combine the advantages of differential recombination operators. In [40], a cooperative DE framework is designed for constrained MOPs, in which multiple DE operators are run in different subpopulations to optimize its own constrained subproblem. In [41], four DE operators are combined and a sliding window is adopted in [42] to provide the reward for each DE operator according to the enhancement on subproblems. Similarly, four DE operator pools are presented in [43] including two DE operators with complementary search patterns in each pool to provide an improved search ability. In ACGDE [44], an adaptive cross-generation DE operator is designed by

exploiting information from individuals across generations to adapt the parameters.

The estimation of distribution algorithms (EDAs) [45] exploit the probabilistic models extracted from the population's distribution to run variation [46, 47]. Unlike the above traditional recombination operators, no crossover or mutation procedures will be run in EDAs. Instead, the globally statistical information of the selected parents is used in EDAs to build a posterior probability distribution model and then offspring are sampled from this built model. Several EDAs are studied for solving continuous MOPs in [48, 49], while the Gaussian distribution model, mixture Gaussian distribution model, or mixed Gaussian with the principal component analysis (PCA) model are introduced in [50] for offspring variation. Different from the individual's information used in traditional recombination operators, the local PCA operator is employed in [45, 47] to generate new offspring. Different from [47], the PCA model was replaced by the locally linear embedding (LLE) [51] model in [52]. In this way, this model only considers the decision space, without considering too much about the target MOP itself.

The inverse model methods for variation utilize machine learning approaches to capture the connection from the objective space to the decision space by exploiting the characteristics of the target MOP. The representative algorithms like IM-MOEA [53] and E-IM-MOEA [54] use the Gaussian process-based inverse model to complete crossover. In another work [55] as inspired by LLE manifold learning idea, a new LLE modeling approach is introduced to use the mapping function known in the MOP that the decision space is considered as the high-dimensional space and the objective space is regarded as a low-dimensional space. Thus, this new modeling method is no longer to build the overall low dimensional space of the sample, which is then reflected back to the high dimensional space, but it replaces directly by constructing new samples in high dimensional space. In this way, the model mapping from the objective space to the decision space is built based on the obtained approximated Pareto set during the evolution. In other words, this model can use the probability model or surrogate model to build a bridge from the objective space to the decision space.

In the above three kinds of variation operators, traditional recombination operators are often used in many MOEAs. However, as these operators are executed based on the individuals, they only provide a finite number of search patterns and are also criticized for lack of mathematic explanation. For the inverse model methods for variation, it is very challenging for mapping from the objective space into the decision space when tackling some complicated MOPs. EDAs usually adopt one standard Gaussian process model with fixed variance, which may not work well for various kinds of MOPs. To enhance the search capability of EDAs, this paper introduces a decomposition-based MOEA with adaptive multiple Gaussian process models, called MOEA/D-AMG, which is effective in generating more superior offspring. Based on the performance enhancements on a number of decomposed subproblems, one suitable Gaussian

process model will be selected accordingly. In this way, our method is more intelligent and can provide more search patterns to produce the diversified solutions. After evaluating our performance on some well-known F, UF, and WFG test instances, the experimental results have validated the superiorities of our algorithm over six competitive MOEAs (MOEA/D-SBX [9], MOEA/D-DE [56], MOEA/D-GM [57], RM-MEDA [47], IM-MOEA [53], and AMEDA [50]).

The rest of this paper is organized as follows. Section 2 presents some background information of MOPs, Gaussian process model, and some Gaussian process model-based MOEAs. Section 3 introduces the details of MOEA/D-AMG. At last, Section 4 provides and discusses the simulation results, while Section 5 gives the conclusions and some future work.

2. Preliminaries

2.1. Multiobjective Optimization Problems. In real life, there exist a number of engineering problems with multiple complicated optimization objectives, which are often called multiobjective optimizations problems (MOPs). This paper considers to solve continuous MOPs, as formulated below:

$$\begin{aligned} \min \quad & F(x) = (f_1(x), \dots, f_m(x))^T \\ \text{subject to} \quad & x \in \prod_{i=1}^n [a_i, b_i], \end{aligned} \quad (1)$$

where n and m are, respectively, the numbers of decision variables and objectives, a_i and b_i are, respectively, the lower and upper bounds of i_{th} dimensional variable of x in the decision space, $x = (x_1, \dots, x_n)^T \in R^n$ is a decision variable vector, $\prod_{i=1}^n [a_i, b_i] \subset R^n$ is the feasible search space, $f_i: R^n \rightarrow R, i = 1, \dots, m$, is a continuous mapping, and $F(x)$ consists of m continuous objective functions.

In MOPs, the conflicts often exist among different objectives, i.e., improvement of one objective results in deterioration of another. Generally, there does not exist an optimal solution that can minimize all the objectives in (1) at the same time. A set of trade-off solutions among the objectives can be found for solving MOPs, which are equally optimal when considering all the objectives. Suppose that there are two vectors $p = (p_1, \dots, p_m)^T$ and $q = (q_1, \dots, q_m)^T \in R^m$, where m is the number of objectives. p is said to dominate q in equation (1), denoted by $p < q$, if $p_i \leq q_i$ for all $i = 1, \dots, m$, and $p \neq q$. A solution $x^* \in \Omega$, where Ω is the feasible search space in equation (1), is called Pareto optimal solution if and only if there is no another $x \in \Omega$ such that $F(x) < F(x^*)$. The collection of all the Pareto optimal solutions is called Pareto optimal set (PS), and the projection of PS in the objective space is called Pareto optimal front (PF).

2.2. Gaussian Process Model. The recombination operator based on the Gaussian process model belongs to EDA. Different from the classical recombination operators, the recombination using Gaussian process model uses the

distribution information from the whole population to generate offspring, which can ensure the diversity of search patterns.

The Gaussian model is one of the most widely used probability models in scientific research and practical applications [58–61]. In general, a random variable $x = (x_1, x_2, \dots, x_n)^T$ with a Gaussian distribution can be expressed as

$$x \sim N(\mu, \Sigma), \quad (2)$$

where μ is an n dimensional mean vector and Σ is the covariance matrix. The probability density function of the random variable is expressed as

$$f(x) = (2\pi)^{-(n/2)} |\Sigma| \exp\left\{-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\right\}. \quad (3)$$

For a given set of data x^1, x^2, \dots, x^K , the mean vector and covariance matrix are, respectively, obtained by

$$\begin{aligned} \mu &= \frac{1}{K} \sum_{k=1}^K x^k, \\ \Sigma &= \frac{1}{K-1} \sum_{k=1}^K (x^k - \mu)(x^k - \mu)^T. \end{aligned} \quad (4)$$

Hence, a new solution $x = x_1, x_2, \dots, x_n$ can be generated using the Gaussian model, which can be divided into three steps of Algorithm 1. At first, decompose the covariance matrix Σ into a lower triangular matrix A by using the Cholesky decomposition method [62] in line 1, where $\Sigma = AA^T$. Then, generate a vector $y = (y_1, \dots, y_n)^T$ in line 2, of which the element $y_i, y = 1, \dots, n$, is sampled from a standard Gaussian distribution $N(0, 1)$. After that, a new trial solution x can be yielded by $x = \mu + Ay$ in line 3. Generally, to improve the search capability of the Gaussian model, a small variation is added to mutate the new solution y by the polynomial mutation operator [2].

2.3. Gaussian Process Model-Based MOEAs. In recent years, several Gaussian process model-based MOEAs have been proposed. Their details are, respectively, introduced in the following paragraphs. At last, the motivation of this paper is clarified.

In [63], a mixture-based multiobjective iterated density estimation evolutionary algorithm (MIDEA) with both discrete and continuous representations is proposed. This approach employs clustering analysis to discover the nonlinear distribution structure, which validates that a simple model is feasible to describe a cluster, and the population can be adequately represented by a mixture of simple models. After that, a mixture of Gaussian models is proposed to produce new solutions for continuous real-valued MOPs in MIEDA.

Although MIEDA is very effective for solving certain MOPs, the regularity property of MOPs is not considered so that it may perform poorly in some problems. Thus, a multiobjective evolutionary algorithm based on decomposition and probability model (MOEA/D-MG) is designed in [57]. In this approach, multivariate Gaussian models are

- (1) Obtain a lower triangular matrix A through decomposing the covariance matrix using Cholesky, and $\Sigma = AA^T$
- (2) Generate a single Gaussian distribution variable $y = (y_1, y_2, \dots, y_n)^T$, $y \sim N(0, 1)$, $j = 1, 2, \dots, n$
- (3) $x = \mu + Ay$.

ALGORITHM 1: Sampling model.

embedded into MOEA/D [9] for continuous multiobjective optimization. Either a local Gaussian distribution model is built around a subproblem based on the neighborhood or a global model is constructed based on the whole population. Thus, the population distribution can be captured by all the probability models working together. However, MOEA/D-MG has to reconstruct a Gaussian model for each subproblem, resulting in a large computational cost in the process of building the model. Moreover, when building the Gaussian model for the similar subproblems, some individuals may be repeatedly used, which aggravates the computational resources.

To reduce the computational cost for the modeling process, an improved MOEA/D-MG with high modeling efficiency (MOEA/D-MG2) is reported in [64], where the neighboring subproblems can share the same covariance matrix to build Gaussian model for sampling solutions. At first, a process of sorting population is used to adjust the sampling orders of the subproblems, which tries to avoid the insufficient diversity caused by the reuse of the covariance matrix and can obtain the uniformly distributed offspring set. Then, in global search, only some subproblems are selected randomly to construct the Gaussian model. Although MOEA/D-MG2 performs well for solving some MOPs, it just builds the Gaussian model for solution sampling in the neighborhoods as defined in MOEA/D, which is only used in the MOEA/D paradigm.

Moreover, an adaptive multiobjective estimation of distribution algorithm with a novel Gaussian sampling strategy (AMEDA) is presented in [50]. In this work, a clustering analysis approach is adopted to reveal the structure of the population distribution. Based on these clusters, a local multivariate Gaussian model or a global model is built for each solution to sample a new solution, which can enhance the accuracy of modeling and the searching ability of AMEDA. Moreover, an adaptive update strategy of the probability is developed to control the contributions for two types of Gaussian model.

From the above studies, it can be observed that their modeling differences mainly focus on the methods of selecting sampling solutions. However, the above Gaussian models in [50, 63, 64] only adopt the standard one with fixed variance, which cannot adaptively adjust the search step sizes according to the different characteristics of MOPs. To alleviate the above problem, a decomposition-based MOEA with adaptive multiple Gaussian process models (called MOEA/D-AMG) is proposed in this paper. Multiple Gaussian models are used in our algorithm by using a set of different variances. Then, based on the performance enhancements of decomposed subproblems, a suitable Gaussian model will be

adaptively selected, which helps to enhance the search capability of MOEA/D-AMG and can well handle various kinds of MOPs as validated in the experimental section.

3. The Proposed Algorithm

In this section, our proposed algorithm MOEA/D-AMG is introduced in detail. At first, the adaptive multiple Gaussian process models are described. Then, the details of MOEA/D-AMG are demonstrated.

3.1. Adaptive Multiple Gaussian Process Models. Many approaches have been designed by taking advantages of the regularity in distributions of Pareto optimal solutions in both the decision and objective spaces to estimate the population distribution. Considering the mild conditions, it can be induced from the Karush–Kuhn–Tucker (KKT) condition that the PF is $(m-1)$ -dimensional piecewise continuous manifolds [65] for an m -objective optimization problem. That is to say, the PF of a continuous biobjective MOP is a piecewise continuous curve, while the PF of a continuous three-objective MOP is a piecewise continuous surface. Thus, the Gaussian process model has been widely studied for both single-objective and multiobjective optimization [66–72]. However, a single Gaussian process model is not so effective for modeling the population distribution when tackling some complicated MOEAs as studied in [57]. In consequence, multiple Gaussian models are used in this paper, which can explicitly exploit the ability of different Gaussian models with various distributions. By using multiple Gaussian process models with good diversity, a more suitable one should be adaptively selected to capture the population structure for sampling new individuals more accurately. Thus, five types of Gaussian models are used in this paper, which have the same mean value 0 with different standard deviations, i.e., 0.6, 0.8, 1.0, 1.2, and 1.4. The distributions of five Gaussian models (as, respectively, represented by g_1 , g_2 , g_3 , g_4 , and g_5) are plotted in Figure 1.

To select a suitable one from multiple Gaussian process models, an adaptive strategy is proposed in this paper to improve the comprehensive search capability. The probabilities of selecting these different models are dependent on their performance to optimize the subproblems. Once a Gaussian process model is selected, it will be used to generate the Gaussian distribution variable y in line 3 of Algorithm 1. Before the adaptive strategy comes into play, these predefined Gaussian models are selected with an equal probability. In our approach, a set of offspring solutions will be produced by using different Gaussian distribution variables y . After analyzing the quality of these new offspring solutions, the contribution rate of each Gaussian process model can be calculated. It should be noted that the quality of the new offspring is determined by their fitness value, which can be calculated by many available methods. In our work, the Tchebycheff approach in [9] is adopted, as follows:

$$g^i(x) = g(x | \lambda_i, z) = \max_{1 \leq j \leq m} \lambda_j^i |f_j(x) - z_j|, \quad (5)$$

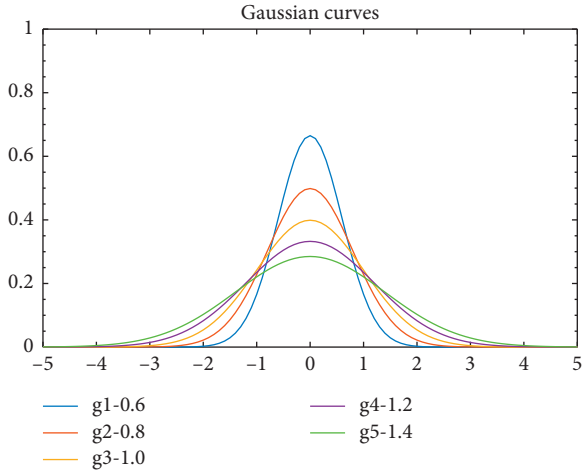


FIGURE 1: The distributions of five Gaussian models.

where z is a reference point, λ^i is the i th weight vector, and m is the number of objectives. A smaller fitness value of new offspring which is generated by Gaussian distribution variable indicates a greater contribution for the corresponding Gaussian model. Hence, for each Gaussian model, the improvement of fitness (IoF) value can be obtained as follows:

$$\text{IoF}_{k,G} = \begin{cases} \text{Fe}_{k,G-LP} - \text{Fe}_{k,G}, & \text{Fe}_{k,G} < \text{Fe}_{k,G-LP} \\ 0, & \text{otherwise,} \end{cases} \quad (6)$$

where $\text{Fe}_{k,G}$ is the fitness of new offspring with the k th Gaussian distribution in generation G . In order to maximize the effectiveness of the proposed adaptive strategy, this strategy is executed by each of LP generations in the whole process of our algorithm. Afterwards, the contribution rates (Cr) of different Gaussian distributions can be calculated as follows:

$$\text{Cr}_{k,G} = \frac{\text{IoF}_{k,G}}{\sum_{i=1}^K \text{IoF}_{i,G}} + \varepsilon, \quad (7)$$

where ε is a very small value, which works when IoF is zero. Then, the probabilities (Pr) of different distributions being selected are updated by the following formula:

$$\text{Pr}_{k,G} = \frac{\text{Cr}_{k,G}}{\sum_{i=1}^K \text{Cr}_{i,G}}, \quad (8)$$

where K is the total number of Gaussian models. As described above, we can adaptively adjust the probability that the k th Gaussian distribution is selected by updating the value of $\text{Pr}_{k,G}$.

3.2. The Details of MOEA/D-AMG. The above adaptive multiple Gaussian process models are just used as the recombination operator, which can be embedded into a state-of-the-art MOEA based on decomposition (MOEA/D [9]), giving the proposed algorithm MOEA/D-AMG. In this section, the details of the proposed MOEA/D-AMG are introduced.

To clarify the running of MOEA/D-AMG, its pseudo-code is given in Algorithm 2 and some used parameters are introduced below:

- (1) N indicates the number of subproblems and also the size of population.
- (2) K is the number of Gaussian process models.
- (3) G is the current generation.
- (4) ξ is the parameter which is applied to control the balance between exploitation and exploration.
- (5) gss is the number of randomly selected subproblems for constructing global Gaussian model.
- (6) LP is the parameter to control the frequency of using the proposed adaptive strategy.
- (7) n_r is the maximal number of old solutions which are allowed to be replaced by a new one.
- (8) $perm(\cdot)$ randomly shuffles the input values, and $rand()$ produces a random real number in $[0, 1]$.

Lines 1-2 of Algorithm 2 describe the initialization process by setting the population size N and generating N weight vectors using the approach in [56] to define N subproblems in (5). Then, an initial population is randomly generated to include N individuals and the neighborhood size is set to select T neighboring subproblems for constructing the Gaussian model. Then, the reference point $z = \{z_1, z_2, \dots, z_m\}$ can be obtained by including all the minimal value of each objective. In line 3, multiple Gaussian distributions are defined, and the corresponding initial probabilities are set to an equal value. Then, a maximum number of generations is used as the termination condition in line 4 and all the subproblems are randomly selected once at each generation in line 5. After that, the main evolutionary loop of the proposed algorithm is run in lines 5–26. For the subproblem selected at each generation, it will go through three components, including model construction, model update, and population update as observed from Algorithm 2.

Lines 6–10 run the process of model construction. As controlled by the parameter ξ , a set of subproblems B is either the neighborhood of current subproblem B^i in line 7 or gss subproblems selected randomly in line 9. If the neighborhood of current subproblem is used, the local multiple Gaussian models are constructed for running exploitation; otherwise, the global models are constructed for running exploration.

Lines 11–16 implement the process of model update. The adaptive strategy described in Section 3.1 is used to update the probability of five different Gaussian models at each of LP generations, as shown in line 12. Then, a Gaussian model can be selected in line 14 and the selected model is used to update y in line 15. At last, Algorithm 1 is performed to generate a new offspring in line 16 with the selected subproblems B for model construction and the generated Gaussian variable y .

Lines 17–23 give the process of population update. For each individual of B in line 18, if the aggregated function value using equation (5) for subproblem is improved by the

```

(1) Initialize  $N$  subproblems including weight vector  $w$ ,  $N$  individuals, and neighborhood size  $T$ 
(2) Initialize the reference point  $z = \{z_1, z_2, \dots, z_m\}$ ,  $j = 1, 2, \dots, m$ 
(3) Predefine  $y \sim \{N(0, \sigma), \dots, N(0, \sigma)\}$ ,  $Pr_{k,0} = 1/K$ 
(4) while not terminate do
(5)   for  $i \in \text{perm}(\{x_1, x_2, \dots, x_N\})$  do
      Model construction
(6)     if  $\text{rand}() < \xi$  then
(7)       Define neighborhood individuals  $B = B^i$ 
(8)     else
(9)       Select  $gss$  individuals from  $\{x_1, x_2, \dots, x_N\}$  to construct  $B$ 
(10)    end
      Model updation
(11)   if  $\text{mod}(G, LP) == 0$  then
(12)     Update  $Pr_{k,G}$  using equations (5)–(8)
(13)   end
(14)   Select a Gaussian model according to  $Pr_{k,G}$ 
(15)   Generate  $y$  based on the selected Gaussian model
(16)   Add  $B$  and  $y$  into Algorithm 1 to generate a new offspring  $x$ 
      Population updating
(17)   Set counter  $c = 0$ 
(18)   for each  $j \in B$  do
(19)     if  $g_i(x) < g^i(x^j)$  and  $c < n_r$  then
(20)        $x^j$  is replaced by  $x$ 
(21)       Set  $c = c + 1$ 
(22)     end
(23)   end
(24) end
(25)  $G = G + 1$ 
(26) Update the reference point  $z$ 
(27) end
(28) Output the final population

```

ALGORITHM 2: MOEA/D-AMG.

new offspring x in line 19, the original individual will be replaced by x in line 20. The c value is increased by 1 in line 21 to prevent premature convergence, such that at most c individuals in B can be replaced.

At last, the generation counter G is added by 1 and the reference point z is updated by including the new minimal value for each objective. If the termination in line 4 is not satisfied, the above evolutionary loop in lines 5–26 will be run again; otherwise, the final population will be outputted in line 28.

4. Experimental Studies

4.1. Test Instances. Twenty-eight unconstrained MOP test instances are employed here as the benchmark problems for empirical studies. To be specific, UF1–UF10 are used as the benchmark problems in CEC2009 MOEA competition and F1–F9 are proposed in [48]. These test instances have complicated PS shapes. We also consider WFG test suite [49] with different problem characteristics, including no separable, deceptive, degenerate problems, mixed PF shape, and variable dependencies. The number of decision variables is set to 30 for UF1–UF10 and F1–F9; for WFG1–WFG9, the numbers of position and distance-related decision variable are, respectively, set to 2 and 4, while the number of objectives is set to 2.

4.2. Performance Metrics

4.2.1. The Inverted Generational Distance (IGD) [73]. Here, assume that P^* is a set of solutions uniformly sampled from the true PF and S represents a solution set obtained by a MOEA [73]. The IGD value from P^* to S will calculate the average distance from each point of P^* to the nearest solution of S in the objective space, as follows:

$$\text{IGD}(S, P^*) = \frac{\sum_{x \in P^*} \text{dist}(x, S)}{|P^*|}, \quad (9)$$

where $\text{dist}(x, S)$ returns the minimal distance from one solution x in P^* to one solution in S and $|P^*|$ returns the size of P^* .

4.2.2. Hypervolume (HV) [74]. Here, assume that a reference point $Z^r = (z_1^r, z_2^r, \dots, z_m^r)$ in the objective space is dominated by all Pareto-optimal objective vectors [74]. Then, the HV metric will measure the size of the objective space dominated by the solutions in S and bounded by Z^r , as follows:

$$\text{HV}(S) = \text{VOL}(\cup_{x \in S} [f_1(x), z_1^r] \times \dots \times [f_m(x), z_m^r]), \quad (10)$$

where $VOL(\cdot)$ indicates the Lebesgue measure. In our empirical studies, $Z^r = (1.1, 1.1)^T$ and $Z^r = (1.1, 1.1, 1.1)^T$ are, respectively, set for the biobjective and three-objective cases in UF and F test problems. Due to the different scaled values in different objectives, $Z^r = (2.0, 4.0)^T$ is set for WFG test suite.

Both IGD and HV metrics can reflect the convergence and diversity for the solution set S simultaneously. The lower IGD value (or the larger HV value) indicates the better quality of S for approximating the entire true PF. In this paper, six competitive MOEAs, including MOEA/D-SBX [9], MOEA/D-DE [56], MOEA/D-GM [57], RM-MEDA [47], IM-MOEA [53], and AMEDA [50], are included to validate the performance of our proposed MOEA/D-AMG. All the comparison results obtained by these MOEAs regarding IGD and HV are presented in the corresponding tables, where the best mean metric values are highlighted in bold and italics. In order to have statistically sound conclusions, Wilcoxon's rank sum test at a 5% significance level is conducted to compare the significance of difference between MOEA/D-AMG and each compared algorithm.

4.3. The Parameter Settings. The parameters of MOEA/D-SBX, MOEA/D-DE, MOEA/D-GM, RM-MEDA, IM-MOEA, and AMEDA are, respectively, set according to references [9, 47, 53, 56, 57, 50]. All these MOEAs are implemented in MATLAB. The detailed parameter settings of all the algorithms are summarized as follows.

4.3.1. Public Parameters. For population size N , we set $N=300$ for biobjective problems and $N=600$ for three-objective problems. For number of runs and termination condition, each algorithm is independently launched by 20 times on each test instance, and the termination condition of an algorithm is the predefined maximal number of function evaluations, which is set to 300000 for UF instances, 150000 for F instances, and 200000 for WFG instances. We set the neighborhood size $T=20$ and the mutation probability $p_m = 1/n$, where n is the number of decision variables for each test problem and its distribution index is $\mu_m = 20$.

4.3.2. Parameters in MOEA/D-SBX. The crossover probability and distribution index are, respectively, set to 0.9 and 30.

4.3.3. Parameters in MOEA/D-DE. The crossover rate $CR=1$ and the scaling factor $F=0.5$ in DE as recommended in [9], the maximal number of solution replacement $n_r=2$, and the probability of selecting the neighboring subproblems $\delta=0.9$.

4.3.4. Parameters in MOEA/D-GM. The neighborhood size for each subproblem $K=15$ for biobjective problems and $K=30$ for triobjective problems (this neighborhood size K is crucial for generating offspring and updating parents in evolutionary process), the parameter to balance the

exploitation and exploration $p_n=0.8$, and the maximal number of old solutions which are allowed to be replaced by a new one $C=2$.

4.3.5. Parameters in RM-MEDA. The number of clusters K is set to 5 (this value indicates the number of disjoint clusters obtained by using local principal component analysis on population) and the maximum number of iterations in local PCA algorithm is set to 50.

4.3.6. Parameters in IM-MOEA. The number of reference vectors K is set to 10 and the model group size L is set to 3.

4.3.7. Parameters in AMEDA. The initial control probability $\beta_0=0.9$, the history length $H=10$, and the maximum number of clusters $K=5$ (this value indicates the maximum number of local clusters obtained by using hierarchical clustering analysis approach on population K).

4.3.8. Parameters in MOEA/D-AMG. The number of Gaussian process models $K=5$, the initial Gaussian distribution is $(0, 0.6^2)$, $(0, 0.8^2)$, $(0, 1.0^2)$, $(0, 1.2^2)$, and $(0, 1.4^2)$, $LP=10$, $n_r=2$, $\xi=0.9$, and $gss=40$.

4.4. Comparison Results and Discussion. In this section, the proposed MOEA/D-AMG is compared with six competitive MOEAs, including MOEA/D-SBX, MOEA/D-DE, MOEA/D-GM, RM-MEDA, IM-MOEA, and AMEDA. The results obtained by these algorithms on the 28 test problems are given in Tables 1 and 2, regarding the IGD and HV metrics, respectively. When compared to other algorithms, the performance of MOEA/D-AMG has a significant improvement when the multiple Gaussian process models are adaptively used. It is observed that the proposed algorithm can improve the performance of MOEA/D in most of the test problems. Table 1 summarizes the statistical results in terms of IGD values obtained by these compared algorithms, where the best result of each test instance is highlighted. The Wilcoxon rank sum test is also adopted at a significance level of 0.05, where symbols "+," "-", and "~" indicate that results obtained by other algorithms are significantly better, significantly worse, and no difference to that obtained by our algorithm MOEA/D-AMG. To be specific, MOEA/D-AMG shows the best results on 12 out of the 28 test instances, while the other compared algorithms achieve the best results on 3, 3, 5, 2, 2, and 1 out of the 28 problems in Table 1. When compared with MOEA/D-SBX, MOEA/D-DE, MOEA/D-GM, RM-MEDA, IM-MOEA, and AMEDA, MOEA/D-AMG yields 23, 21, 18, 22, 24, and 25 significantly better mean IGD metric values, respectively. When compared to RM-MEDA, IM-MOEA, and AMEDA, MOEA/D-AMG has shown an absolute advantage, as it outperforms them on 22, 24, and 25 test problems regarding IGD values, respectively, whereas RM-MEDA outperforms MOEA/D-AMG on F8, UF3, WFG2, and WFG6, IM-MOEA outperforms MOEA/D-AMG on F1, UF10, WFG4, and WFG6, and AMEDA

TABLE 1: Performance comparison of MOEA/D-SBX, MOEA/D-DE, MOEA/D-GM, RM-MEDA, IM-MOEA, AMEDA, and MOEA/D-AMG in terms of the average IGD values and standard deviation on F, UF, and WFG problems.

Problems	MOEA/D-SBX	MOEA/D-DE	MOEA/D-GM	RM-MEDA	IM-MOEA	AMEDA	MOEA/D-AMG
F1	3.43E-03 (8.00E-04)-	3.66E-03 (1.94E-04)-	1.36E-03 (1.36E-05)~	1.57E-03 (2.95E-05)-	1.27E-03 (2.16E-05)+	2.41E-02 (2.52E-02)-	1.35E-03 (1.28E-05)
F2	1.05E-01 (5.13E-02)-	1.26E-02 (1.34E-03)-	3.47E-03 (1.84E-04)-	4.03E-02 (4.90E-03)-	6.86E-02 (7.76E-03)-	5.21E-02 (1.47E-02)-	2.85E-03 (2.95E-04)
F3	4.93E-02 (3.03E-02)-	1.32E-02 (2.59E-03)-	3.07E-03 (2.21E-04)-	1.37E-02 (2.30E-03)-	2.83E-02 (5.86E-03)-	1.38E-02 (4.10E-03)-	2.55E-03 (3.59E-04)
F4	6.07E-02 (4.18E-02)-	1.30E-02 (1.16E-03)-	3.16E-03 (5.22E-04)-	2.11E-02 (4.25E-03)-	2.09E-02 (5.43E-03)-	1.00E-02 (3.71E-03)-	2.93E-03 (7.23E-04)
F5	4.21E-02 (2.45E-02)-	1.42E-02 (1.24E-03)-	8.47E-03 (8.86E-04)-	1.43E-02 (1.91E-03)-	2.30E-02 (5.78E-03)-	1.57E-02 (3.71E-03)-	7.94E-03 (1.27E-03)
F6	7.30E-02 (1.75E-02)-	8.45E-02 (2.90E-03)-	6.26E-02 (7.65E-03)+	2.18E-01 (6.89E-02)-	2.20E-01 (3.00E-02)-	8.69E-02 (5.60E-03)-	6.54E-02 (8.13E-03)
F7	2.79E-01 (2.08E-02)-	2.02E-01 (4.70E-02)-	1.42E-02 (3.11E-02)+	1.38E+00 (4.11E-01)-	3.01E-01 (4.48E-02)-	3.11E-01 (4.50E-02)-	4.21E-02 (7.84E-02)
F8	2.10E-01 (5.24E-02)-	8.31E-02 (1.82E-02)-	8.32E-03 (4.24E-03)+	1.20E-02 (5.14E-03)+	7.74E-02 (8.89E-03)-	6.47E-02 (2.17E-02)-	1.55E-02 (1.07E-02)
F9	1.11E-01 (3.84E-02)-	1.53E-02 (2.30E-03)-	4.87E-03 (1.17E-03)-	3.95E-02 (4.54E-03)-	6.22E-02 (1.23E-02)-	5.49E-02 (1.57E-02)-	5.14E-03 (2.12E-03)
UF1	1.05E-01 (2.71E-02)-	1.27E-02 (9.03E-04)-	2.88E-03 (5.02E-04)-	3.89E-02 (2.91E-03)-	6.51E-02 (1.16E-02)-	4.68E-02 (1.05E-02)-	2.45E-03 (1.42E-03)
UF2	3.95E-02 (2.14E-02)-	1.40E-02 (1.17E-03)-	7.08E-03 (1.05E-03)-	1.30E-02 (1.40E-03)-	2.55E-02 (7.06E-03)-	1.67E-02 (5.66E-03)-	6.68E-03 (8.37E-04)
UF3	2.16E-01 (4.31E-02)-	8.60E-02 (1.54E-02)-	5.41E-03 (3.30E-03)+	1.53E-02 (6.36E-03)+	8.47E-02 (1.21E-02)-	6.62E-02 (1.74E-02)-	1.74E-02 (1.99E-02)
UF4	5.19E-02 (3.61E-03)+	4.86E-02 (2.91E-03)+	5.39E-02 (2.94E-03)+	9.46E-02 (9.84E-03)-	7.68E-02 (8.00E-03)-	9.30E-02 (1.05E-02)-	6.98E-02 (5.78E-03)
UF5	4.41E-01 (8.29E-02)-	2.78E-01 (7.49E-02)+	3.14E-01 (5.19E-02)+	1.04E+00 (5.06E-01)-	9.41E-01 (1.63E-01)-	7.26E-01 (1.34E-01)-	4.08E-01 (9.50E-02)
UF6	3.29E-01 (1.32E-01)-	1.56E-01 (4.89E-02)~	2.91E-01 (2.66E-01)-	4.26E-01 (3.72E-02)-	2.28E-01 (3.57E-02)-	1.84E-01 (1.22E-01)-	1.62E-01 (2.06E-01)
UF7	2.53E-01 (2.27E-01)-	1.12E-02 (2.21E-03)-	4.12E-03 (3.71E-04)+	1.77E-02 (2.62E-03)-	3.47E-02 (3.37E-02)-	4.89E-02 (1.05E-01)-	5.13E-03 (6.70E-04)
UF8	7.02E-02 (6.44E-03)-	8.80E-02 (3.73E-03)-	6.50E-02 (4.84E-03)-	1.56E-01 (3.56E-02)-	1.65E-01 (3.58E-03)-	8.85E-02 (4.38E-03)-	5.74E-02 (8.29E-03)
UF9	1.13E-01 (5.02E-02)-	8.34E-02 (4.49E-03)-	3.32E-02 (3.92E-03)-	1.89E-01 (5.92E-02)-	1.56E-01 (5.27E-02)-	7.62E-02 (1.06E-02)-	2.76E-02 (4.37E-03)
UF10	2.79E-01 (8.23E-02)+	4.69E-01 (7.35E-02)+	1.79E+00 (2.41E-01)-	2.49E+00 (1.16E-01)-	1.77E-01 (4.32E-03)+	1.80E+00 (3.11E-01)-	8.03E-01 (1.71E-01)
WFG1	7.71E-01 (5.32E-02)+	1.06E+00 (7.62E-03)+	1.16E+00 (9.55E-03)-	1.17E+00 (9.36E-03)-	1.44E+00 (9.80E-02)-	1.83E+00 (1.86E-01)-	1.11E+00 (1.17E-02)
WFG2	5.79E-02 (2.12E-02)-	1.64E-02 (5.10E-04)-	1.57E-02 (3.63E-04)~	6.27E-03 (1.04E-03)+	1.71E-02 (1.78E-03)-	1.26E-02 (1.75E-02)+	1.55E-02 (3.24E-04)
WFG3	6.68E-03 (5.70E-04)-	7.45E-03 (5.04E-04)-	5.05E-03 (1.71E-04)-	8.29E-03 (2.31E-04)-	1.22E-02 (1.26E-03)-	8.55E-03 (2.38E-04)-	4.45E-03 (1.42E-04)
WFG4	5.71E-03 (2.45E-04)+	7.84E-03 (8.40E-04)+	5.91E-02 (2.39E-03)-	4.54E-02 (7.98E-04)~	3.89E-02 (4.07E-03)+	2.51E-02 (5.94E-03)+	4.81E-02 (1.64E-03)
WFG5	6.34E-02 (2.94E-03)+	6.46E-02 (2.00E-03)+	6.53E-02 (4.09E-05)~	6.92E-02 (1.44E-04)-	6.94E-02 (2.95E-03)-	6.81E-02 (3.23E-04)-	6.52E-02 (4.14E-04)
WFG6	6.73E-02 (1.29E-02)-	2.07E-01 (6.98E-02)-	5.53E-02 (9.88E-02)-	7.71E-03 (2.35E-04)+	1.04E-02 (1.35E-03)+	6.73E-03 (3.48E-04)+	1.62E-02 (5.02E-02)
WFG7	6.10E-03 (1.55E-04)-	6.65E-03 (1.94E-04)-	5.79E-03 (5.83E-05)-	7.32E-03 (3.43E-04)-	1.11E-02 (8.98E-04)-	7.25E-03 (2.37E-04)-	5.53E-03 (2.13E-05)
WFG8	9.08E-02 (6.66E-03)-	6.60E-02 (4.96E-03)-	5.94E-02 (9.81E-04)-	6.65E-02 (1.36E-03)-	8.86E-02 (3.06E-03)-	6.49E-02 (2.04E-03)-	5.77E-02 (1.60E-03)
WFG9	1.72E-02 (1.99E-02)-	5.45E-02 (9.49E-02)-	1.73E-02 (2.40E-04)-	1.51E-02 (1.83E-04)~	1.83E-02 (2.60E-03)-	1.61E-02 (8.66E-04)-	1.52E-02 (2.92E-04)
Total	5/0/23	6/1/21	7/3/18	4/2/22	4/0/24	3/0/25	

The best average values obtained by these algorithms for each instance are given in bold and italics.

TABLE 2: Performance comparison of MOEA/D-SBX, MOEA/D-DE, MOEA/D-GM, RM-MEDA, IM-MOEA, AMEDA, and MOEA/D-AMG in terms of the average HV values and standard deviation on F, UF, and WFG problems.

Problems	MOEA/D-SBX	MOEA/D-DE	MOEA/D-GM	RM-MEDA	IM-MOEA	AMEDA	MOEA/D-AMG
F1	8.70E-01 (1.54E-03)-	8.71E-01 (2.70E-04)-	8.74E-01 (3.36E-05)-	8.74E-01 (7.14E-05)-	8.75E-01 (4.73E-05)~	8.71E-01 (2.26E-04)-	8.75E-01 (3.88E-05)
F2	7.20E-01 (5.49E-02)-	8.54E-01 (2.49E-03)-	8.66E-01 (5.73E-04)-	7.98E-01 (8.48E-03)-	7.60E-01 (1.87E-02)-	7.88E-01 (2.12E-02)-	8.70E-01 (7.14E-04)
F3	8.15E-01 (3.06E-02)-	8.55E-01 (3.68E-03)-	8.70E-01 (4.58E-04)~	8.55E-01 (2.22E-03)-	8.39E-01 (5.94E-03)-	8.53E-01 (5.18E-03)-	8.72E-01 (6.95E-04)
F4	8.11E-01 (3.32E-02)-	8.55E-01 (2.17E-03)-	8.65E-01 (1.28E-03)-	8.43E-01 (5.58E-03)-	8.48E-01 (7.24E-03)-	8.60E-01 (3.73E-03)-	8.71E-01 (1.91E-03)
F5	8.29E-01 (2.00E-02)-	8.53E-01 (2.22E-03)-	8.60E-01 (2.36E-03)-	8.54E-01 (2.85E-03)-	8.44E-01 (7.85E-03)-	8.51E-01 (3.32E-03)-	8.63E-01 (1.93E-03)
F6	6.39E-01 (2.89E-02)-	5.93E-01 (1.23E-02)-	6.60E-01 (1.19E-02)+	4.37E-01 (6.99E-02)-	4.80E-01 (3.16E-02)-	6.07E-01 (1.37E-02)-	6.56E-01 (4.48E-03)
F7	4.66E-01 (3.62E-02)-	5.91E-01 (6.00E-02)-	8.56E-01 (4.41E-02)+	1.99E-02 (4.58E-02)-	3.73E-01 (5.14E-02)-	3.60E-01 (5.04E-02)-	8.19E-01 (1.04E-01)
F8	5.00E-01 (5.69E-02)-	7.31E-01 (3.17E-02)-	8.63E-01 (6.60E-03)+	8.39E-01 (1.30E-02)-	7.48E-01 (1.28E-02)-	7.47E-01 (3.07E-02)-	8.51E-01 (1.69E-02)
F9	3.78E-01 (5.67E-02)-	5.14E-01 (4.36E-03)-	5.31E-01 (1.62E-03)-	4.69E-01 (4.37E-03)-	4.31E-01 (1.93E-02)-	4.55E-01 (2.16E-02)-	5.35E-01 (2.86E-03)
UF1	7.09E-01 (3.82E-02)-	8.54E-01 (1.80E-03)-	8.61E-01 (7.80E-04)-	8.01E-01 (4.86E-03)-	7.64E-01 (2.13E-02)-	7.94E-01 (2.29E-02)-	8.69E-01 (2.66E-03)
UF2	8.29E-01 (1.91E-02)-	8.54E-01 (1.61E-03)-	8.65E-01 (2.05E-03)-	8.56E-01 (1.65E-03)-	8.42E-01 (6.25E-03)-	8.52E-01 (4.95E-03)-	8.70E-01 (1.50E-03)
UF3	5.12E-01 (4.39E-02)-	7.25E-01 (2.26E-02)-	8.67E-01 (5.32E-03)+	8.29E-01 (1.75E-02)-	7.34E-01 (1.71E-02)-	7.42E-01 (2.49E-02)-	8.50E-01 (2.77E-02)
UF4	4.45E-01 (4.52E-03)+	4.60E-01 (5.13E-03)+	4.49E-01 (4.95E-03)+	3.75E-01 (1.58E-02)-	4.06E-01 (1.12E-02)-	3.79E-01 (1.25E-02)-	4.24E-01 (8.22E-03)
UF5	1.71E-01 (6.28E-02)+	2.07E-01 (1.02E-01)+	1.30E-01 (5.09E-02)+	9.14E-03 (2.89E-02)-	4.72E-03 (8.86E-03)-	1.97E-02 (4.83E-02)-	9.30E-02 (4.48E-02)
UF6	3.28E-01 (4.60E-02)-	3.18E-01 (9.71E-02)-	2.73E-01 (1.22E-01)-	1.15E-01 (2.08E-02)-	3.87E-01 (3.80E-02)+	4.28E-01 (6.99E-02)+	3.63E-01 (9.42E-02)
UF7	4.43E-01 (2.10E-01)-	6.88E-01 (4.63E-03)-	7.01E-01 (1.31E-03)+	6.77E-01 (4.88E-03)-	6.51E-01 (4.81E-02)-	6.48E-01 (1.07E-01)-	6.99E-01 (1.93E-03)
UF8	6.42E-01 (1.91E-02)-	5.89E-01 (1.69E-02)-	6.61E-01 (9.41E-03)-	4.49E-01 (5.22E-02)-	4.68E-01 (5.44E-03)-	6.03E-01 (7.93E-03)-	6.66E-01 (1.36E-02)
UF9	9.00E-01 (7.35E-02)-	9.51E-01 (1.05E-02)-	1.04E+00 (8.35E-03)-	6.97E-01 (1.27E-01)-	7.74E-01 (8.95E-02)-	9.60E-01 (2.11E-02)-	1.04E+00 (1.29E-02)
UF10	2.49E-01 (9.28E-02)+	5.23E-02 (2.76E-02)+	0.00E+00 (0.00E+00)-	0.00E+00 (0.00E+00)-	4.62E-01 (8.57E-04)+	0.00E+00 (0.00E+00)-	1.13E-02 (2.17E-02)
WFG1	2.08E+00 (1.66E-01)+	1.29E+00 (1.90E-02)+	1.04E+00 (2.63E-02)-	1.01E+00 (2.45E-02)-	4.57E-01 (1.89E-01)-	7.22E-02 (1.53E-01)-	1.12E+00 (2.50E-02)
WFG2	4.41E+00 (1.47E-02)-	4.45E+00 (2.17E-03)-	4.46E+00 (6.76E-04)~	4.45E+00 (1.33E-03)~	4.41E+00 (1.01E-02)-	4.45E+00 (1.65E-02)~	4.46E+00 (8.70E-04)
WFG3	3.96E+00 (2.94E-03)-	3.96E+00 (3.05E-03)-	3.97E+00 (6.33E-04)-	3.96E+00 (1.01E-03)-	3.93E+00 (5.72E-03)-	3.95E+00 (1.09E-03)-	3.98E+00 (5.80E-04)
WFG4	1.70E+00 (1.32E-03)+	1.69E+00 (3.06E-03)+	1.46E+00 (7.99E-03)-	1.50E+00 (4.23E-03)~	1.54E+00 (2.10E-02)+	1.62E+00 (2.61E-02)+	1.50E+00 (5.60E-03)
WFG5	1.45E+00 (1.07E-02)~	1.45E+00 (7.25E-03)~	1.45E+00 (3.20E-04)~	1.43E+00 (7.76E-04)-	1.43E+00 (1.24E-02)-	1.44E+00 (9.12E-04)~	1.45E+00 (2.75E-04)
WFG6	1.43E+00 (4.82E-02)-	9.99E-01 (2.36E-01)-	1.52E+00 (3.40E-01)-	1.68E+00 (9.49E-04)+	1.67E+00 (7.07E-03)+	1.69E+00 (1.52E-03)+	1.62E+00 (2.45E-01)
WFG7	1.70E+00 (6.59E-04)~	1.69E+00 (1.35E-03)~	1.70E+00 (4.60E-04)-	1.68E+00 (1.44E-03)-	1.67E+00 (2.47E-03)-	1.69E+00 (8.86E-04)-	1.70E+00 (2.40E-04)
WFG8	1.34E+00 (1.38E-02)-	1.39E+00 (1.25E-02)-	1.41E+00 (2.07E-03)-	1.39E+00 (3.16E-03)-	1.34E+00 (6.77E-03)-	1.39E+00 (4.88E-03)-	1.41E+00 (2.98E-03)
WFG9	1.54E+00 (7.89E-02)-	1.52E+00 (3.15E-01)-	1.63E+00 (1.30E-03)-	1.64E+00 (8.96E-04)~	1.63E+00 (1.12E-02)-	1.64E+00 (3.86E-03)~	1.64E+00 (1.46E-03)
Total	5/2/21	5/2/21	7/3/18	1/3/24	4/1/23	3/3/22	

The best average values obtained by these algorithms for each instance are given in bold and italics.

outperforms MOEA/D-AMG on WFG2, WFG4, and WFG6 when considering IGD values. Regarding the comparisons to MOEA/D-SBX and MOEA/D-DE, MOEA/D-AMG also shows some advantages, as it outperforms them on at least 21 test problems. Except for UF4-UF5, UF10, WFG1, and WFG4-WFG5, MOEA/D-AMG performs better on most cases regarding the IGD values. Since a multivariate Gaussian model is used in MOEA/D-GM to capture the population distribution from a global view, MOEA/D-GM can give the best IGD values on 7 cases of test instances. However, MOEA/D-AMG performs better than MOEA/D-AMG on 18 test problems. These statistical results about IGD indicate MOEA/D-AMG has the better optimization performance when compared with other algorithms on most of the test instances.

For HV metric shown in Table 2, the experimental results also demonstrate the superiority of MOEA/D-AMG on these test problems. It is clear from Table 2 that MOEA/D-AMG obtains the best results in 13 out of 28 cases, while MOEA/D-SBX, MOEA/D-DE, MOEA/D-GM, RM-MEDA, IM-MOEA, and AMEDA perform, respectively, best in 2, 3, 5, 1, 2, and 2 cases. As observed from Table 2, considering the F instances, except for MOEA/D-GM, MOEA/D-AMG is only beaten by IM-MOEA on F1, while MOEA-GM is a little better than MOEA/D-AMG on F6–F8. On UF problems, MOEA/D-AMG obtains significantly better results than MOEA/D-SBX and RM-MEDA on all cases in terms of HV. MOEA/D-DE performs better on UF4 and UF5, and MOEA/D-GM performs better on UF3-UF5 and UF7. When compared to IM-MEDA and AMEDA, they only perform better than MOEA/D-AMG on UF10 and UF6, respectively. For WFG instances, MOEA/D-DE and RM-MEDA only obtain statistically similar results with MOEA/D-AMG on WFG5 and WFG9, respectively. MOEA/D-AMG significantly outperforms MOEA/D-GM on WFG1-WFG9. Except for WFG1 and WFG4, MOEA/D-AMG shows the superior performance on most WFG problems. AMEDA only has the better HV result on WFG6. Therefore, it is reasonable to draw a conclusion that the proposed MOEA/D-AMG presents a superior performance over each compared algorithm when considering all the test problems.

Moreover, we also use the R2 indicator to further show the superior performance of MOEA/D-AMG, and the similar conclusions can be obtained from Table 3.

To examine the convergence speed of the seven algorithms, the mean IGD metric values versus the fitness evaluation numbers for all the compared algorithms over 20 independent runs are plotted in Figures 2–4, respectively, for some representative problems from F, UF, and WFG test suites. It can be observed from these figures that the curves of the mean IGD values obtained by MOEA/D-AMG reach the lowest positions with the fastest searching speed on most cases, including F2–F5, F9, UF1-UF2, UF8-UF9, WFG3, and WFG7-WFG8. Even for F1, F6–F7, UF6-UF7, and WFG9, MOEA/D-AMG achieves the second lowest mean IGD values in our experiment. The promising convergence speed of the proposed MOEA/D-AMG might be attributed to the adaptive strategy used in the multiple Gaussian process models.

To observe the final PFs, Figures 5 and 6 present the final nondominated fronts with the median IGD values found by each algorithm over 20 independent runs on F4–F5, F9, UF2, and UF8-UF9. Figure 5 shows that the final solutions of F4 yielded by RM-MEDA, IM-MOEA, and AMEDA do not reach the PF, while MOEA/D-GM and MOEA/D-AMG have good approximations to the true PF. Nevertheless, the solutions achieved by MOEA/D-AMG on the right end of the nondominated front have better convergence than those achieved by MOEA/D-GM. In Figure 5, it seems that F5 is a hard instance for all compared algorithms. This might be due to the fact that the optimal solutions to two neighboring subproblems are not very close to each other, resulting in little sense to mate among the solutions to these neighboring problems. Therefore, the final nondominated fronts of F5 are not uniformly distributed over the PF, especially in the right end. However, the proposed MOEA/D-AMG outperforms other algorithms on F5 in terms of both convergence and diversity. For F9 that is plotted in Figure 5, except for MOEA/D-GM and MOEA/D-AMG, other algorithms show the poor performance to search solutions which can approximate the true PF. With respect to UF problems in Figure 6, the final solutions with median IGD obtained by MOEA/D-AMG have better convergence and uniformly spread on the whole PF, when compared with the solutions obtained by other compared algorithms. These visual comparison results reveal that MOEA/D-AMG has much more stable performance to generate satisfactory final solutions with better convergence and diversity for the test instances.

Based on the above analysis, it can be concluded that MOEA/D-AMG performs better than MOEA/D-SBX, MOEA/D-DE, MOEA/D-GM, RM-MEDA, IM-MOEA, and AMEDA when considering all the F, UF, and WFG instances, which is attributed to the use of adaptive strategy to select a more suitable Gaussian model for running recombination. There may be different causes to let MOEA/D-SBX, MOEA/D-DE, MOEA/D-GM, RM-MEDA, IM-MOEA, and AMEDA perform not so well on the test suites. For RM-MEDA and AMEDA, they select individuals to construct the model by using the clustering method, which cannot show good diversity as that in the MOEA/D framework. Both algorithms using the EDA method firstly classify the individuals and then use all the individuals in the same class as samples for training model. Implicitly, an individual in the same class can be considered as a neighboring individual. However, on the boundary of each class, there may exist individuals that are far away from each other, and there may exist individuals from other classes that are closer to this boundary individual, which may result in the fact that their model construction is not so accurate. For IM-MOEA that uses reference vectors to partition the objective space, its training data are too small when dealing with the problems with low decision space, like some test instances used in our empirical studies.

4.5. Effectiveness of the Proposed Adaptive Strategy. According to the experimental results in the previous section, MOEA/D-AMG with the adaptive strategy shows great

TABLE 3: Performance comparison of MOEA/D-SBX, MOEA/D-DE, MOEA/D-GM, RM-MEDA, IM-MOEA, AMEDA, and MOEA/D-AMG in terms of the average R2 values and standard deviation on F, UF, and WFG problems.

Problems	MOEA/D-SBX	MOEA/D-DE	MOEA/D-GM	RM-MEDA	IM-MOEA	AMEDA	MOEA/D-AMG
F1	1.31E-01 (1.91E-04)-	1.32E-01 (5.73E-05)-	1.31E-01 (8.83E-06)~	1.31E-01 (3.00E-05)-	1.31E-01 (1.27E-05)~	1.31E-01 (5.01E-05)-	1.31E-01 (8.34E-06)
F2	1.82E-01 (2.88E-02)-	1.34E-01 (5.40E-04)-	1.31E-01 (8.11E-05)-	1.44E-01 (1.76E-03)-	1.57E-01 (7.50E-03)-	1.49E-01 (4.50E-03)-	1.31E-01 (7.95E-05)
F3	1.52E-01 (1.58E-02)-	1.34E-01 (5.82E-04)-	1.31E-01 (7.06E-05)-	1.34E-01 (6.91E-04)-	1.38E-01 (2.25E-03)-	1.35E-01 (1.57E-03)-	1.31E-01 (7.05E-05)
F4	1.55E-01 (1.59E-02)-	1.34E-01 (4.52E-04)-	1.31E-01 (1.17E-04)-	1.35E-01 (4.76E-04)-	1.37E-01 (3.14E-03)-	1.33E-01 (1.45E-03)-	1.31E-01 (1.70E-04)
F5	1.46E-01 (1.03E-02)-	1.34E-01 (4.84E-04)-	1.32E-01 (6.57E-04)-	1.34E-01 (8.93E-04)-	1.37E-01 (3.04E-03)-	1.35E-01 (1.33E-03)-	1.32E-01 (3.81E-04)
F6	8.17E-02 (6.68E-03)-	8.33E-02 (6.11E-04)-	7.91E-02 (8.59E-04)+	1.21E-01 (2.49E-02)-	1.18E-01 (1.67E-02)-	7.95E-02 (8.90E-04)-	7.89E-02 (5.88E-04)
F7	3.15E-01 (1.99E-02)-	2.33E-01 (4.55E-02)-	1.36E-01 (1.47E-02)+	7.13E-01 (1.38E-01)-	3.07E-01 (2.12E-02)-	3.51E-01 (2.65E-02)-	1.50E-01 (3.87E-02)
F8	2.95E-01 (3.34E-02)-	1.59E-01 (5.86E-03)-	1.33E-01 (1.54E-03)+	1.36E-01 (2.27E-03)-	1.58E-01 (3.17E-03)-	1.60E-01 (1.33E-02)-	1.35E-01 (3.33E-03)
F9	2.50E-01 (2.34E-02)-	2.00E-01 (7.50E-04)-	1.97E-01 (2.04E-04)~	2.06E-01 (6.14E-04)-	2.18E-01 (4.81E-03)-	2.10E-01 (3.38E-03)-	1.97E-01 (3.76E-04)
UF1	1.81E-01 (1.78E-02)-	1.34E-01 (3.13E-04)-	1.31E-01 (7.74E-05)~	1.43E-01 (7.91E-04)-	1.56E-01 (9.45E-03)-	1.48E-01 (6.88E-03)-	1.32E-01 (6.57E-04)
UF2	1.46E-01 (9.19E-03)-	1.34E-01 (4.29E-04)-	1.32E-01 (4.82E-04)~	1.34E-01 (4.08E-04)-	1.37E-01 (2.98E-03)-	1.35E-01 (2.24E-03)-	1.32E-01 (3.71E-04)
UF3	2.88E-01 (2.50E-02)-	1.61E-01 (4.57E-03)-	1.32E-01 (1.21E-03)+	1.37E-01 (3.92E-03)+	1.63E-01 (7.28E-03)-	1.63E-01 (1.15E-02)-	1.38E-01 (1.02E-02)
UF4	2.12E-01 (1.59E-03)+	2.09E-01 (9.04E-04)+	2.10E-01 (9.10E-04)+	2.23E-01 (2.71E-03)-	2.18E-01 (1.93E-03)-	2.23E-01 (2.19E-03)-	2.15E-01 (1.50E-03)
UF5	4.51E-01 (6.76E-02)-	2.63E-01 (3.21E-02)+	2.79E-01 (1.76E-02)+	5.75E-01 (2.32E-01)-	5.69E-01 (5.63E-02)-	4.83E-01 (7.24E-02)-	3.34E-01 (6.39E-02)
UF6	3.42E-01 (6.58E-02)-	2.42E-01 (1.96E-02)+	3.25E-01 (1.34E-01)-	3.16E-01 (2.19E-02)-	2.68E-01 (1.41E-02)-	2.53E-01 (6.35E-02)-	2.47E-01 (8.10E-02)
UF7	3.09E-01 (1.28E-01)-	1.68E-01 (6.60E-04)-	1.66E-01 (1.98E-04)~	1.70E-01 (9.38E-04)-	1.77E-01 (1.24E-02)-	1.90E-01 (6.55E-02)-	1.66E-01 (3.39E-04)
UF8	8.01E-02 (3.55E-03)-	8.31E-02 (7.51E-04)-	7.93E-02 (7.24E-04)-	1.09E-01 (2.20E-02)-	1.24E-01 (6.95E-03)-	7.97E-02 (4.53E-04)-	7.83E-02 (5.07E-04)
UF9	8.20E-02 (1.45E-02)-	6.60E-02 (1.80E-03)-	5.85E-02 (9.50E-04)-	8.76E-02 (1.08E-02)-	9.33E-02 (1.66E-02)-	6.46E-02 (2.95E-03)-	5.94E-02 (2.09E-03)
UF10	1.91E-01 (5.10E-02)+	2.16E-01 (1.29E-02)+	4.55E-01 (3.38E-02)-	6.76E-01 (2.13E-02)-	1.28E-01 (1.09E-04)+	4.92E-01 (5.74E-02)-	3.05E-01 (2.13E-02)
WFG1	6.98E-01 (2.59E-02)-	8.27E-01 (4.77E-03)+	8.81E-01 (1.30E-02)-	9.33E-01 (9.07E-03)-	1.07E+00 (4.40E-02)-	1.20E+00 (6.37E-02)-	8.56E-01 (4.93E-03)
WFG2	5.24E-01 (5.00E-02)-	4.28E-01 (1.17E-04)~	4.28E-01 (3.25E-05)~	4.28E-01 (9.78E-05)~	4.31E-01 (1.04E-03)-	4.49E-01 (4.45E-02)-	4.28E-01 (2.66E-05)
WFG3	4.52E-01 (2.12E-04)+	4.52E-01 (1.57E-04)-	4.51E-01 (3.09E-05)~	4.53E-01 (7.76E-05)-	4.54E-01 (3.97E-04)-	4.53E-01 (1.10E-04)-	4.51E-01 (2.29E-05)
WFG4	5.81E-01 (8.19E-05)+	5.82E-01 (1.75E-04)+	5.96E-01 (3.71E-04)-	5.95E-01 (4.35E-04)-	5.91E-01 (1.60E-03)+	5.86E-01 (1.60E-03)+	5.93E-01 (5.36E-04)
WFG5	6.02E-01 (1.66E-03)~	6.03E-01 (5.86E-04)+	6.03E-01 (7.01E-04)-	6.04E-01 (7.93E-04)-	6.02E-01 (2.42E-03)-	6.04E-01 (1.68E-03)-	6.02E-01 (9.62E-04)
WFG6	5.99E-01 (3.55E-03)-	6.41E-01 (2.05E-02)-	5.96E-01 (2.91E-02)-	5.82E-01 (5.91E-05)+	5.83E-01 (5.16E-04)+	5.82E-01 (1.22E-04)+	5.88E-01 (2.10E-02)
WFG7	5.82E-01 (4.11E-05)-	5.82E-01 (3.76E-05)-	5.81E-01 (2.57E-05)-	5.82E-01 (1.06E-04)-	5.83E-01 (1.31E-04)-	5.82E-01 (8.75E-05)-	5.81E-01 (1.21E-05)
WFG8	6.28E-01 (8.08E-04)-	6.26E-01 (5.35E-04)-	6.25E-01 (1.71E-04)-	6.30E-01 (1.06E-03)-	6.31E-01 (1.03E-03)-	6.27E-01 (1.47E-03)-	6.24E-01 (6.73E-05)
WFG9	5.91E-01 (6.25E-03)-	6.03E-01 (3.17E-02)-	5.91E-01 (1.94E-04)-	5.92E-01 (6.75E-04)-	5.91E-01 (7.84E-04)-	5.91E-01 (9.08E-04)-	5.90E-01 (2.30E-04)
Total	4/1/23	7/1/20	6/7/15	2/1/25	3/1/24	2/0/26	

The best average values obtained by these algorithms for each instance are given in bold and italics.

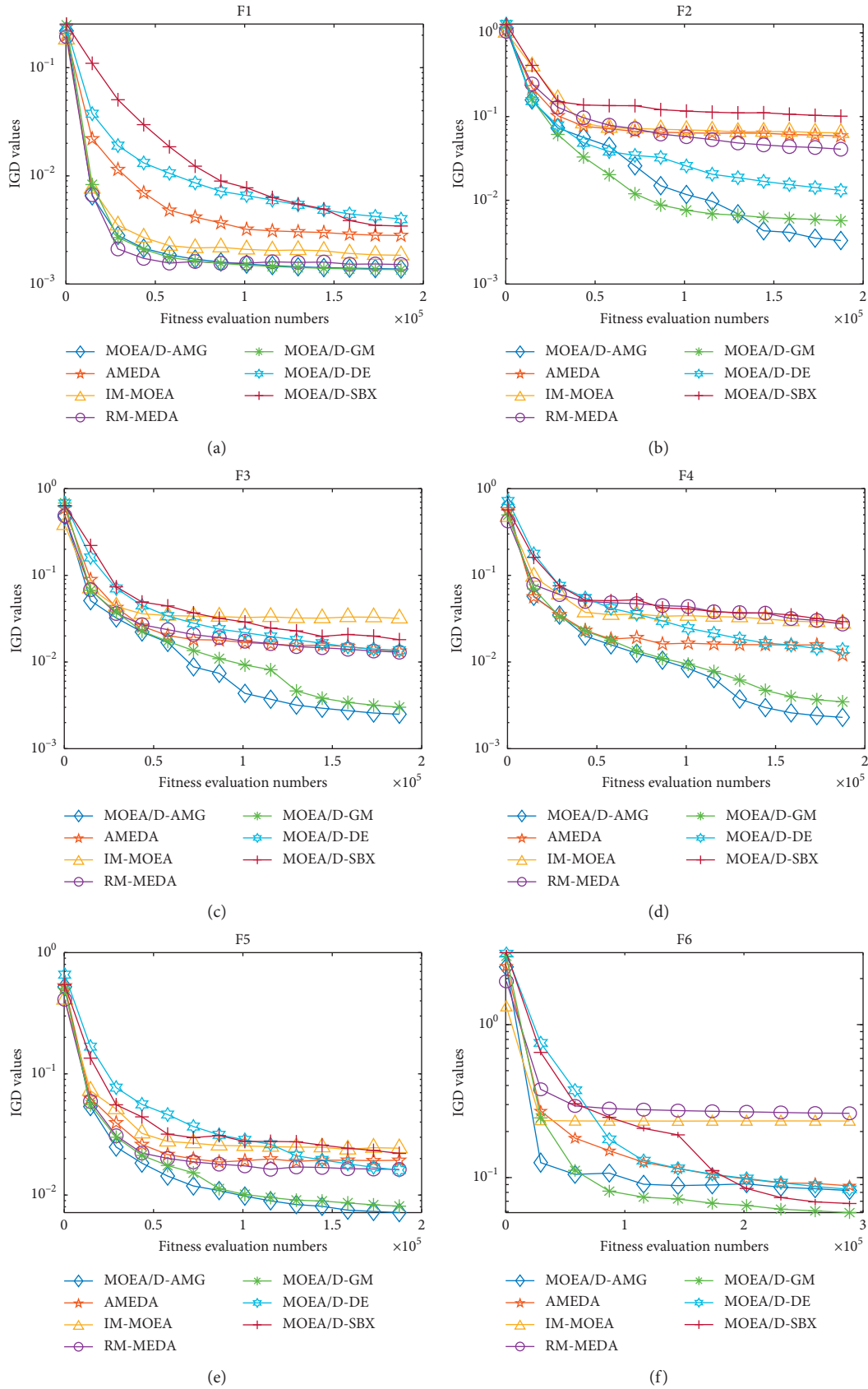


FIGURE 2: Continued.

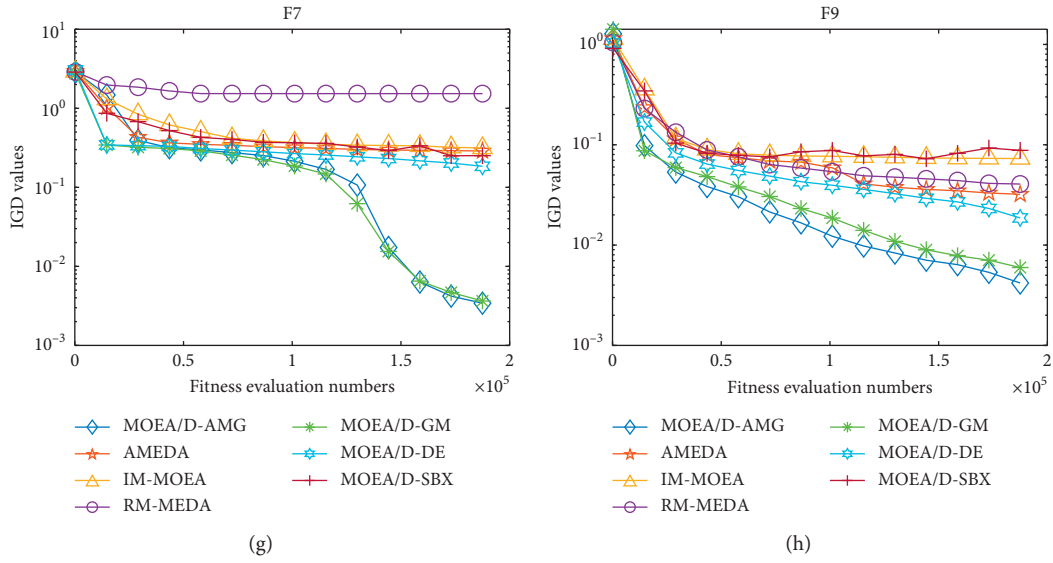


FIGURE 2: The convergence curves of all the compared algorithms on F1–F7 and F9.

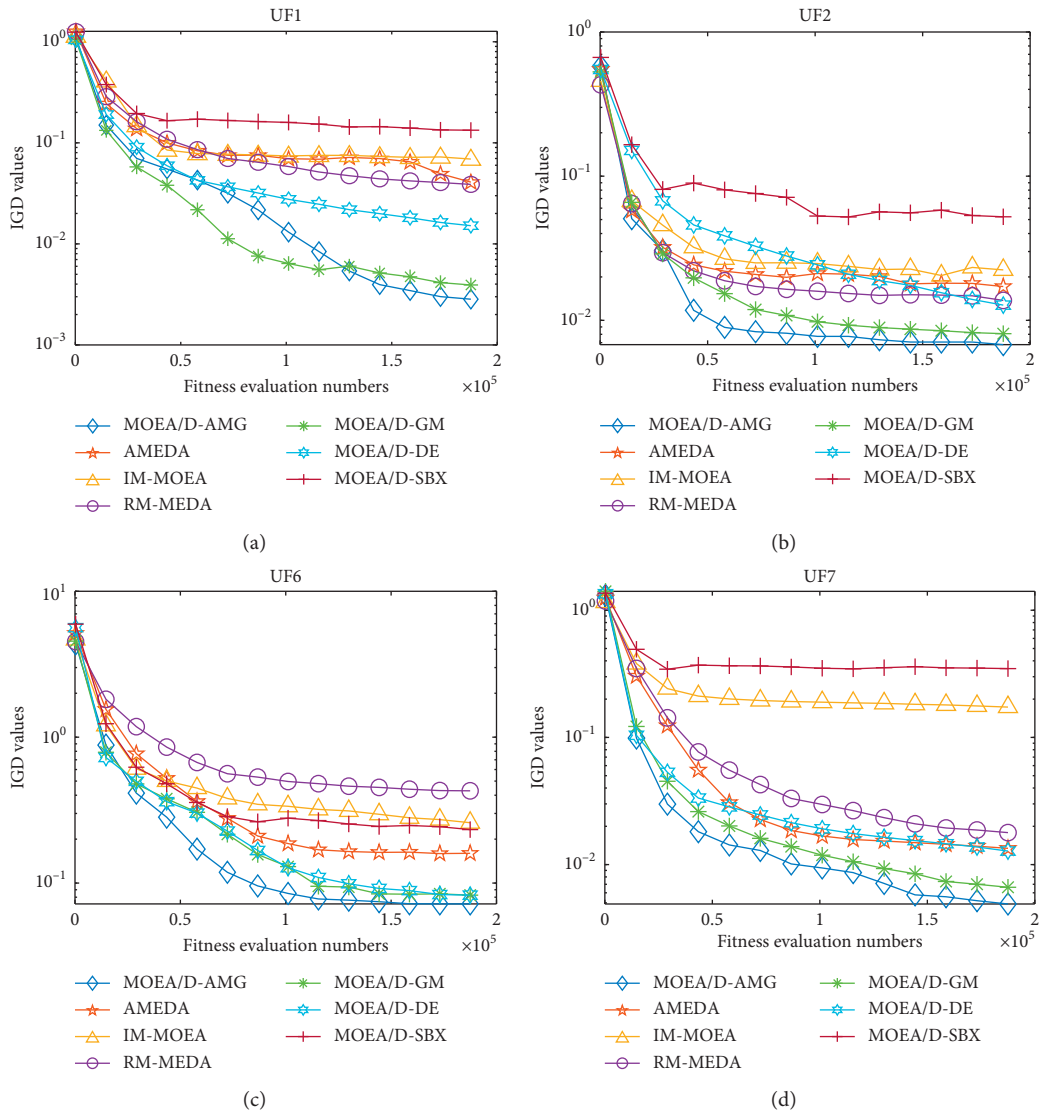


FIGURE 3: Continued.

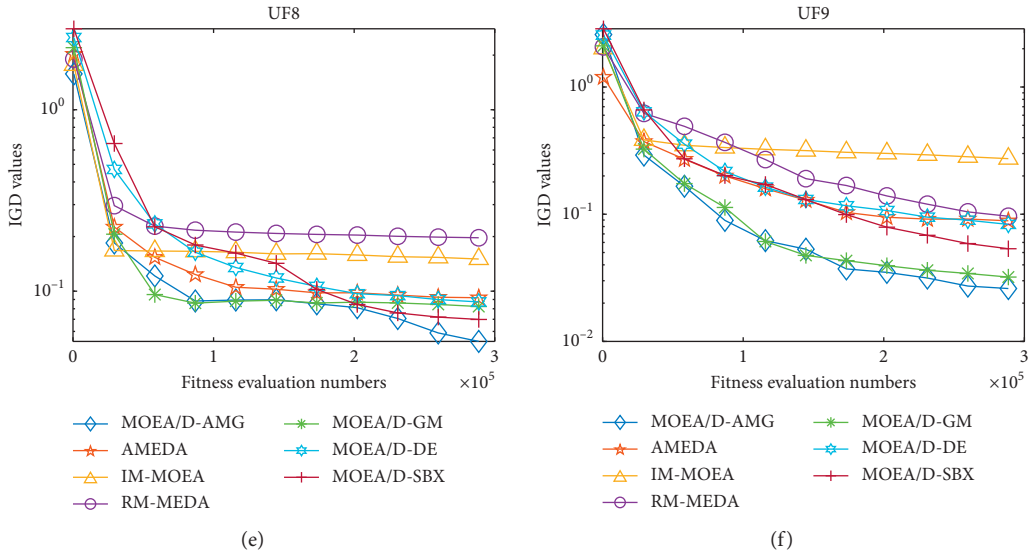


FIGURE 3: The convergence curves of all the compared algorithms on UF1-UF2 and UF6-UF9.

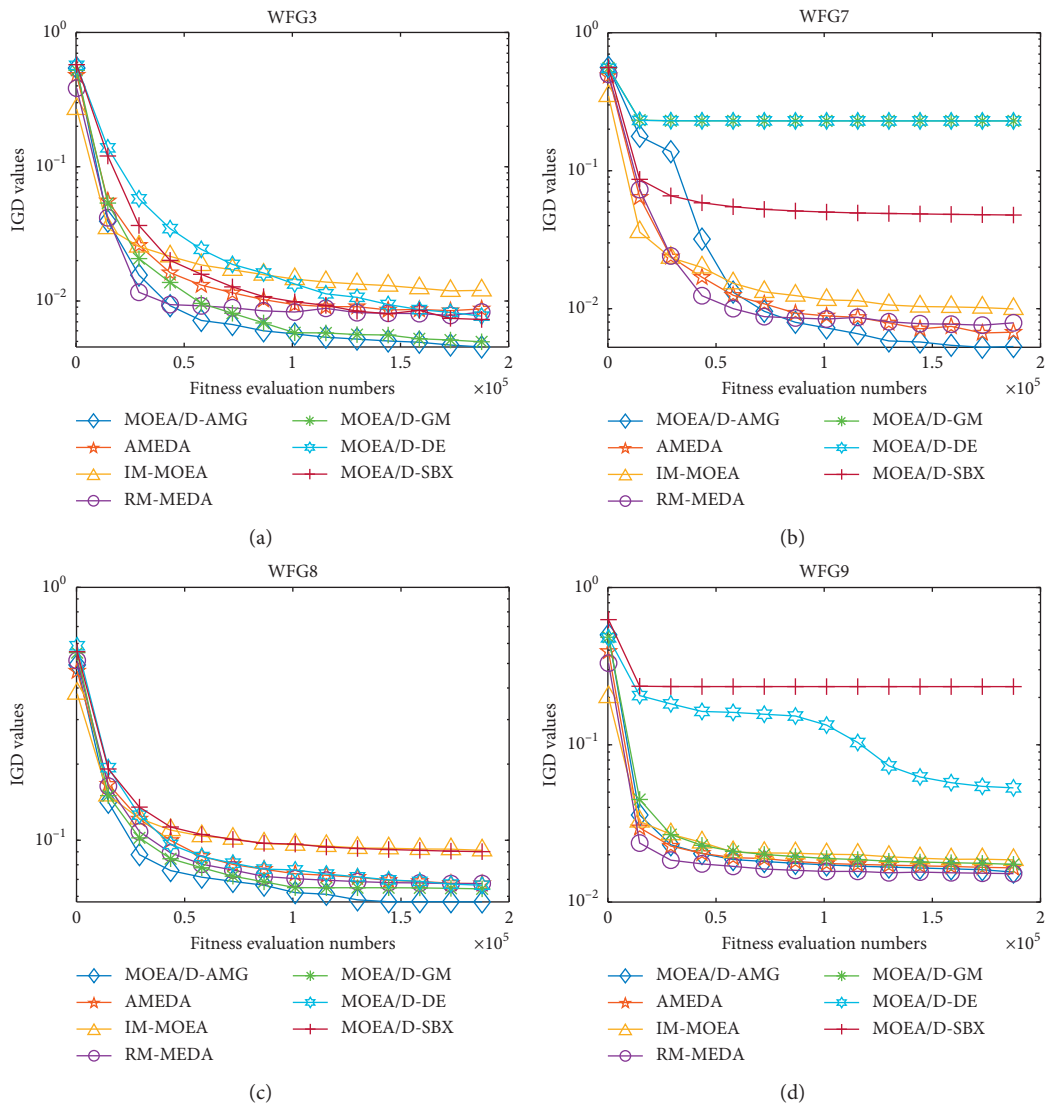


FIGURE 4: The convergence curves of all the compared algorithms on WFG3 and WFG7-WFG9.

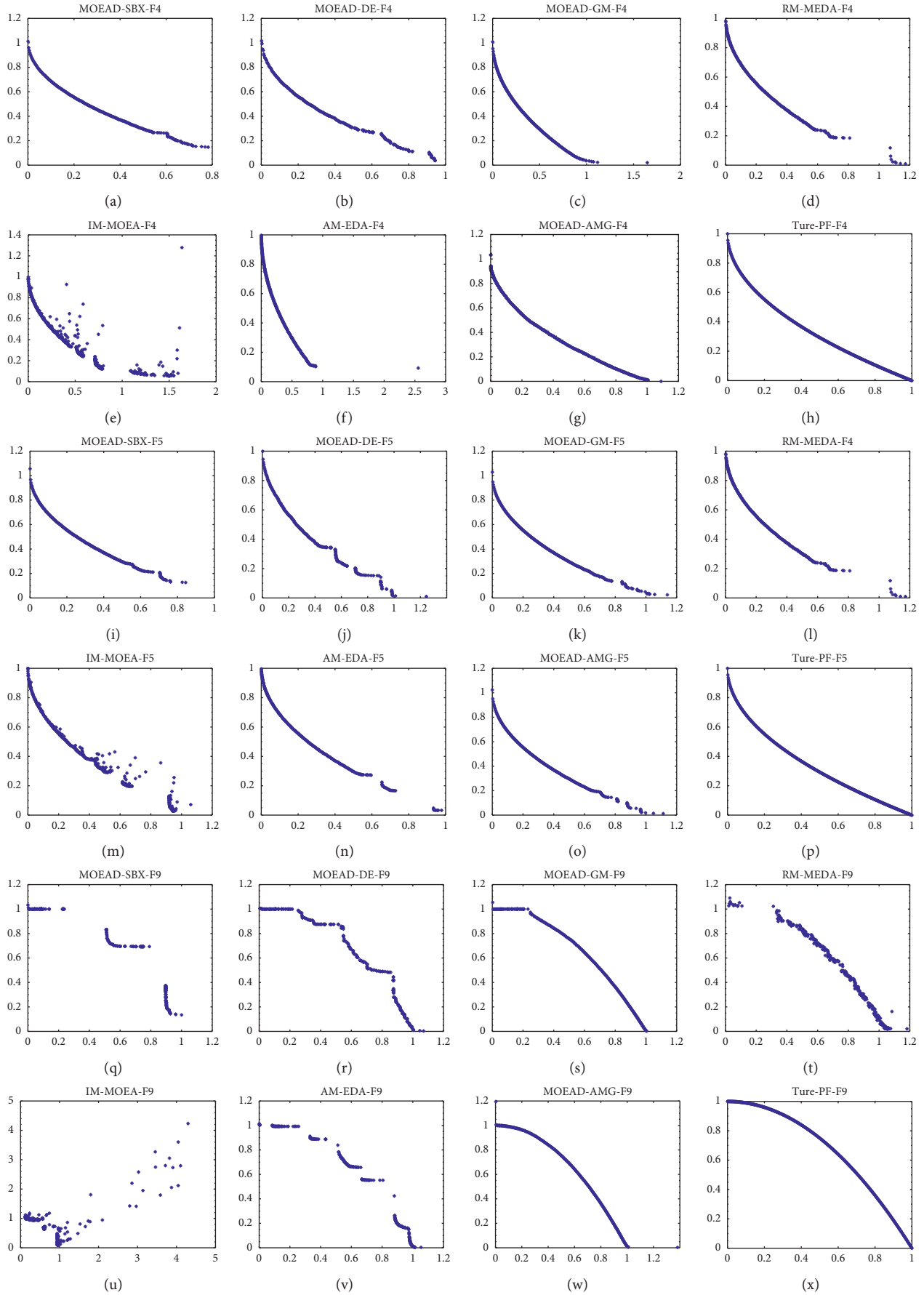


FIGURE 5: The final populations obtained by all the compared algorithms on F4, F5, and F9.

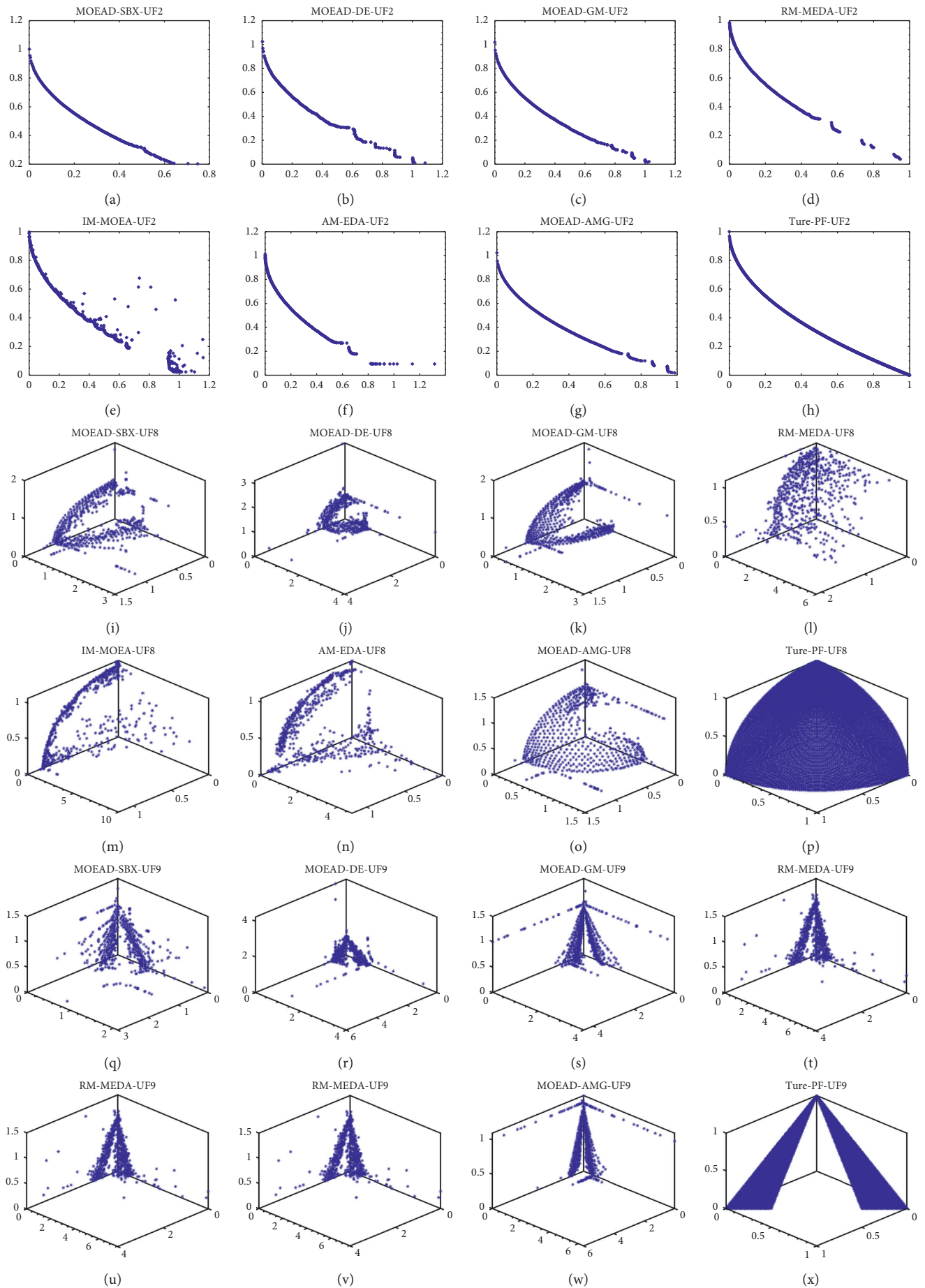


FIGURE 6: The final populations obtained by all the compared algorithms on UF2, UF8, and UF9.

TABLE 4: Performance comparison of MOEA/D-FMG with various Gaussian distributions and MOEA/D-AMG in terms of the average IGD values and standard deviations on F, UF, and WFG problems.

Problems	MOEA/D-FMG (0, 0.6 ²)	MOEA/D-FMG (0, 0.8 ²)	MOEA/D-FMG (0, 1.0 ²)	MOEA/D-FMG (0, 1.2 ²)	MOEA/D-FMG (0, 1.4 ²)	MOEA/D-AMG
F1	1.29E-03 (8.00E-04)+	1.44E-03 (1.04E-04)-	1.46E-03 (2.36E-05)-	1.60E-03 (3.15E-05)-	2.10E-03 (2.04E-05)-	1.35E-03 (1.28E-05)
F2	5.05E-03 (5.13E-02)-	4.03E-03 (2.24E-03)-	5.17E-03 (1.24E-04)-	6.30E-03 (5.10E-03)-	9.40E-03 (6.51E-03)-	2.85E-03 (2.95E-04)
F3	5.53E-03 (3.03E-02)-	3.45E-03 (1.59E-03)-	3.81E-03 (2.11E-04)-	5.20E-03 (1.90E-03)-	7.40E-03 (5.86E-03)-	2.55E-03 (3.59E-04)
F4	6.10E-03 (4.18E-02)-	4.72E-03 (1.19E-03)-	5.76E-03 (5.22E-04)-	6.60E-03 (4.15E-03)-	9.70E-03 (4.03E-03)-	2.93E-03 (7.23E-04)
F5	1.21E-02 (2.45E-02)-	8.80E-03 (1.24E-03)-	8.97E-03 (1.46E-04)-	1.03E-02 (1.21E-03)-	1.31E-02 (4.72E-03)-	7.94E-03 (1.27E-03)
F6	5.93E-02 (1.75E-02)-	5.93E-02 (2.80E-03)-	4.70E-02 (7.65E-03)+	6.96E-02 (6.89E-02)-	8.44E-02 (3.00E-02)-	6.54E-02 (8.13E-03)
F7	2.59E-01 (2.08E-02)-	1.27E-01 (4.20E-02)-	2.57E-02 (3.11E-02)+	3.72E-02 (4.01E-01)+	1.13E-01 (7.48E-02)-	4.21E-02 (7.84E-02)
F8	5.10E-02 (5.24E-02)-	1.25E-02 (1.82E-02)+	1.15E-02 (2.94E-03)+	1.38E-02 (5.44E-03)+	2.88E-02 (8.89E-03)-	1.55E-02 (1.07E-02)
F9	8.31E-03 (3.84E-02)-	8.93E-03 (2.190E-03)-	8.00E-03 (1.17E-03)-	1.04E-02 (3.54E-03)-	1.47E-02 (2.61E-02)-	5.14E-03 (2.12E-03)
UF1	2.45E-03 (2.71E-02)-	2.77E-03 (8.93E-04)-	2.88E-03 (5.02E-04)-	3.60E-03 (3.21E-03)-	5.00E-03 (1.07E-02)-	2.45E-03 (1.42E-03)
UF2	6.95E-03 (2.14E-02)-	6.40E-03 (1.27E-03)-	6.98E-03 (1.35E-03)-	7.50E-03 (1.59E-03)-	9.00E-03 (2.96E-03)-	6.68E-03 (8.37E-04)
UF3	2.06E-03 (4.31E-02)-	5.43E-03 (1.94E-02)+	5.60E-03 (3.10E-03)+	5.40E-03 (6.02E-03)+	6.51E-03 (1.21E-02)+	1.74E-02 (1.99E-02)
UF4	7.09E-02 (3.61E-03)-	6.26E-02 (2.89E-03)-	5.49E-02 (2.81E-03)+	5.35E-02 (9.49E-03)+	5.13E-02 (6.77E-03)+	6.98E-02 (5.78E-03)
UF5	3.41E-01 (8.29E-02)+	3.08E-01 (7.50E-02)+	3.17E-01 (5.20E-02)+	3.31E-01 (5.26E-01)+	3.85E-01 (3.63E-01)-	4.08E-01 (9.50E-02)
UF6	2.33E-01 (1.32E-01)-	2.26E-01 (4.81E-02)-	1.65E-01 (2.66E-01)~	1.15E-01 (3.70E-02)+	1.41E-01 (3.57E-02)+	1.62E-01 (2.06E-01)
UF7	5.93E-03 (2.27E-01)-	5.82E-03 (2.11E-03)-	6.00E-03 (3.71E-04)-	6.90E-03 (2.62E-03)-	6.50E-03 (3.27E-02)-	5.13E-03 (6.70E-04)
UF8	6.09E-02 (6.44E-03)-	6.30E-02 (3.23E-03)-	6.59E-02 (4.34E-03)-	7.65E-02 (3.53E-02)-	8.73E-02 (3.58E-03)-	5.74E-02 (8.29E-03)
UF9	7.13E-02 (5.02E-02)-	4.34E-02 (4.99E-03)-	3.22E-02 (3.12E-03)-	4.50E-02 (5.92E-02)-	4.60E-02 (4.87E-02)-	2.76E-02 (4.37E-03)
UF10	6.19E-01 (8.23E-02)+	8.89E-01 (7.95E-02)+	1.70E+00 (2.21E-01)-	2.43E+00 (1.19E-01)-	2.58E+00 (4.09E-03)-	8.03E-01 (1.71E-01)
WFG1	1.12E+00 (5.32E-02)-	1.16E+00 (8.02E-03)-	1.16E+00 (8.15E-03)-	1.19E+00 (9.32E-03)-	1.22E+00 (8.90E-02)-	1.11E+00 (1.17E-02)
WFG2	1.55E-02 (2.12E-02)~	1.55E-02 (5.20E-04)~	1.57E-02 (3.63E-04)-	1.59E-02 (1.31E-03)-	1.62E-02 (1.78E-03)-	1.55E-02 (3.24E-04)
WFG3	4.45E-01 (5.70E-04)~	4.45E-01 (5.04E-04)~	4.59E-01 (1.71E-04)-	4.62E-01 (2.21E-04)-	4.64E-01 (1.76E-03)-	4.45E-03 (1.42E-04)
WFG4	5.00E-02 (2.45E-04)-	5.34E-02 (8.10E-04)-	6.03E-02 (2.39E-03)-	6.70E-02 (6.18E-04)-	7.50E-02 (4.31E-03)-	4.81E-02 (1.64E-03)
WFG5	6.55E-02 (2.94E-03)-	6.55E-02 (2.10E-03)-	6.56E-02 (4.29E-05)-	6.57E-02 (1.94E-04)-	6.56E-02 (3.02E-03)-	6.52E-02 (4.14E-04)
WFG6	5.13E-03 (1.29E-02)+	2.77E-02 (6.98E-02)-	6.21E-02 (9.88E-02)-	6.44E-02 (2.15E-04)-	1.31E-01 (1.25E-03)-	1.62E-02 (5.02E-02)
WFG7	5.50E-03 (1.55E-04)~	5.60E-03 (2.94E-04)~	5.81E-03 (5.83E-05)-	6.40E-03 (3.49E-04)-	8.30E-03 (7.98E-04)-	5.53E-03 (2.13E-05)
WFG8	8.28E-02 (6.66E-03)-	8.20E-02 (1.96E-03)-	8.34E-02 (9.81E-04)-	8.59E-02 (1.38E-03)-	8.96E-02 (3.16E-03)-	5.77E-02 (1.60E-03)
WFG9	1.59E-02 (1.99E-02)-	1.63E-02 (8.29E-02)-	1.73E-02 (2.40E-04)-	1.90E-02 (1.01E-04)-	2.11E-02 (2.64E-03)-	1.52E-02 (2.92E-04)
Total	4/3/21	4/3/21	6/1/21	6/0/22	3/0/25	

The better average values of these algorithms for each instance are highlighted in bold italics.

TABLE 5: Performance comparison of MOEA/D-FMG with various Gaussian distributions and MOEA/D-AMG in terms of the average IGD values and standard deviations on F, UF, and WFG problems.

Problems	MOEA/D-FMG (0, 1.6 ²)	MOEA/D-FMG (0, 1.8 ²)	MOEA/D-FMG (0, 2.0 ²)	MOEA/D-AMG
F1	2.16E-03 (2.06E-05)-	2.21E-03 (3.03E-05)-	2.20E-03 (7.32E-05)-	1.35E-03 (1.28E-05)
F2	1.03E-02 (2.14E-04)-	1.10E-02 (5.72E-03)-	1.40E-02 (2.01E-03)-	2.85E-03 (2.95E-04)
F3	7.91E-03 (2.31E-04)-	8.81E-03 (2.81E-03)-	9.40E-03 (3.86E-03)-	2.55E-03 (3.59E-04)
F4	9.93E-03 (7.19E-04)-	1.04E-02 (4.49E-03)-	1.43E-02 (6.03E-03)-	2.93E-03 (7.23E-04)
F5	9.67E-03 (2.96E-04)-	1.33E-02 (4.21E-03)-	1.49E-02 (4.62E-03)-	7.94E-03 (1.27E-03)
F6	9.01E-01 (9.93E-03)-	9.96E-02 (2.89E-02)-	1.09E-01 (3.97E-02)-	6.54E-02 (8.13E-03)
F7	1.23E-01 (7.21E-02)-	9.84E-02 (2.21E-01)-	1.29E-01 (8.08E-02)-	4.21E-02 (7.84E-02)
F8	2.95E-02 (9.21E-03)-	2.85E-02 (5.31E-03)-	2.95E-02 (5.89E-03)-	1.55E-02 (1.07E-02)
F9	1.54E-02 (2.41E-03)-	1.55E-02 (2.34E-03)-	1.66E-02 (2.11E-02)-	5.14E-03 (2.12E-03)
UF1	5.42E-03 (5.12E-04)-	5.68E-03 (3.57E-03)-	5.90E-03 (6.20E-02)-	2.45E-03 (1.42E-03)
UF2	8.78E-03 (5.35E-03)-	9.50E-03 (5.55E-03)-	9.71E-03 (2.77E-03)-	6.68E-03 (8.37E-04)
UF3	6.55E-03 (3.00E-03)+	6.60E-03 (5.52E-03)+	1.75E-02 (2.29E-02)~	1.74E-02 (1.99E-02)
UF4	5.60E-02 (3.19E-03)+	7.03E-02 (7.12E-03)-	7.13E-02 (6.09E-03)-	6.98E-02 (5.78E-03)
UF5	3.90E-01 (6.31E-02)+	4.56E-01 (5.06E-01)-	5.85E-01 (5.02E-01)-	4.08E-01 (9.50E-02)
UF6	1.63E-01 (2.10E-01)~	1.45E-01 (3.00E-02)+	1.84E-01 (2.57E-02)-	1.62E-01 (2.06E-01)
UF7	6.77E-03 (4.22E-04)-	6.94E-03 (4.02E-03)-	6.99E-03 (3.17E-02)-	5.13E-03 (6.70E-04)
UF8	8.00E-02 (4.54E-03)-	9.05E-02 (4.04E-02)-	9.73E-02 (3.08E-03)-	5.74E-02 (8.29E-03)
UF9	3.70E-02 (2.99E-03)-	4.52E-02 (3.20E-02)-	4.76E-02 (4.45E-02)-	2.76E-02 (4.37E-03)
UF10	1.92E+00 (2.71E-01)-	2.53E+00 (1.09E-01)-	2.50E+00 (4.94E-03)-	8.03E-01 (1.71E-01)
WFG1	1.24E+00 (7.25E-03)-	1.28E+00 (7.02E-03)-	2.34E+00 (8.10E-02)-	1.11E+00 (1.17E-02)
WFG2	1.77E-02 (2.93E-04)-	1.79E-02 (4.00E-03)-	1.92E-02 (2.78E-03)-	1.55E-02 (3.24E-04)
WFG3	4.60E-01 (2.22E-04)-	4.69E-01 (5.01E-04)-	4.97E-01 (1.01E-03)-	4.45E-03 (1.42E-04)
WFG4	6.10E-02 (3.29E-03)-	7.67E-02 (4.28E-04)-	7.90E-02 (2.01E-03)-	4.81E-02 (1.64E-03)
WFG5	6.66E-02 (3.97E-05)-	6.57E-02 (1.93E-04)-	6.76E-02 (8.02E-03)-	6.52E-02 (4.14E-04)
WFG6	8.47E-02 (4.78E-02)-	1.44E-01 (4.02E-04)-	1.92E-01 (3.09E-03)-	1.62E-02 (5.02E-02)
WFG7	8.39E-03 (4.93E-05)-	8.41E-03 (2.93E-04)-	8.62E-03 (8.08E-04)-	5.53E-03 (2.13E-05)
WFG8	8.45E-02 (8.90E-04)-	9.06E-02 (3.08E-03)-	1.12E-01 (3.16E-03)-	5.77E-02 (1.60E-03)
WFG9	3.54E-02 (3.40E-04)-	3.71E-02 (2.21E-04)-	3.77E-02 (1.94E-03)-	1.52E-02 (2.92E-04)
Total	3/1/24	2/0/26	0/1/27	

The better average values of these algorithms for each instance are given in bold and italics.

TABLE 6: The standard deviations of Gaussian models adopted by each K value.

K	Standard deviation
3	0.8, 1.0, 1.2
5	0.6, 0.8, 1.0, 1.2, 1.4
7	0.6, 0.8, 0.9, 1.0, 1.1, 1.2, 1.4
10	0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4

advantages when compared with other algorithms. In order to illustrate the effectiveness of the proposed adaptive strategy, we design some experiments for in-depth analysis.

In this section, an algorithm called MOEA/D-FMG is used to run this comparison experiment. Note that the difference between MOEA/D-FMG and MOEA/D-AMG lies in that MOEA/D-FMG adopts a fixed Gaussian distribution to control the generation of new offspring without adaptive strategy during the whole evolution. Here, MOEA/D-FMG with five different distributions (0, 0.6²), (0, 0.8²), (0, 1.0²), (0, 1.2²), and (0, 1.4²) are conducted to solve all the F, UF, and WFG instances. To have a fair comparison, other components of MOEA/D-FMG are kept the same as that in MOEA/D-AMG. The statistical results of the IGD metric values obtained by MOEA/D-AMG and MOEA/D-FMG with different distributions over 20 independent runs are presented in Table 4 for the used test problems.

As observed from Table 4, MOEA/D-AMG obtains the best mean IGD values on 17 out of 28 cases, while all the MOEA/D-FMG variants totally achieve the best mean IGD values on the remaining 11 cases. More specifically, for five MOEA/D-FMG variants with the distributions (0, 0.6²), (0, 0.8²), (0, 1.0²), (0, 1.2²), and (0, 1.4²), they, respectively, obtain the best HV results on 4, 1, 3, 2, and 1 out of 28 comparisons. As observed from the one-to-one comparisons in the last row of Table 4, MOEA/D-AMG performs better than five MOEA/D-FMG variants with the distributions (0, 0.6²), (0, 0.8²), (0, 1.0²), (0, 1.2²), and (0, 1.4²), respectively, on 21, 21, 21, 22, and 25 out of 28 comparisons whereas MOEA/D-AMG is only beaten by these five MOEA/D-FMG variants on 4, 4, 6, 6, and 3 comparisons, respectively. Therefore, it can be concluded from Table 4 that the proposed adaptive strategy for selecting a suitable Gaussian model significantly contributes to the superior performance of MOEA/D-AMG on solving MOPs.

The above analysis only considers five different distributions with variance from 0.6² to 1.4². The performance with a Gaussian distribution with bigger variances is further studied here. Thus, several variants of MOEA/D-FMG with three fixed (0, 1.6²), (1, 1.8²), and (0, 2.0²) are used for comparison. Table 5 lists the IGD comparative results on all instances adopted. As observed from the results, MOEA/D-AMG also shows the obvious advantages, as it performs best

TABLE 7: Performance comparison of MOEA/D-AMG with different values of K in terms of the average IGD values and standard deviations on F, UF, and WFG problems.

Problems	$K=3$	$K=5$	$K=7$	$K=10$
F1	1.36E-03 (1.25E-05)	1.35E-03 (1.28E-05)	1.34E-03 (6.96E-06)	1.33E-03 (1.01E-05)
F2	3.29E-03 (3.50E-04)	2.85E-03 (2.95E-04)	3.48E-03 (1.36E-03)	3.25E-03 (7.55E-04)
F3	3.09E-03 (2.70E-04)	2.55E-03 (3.59E-04)	3.29E-03 (7.85E-04)	3.57E-03 (1.02E-03)
F4	3.18E-03 (7.57E-04)	2.93E-03 (7.23E-04)	3.61E-03 (1.48E-03)	3.15E-03 (1.21E-03)
F5	9.42E-03 (1.56E-03)	7.94E-03 (1.27E-03)	7.40E-03 (7.13E-04)	7.82E-03 (1.37E-03)
F6	6.14E-02 (7.57E-03)	6.54E-02 (8.13E-03)	5.99E-02 (8.70E-03)	6.36E-02 (7.34E-03)
F7	6.71E-03 (7.04E-03)	4.21E-02 (7.84E-02)	4.41E-02 (6.07E-02)	9.36E-02 (1.21E-01)
F8	7.89E-03 (5.53E-03)	1.55E-02 (1.07E-02)	1.28E-02 (1.44E-02)	2.14E-02 (2.46E-02)
F9	5.77E-03 (1.84E-03)	5.14E-03 (2.12E-03)	5.54E-03 (2.23E-03)	5.01E-03 (2.07E-03)
UF1	3.06E-03 (1.24E-03)	2.45E-03 (1.42E-03)	2.12E-03 (3.92E-04)	2.15E-03 (6.25E-04)
UF2	7.08E-03 (1.27E-03)	6.68E-03 (8.37E-04)	6.73E-03 (5.91E-04)	8.19E-03 (3.03E-03)
UF3	1.01E-02 (5.71E-03)	1.74E-02 (1.99E-02)	1.51E-02 (2.47E-02)	1.30E-02 (1.14E-02)
UF4	6.92E-02 (6.78E-03)	6.98E-02 (5.78E-03)	7.41E-02 (5.73E-03)	7.38E-02 (4.95E-03)
UF5	3.70E-01 (7.27E-02)	4.08E-01 (9.50E-02)	3.59E-01 (6.42E-02)	3.12E-01 (4.57E-02)
UF6	1.71E-01 (2.32E-01)	1.62E-01 (2.06E-01)	1.13E-01 (5.06E-02)	2.34E-01 (2.70E-01)
UF7	5.02E-02 (1.41E-01)	5.13E-03 (6.70E-04)	5.33E-03 (4.75E-04)	5.26E-03 (7.49E-04)
UF8	5.84E-02 (6.25E-03)	5.74E-02 (8.29E-03)	5.96E-02 (1.13E-02)	5.59E-02 (9.79E-03)
UF9	3.29E-02 (1.39E-02)	2.76E-02 (4.37E-03)	3.07E-02 (8.23E-03)	3.38E-02 (1.40E-02)
UF10	1.45E+00 (2.84E-01)	8.03E-01 (1.71E-01)	7.24E-01 (9.71E-02)	6.10E-01 (7.81E-02)
WFG1	1.16E+00 (7.93E-03)	1.11E+00 (1.17E-02)	1.14E+00 (5.25E-03)	1.12E+00 (7.77E-03)
WFG2	1.58E-02 (4.26E-04)	1.55E-02 (3.24E-04)	1.56E-02 (3.13E-04)	1.56E-02 (3.37E-04)
WFG3	5.07E-03 (1.82E-04)	4.45E-03 (1.42E-04)	4.74E-03 (1.80E-04)	4.79E-03 (1.68E-04)
WFG4	5.73E-02 (2.02E-03)	4.81E-02 (1.64E-03)	5.45E-02 (1.64E-03)	5.20E-02 (2.15E-03)
WFG5	6.55E-02 (4.28E-04)	6.52E-02 (4.14E-04)	6.53E-02 (1.44E-04)	6.53E-02 (3.18E-04)
WFG6	1.18E-01 (1.18E-01)	1.62E-02 (5.02E-02)	5.11E-02 (8.14E-02)	5.55E-02 (9.32E-02)
WFG7	5.92E-03 (1.56E-04)	5.53E-03 (2.13E-05)	5.70E-03 (6.29E-05)	5.63E-03 (4.28E-05)
WFG8	5.88E-02 (9.04E-04)	5.77E-02 (1.60E-03)	5.81E-02 (1.59E-03)	6.08E-02 (1.35E-03)
WFG9	1.66E-02 (4.21E-04)	1.52E-02 (2.92E-04)	1.61E-02 (6.51E-04)	1.57E-02 (4.72E-04)
Best	3	16	6	3

The better average values of these algorithms for each instance are given in bold and italics.

on most of the comparisons, i.e., in 24 out of 28 cases for IGD results, while other three competitors are not able to obtain a good performance. In addition, we can also observe that with the increase of variances used in competitors, the performance becomes worse. The reason for this observation may be that if we adopt a too larger variance in the whole evolutionary process, the offspring will become irregular leading to the poor convergence. Therefore, it is not surprising that the performance with bigger variances will degrade. It should be noted that the results from Table 5 also explain why we only choose Gaussian distributions from $(0, 0.6^2)$ to $(0, 1.4^2)$ for MOEA/D-AMG.

4.6. Parameter Analysis. In the above comparison, our algorithm MOEA/D-AMG is initialized with $K=5$, including five types of Gaussian models, i.e., $(0, 0.6^2)$, $(0, 0.8^2)$, $(0, 1.0^2)$, $(0, 1.2^2)$, and $(0, 1.4^2)$. In this section, we test the proposed algorithm using different K values (3, 5, 7, and 10), which will use K kinds of Gaussian distributions. To clarify the experimental setting, the standard deviations of Gaussian models adopted by each K value are listed in Table 6. As shown by the obtained IGD results presented in Table 7, it is clear that $K=5$ is a preferable number of Gaussian models for MOEA/D-AMG since it achieves the best IGD results in 16 of 28 cases. It also can be found that as K increases, the

average IGD values also increase for most of test problems. This is due to the fact that in our algorithm MOEA/D-AMG, the population size is often relatively small, which makes little sense to set a large K value. Thus, a large K value has little effect on improving the performance of the algorithm. However, $K=3$ seems to be too small, which is not able to take full advantage of the proposed adaptive multiple Gaussian process models. According to the above analysis, we recommend that $K=5$.

5. Conclusions and Future Work

In this paper, a decomposition-based multiobjective evolutionary algorithm with adaptive multiple Gaussian process models (called MOEA/D-AMG) has been proposed for solving MOPs. Multiple Gaussian process models are used in MOEA/D-AMG, which can help to solve various kinds of MOPs. In order to enhance the search capability, an adaptive strategy is developed to select a more suitable Gaussian process model, which is determined based on the contributions to the optimization performance for all the decomposed subproblems. To investigate the performance of MOEA/D-AMG, twenty-eight test MOPs with complicated PF shapes are adopted, and the experiments show that MOEA/D-AMG has superior advantages on most cases when compared to six competitive MOEAs. In addition,

other experiments also have verified the effectiveness of the used adaptive strategy to significantly improve the performance of MOEA/D-AMG.

When comparing to generic recombination operators and other Gaussian process-based recombination operators, our proposed method based on multiple Gaussian process models is effective to solve most of the test problems. However, in MOEA/D-AMG, the number of Gaussian models built in each generation is the same as the size of subproblems, which still needs a lot of computational cost. In our future work, we will try to enhance the computational efficiency of the proposed MOEA/D-AMG, and it is also interesting to study the potential application of MOEA/D-AMG in solving some many-objective optimization problems and engineering problems.

Data Availability

The source code and data are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Natural Science Foundation of China under grant nos. 61876110, 61836005, and 61672358, Joint Funds of the National Natural Science Foundation of China under Key Program Grant U1713212, Natural Science Foundation of Guangdong Province under grant no. 2017A030313338, and Shenzhen Technology Plan under grant no. JCYJ20170817102218122. This study was also supported by the National Engineering Laboratory for Big Data System Computing Technology and the Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen University.

References

- [1] S. JD, "Multiple objective optimization with vector valued genetic algorithms," in *Proceedings of the First International Conference on Genetic Algorithms and their Applications*, pp. 93–100, 1985.
- [2] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, vol. 16, John Wiley & Sons, New York, NY, USA, 2001.
- [3] C. A. Coello, "An updated survey of GA-based multiobjective optimization techniques," *ACM Computing Surveys*, vol. 32, no. 2, pp. 109–143, 2000.
- [4] 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 and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.
- [5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [6] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: improving the strength Pareto evolutionary algorithm," *TIK-report*, vol. 103, 2001.
- [7] J. D. Knowles and D. W. Corne, "Approximating the non-dominated front using the Pareto archived evolution strategy," *Evolutionary Computation*, vol. 8, no. 2, pp. 149–172, 2000.
- [8] J. Bader and E. Zitzler, "HypE: an algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.
- [9] Q. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [10] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evolutionary Computation*, vol. 10, no. 3, pp. 263–282, 2002.
- [11] D. Hadka and P. Reed, "Borg: an auto-adaptive many-objective evolutionary computing framework," *Evolutionary Computation*, vol. 21, no. 2, pp. 231–259, 2013.
- [12] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 721–736, 2013.
- [13] G. Wang and H. Jiang, "Fuzzy-dominance and its application in evolutionary many objective optimization," in *Proceedings of the International Conference on Computational Intelligence and Security Workshops*, pp. 195–198, Heilongjiang, China, 2007.
- [14] F. di Pierro, S.-T. Khu, and D. A. Savi, "An investigation on preference order ranking scheme for multiobjective evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 1, pp. 17–45, 2007.
- [15] C. Zhu, L. Xu, and E. D. Goodman, "Generalization of Pareto-optimality for many-objective evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 2, pp. 299–315, 2016.
- [16] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.
- [17] 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 Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.
- [18] Y. Liu, D. Gong, X. Sun, and Y. Zhang, "Many-objective evolutionary optimization based on reference points," *Applied Soft Computing*, vol. 50, pp. 344–355, 2017.
- [19] S. Jiang and S. Yang, "A strength Pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 3, pp. 329–346, 2017.
- [20] L. Pan, C. He, Y. Tian, Y. Su, and X. Zhang, "A region division based diversity maintaining approach for many-objective optimization," *Integrated Computer-Aided Engineering*, vol. 24, no. 3, pp. 279–296, 2017.
- [21] W. Lin, Q. Lin, Z. Zhu, J. Li, J. Chen, and Z. Ming, "Evolutionary search with multiple utopian reference points in decomposition-based multiobjective optimization," *Complexity*, vol. 2019, Article ID 7436712, 22 pages, 2019.
- [22] C. Dai and X. Lei, "A multiobjective brain storm optimization algorithm based on decomposition," *Complexity*, vol. 2019, p. 11, 2019.

- [23] Y. Su, Q. Lin, J. Wang, J. Li, J. Chen, and Z. Ming, "A constrained solution update strategy for multiobjective evolutionary algorithm based on decomposition," *Complexity*, vol. 2019, Article ID 3251349, 11 pages, 2019.
- [24] X. Wang, K. Xing, C. Yan, and M. Zhou, "A novel MOEA/D for multiobjective scheduling of flexible manufacturing systems," *Complexity*, vol. 2019, Article ID 5734149, 14 pages, 2019.
- [25] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," *Lecture Notes in Computer Science*, Springer, in *Proceedings of the International Conference on Parallel Problem Solving from Nature*, pp. 832–842, Springer, Birmingham, UK, 2004.
- [26] R. H. Gómez and C. A. C. Coello, "Improved metaheuristic based on the R2 indicator for many-objective optimization," in *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pp. 679–686, Madrid, Spain, 2015.
- [27] A. Menchaca-Mendez and C. A. C. Coello, "GD-MOEA: a new multiobjective evolutionary algorithm based on the generational distance indicator," in *Proceedings of the 8th International Conference on Evolutionary Multi-Criterion Optimization*, pp. 156–170, Springer, Guimarães, Portugal, 2015.
- [28] D. H. Phan and J. Suzuki, "R2-IBEA: R2 indicator based evolutionary algorithm for multiobjective optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '13)*, pp. 1836–1845, Cancun, Mexico, June 2013.
- [29] C. A. R. Villalobos and C. A. C. Coello, "A new multi-objective evolutionary algorithm based on a performance assessment indicator," in *Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference-GECCO '12*, pp. 505–512, ACM, Philadelphia, PA, USA, 2012.
- [30] K. Deb, A. Sinha, and S. Kukkonen, "Multi-objective test problems, linkages, and evolutionary methodologies," in *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pp. 1141–1148, ACM, Seattle, WA, USA, 2006.
- [31] Q. Zhu, Q. Lin, Z. Du et al., "A novel adaptive hybrid crossover operator for multiobjective evolutionary algorithm," *Information Sciences*, vol. 345, pp. 177–198, 2016.
- [32] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 2, pp. 115–148, 1995.
- [33] K. Deep and M. Thakur, "A new crossover operator for real coded genetic algorithms," *Applied Mathematics and Computation*, vol. 188, no. 1, pp. 895–911, 2007.
- [34] K. Deb, A. Anand, and D. Joshi, "A computationally efficient evolutionary algorithm for real-parameter optimization," *Evolutionary Computation*, vol. 10, no. 4, pp. 371–395, 2002.
- [35] L. J. Eshelman and J. D. Schaffer, "Real-coded genetic algorithms and interval-schemata," in *Foundations of Genetic Algorithms*, vol. 2, pp. 187–202, Elsevier, Amsterdam, Netherlands, 1993.
- [36] I. Ono, H. Kita, and S. Kobayashi, "A robust real-coded genetic algorithm using unimodal normal distribution crossover augmented by uniform crossover: effects of self-adaptation of crossover probabilities," in *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1*, vol. 1, pp. 496–503, Morgan Kaufmann Publishers Inc., Orlando, FL, USA, 1999.
- [37] S. Tsutsui, M. Yamamura, and T. Higuchi, "Multi-parent recombination with simplex crossover in real coded genetic algorithms," in *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1*, vol. 1, pp. 657–664, Morgan Kaufmann Publishers Inc., Orlando, FL, USA, 1999.
- [38] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [39] X. He, Y. Zhou, and Z. Chen, "An evolution path based reproduction operator for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 1, pp. 29–43, 2017.
- [40] J. Wang, G. Liang, and J. Zhang, "Cooperative differential evolution framework for constrained multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 10, no. 11, pp. 1–13, 2018.
- [41] K. Li and H. Tian, "Adaptive differential evolution with evolution memory for multiobjective optimization," *IEEE Access*, vol. 11, no. 17, pp. 866–876, 2018.
- [42] K. Li, A. Fialho, S. Kwong, and Q. Zhang, "Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 114–130, 2014.
- [43] Q. Lin, Z. Liu, Q. Yan et al., "Adaptive composite operator selection and parameter control for multiobjective evolutionary algorithm," *Information Sciences*, vol. 339, pp. 332–352, 2016.
- [44] X. Qiu, J. X. Xu, K. C. Tan, and H. A. Abbass, "Adaptive cross-generation differential evolution operators for multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 2, pp. 232–244, 2015.
- [45] Y. Sun, G. G. Yen, and Z. Yi, "Improved regularity model-based EDA for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 5, pp. 662–678, 2018.
- [46] A. Zhou, Q. Zhang, and Y. Jin, "Approximating the set of Pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1167–1189, 2009.
- [47] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: a regularity model-based multiobjective estimation of distribution algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 41–63, 2008.
- [48] Q. Zhang, A. Zhou, S. Zhao, and P. N. Suganthan, "Multi-objective optimization test instances for the CEC 2009 special session and competition," *Special Session on Performance Assessment of Multi-Objective Optimization Algorithms*, Technical report, vol. 264, University of Essex, Colchester, UK, 2008.
- [49] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.
- [50] T. Lin, H. Zhang, K. Zhang, Z. Tu, and N. Cui, "An adaptive multiobjective estimation of distribution algorithm with a novel Gaussian sampling strategy," *Soft Computing*, vol. 21, no. 20, pp. 6043–6061, 2017.
- [51] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [52] Y. Z. G. Dai, L. Peng, and M. Wang, "HMOEDA_LLE: a hybrid multi-objective estimation of distribution algorithm combining locally linear embedding," *IEEE Congress on Evolutionary Computation*, pp. 707–714, 2014.

- [53] R. Cheng, Y. Jin, K. Narukawa, and B. Sendhoff, "A multi-objective evolutionary algorithm using Gaussian process-based inverse modeling," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 838–856, 2015.
- [54] Y. Lin, H. Liu, and Q. Jiang, "Dynamic reference vectors and biased crossover use for inverse model based evolutionary multi-objective optimization with irregular Pareto fronts," *Applied Intelligence*, vol. 48, no. 9, pp. 3116–3142, 2018.
- [55] Q. Yuan, G. Dai, and Y. Zhang, "A novel multi-objective evolutionary algorithm based on LLE manifold learning," *Engineering with Computers*, vol. 33, no. 2, pp. 293–305, 2017.
- [56] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.
- [57] A. Zhou, Q. Zhang, and G. Zhang, "A multiobjective evolutionary algorithm based on decomposition and probability model," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 1–8, Brisbane, Australia, June 2012.
- [58] C. Liu, D. Zhou, Z. Wang, D. Yang, and G. Song, "Damage detection of refractory based on principle component analysis and Gaussian mixture model," *Complexity*, vol. 2018, Article ID 7356189, 9 pages, 2018.
- [59] M. Lungaroni, A. Murari, E. Peluso, P. Gaudio, and M. Gelfusa, "Geodesic distance on Gaussian manifolds to reduce the statistical errors in the investigation of complex systems," *Complexity*, vol. 2019, Article ID 5986562, 24 pages, 2019.
- [60] J. Li, B. Zhang, G. Lu, H. Ren, and D. Zhang, "Visual classification with multikernel shared Gaussian process latent variable model," *IEEE Transactions on Cybernetics*, vol. 49, no. 8, pp. 2886–2899, 2018.
- [61] Y. Zhou, J. He, and H. Gu, "Partial label learning via Gaussian processes," *IEEE Transactions on Cybernetics*, vol. 47, no. 12, pp. 4443–4450, 2017.
- [62] M. C. Seiler and F. A. Seiler, "Numerical recipes in C: the art of scientific computing," *Risk Analysis*, vol. 9, no. 3, pp. 415–416, 1989.
- [63] P. A. N. Bosman and D. Thierens, "Multi-objective optimization with diversity preserving mixture-based iterated density estimation evolutionary algorithms," *International Journal of Approximate Reasoning*, vol. 31, no. 3, pp. 259–289, 2002.
- [64] A. Zhou, Q. Zhang, and G. Zhang, "Multiobjective evolutionary algorithm based on mixture Gaussian models," *Journal of Software*, vol. 25, no. 5, pp. 913–928, 2014.
- [65] M. Dellnitz, O. Schütze, and T. Hestermeyer, "Covering Pareto sets by multilevel subdivision techniques," *Journal of Optimization Theory and Applications*, vol. 124, no. 1, pp. 113–136, 2005.
- [66] P. Larránaga and J. A. Lozano, Eds., *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, Kluwer Academic Publishers, vol. 2, Boston, MA, USA, 2002.
- [67] P. A. N. Bosman and D. Thierens, "Continuous iterated density estimation evolutionary algorithms within the idea framework," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2000)*, pp. 197–200, Las Vegas, NV, USA, July 2000.
- [68] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.
- [69] C. Igel, N. Hansen, and S. Roth, "Covariance matrix adaptation for multi-objective optimization," *Evolutionary Computation*, vol. 15, no. 1, pp. 1–28, 2007.
- [70] T. Paul and H. Iba, "Real-coded estimation of distribution algorithm," in *Proceedings of the 5th Metaheuristics International Conference*, pp. 61–66, Kyoto, Japan, 2003.
- [71] M. R. Wagner, A. Auger, and M. Schoenauer, "EEDA: a new robust estimation of distribution algorithm," Rapport de Recherche (Research Report) RR-5190, Institut National de Recherche en Informatique et en Automatique (INRIA), Rocquencourt, France, 2004.
- [72] W. Dong and X. Yao, "Unified eigen analysis on multivariate Gaussian based estimation of distribution algorithms," *Information Sciences*, vol. 178, no. 15, pp. 3000–3023, 2008.
- [73] P. A. N. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 174–188, 2003.
- [74] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.