SPECIAL SECTION ON RECENT COMPUTATIONAL METHODS IN KNOWLEDGE ENGINEERING AND INTELLIGENCE COMPUTATION

Received March 23, 2018, accepted April 18, 2018, date of publication April 23, 2018, date of current version May 16, 2018.

*Digital Object Identifier 10.1109/ACCESS.2018.2829262*

# A Multi-Objective Genetic Algorithm Based on Fitting and Interpolation

**CHUANG HAN, LING WANG, ZHAOLIN ZHANG, JIAN XIE, AND ZIJIAN XING**
School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072, China

Corresponding author: Chuang Han (hanchuang357@163.com)

This work was supported by the National Natural Science Foundation of China under Grant 61771404, Grant 61601372, and Grant 61301093.

**ABSTRACT** Considering the diversity of uniform distribution for the solutions of multi-objective optimization problems, we propose the multi-objective genetic algorithm based on fitting (MOGA/F) and interpolation (MOGA/I). The selected operator is based on the optimal reference points uniformly distributed in the objective space, which is calculated by applying a fitting function or interpolation method from a finite set of objective values. After sorting the ranks of the population, the objective space for the last front can be easily calculated by using fitting and interpolation functions, and the uniformly distributed points can be obtained without parameter setting. The individuals with the shortest Euclidean distance to the reference points are chosen according to the error matrix. This method can maintain the diversity and spread of the solutions without destroying the convergence. In this paper, MOGA/F and MOGA/I are compared with the traditional methods, non-dominated sorting genetic algorithm-II and multi-objective evolutionary algorithm based on decomposition, by optimizing the mathematical problems. The numerical examples show that MOGA/F and MOGA/I have a much higher performance in terms of diversity and convergence of the final solutions.

**INDEX TERMS** Multi-objective optimization, diversity, fitting, interpolation, genetic algorithm.

## I. INTRODUCTION

Multi-objective evolutionary algorithms (MOEAs) have been widely used in optimization problems, and their results are mostly capable of solving real-world problems [1]–[8]. With the increasing quality of solutions, diversity assessment and convergence performance have become the primary focus in the evolutionary process. The balance between these two aspects has been deeply and broadly discussed. To maintain the performance in terms of diversity and convergence, three classes of MOEAs have been used to solve multi-objective optimization problems (MOPs).

The non-dominated sorting genetic algorithm-II (NSGA-II) has a high performance for the MOPs based on dominance [9], [10]. NSGA-II computes the crowding distance for every last level member as the summation of the objective-wise normalized distance between two neighboring solutions. Two mechanisms for managing diversity, namely, management of diversity promotion during the selection of individuals and mutation range adjustment for each decision variable, are introduced and hybridized with NSGA-II in [11]. $\varepsilon$-MOEA [12] is a steady-state algorithm based on the $\varepsilon$-dominance relation. It is designed to replace Pareto dominance and divides the objective space into hyperboxes by a size of $\varepsilon$. This class of MOEAs can promote the diversity of solutions to approximate the front. However, the spread of the population and the uniform diversity cannot be improved [13].

Another widely used method is the multi-objective evolutionary algorithm based on decomposition (MOEA/D) [14], [32], which decomposes an MOP into a number of scalar optimization subproblems that are then optimized simultaneously. The diversity of the population is maintained by using a group of well-distributed weight vectors or reference points. In the evolutionary process, each weight vector guides a different population toward a different position in the true Pareto front (PF) [15], [16]. Reference points are generated by applying uniform systematic sampling [17] and weight vectors with uniform distribution over the predefined design space [18]–[20]. These methods are emended to MOEA/D separately to sort the solutions for each subproblem to uniformly converge to the PF. Further, the objective space is decomposed into a set of sub-objective spaces, and three grid-based criteria (i.e., grid ranking, grid crowding distance, and grid coordinate point distance) are incorporated into the fitness of individuals to distinguish them in both mating and environmental selection processes [16], [21]. The power of individual selection can be defined as the perpendicular distance [18] or Hamiltonian path [22] from the individual to the reference point or weight vector in the objective space. However, the performance of this kind of algorithm is highly dependent on the chosen parameters. The difficulties for the number of weight vectors, neighborhood size, and the choice of aggregation method, make it harder to obtain a good convergence performance with this algorithm. Meanwhile, having well-distributed parameters does not ensure that their corresponding optimal solutions are also well distributed [23].

In some MOEAs, diversity becomes the primary focus in the evolutionary process. Multiple well-distributed predefined reference points are generated on the hyperplane in NSGA-III [24]. NSGA-III uses a predefined set of reference points to ensure diversity in obtained solutions. The chosen reference points can either be predefined in a structured manner or supplied preferentially by the user. But these points are uniformly distributed in every axis instead of the PF. Many algorithms apply modified quality indicators to directly assign each individual a fitness value, which should simultaneously reflect the convergence and diversity performance of individuals [25]–[27]. In this class of MOEAs, convergence is ensured by Pareto dominance or other approaches that move the individuals into a more crowded area [24]–[30].

The improved genetic algorithm (IGA) is proposed for the array pattern synthesis [31]. IGA makes three modifications, which has a high performance just for the special optimization of array pattern synthesis but not all the MOPs. The front uniformly distributed strategy can find the optimal solution with a good diversity [31]. The whole length of the front is simply calculated by Euclidean distance, and then divided by the expected number. There are no more advanced methods, such as fitting or interpolation, for calculating the distribution of population. Once the optimal reference points are found, the error-comparison operator mentioned in [31] is used for choosing the selected individuals in this paper.

In this paper, we define the optimal reference points uniformly distributed in the objective space, which is calculated by applying a fitting function or interpolation method from a finite set of objective function values. After sorting the ranks of the population, the objective space for the last front can be easily calculated, and the uniformly distributed points can be found without parameter setting. The solutions with the shortest Euclidean distance to the reference points are chosen according to the error matrix. This method can maintain the diversity and spread of the solutions without destroying the convergence.

The remaining part of this paper is organized as follows. In Section II, the multi-objective optimization problems are described. The multi-objective genetic algorithms based on fitting and interpolation (MOGA/F and MOGA/I) are proposed Section III. Section IV presents the metric used to evaluate the performance of optimal solutions and discusses the comparative performance of the proposed technique by simulating numerical examples. The conclusions are given in Section V.

## II. MULTI-OBJECTIVE OPTIMIZATION PROBLEMS

The multi-objective optimization problem can be expressed as

$$\begin{aligned} &\min\ F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x}))^{\mathrm{T}} \\ &\text{subject to } \mathbf{x} = (x_1, x_2, \ldots, x_n) \in \Omega \end{aligned} \tag{1}$$

where $\mathbf{x}$ is the individuals of optimized parameters, and $\Omega$ is the constrained region for $\mathbf{x}$. In (1), the parameter $M$ is the size of the objective functions, and $n$ is the number of optimized parameter in one individual.

Every objective in MOPs contradicts each other, so we cannot find one individual in the constrained region to minimize all the objectives simultaneously. Hence, the main goal for MOEAs is to find a set of solutions, which considers all the objectives in the process of optimization. Pareto optimality can have the best balance between all the objectives.

In (1), the MOP is a minimization problem, which aims to enable every objective to have a minimum value. Here $\mathbf{u}, \mathbf{v} \in R^M$, $\mathbf{u}$ dominates $\mathbf{v}$ (shown as $\mathbf{u} \succ \mathbf{v}$) only if $u_i \leq v_i$ for every $i \in \{1, \ldots, M\}$ and $u_i < v_i$ not less than one objective. If there is no one individual $\mathbf{x} \in \Omega$ that can make $F(\mathbf{x}) \succ F(\mathbf{x}^*)$, then the individual $\mathbf{x}^* \in \Omega$ is one Pareto optimal solution for the MOP of (1). The value of objective function $F(\mathbf{x}^*)$ is defined as a Pareto objective vector. There are no other individuals which can make all the objective values of the Pareto optimal solution better. The set of all the Pareto optimal solutions is defined as the Pareto set (PS), and the set of all the Pareto optimal objective vectors is the PF [31], [32]. The MOEAs aim to find the optimum solutions approximating the true PF.

In many real-life applications of multi-objective optimization, an approximation to the PF is required by a decision maker for selecting a final preferred solution. Most MOPs may have many or even infinite Pareto optimal vectors. It is very time-consuming, if not impossible, to obtain the complete PF. On the other hand, the decision maker may not be interested in having an unduly large number of Pareto optimal vectors to deal with due to overflow of information. Therefore, many multi-objective optimization algorithms are to find a manageable number of Pareto optimal vectors which are evenly distributed along the PF, and thus good representatives of the entire PF [32].

Unlike single objective problems, for which the PF is but a single point, PFs for multi-objective problems can have a wide variety of geometries [33], [34]. In this section, we review the ZDT test suite of Zitzler *et al.* [35]. This suite of six test problems is perhaps the most widely applied suite of benchmark multi-objective problems in the EA literature. It should also be noted that ZDT4 only uses one parameter of dissimilar domain; that is, the single position parameter $x_1$ has domain [0,1], whereas all other parameters have domain [5,5]. The PFs of its problems are well defined, and the test results from various other studies are commonly available, thus facilitating comparisons with new algorithms.

In this paper, ZDT1, ZDT2, and ZDT4 test instances are used to compare the proposed methods with MOEA/D and NSGA-II [10], one of the most successful non-decomposition MOEAs. All these test instances are minimizations of the objectives.

- ZDT1

$f_1(x) = x_1$
$f_2(x) = g(x)[1 - \sqrt{f_1(x)/g(x)}]$
$g(x) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$

and $x = (x_1, \ldots, x_n)^T \in [0,1]^n$. The PF is convex, and $n = 30$ is chosen for our experiments.

- ZDT2

$f_1(x) = x_1$
$f_2(x) = g(x)[1 - (f_1(x)/g(x))^2]$

where $g(x)$ and the range and dimensionality of $x$ are the same as in ZDT1. The PF is non-convex.

- ZDT4

$f_1(x) = x_1$
$f_2(x) = g(x)[1 - \sqrt{f_1(x)/g(x)}]$

where $g(x) = 1 + 10(n-1) + \sum_{i=2}^{n}[x_i^2 - 10\cos(4\pi x_i)]$ and $x = (x_1, \ldots, x_n)^T \in [0,1]^n \times [-5,5]^{n-1}$. This test instance has many local PFs, and $n = 10$ is chosen for our experiments.

## III. MULTI-OBJECTIVE GENETIC ALGORITHM BASED ON FITTING AND INTERPOLATION

In practice engineering application, only some discrete points can be observed, and we need to find several functions with established parameters to approximate these points. There are two common methods of interpolation and fitting used to achieve this [36]–[39]. We can get one proximate curve for two-dimensional data or a hook face for three-dimensional data. For high-dimensional data, the suitable functions and parameters can also be defined. For the method of interpolation, the proximate function can go through each point; for the method of fitting, the proximate function can well present the law of the points.

Proximity to the front and diversity of solutions within the approximation set are important requirements for MOPs. For the objective function value space, the solutions are the finite set of points. Based on the fitting and interpolation function, we can know the distributed situation of the population in current front. The expected points with uniform distribution can be directly calculated by applying the fitting and interpolation function. Then, the optimal individuals can be easily found by using the error-comparison operator. The concrete process is presented in Algorithm 1.

First, the population should be ranked by nondominance comparison, and those with lower ranks are copied into the new population until its size equals $N$. The population in the front with the last rank is a non-dominated set $P$, and the number of individuals is $L$. Every individual in $P$ can be assigned an expected error by using error-comparison operator. The new population will copy the individuals with smaller expected error to the uniformly distributed points until the size of the new population reaches $N$. The values of objective function can be expressed as $\mathbf{f}_L = (f_1, \ldots, f_m, \ldots, f_M)_L$ for $P$. Here, the number of uniformly distributed points is set as $K$. $K$ should not exceed $N$ for a good approximation to the PF, and the constraint of the size of population. $K$ can be shown as

$$K = \begin{cases} L & if\ L \leq N \\ N & if\ L > N \end{cases} \tag{2}$$

**Algorithm 1** The Procedure of MOGA/I and MOGA/F

**Input:** MOP, the maximum number of generation $MaxGen$, the population size $N$
**Output:** The optimized solution $EP$

Set the generation number $t = 0$, $i = 1$, $j = 1$;
Generate an initial population $P_0$ with size $N$.
**if** $t = MaxGen$ **then**
  $EP = P_{t+1}$ and break
**else**
  $Q_t$ = Elitist-preserving+Recombination+Mutation $(P_t)$
  $R_t = P_t \cup Q_t$
  $(F_1, F_2, \ldots)$ = Non-dominated-sort $(R_t)$
  **repeat**
    $S_t = S_t \cup F_i$ and $i = i + 1$
  **until** $|S_t| \geq N$
  Last front to be included: $F_l = F_i$
  **if** $|S_t| = N$ **then**
    $P_{t+1} = S_t$
  **else**
    $P_{t+1} = \cup_{i=1}^{l-1} F_j$
    Calculate the number of individuals in $F_l$: $L = |F_l|$
    Points uniformly distributed in the front: $K = L$ **if** $L \leq N$ **else** $K = N$
    The expected points $\hat{\mathbf{f}}_K$ = Interpolation or Fitting $(F_l, K)$
    Assign expected error order for every individuals in $F_l$: $(f_1, f_2, \ldots)$= Error-comparison operator $(F_l, \hat{\mathbf{f}}_K)$
    **repeat**
      $P_{t+1} = P_{t+1} \cup f_j$ and $j = j + 1$
    **until** $|P_{t+1}| = N$
  **end if**
  $t = t + 1$
**end if**

For the method of interpolation, there are several kinds of interpolation functions, such as Lagrange, Piecewise Cubic Hermite Interpolation Polynomial (PCHIP), Cubic Spline, Piecewise Linear (PL), and Fractal Interpolation (FI) [40], [41]. In this paper, PL, PCHIP, and FI are used to obtain the interpolation function according to the population in the current front. The interpolation function can be shown as

$$\mathrm{H}_{\mathrm{I}}(\mathbf{f}_L, M) = \mathrm{I}([f_1, f_2, \ldots, f_M]_L) \tag{3}$$

In the interpolation process, every two points should carry out one operator on the basis of the sorted points. Because the operator of interpolation is behind the ranking operator, in the current front, other objective values will decrease gradually with the increasing values of one objective. Therefore, the $M$-th objective values are sorted, and the interpolation function can be calculated. For the interpolation methods, the higher the number of interpolation points, the better the approximate performance.

When the approximate function for the population is known, the uniformly distributed points can be calculated directly. The uniformly results can be written as

$$\hat{\varsigma}_{\mathrm{I}} = \frac{\iint\limits_{[\mathbf{f}_1, \mathbf{f}_L]} \ldots \int \mathrm{H}_{\mathrm{I}}(\mathbf{f}_L, M)\mathrm{d}\mathbf{f}}{K-1} \tag{4}$$

The integral can be understood as the whole length of the curve for the two-dimensional objective space and as the whole area of the hook face for the three-dimensional objective space. After determining the global distribution of the population, the expected points $\hat{\mathbf{f}}_K = (\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_M)_K$ can be calculated according to the uniformly results $\hat{\varsigma}_{\mathrm{I}}$, which are uniformly divided by the average length or area.

For the method of fitting, one polynomial with undetermined coefficients can approximate the discrete points. The fitting function can be defined by the user or matched with the characteristic of the points. There are several common fitting functions, such as Exponential, Fourier, Gaussian, Polynomial, Power, and Rational. One fitting function can be presented as

$$\begin{aligned}\mathrm{H}_{\mathrm{F}}&(f_1, f_2, \ldots, f_{M-1}) \\ &= a_1 r_1(f_1, f_2, \ldots, f_{M-1}) + a_2 r_2(f_1, f_2, \ldots, f_{M-1}) \\ &\quad + \ldots + a_Q r_Q(f_1, f_2, \ldots, f_{M-1})\end{aligned} \tag{5}$$

where $r_1, r_2, \ldots, r_Q$ refers to the designed functions with the number $Q < L$, and $\mathbf{a} = (a_1, a_2, \ldots, a_Q)^{\mathrm{T}}$ denotes the undetermined coefficients, which are calculated by using the least squares criterion as

$$\begin{aligned}J&(a_1, a_2, \ldots, a_Q) \\ &= \sum_{l=1}^{L} \delta_i^2 = \sum_{l=1}^{L} [\mathrm{H}_{\mathrm{F}}(f_1^l, f_2^l, \ldots, f_{M-1}^l) - f_M^l]^2 \\ &= \sum_{l=1}^{L} [\sum_{q=1}^{Q} a_q r_q(f_1^l, f_2^l, \ldots, f_{M-1}^l) - f_M^l]^2\end{aligned} \tag{6}$$

The undetermined coefficients can be obtained by minimizing $J(a_1, a_2, \ldots, a_Q)$. Then, $\mathbf{a}$ can be calculated by matrix left division as

$$\begin{aligned}\mathbf{a} = [&r_1(f_1^l, f_2^l, \ldots, f_{M-1}^l), r_1(f_1^l, f_2^l, \ldots, f_{M-1}^l), \\ &\ldots, r_1(f_1^l, f_2^l, \ldots, f_{M-1}^l)]_L \backslash [f_M^l]_L\end{aligned} \tag{7}$$

The uniformly results from the fitting function can be written as

$$\hat{\varsigma}_{\mathrm{F}} = \frac{\iint\limits_{\Upsilon} \ldots \int \mathrm{H}_{\mathrm{F}}(f_1, f_2, \ldots, f_{M-1})\mathrm{d}(f_1, f_2, \ldots, f_{M-1})}{K-1} \tag{8}$$

where $\Upsilon = [(f_1^1, f_2^1, \ldots, f_{M-1}^1), (f_1^L, f_2^L, \ldots, f_{M-1}^L)]$. Thus, the uniformly distributed points $\hat{\mathbf{f}}_K$ in the fitting function $\mathrm{H}_{\mathrm{F}}$ can be found according to the uniformly results $\hat{\varsigma}_{\mathrm{F}}$. The process of interpolation and fitting is presented in Algorithm 2.

**Algorithm 2** The Procedure of Interpolation and Fitting

**Input:** The population in the last front $F_l$, the number of expected points $K$
**Output:** The uniformly distributed points $\hat{\mathbf{f}}_K$
1: Calculate the number of individuals in $F_l$: $L = |F_l|$
2: Calculate the objective value of the population $\mathbf{f}_L$
3: Calculate the interpolation function or the fitting funtion according to $\mathbf{f}_L$: $\mathrm{H}_{\mathrm{I}}(\mathbf{f}_L, M)$ or $\mathrm{H}_{\mathrm{F}}(\mathbf{f}_L, M)$
4: Divide the integral of interpolation function or fitting funtion by $K-1$
5: The uniformly distributed points $\hat{\mathbf{f}}_K$ can be obtained

In order to find the best solution uniformly distributed in the PF, the error-comparison operator mentioned in [31] is used in this paper. The expected error order between the expected individuals $\hat{\mathbf{f}}_K$ and the individuals in the last front should be assigned. After calculating the expected individuals $\hat{\mathbf{f}}_K$ by applying the interpolation or fitting function, every individual in the last front can be identified with an order. The distances between $\hat{\mathbf{f}}_K$ and each individual in $P$ can be obtained by calculating Euclidean distance. Then the orders for every individual can be determined by the distances, and the individuals with smaller orders will be copied to the new population.

Once the $\hat{\mathbf{f}}_K$ is obtained, the expected error between the $i$-th element in $P$ and the $j$-th element in $\hat{\mathbf{f}}_K$ can be defined as

$$\begin{aligned}\hat{\Delta}_{ij} &= \mathrm{E}[\mathbf{f}_i, \hat{\mathbf{f}}_j] \\ &= \sqrt{(f_1^i - \hat{f}_1^j)^2 + (f_2^i - \hat{f}_2^j)^2 + \ldots + (f_M^i - \hat{f}_M^j)^2}\end{aligned} \tag{9}$$

According to (9), we can calculate all the expected errors, which can make up an expected error matrix $\hat{R}$ with $L$ rows and $K$ columns. The expected error order will be calculated by using the error-comparison operator [31]. After that, $K$ individuals with the lower values in the last front can be selected.

The procedure of error-comparison operator is shown in Algorithm 3. In the $t$-th selection, the minimum value $\hat{\sigma}_t$ is located in $\hat{R}$. Then the expected error order of individual in $P$ associated with the row of $\hat{\sigma}_t$ is set as $t$. At the same time,

**Algorithm 3** The Procedure of Error-Comparison Operator

**Input:** The population in the last front $F_l$, the uniformly distributed points $\hat{\mathbf{f}}_K$, the number of expected points $K$

**Output:** The individuals in $F_l$ with expected error order: $(f_1, f_2, \ldots)$

1: Calculate the number of individuals in $F_l$: $L = |F_l|$
2: Calculate the objective value $\hat{\mathbf{f}}_L$ of the population $\mathbf{f}_L$
3: **for** $i = 1$ **to** $L$ **do**
4: **for** $j = 1$ **to** $K$ **do**
5: Calculate the expected error: $\hat{\Delta}_{ij} = \mathrm{E}[\mathbf{f}_i, \hat{\mathbf{f}}_j]$
6: **end for**
7: **end for**
8: The expected error matrix $\hat{R}$ is constructed by $\hat{\Delta}_{ij}$
9: **for** $t = 1$ **to** $K$
10: Find the smallest value $\hat{\sigma}_t$ in $\hat{R}$ with row number of $m$ and column number of $n$
11: Assign the $m$th individual in $F_l$ with expected error order of $t$: $f_t = \{F_l(m), t\}$
12: Assign the $m$th row and $n$the column in $\hat{R}$ with infinite values: $\hat{R}(m, :) = \inf$, and $\hat{R}(:, n) = \inf$
13: **end for**

the elements in the associated row and column in the expected error matrix $\hat{R}$ must be set as infinite values, to prevent the individual from being repeatedly selected.

In the comparison process, every individual in $P$ can be assigned an expected error value from only one nearest expected position in $\hat{\mathbf{f}}_K$. The individuals on the edge of distributed space are definitely selected because the expected error values, $\hat{\Delta}_{11}$ and $\hat{\Delta}_{lk}$, are the smallest.

Both MOGA/F and MOGA/I can improve the diversity of the final solution. The fitting function and interpolation function aim to express the distribution character of the population in the last front. The uniformly distributed points can be achieved by these two functions. The main difference between the two methods is whether the function passes through each individual. MOGA/F uses fitting function which just reflects the distribution character of the population. The type of fitting function has a great influence on the optimization result. The convergence is reduced to some extent by using MOGA/F. On the contrary, interpolation function passes every individual in the last front. This method has a fast convergence by different types of interpolation functions.

## IV. SIMULATION RESULTS

### A. EVALUATION METRICS

There should be some metrics that have the abilities to evaluate the quality of the population. In many MOPs, the diversity and the convergence of the optimized solution are the main concern for the performance of the algorithm. In [32], the $D$-metric measures the error sum between the optimized solution and the expected points uniformly distributed in the PF. In this metric, the size of the expected positions is much large than $N$, and there is no constraint on the one-to-one correspondence between the desired position and the individual in the final solution. As a result, the smaller value of $D$-value cannot guarantee the results with a good uniformity.

Here, we define a metric of expectation value ($E$-metric) [31] to show the quality of the optimized results at two areas: convergence to the PS and maintenance of diversity in the solutions of the PS. For mathematical optimization problems, the true PFs have been previously described in detail. The PFs for each instance can be directly used in the result analysis, so the expected positions can be easily calculated for the $E$-metric. The uniformly distributed points $\mathbf{f}_N$ with number of $N$ can be found from the true PF in the objective space. Then, the minimum Euclidean distances of the solutions from the chosen solutions in the PF can be calculated by finding the minimum element $\sigma_t$ in the error matrix $R$.

The optimal solution MOEAs desire is exactly the same as the expected points $\mathbf{f}_K$, which is uniformly distributed in the PF. In order to consider the diversity and convergence of the final solution $Q^*$, all the individuals in $Q^*$ should be have only one expected position. The expectation value of $E$-metric can be obtained by calculating the expected error between $Q^*$ and $\mathbf{f}_K$, and the $E$-metric can be expressed as

$$E(P^*, \mathbf{f}_K) = \frac{\sum_{t=1}^{K} \sigma_t}{|P^*|} \tag{10}$$

where $K$ is the size of $\mathbf{f}_K$, which is calculated by (2), and the values $\sigma_t$ for each individual can be directly adopted by the error-comparison operator. The solution with both a high diversity and a good convergence can have a low expectation value of $E$-metric. Because the reference points come from the true PF, this metric can directly evaluate the performance of the solutions.

### B. SIMULATION RESULTS AND ANALYSIS

In order to show the high performance of the proposed methods, MOGA/F and MOGA/I are compared with the traditional methods of MOEA/D, NSGA-II, and NSGA-III. MOGA/I is implemented with PL, PCHIP, and FI. Several mathematical optimization problems of ZDT1, ZDT2, and ZDT4 are implemented. Only two-objective optimization problems are considered herein.

In this paper, the same conditions as used in NSGA-II, NSGA-III and MOEA/D in [10], [24], and [32] are adopted for the simulation. The population size will be fixed at 100 in each method. The maximum generation is set to 2500. The populations in the first generation are randomly generated in the constrained region, and the initial population with the same condition is used for each algorithm. For the crossover and mutation operation, the same parameters and processes as used in [10] and [32] are adopted in this paper. The crossover probabilities is set to $p_c = 0.9$, and the mutation probabilities is set to $p_m = 0.1$ in the evolutionary operator. The same distribution indices are applied for the crossover and mutation

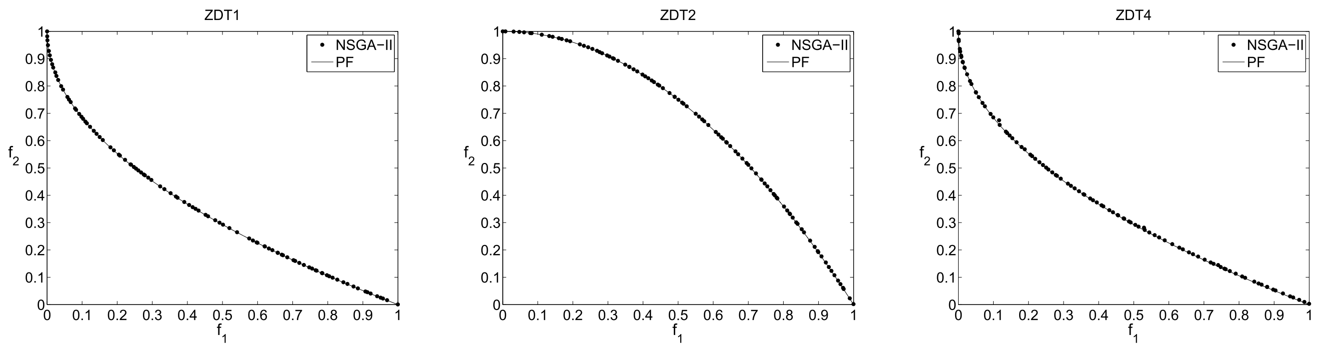



**FIGURE 1.** **The objective function values of the optimal population in the best simulation found by using NSGA-II.**

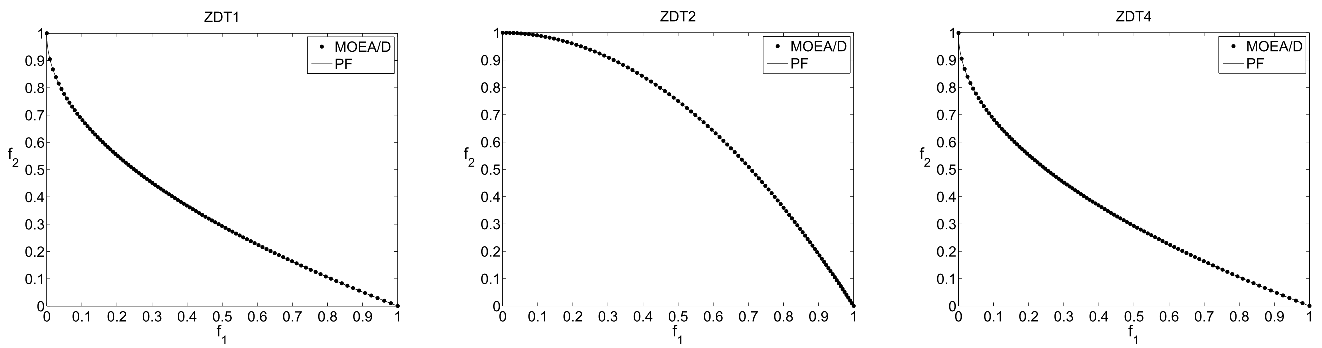



**FIGURE 2.** **The objective function values of the optimal population in the best simulation found by using MOEA/D.**

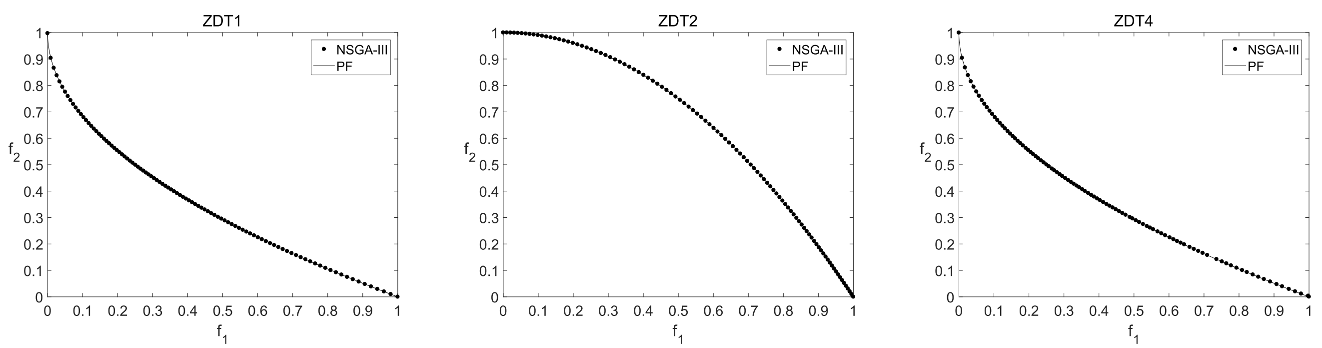



**FIGURE 3.** **The objective function values of the optimal population in the best simulation found by using NSGA-III.**

operators as $\eta_c = 20$ and $\eta_m = 20$, respectively [10]. MOEA/D is applied with Tchebycheff approach in this paper, which can indicate the performance of the evolutionary algorithms based on decomposition.

In order to avoid contingency of a single process, thirty independent simulations are carried out for all the optimization problems. For the interpolation method, the number of interpolation points is set to 2. The Polynomial fitting function is applied.

Fig. 1 to Fig. 7 give the objective function values of the optimal population obtained by using NSGA-II, MOEA/D, NSGA-III, MOGA/F, and MOGA/I with PL, PCHIP, and FI, which are the best results selected from the multiple simulations. Regarding the uniformity of the final solutions, the proposed algorithms have significant advantages over the other algorithm in different conditions.

NSGA-II can easily evolve the population approximating the PF, but there are some individuals deviating from the optimal position, and the corresponding results have a bad uniformity. Although NSGA-II can improve the diversity of the solutions by crowding distance, its ability to drive solutions toward uniformness may be limited. The solutions optimized by MOEA/D showed good diversity of uniform distribution and achieved a perfect match with the Tchebycheff vectors. Obviously, the individuals at the edge of ZDT1 and ZDT4 and those in the middle of ZDT2 have a larger distance from their neighbor. It should be noted that the individuals in this scale have a more important role in MOPs. This kind of MOEAs

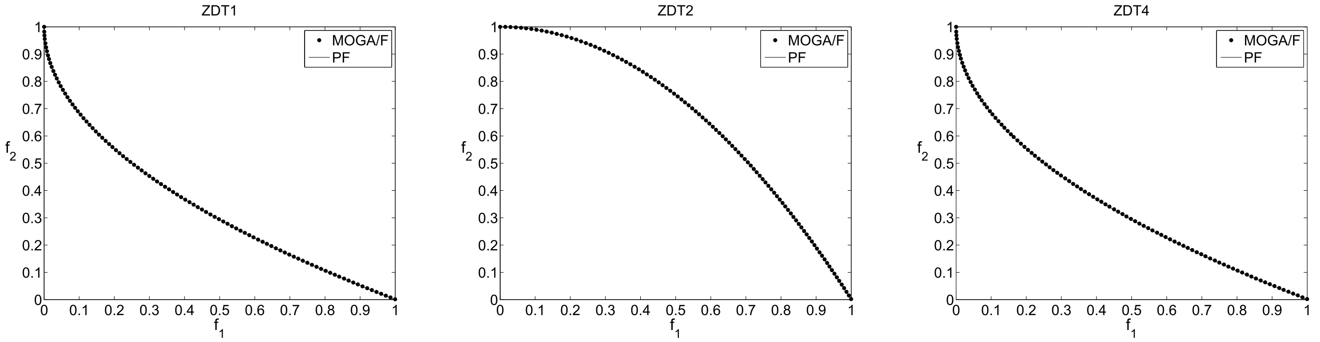



**FIGURE 4.** The objective function values of the optimal population in the best simulation found by using MOGA/F.

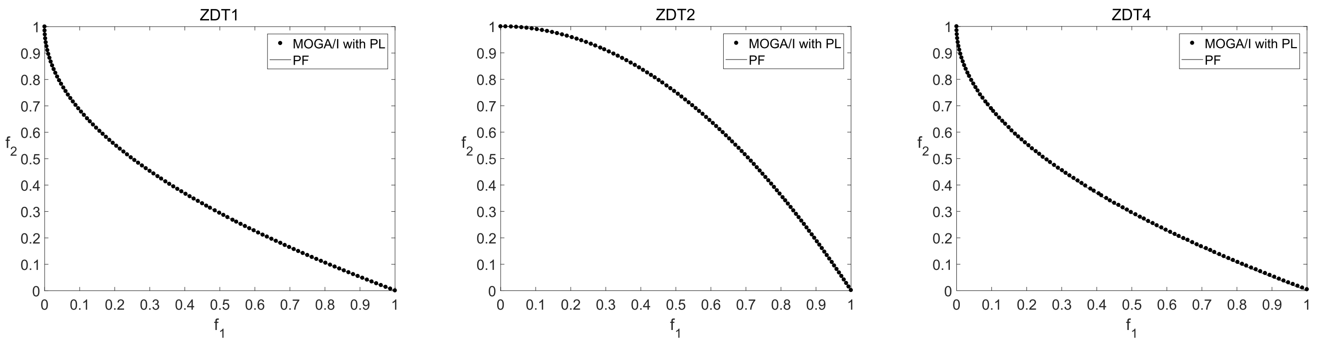



**FIGURE 5.** The objective function values of the optimal population in the best simulation found by using MOGA/I with PL.

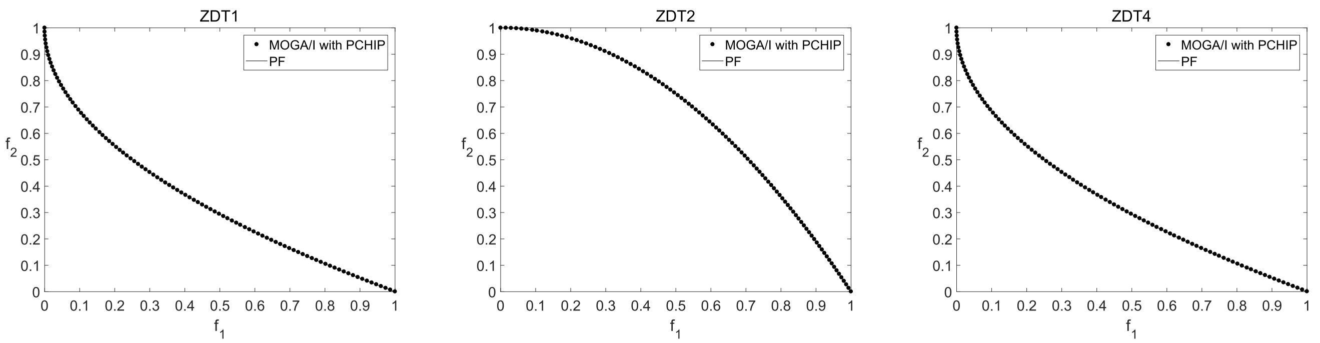



**FIGURE 6.** The objective function values of the optimal population in the best simulation found by using MOGA/I with PCHIP.

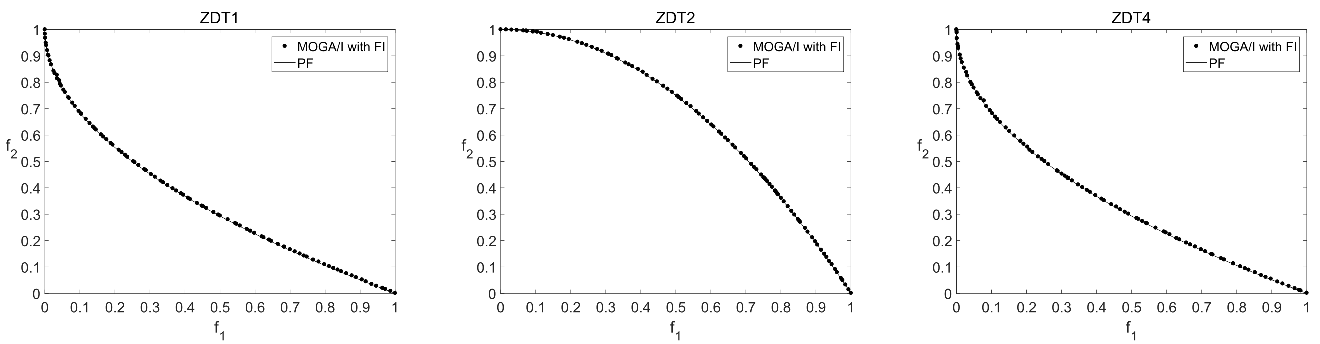



**FIGURE 7.** The objective function values of the optimal population in the best simulation found by using MOGA/I with FI.

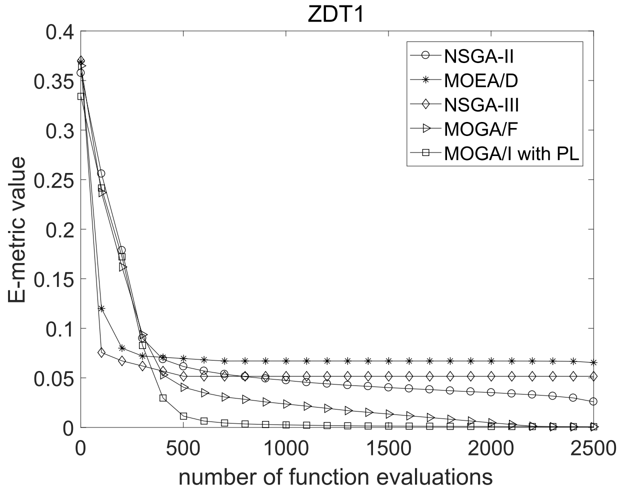



**FIGURE 8.** Evolution of the *E*-metric value in NSGA-II, MOEA/D, NSGA-III, MOGA/F, and MOGA/I with PL for ZDT1.

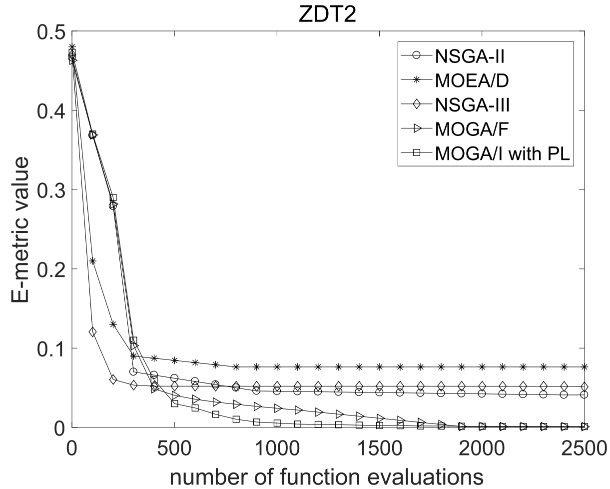



**FIGURE 9.** Evolution of the *E*-metric value in NSGA-II, MOEA/D, NSGA-III, MOGA/F, and MOGA/I with PL for ZDT2.

similar to MOEA/D and predefined reference points or vectors may also have this appearance because of the concave and convex of the PF. In Fig. 3, the final solutions obtained by NSGA-III are similar to those of MOEA/D. For NSGA-III, reference points are uniformly distributed in every axis instead of the PF. As a result, the individuals in the areas with big curvature may loss the uniformity.

The solutions optimized by MOGA/F and MOGA/I showed a better performance in diversity of uniformness. Each individual has a relatively equal distance from its neighbor and can approximate the PF very well.

To compare the performances of the solutions obtained by these algorithms, the performance index of the expectation value is implemented for all the optimization problems. The size of expected points $\hat{\mathbf{f}}_K$ is set to $N$. Then every individual can be corresponding to one single expected position, and the evaluation of its performance can be obtained.

**TABLE 1.** Average expectation values of the solutions found by NSGA-II, MOEA/D, NSGA-III, MOGA/F, and MOGA/I.

| Instance | ZDT1 | ZDT2 | ZDT4 |
|---|---|---|---|
| NSGA-II | 0.0259 | 0.0410 | 0.0155 |
| MOEA/D | 0.0669 | 0.0763 | 0.0671 |
| NSGA-III | 0.0516 | 0.0512 | 0.0642 |
| MOGA/F | 6.6888e-4 | 0.0010 | 0.0016 |
| MOGA/I with PL | 7.3044e-4 | 0.0011 | 0.0028 |
| MOGA/I with PCHIP | 4.6034e-4 | 4.5118e-4 | 7.7251e-4 |
| MOGA/I with FI | 0.0159 | 0.0214 | 0.0276 |

Table 1 presents the average expectation values obtained from each method. NSGA-II clearly achieved a better performance than NSGA-III and MOEA/D, which is contrary to the simulation and analysis in [32], due to the different number of expected points in the calculation of expectation value. This parameter in [32] is much larger, so the population

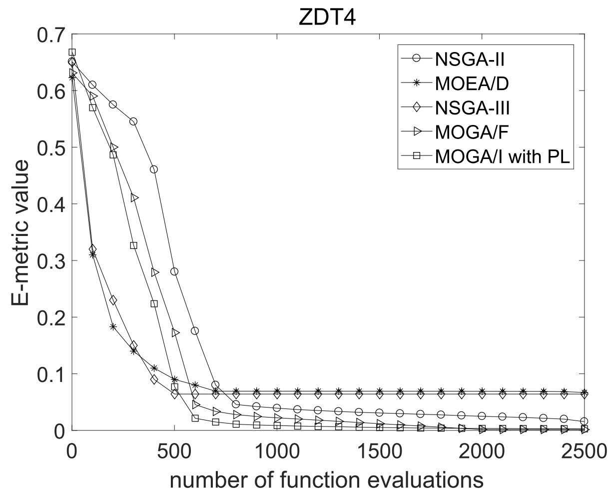



**FIGURE 10.** Evolution of the *E*-metric value in NSGA-II, MOEA/D, NSGA-III, MOGA/F, and MOGA/I with PL for ZDT4.

with bad uniformity can have a good expectation value. The data show that the simulation results calculated by MOGA/F and MOGA/I can achieve a considerably higher performance by several orders of magnitude. The obtained population with minimum error to the expected points has a higher performance of uniformity and a good approximation of the PF. The results show the fact that MOGA/F achieves a better performance than MOGA/I. In the application, the fitting function has better robustness because when one individual deviates from the PF, the interpolation function cannot accurately reflect the law of PF. For the method PCHIP, higher order of polynomial can be a better representation for the PF, whereas for MOGA/I with FI, there is no corresponding performance improvement. FI function introduces new errors by the original advantages of handling volatile data. The optimum solutions of ZDTs have the character of polynomial. Therefore, the expectation values of the final population obtained by MOGA/I with FI are worse than those of other two interpolation methods.

Fig. 8, Fig.9 and Fig. 10 present the evolution of the average $E$-metric value of the current population to the expected positions $\mathbf{f}_N$ with the number of function evaluations in each algorithm for each instance. These results indicate that, in terms of the number of function evaluations, MOEA/D and NSGA-III converge much faster than the other algorithms in minimizing the $E$-metric value but cannot converge to the smallest value. MOGA/F and MOGA/I with PL have similar convergence in the initial process because they have the same evolutional structure. MOGA/I with PL converges faster than MOGA/F, whereas MOGA/F has a smaller $E$-metric value in the end process. The interpolation function can approximate the PF faster, and the fitting function can approximate the PF more precisely. For different interpolation functions, they have the similar convergence. Hence, only the result of MOGA/I with PL is listed.

## V. CONCLUSION

The multi-objective optimization methods of MOGA/F and MOGA/I have been proposed to find the best population with a high performance of diversity and a good approximation of the PF. By using a fitting and an interpolation function, the reference points uniformly distributed in the objective space can be easily obtained. Contributing to the error-comparison operator, the individuals with lower distance to the expected points calculated by MOGA/F and MOGA/I can be easily chosen through the error matrix. The results of the three mathematical optimization problems indicate that the optimal population found by using MOGA/F and MOGA/I has a good uniformity in the PF and can converge to a desired result. Regarding the convergence of the average $E$-metric value of the current population to the expected positions, the interpolation function can approximate the PF faster, while the fitting function can approximate the PF more precisely. Based on the objective function values of the optimal population found in the best simulations found by using different methods, MOGA/F and MOGA/I have a much higher performance on diversity of uniformness.

## REFERENCES

[1] L. Liu, F. Zhou, M. Tao, and Z. Zhang, "A novel method for multi-targets ISAR imaging based on particle swarm optimization and modified CLEAN technique," *IEEE Sensors J.*, vol. 16, no. 1, pp. 97–108, Jan. 2016.
[2] Y. Xu, K. Li, J. Hu, and K. Li, "A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues," *Inf. Sci.*, vol. 270, pp. 255–287, Jun. 2014.
[3] C. W. Ahn and R. S. Ramakrishna, "A diversity preserving selection in multiobjective evolutionary algorithms," *Appl. Intell.*, vol. 32, no. 3, pp. 231–248, Jun. 2010.
[4] F. V. C. Martins, E. G. Carrano, E. F. Wanner, R. H. C. Takahashi, and G. R. Mateus, "A hybrid multiobjective evolutionary approach for improving the performance of wireless sensor networks," *IEEE Sensors J.*, vol. 11, no. 3, pp. 545–554, Mar. 2011.
[5] S. H. Hojjati, A. Ebrahimzadeh, M. Najimi, and A. Reihanian, "Sensor selection for cooperative spectrum sensing in multiantenna sensor networks based on convex optimization and genetic algorithm," *IEEE Sensors J.*, vol. 16, no. 10, pp. 3486–3487, May 2016.
[6] Q. Cai, M. Gong, S. Ruan, Q. Miao, and H. Du, "Network structural balance based on evolutionary multiobjective optimization: A two-step approach," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 903–916, Dec. 2015.
[7] L. Zhang, K. Li, C. Li, and K. Li, "Bi-objective workflow scheduling of the energy consumption and reliability in heterogeneous computing systems," *Inf. Sci.*, vol. 379, pp. 241–256, Feb. 2017.
[8] H.-C. Huang and C.-H. Chiang, "An evolutionary radial basis function neural network with robust genetic-based immunecomputing for online tracking control of autonomous robots," *Neural Process. Lett.*, vol. 44, no. 1, pp. 19–35, Aug. 2016.
[9] N. Baatar, K.-Y. Jeong, and C.-S. Koh, "Adaptive parameter controlling non-dominated ranking differential evolution for multi-objective optimization of electromagnetic problems," *IEEE Trans. Magn.*, vol. 50, no. 2, Feb. 2014, Art. no. 7017504.
[10] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[11] S. F. Adra and P. J. Fleming, "Diversity management in evolutionary many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 15, no. 2, pp. 183–195, Apr. 2011.
[12] K. Deb, M. Mohan, and S. Mishra, "Evaluating the $\varepsilon$-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions," *Evol. Comput.*, vol. 13, no. 4, pp. 501–525, 2005.
[13] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evol. Comput.*, vol. 10, no. 3, pp. 263–282, 2002.
[14] R. C. Liu, R. Wang, W. Feng, J. Huang, and L. Jiao, "Interactive reference region based multi-objective evolutionary algorithm through decomposition," *IEEE Access*, vol. 4, pp. 7331–7346, 2016.
[15] 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.
[16] C. Dai and Y. Wang, "A new multiobjective evolutionary algorithm based on decomposition of the objective space for multiobjective optimization," *J. Appl. Math.*, Jan. 2014, Art. no. 906147, doi: 10.1155/2014/906147.
[17] M. Asafuddoula, T. Ray, and R. Sarker, "A decomposition-based evolutionary algorithm for many objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 445–460, Jun. 2015.
[18] Y. Yuan, H. Xu, B. Wang, B. Zhang, and X. Yao, "Balancing convergence and diversity in decomposition-based many-objective optimizers," *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 180–198, Apr. 2016.
[19] Y.-Y. Tan, Y.-C. Jiao, H. Li, and X.-K. Wang, "A modification to MOEA/D-DE for multiobjective optimization problems with complicated Pareto sets," *Inf. Sci.*, vol. 213, pp. 14–38, Dec. 2012.
[20] R. Carvalho, R. R. Saldanha, B. N. Gomes, A. C. Lisboa, and A. X. Martins, "A multi-objective evolutionary algorithm based on decomposition for optimal design of Yagi–Uda antennas," *IEEE Trans. Magn.*, vol. 48, no. 2, pp. 803–806, Feb. 2012.
[21] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 5, pp. 721–736, Oct. 2013.
[22] K. B. Lari and A. Hamzeh, "A diversity control mechanism in many objective optimizations," *Appl. Intell.*, vol. 45, no. 4, pp. 953–975, Dec. 2016.
[23] Z. He and G. G. Yen, "Many-objective evolutionary algorithm: Objective space reduction and diversity improvement," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 145–160, Feb. 2016.
[24] 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.
[25] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, Apr. 2011.
[26] M. Li, S. Yang, and X. Liu, "Diversity comparison of Pareto front approximations in many-objective optimization," *IEEE Trans. on*, vol. 44, no. 12, pp. 2568–2584, Dec. 2014.
[27] C. Segura, C. A. C. Coello, E. Segredo, and A. H. Aguirre, "A novel diversity-based replacement strategy for evolutionary algorithms," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 3233–3246, Dec. 2016.
[28] Y. Xu, K. Li, L. He, L. Zhang, and K. Li, "A hybrid chemical reaction optimization scheme for task scheduling on heterogeneous computing systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 12, pp. 3208–3222, Dec. 2015.
[29] M. Tang and S. Pan, "A hybrid genetic algorithm for the energy-efficient virtual machine placement problem in data centers," *Neural Process. Lett.*, vol. 41, no. 2, pp. 211–221, Apr. 2015.

[30] H. Ganjidoost, S. J. Mousavi, and A. Soroush, "Adaptive network-based fuzzy inference systems coupled with genetic algorithms for predicting soil permeability coefficient," *Neural Process. Lett.*, vol. 44, no. 1, pp. 53–79, Aug. 2016.

[31] C. Han, L. Wang, Z. Zhang, J. Xie, and Z. Xing, "Linear array pattern synthesis using an improved multiobjective genetic algorithm," *Radioengineering*, vol. 26, no. 4, pp. 1048–1059, Dec. 2017.

[32] 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.

[33] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, Oct. 2006.

[34] Y. Zhou, L. Li, and M. Ma, "A complex-valued encoding bat algorithm for solving 0–1 knapsack problem," *Neural Process. Lett.*, vol. 44, no. 2, pp. 407–430, Oct. 2016.

[35] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.

[36] A. Toriello and J. P. Vielma, "Fitting piecewise linear continuous functions," *Eur. J. Oper. Res.*, vol. 219, no. 1, pp. 86–95, May 2012.

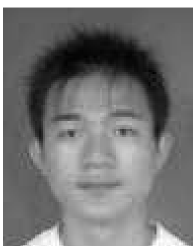
[37] A. Magnani and S. Boyd, "Convex piecewise-linear fitting," *Optim. Eng.*, vol. 10, no. 1, pp. 1–17, Mar. 2009.

[38] I. Gijbels, A. Lambert, and P. Qiu, "Jump-preserving regression and smoothing using local linear fitting: A compromise," *Ann. Inst. Stat. Math.*, vol. 59, no. 2, pp. 235–272, Jun. 2007.

[39] H. Q. Fu, J. J. Zhu, C. C. Wang, H. Q. Wang, and R. Zhao, "A wavelet decomposition and polynomial fitting-based method for the estimation of time-varying residual motion error in airborne interferometric SAR," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 49–59, Jan. 2018.

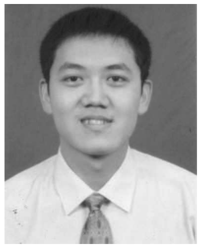
[40] H.-Y. Wang and J.-S. Yu, "Fractal interpolation functions with variable parameters and their analytical properties," *J. Approx. Theory*, vol. 175, pp. 1–18, Nov. 2013.

[41] P. Jiang, F. Liu, J. Wang, and Y. Song, "Cuckoo search-designated fractal interpolation functions with winner combination for estimating missing values in time series," *Appl. Math. Model.*, vol. 40, nos. 23–24, pp. 9692–9718, Dec. 2016.

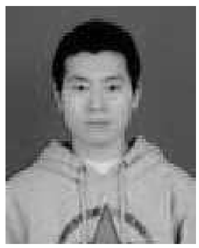


**CHUANG HAN** was born in Xinji, Hebei, China, in 1989. He received the B.Sc. and M.Sc. degrees in electronic engineering from Northwestern Polytechnical University in 2012 and 2015, respectively, where he is currently pursuing the Ph.D. degree. His recent research interests include array signal process, antenna analysis and synthesis, satellite communication systems, and satellite navigation systems.

**LING WANG** was born in Renhuai, Guizhou, China, in 1978. He received the B.Sc., M.Sc., and Ph.D. degrees from Xidian University, Xi'an, China, in 1999, 2002, and 2004, respectively. From 2004 to 2007, he was with Siemens Academy in the Wireless Communication Unit and Nokia Siemens in the Networks Research Center. Since 2007, he has been with the School of Electronic and Information, Northwestern Polytechnical University. His current research interests include B3G/4G mobile communications; vehicle tracking, telemetry, and command; antijamming for communications and navigation systems; and cognitive radio.



**ZHAOLIN ZHANG** received the B.Sc., M.Sc., and Ph.D. degrees from Northwestern Polytechnical University, Xi'an, China, in 2000, 2005, and 2012, respectively. Since 2000, he has been a Teacher with the School of Electronic and Information, Northwestern Polytechnical University, where he was promoted to be an Associate Professor. His current research interests include antijamming for communications and navigation systems, array signal process, and multimedia communication.



**JIAN XIE** was born in Nantong, Jiangsu, China, in 1986. He received the M.Sc. and Ph.D. degrees from the School of Electronic Engineering, Xidian University, in 2012 and 2015, respectively. He is currently an Assistant Professor with Northwestern Polytechnical University. His research interests include antenna array processing and radar signal processing.

**ZIJIAN XING** received the B.Sc., M.Sc., and Ph.D. degrees from Northwestern Polytechnical University in 2008, 2011, and 2014, respectively. Since 2014, he has been an Assistant Professor with Northwestern Polytechnical University. His research interests include antenna array, circular polarization antenna, and antennas for radio frequency identification devices application.