DMOEA-εC: Decomposition-Based Multiobjective Evolutionary Algorithm With the ε-Constraint Framework

Jie Chen, *Senior Member, IEEE*, Juan Li, and Bin Xin, *Member, IEEE*

***Abstract*—Decomposition is an efficient and prevailing strategy for solving multiobjective optimization problems (MOPs). Its success has been witnessed by the multiobjective evolutionary algorithm MOEA/D and its variants. In decomposition-based methods, an MOP is decomposed into a number of scalar subproblems by using various scalarizing functions. Most decomposition schemes adopt the weighting method to construct scalarizing functions. In this paper, another classical generation method in the field of mathematical programming, that is the ε-constraint method, is adopted for the multiobjective optimization. It selects one of the objectives as the main objective and converts other objectives into constraints. We incorporate the ε-constraint method into the decomposition strategy and propose a new decomposition-based multiobjective evolutionary algorithm with the ε-constraint framework (DMOEA-εC). It decomposes an MOP into a series of scalar constrained optimization subproblems by assigning each subproblem with an upper bound vector. These subproblems are optimized simultaneously by using information from neighboring subproblems. Besides, a main objective alternation strategy, a solution-to-subproblem matching procedure, and a subproblem-to-solution matching procedure are proposed to strike a balance between convergence and diversity. DMOEA-εC is compared with a number of state-of-the-art multiobjective evolutionary algorithms. Experimental studies demonstrate that DMOEA-εC outperforms or performs competitively against these algorithms on the majority of 34 continuous benchmark problems, and it also shows obvious advantages in solving multiobjective 0-1 knapsack problems.**

***Index Terms*—ε-constraint method, decomposition, main objective alternation strategy, multiobjective optimization, solution-to-subproblem matching procedure, subproblem-to-solution matching procedure.**

Manuscript received July 22, 2016; revised November 20, 2016 and January 18, 2017; accepted February 13, 2017. Date of publication February 17, 2017; date of current version September 28, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61673058, Grant U1609214, and Grant 61304215, in part by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China under Grant 61321002, in part by the Projects of Major International (Regional) Joint Research Program NSFC under Grant 61120106010, in part by the Beijing Outstanding Ph.D. Program Mentor under Grant 20131000704, and in part by the Research Fund for the Doctoral Program of Higher Education of China under Grant 20131101120033. *(Corresponding author: Bin Xin.)*

The authors are with the School of Automation, Beijing Institute of Technology, Beijing 100081, China, also with the State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology, Beijing 100081, China, and also with the Beijing Advanced Innovation Center for Intelligent Robots and Systems, Beijing Institute of Technology, Beijing 100081, China (e-mail: brucebin@bit.edu.cn).

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2017.2671462

I. Introduction

MANY real-world practical problems have two or more objectives which are usually conflicted. These problems to be handled are named multiobjective optimization problems (MOPs). An MOP can be stated as follows:

$$\begin{aligned} P0 : \text{minimize } & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})) \\ \text{subject to } & \mathbf{x} = (x_1, x_2, \ldots, x_n) \in \Omega \end{aligned}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is the decision vector. It belongs to the nonempty feasible set Ω. $\mathbf{F} : \Omega \to R^m$ consists of $m(\geq 2)$ objective functions $f_i : R^n \to R, i = 1, \ldots, m$. The objective functions in $P0$ contradict each other, and no single solution optimizes them simultaneously. The Pareto optimality is a tradeoff among these conflicted objectives [1], [2]. Any improvement in one objective of a Pareto optimal point must lead to deteriorations in at least one other objective. The set of all the Pareto optimal points is called the Pareto set (PS) and the set of all the Pareto optimal objective vectors is called the Pareto front (PF).

Very often, an approximation of the PF with manageable number of points and even distribution along the true PF is needed and presented to decision makers for the purpose of supporting their decision-makings. Recently, various multiobjective evolutionary algorithms (MOEAs) have been widely accepted as major approaches for approximating the true PF [3]–[5]. Based on their selection strategies, these algorithms can be categorized into three classes: 1) Pareto dominance-based [6]–[11]; 2) performance indicator-based [12], [13]; and 3) decomposition-based [14]–[22]. Among them, the decomposition-based approaches are growing in popularity and become major methodologies for the multiobjective optimization thanks to their good properties.

MOEA/D [15] is the most popular decomposition-based MOEA. It decomposes an MOP into a number of scalar optimization subproblems and then applies evolutionary algorithms (EAs) to optimize these subproblems in a collaborative manner. Each subproblem is optimized by using information from its neighboring subproblems, which makes MOEA/D with relative low computational complexity. Neighborhood relations between these subproblems are defined based on Euclidean distances between their weight vectors. The diversity of the population is implicitly maintained by specifying a good spread of the weight vectors in the objective space. In recent years, MOEA/D has attracted increasing research interests and many follow-up studies have been published. In [23]–[30], authors combine MOEA/D with other metaheuristics. Deb and Jain [21] and Jain and Deb [22], and Ishibuchi *et al.* [31], [32] researched on different scalarizing approaches. Besides, some studies address the problem of adjusting weight vectors to make optimal solutions uniformly distributed along the PF [20], [23], [33] and extending applications from benchmark problems to real-world problems [24], [34]–[39].

In general, the weighting method and the ε-constraint method are two basic generation methods [1]. They are often used as elements of more developed methods. MOEA/D [15] takes inspiration from the weighting method. The ε-constraint method selects one of the objectives as the main objective while transforming the other nonmain objectives to constraints and associating each nonmain objective with an upper bound coefficient. Mavrotas [40] first proposed an augmented ε-constraint method (AUGMECON) which divides the range of each nonmain objective function into a fixed number of equal intervals by using the ideal and nadir points obtained from the payoff table and uses these grid points as upper bound vectors. Besides, in order to avoid the production of weakly Pareto optimal solutions, AUGMECON transforms inequality constraints to equality ones by explicitly incorporating slack variables. In the literature, several versions of the ε-constraint method have been put forward. Mavrotas and Florios [41] presented the AUGMECON2 which is an improvement of AUGMECON. It introduces a concept of the bypass coefficient which indicates how many consecutive iterations can be bypassed. Grandinetti *et al.* [42] proposed an approximate ε-constraint method to solve a multiobjective job scheduling problem. The method defines a finite sequence of ε-constraint problems through a progressive reduction of the values of upper bound coefficients. Zhang and Reimann [43] put forward a simple augmented ε-constraint method (SAUGMECON) which is a variant of AUGMECON2. The innovative mechanisms of SAUGMECON include an extension to the acceleration algorithm with an early exit and an addition of an acceleration algorithm with bouncing steps.

In this paper, we synthesize the merits of the decomposition strategy and the ε-constraint method and propose a new decomposition-based MOEA with the ε-constraint framework (DMOEA-εC). DMOEA-εC explicitly decomposes an MOP into a series of scalar constrained optimization subproblems by selecting one of the objectives as the main objective function and associating each subproblem with an upper bound vector. These subproblems are optimized collaboratively by an EA based on the Deb's [44] feasibility rule. And each subproblem is optimized by using information only from its neighboring subproblems. Besides, since the selection of the main objective function is a very important factor under the ε-constraint framework, a main objective alternation strategy is proposed and used periodically. Then in order to tackle problems induced by the main objective alternation strategy, a solution-to-subproblem matching procedure is designed to place the nearest solution to each subproblem and is utilized after the main objective alternation strategy. Finally, for the purpose of further improving performance, a subproblem-to-solution matching procedure is proposed to find a subproblem with the minimum constraint violation value for a new generated solution.

The major innovations and contributions of this paper include the following.

1) To the best of our knowledge, this is the first attempt to incorporate the ε-constraint method into the decomposition strategy and solve an MOP via optimizing a series of scalar constrained subproblems collaboratively using the neighbor information. This idea gives birth to a new efficient MOEA, namely, DMOEA-εC. Numerical results show that DMOEA-εC is superior to or competitive against six state-of-the-art MOEAs on 34 continuous test problems. It also shows obvious advantages over MOEA/D on multiobjective 0-1 knapsack problems (MOKPs).
2) Under the ε-constraint framework, DMOEA-εC tends to retain feasible solutions for each subproblem. This will be bad for the optimization of the main objective function. Thus a main objective alternation strategy is proposed and used periodically to tackle this problem. And the necessity of the main objective alternation strategy is experimentally confirmed in Section VI-A.
3) After the main objective alternation strategy is utilized, a solution which is good for the current subproblem will no longer perform well since the objective function of this subproblem has been changed. Thus a solution-to-subproblem matching procedure is designed to place the nearest solution to each subproblem.
4) When a new solution is generated, it may perform badly for the current subproblem but perform well for another subproblem. In order to avoid wasting potentially useful solutions and make best use of them, the subproblem-to-solution matching procedure is proposed to find a subproblem with the minimum constraint violation value for the new solution. The two matching procedures strike a balance between convergence and diversity. And further experimental results in Section VI-B confirm the effects of the two matching procedures.

The rest of this paper is organized as follows. Section II reformulates MOPs under the ε-constraint framework. Section III describes the algorithmic framework of the DMOEA-εC. Sections IV and V show comparison results about DMOEA-εC against other state-of-the-art MOEAs on continuous benchmark problems and against MOEA/D on MOKPs. Some further experimental studies on the parameter sensitivity, the effectiveness of the solution-to-subproblem matching procedure, the subproblem-to-solution matching procedure, and the farthest-candidate approach in DMOEA-εC are conducted in Section VI. Section VII concludes this paper.

II. Formulation of MOPs Under the ε-Constraint Framework

In the ε-constraint method first introduced by Haimes *et al.* [45], one of the objectives is selected as the main objective function to be optimized and all the other nonmain objectives are converted into constraints by giving

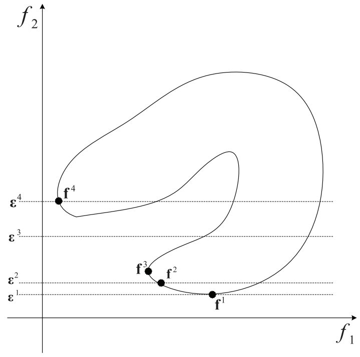

Fig. 1. Different upper bounds for the ε-constraint method.

an upper bound coefficient to each of them. The ε-constraint problem corresponding to the MOP $P0$ is formulated as follows:

$$P1 : \text{minimize } f_{\text{main}} = f_s(\mathbf{x}) + \rho \sum_{i=1}^{m} f_i(\mathbf{x})$$

$$\text{subject to } \begin{cases} \dfrac{f_i(\mathbf{x}) - z_i^*}{z_i^{\text{nad}} - z_i^*} \le \varepsilon_i, \forall i \in \{1, 2, \ldots, m\}/\{s\} \\ \mathbf{x} = (x_1, x_2, \ldots, x_n) \in \Omega \end{cases}$$

where $0 \le \varepsilon = (\varepsilon_1, \ldots, \varepsilon_{s-1}, \varepsilon_{s+1}, \ldots, \varepsilon_m) \le 1$ is the upper bound vector. The main objective index s is randomly selected from $\{1, 2, \ldots, m\}$ or predefined by decision makers. $\rho > 0$ is a very small positive number. $\mathbf{z}^* = (z_1^*, \ldots, z_m^*)$ and $\mathbf{z^{nad}} = (z_1^{\text{nad}}, \ldots, z_m^{\text{nad}})$ are the ideal point and the nadir point, respectively. The exact definitions will be given in the following. An example illustration of different upper bounds for the ε-constraint method is shown in Fig. 1. In Fig. 1, different upper bounds (i.e., different ε values) for the objective function f_2 are given, while the objective function f_1 is selected as the main objective function to be minimized. The Pareto optimal solutions corresponding to the four upper bound vectors are shown by the black points.

Definition 1 (Ideal Point $\mathbf{z}^*$): The component z_i^* of the ideal point $\mathbf{z}^* \in R^m$ is obtained by minimizing each of the objective functions individually subject to the constraints, that is, by solving the following problems:

$$\begin{aligned} &\text{minimize } f_i(\mathbf{x}) \\ &\text{subject to } \mathbf{x} \in \Omega \quad \text{for } i = 1, \ldots, m. \end{aligned}$$

Definition 2 (Nadir Point $\mathbf{z^{nad}}$): The nadir point is the upper bound of the PF. Each element z_i^{nad} of the nadir point $\mathbf{z^{nad}} \in R^m$ is defined as $z_i^{\text{nad}} = \max\{f_i | \mathbf{F} = (f_1, f_2 \ldots, f_m) \in \text{PF}\}$ for $i = 1, \ldots, m$.

Nadir points are much harder to obtain than ideal points. The payoff table provides a way for roughly estimating nadir points. However, it does not guarantee the accurate calculation except for bi-objective problems [46]. Several approaches have been proposed to calculate nadir points [47], [48]. In this paper, we simply replace the ideal point by the minimum value of each objective function in the population, replace the nadir point with the maximum value of each objective function in the external archive, and update them iteratively.

Algorithm 1 Framework of DMOEA-εC

Input: An MOP, related parameters.
Output: An external archive population *EP*.

Initialize N evenly spread upper bound vectors; randomly initialize the evolving population $\mathbf{P} = \{\mathbf{x^1}, \ldots, \mathbf{x^N}\}$ and set $\mathbf{FV}^i = \mathbf{F}(\mathbf{x}^i)$; extract nondominated individuals from P and denote the set of them as *EP*; initialize $\mathbf{z}^*$ and $\mathbf{z^{nad}}$; set $gen = 0, n = N$.
Use the solution-to-subproblem matching procedure (**Algorithm 2**) to match solutions with subproblems.
for $i = 1$ to N **do**
  Set the neighborhood of the ith subproblem $B(i)$.
  $\pi^i = 1$.
end for
if *gen* is a multiple of *DRA_interval* **then**
  Update the indices of the subproblems I that will be processed in next generation by applying the dynamic resource allocation scheme (**Algorithm 3**).
end if
if *gen* is a multiple of IN_m **then**
  Alternate the main objective.
  Use the solution-to-subproblem matching procedure (**Algorithm 2**) to match solutions with subproblems.
end if
while $n \le NFE$ **do**
  for $i \in I$ **do**
    $P = \begin{cases} B(i), & \text{if } rand < \delta \\ \{1, 2, \ldots, N\}, & otherwise \end{cases}$
    Reproduction: select parent individuals from P randomly and apply certain reproduction operator to generate a new solution $\mathbf{y}$.
    $n = n + 1$.
    Repair: if $\mathbf{y}$ is infeasible, repair it.
    Update the approximated ideal point $\mathbf{z}^*$.
    Use the subproblem-to-solution matching procedure (**Algorithm 4**) to find a subproblem k for $\mathbf{y}$.
    Compare $\mathbf{y}$ with neighboring solutions of the subproblem k and update these neighboring solutions by using the feasibility rule.
    Update the external archive *EP* and prune it by using the farthest-candidate approach (**Algorithm 5**).
    Update the approximated nadir point $\mathbf{z^{nad}}$.
  end for
  $gen = gen + 1$.
end while

According to the theoretical results given by Miettinen [1], every Pareto optimal solution of an MOP can be found via the ε-constraint method by altering the upper bounds and the main objective function to be minimized. This result provides a theoretical foundation for the proposed DMOEA-εC.

III. Framework of DMOEA-εC

A. Algorithmic Framework

The DMOEA-εC framework converts an MOP into N scalar constrained subproblems and optimizes them simultaneously

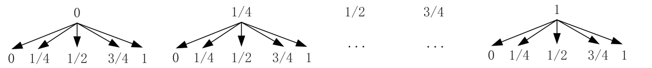

Fig. 2. Illustration of generating uniformly spread upper bound vectors ($m = 3$, $q = 5$).

in a single run. Let $\varepsilon^{\mathbf{1}}, \varepsilon^{\mathbf{2}}, \ldots, \varepsilon^{\mathbf{N}}$ be a set of evenly spread upper bound vectors, and the neighborhood of upper bound vector ε^i is defined as a set of its several closest upper bound vectors in $\{\varepsilon^{\mathbf{1}}, \varepsilon^{\mathbf{2}}, \ldots, \varepsilon^{\mathbf{N}}\}$. The neighborhood of the ith subproblem consists of all subproblems with the upper bound vectors from the neighborhood of ε^i and is denoted as $B(i)$. When optimizing each subproblem, the feasibility rule [44] is adopted as the constraint handling method. When two solutions are compared, the following criteria are followed.

1) Any feasible solution is preferred to any infeasible solution.
2) Among two feasible solutions, the one having better objective function value is preferred.
3) Among two infeasible solutions, the one having smaller constraint violation is preferred.

The following notations will be used in the description of DMOEA-εC.

- *N:* The number of upper bound vectors, which is the same as the population size.
- *T:* Neighborhood size.
- *δ:* Probability of selecting mate solutions from its neighborhood.
- *n_r:* Maximum number of replacement.
- *IN_m:* Iteration interval of alternating the main objective function.
- *DRA_interval:* Iteration interval of utilizing the dynamic resource allocation strategy.
- *S:* Maximum size of the external archive population (EP).
- *NFE:* Maximum number of function evaluations.

The algorithmic description of DMOEA-εC is presented in Algorithm 1. In this algorithm, rand is a uniformly randomly distributed value in [0, 1].

1) Generation of Upper Bound Vectors $\varepsilon^{\mathbf{1}}, \varepsilon^{\mathbf{2}}, \ldots, \varepsilon^{\mathbf{N}}$: A structured set of upper bound vectors are generated by dividing each objective axis with an equal spacing Δ. The process of generating the upper bound vectors is illustrated by using a tri-objective optimization problem ($m = 3$) with a spacing of $\Delta = 1/4 (q = 5)$ in Fig. 2. The process results in the generation of 25 upper bound vectors.

2) Solution-to-Subproblem Matching Procedure (Algorithm 2): After the main objective alternation strategy is utilized, a solution which is good for the current subproblem will no longer perform well since the objective function of this subproblem has been changed. Thus a solution-to-subproblem matching procedure is proposed. In this matching procedure, the solution that has the minimum distance value to certain subproblem among N current solutions is matched with this subproblem. The distance value from solution $(\mathbf{x}^i, \mathbf{FV}^i)$ to the subproblem with ε^l is defined as $d_i^l = \sum_{j=1, j\neq s}^{m} |f_j^i - \varepsilon_j^l|$. This matching procedure will be used after the main objective alternation procedure. It can place the nearest solution to each subproblem at a large degree, which is beneficial to diversity.

Algorithm 2 Solution-to-Subproblem Matching

Input: N solutions $(\mathbf{x}^1, \mathbf{FV}^1), \ldots, (\mathbf{x}^N, \mathbf{FV}^N)$ and N subproblems with upper bound vectors $\varepsilon^{\mathbf{1}}, \varepsilon^{\mathbf{2}} \cdots, \varepsilon^{\mathbf{N}}$.
Output: Matched pairs $(\mathbf{x}^k, \mathbf{FV}^k) \sim \varepsilon^l (k, l = 1, \ldots, N)$.

```
Initialize S = {1, 2, ..., N}.
while S is nonempty do
    Randomly select an upper bound vector ε^l, l ∈ S.
    for i = 1 to N do
        d_i^l = Σ_{j=1, j≠s}^{m} |f_j^i − ε_j^l|.
    end for
    k = argmin_{i=1,...,N}(d_1^l, d_2^l, ..., d_N^l).
    FV^k = Inf; S = S/{l}.
end while
```

Algorithm 3 Dynamic Resource Allocation [17]

Input: Utility values $\pi^1, \pi^2, \ldots, \pi^N$, old and new function value of each subproblem, denoted as $f_{main}^{old\_i}, f_{main}^{new\_i}$, respectively, for all $i = 1, 2, \ldots, N$.
Output: The indices of the selected subproblems I.

1: **for** $i = 1$ to N **do**
2: $\quad \Delta^i = \frac{f_{main}^{old\_i} - f_{main}^{new\_i}}{f_{main}^{old\_i}}$.
3: $\quad \pi^i = \begin{cases} 1, & \text{if } \Delta^i > 0.001 \\ (0.95 + 0.05 \cdot \frac{\Delta^i}{0.001}) \cdot \pi^i, & otherwise \end{cases}$.
4: **end for**
5: Set $I = \emptyset$ and select the indices of the m-subproblems whose epsilon vectors are permutation of $(1, 0, \ldots, 0)$. Choose other $\left\lfloor \frac{N}{5} \right\rfloor - m - 1$ indices using 10-tournament selection according to π^i, and add them to I.

Algorithm 4 Subproblem-to-Solution Matching

Input: New generated solution $\mathbf{y}$ and N subproblems with upper bound vectors $\varepsilon^{\mathbf{1}}, \varepsilon^{\mathbf{2}} \cdots, \varepsilon^{\mathbf{N}}$.
Output: The index of the selected subproblem k.

```
for l = 1 to N do
    CV^l = Σ_{j=1, j≠s}^{m} max((y_j − z_j^*)/(z_j^{nad} − z_j^*) − ε_j^l, 0).
    if CV^l = 0 then
        CV^l = 1 / Σ_{j=1, j≠s}^{m} ((y_j − z_j^*)/(z_j^{nad} − z_j^*) − ε_j^l).
    end if
end for
k = argmin_{i=1,...,N}(CV^1, CV^2, ..., CV^N). // Select the subproblem for which y is feasible and is nearest to y in the objective space.
```

3) Dynamic Resource Allocation Strategy (Algorithm 3): Different subproblems have different computational difficulties, therefore, it is reasonable to assign different amounts of computational effort to them based on their utility values which are defined in the line 3 of Algorithm 3 [17]. The 10-tournament selection in the line 5 is used to select subproblems that need to be processed in the next generation. It

Algorithm 5 Farthest Candidate Approach [49]

Input: A population Φ with size F, the size of selected individuals $K(< F)$.

Output: The selected solutions P_{accept} with size $K(< F)$.

Initialize $P_{accept} = \emptyset, \mathbf{D} = \mathbf{0}$.
for $i = 1$ to m **do**
    $P_{accept} = P_{accept} \cup \underset{\mathbf{x}\in\Phi}{argmin}(f_i(\mathbf{x})) \cup \underset{\mathbf{x}\in\Phi}{argmax}(f_i(\mathbf{x}))$.
end for
for each $\mathbf{x}$ in $\Phi - P_{accept}$ **do**
    $D(\mathbf{x}) = argmin_{\mathbf{x}' \in P_{accept}} dist(\mathbf{x}, \mathbf{x}')$.
end for
while $|P_{accept}| < N$ **do**
    $\mathbf{x_1} = argmax_{\mathbf{x}\in(\Phi - P_{accept})}(D(x))$.
    for each $\mathbf{x_2}$ in $\Phi - P_{accept}$ **do**
        $D(\mathbf{x_2}) = \min(D(\mathbf{x_2}), dist(\mathbf{x_1}, \mathbf{x_2}))$.
        $P_{accept} = P_{accept} \cup \mathbf{x_1}$.
    end for
end while

means drawing ten subproblems from the set of all subproblems randomly, and the subproblem with the best utility value will be selected. This process is repeated $\lfloor N/5 \rfloor - m - 1$ times to obtain the indices of the selected subproblems.

4) Subproblem-to-Solution Matching Procedure (Algorithm 4): When a new solution is generated, it may perform badly for the current subproblem but perform well for another subproblem. In order to avoid wasting potentially useful solutions and make best use of them, the subproblem-to-solution matching procedure is proposed. In this matching procedure, the subproblem that has the minimum constraint violation value is selected for the newly generated solution. The constraint violation value of the solution $\mathbf{y}$ regarding the subproblem with ε^l is defined as described in Algorithm 4. Since the feasibility rule is adopted to handle constrained subproblems, this procedure is good for convergence. The solution-to-subproblem matching procedure and the subproblem-to-solution matching procedure consider diversity and convergence, respectively.

5) Farthest-Candidate Approach (Algorithm 5) [49]: In DMOEA-εC, an EP is maintained in addition to the evolving population. Thus when a new solution is generated, the EP should be updated. And if the number of individuals in EP exceeds S, EP is pruned until its size equals to S.

In many MOEAs, in order to maintain a good spread of obtained nondominated solutions, several crowded comparison mechanisms have been proposed. In NSGA-II [9], a crowding distance-based comparison mechanism is adopted. Kukkonen and Deb [50] put forward an improved pruning of nondominated solutions. This method removes the solution that has the minimum crowding distance value one by one and recalculates the crowding distance value after each removal until the number of the remaining solutions is equal to the population size. In [51], a fast and effective method which is based on the crowding distance using the nearest neighborhood of solutions in the Euclidean sense is proposed. However, Chen *et al* [49] pointed out that these methods are unable to get a good spread result under some situations and presented a particular case for further explanation. Thus, the farthest-candidate approach [49] inspired by the best-candidate sampling algorithm [52] in sampling theory is adopted here to prune EP.

In the farthest-candidate approach, boundary points (solutions with the minimum and maximum objective values) are selected first. Then the candidate point among the unselected points which is farthest from the selected points is chosen iteratively. In this way, a set of evenly distributed nondominated solutions will be selected from a set of alternative nondominated solutions. In Algorithm 5, P_{accept} stores selected solutions, $\mathbf{D}$ stores the minimum Euclidean distance between $\mathbf{x}$ and unselected points, and dist$(\mathbf{x}, \mathbf{x}')$ is a function that calculates the Euclidean distance between $\mathbf{x}$ and $\mathbf{x}'$. The superiority of the farthest-candidate approach over the crowding distance based one used in [9] will be demonstrated in Section VI-B.

B. Discussions

1) Main Differences Between DMOEA-εC and MOEA/D: Both DMOEA-εC and MOEA/D introduce the concept of decomposition into MOEAs. Specifically, MOEA/D decomposes an MOP into N scalar subproblems by a scalarizing function. DMOEA-εC selects one of the objectives as the main objective function and converts the other nonmain objectives into constraints by giving them upper bound coefficients. Based on N evenly distributed upper bound vectors, an MOP is decomposed into N scalar constrained subproblems. Similarly, the neighborhood of each subproblem is defined according to the Euclidean distances from the upper bound vector corresponding to the subproblem to other upper bound vectors for DMOEA-εC. N subproblems are optimized using the neighbor information in parallel. However, there are three special mechanisms in DMOEA-εC for improving the performance. First, Since DMOEA-εC tends to retain feasible solutions for each subproblem, this will be bad for the optimization of the main objective function. Thus a main objective alternation strategy is proposed. In order to tackle problems induced by the main objective alternation strategy, a solution-to-subproblem matching procedure is proposed to place the nearest solution to each subproblem. Lastly, a subproblem-to-solution matching procedure is used find a subproblem with the minimum constraint violation value for the newly generated solution.

2) Main Differences Between DMOEA-εC and AUGMECON: Both DMOEA-εC and variants of AUGMECON convert an MOP into a series of scalar constrained subproblems, but they handle these subproblems in totally different ways. Specifically, variants of AUGMECON optimize them one by one, while DMOEA-εC solves them collaboratively by using the neighbor information. Furthermore, it is worth mentioning that the exact or approximated ideal point and nadir point are needed in advance in various variants of AUGMECON, while there is no such limitation in DMOEA-εC. Actually, the ideal point and the nadir point are updated in each generation.

3) Computational Complexity Analysis: The time complexity analysis of DMOEA-εC is presented in Table I. In summary, the time complexity of DMOEA-εC is

TABLE I
TIME COMPLEXITY ANALYSIS OF DMOEA-εC

Procedure	Time complexity
Solution-to-Subproblem Matching	$O(m \cdot N)$
Extract nondominated solutions	$O(m \cdot N^2)$
Dynamic Resource Allocation	$O(N)$
Generate a new solution	$O(m)$
Update neighborhood solutions	$O(m \cdot T)$
Subproblem-to-Solution Matching	$O(m)$
Update EP using the Algorithm 5	$O(m \cdot S^2) \approx O(m \cdot N^2)$

$O(m \cdot S^2) \approx O(m \cdot N^2)$. Besides, the time complexity of MOEA/D and multiobjective evolutionary algorithm based on decomposition with dynamic resource allocation (MOEA/D-DRA) is $O(m \cdot N \cdot T)$ [15], [17]. And the time complexity of multiobjective evolutionary algorithm based on decomposition with adaptive weight adjustment (MOEA/D-AWA) is $O(m \cdot N^2 \cdot (T + \text{nus}))$ [20]. Compared with MOEA/D and its variants, DMOEA-εC allocates additional computational resources to the solution-to-subproblem matching procedure, the subproblem-to-solution matching procedure and the farthest candidate approach when pruning EP. The computational resources spent on two matching procedures are negligible. And the main time complexity is introduced by the farthest-candidate approach described in Algorithm 6.

IV. COMPARISONS ON MULTIOBJECTIVE CONTINUOUS TEST INSTANCES

This section is devoted to the experimental design for investigating the performance of DMOEA-εC on continuous test instances. First, six state-of-the-art MOEAs for comparison, three performance metrics, and three groups of 34 continuous test instances with various characteristics used here are given. Then parameter settings adopted in this paper are provided. Finally, the experimental results are illustrated.

A. MOEAs for Comparison

Six state-of-the-art MOEAs are considered as competitive candidates, including MOEA/D [15], MOEA/D-DRA [17], MOEA/D-AWA [20], self-organizing multiobjective evolutionary algorithm (SMEA) [53], cellular genetic algorithm for multiobjective optimization (MOCell) [54], and multiobjective optimization particle swarm optimization (SMPSO) [55]. MOEA/D, MOEA/D-DRA, and MOEA/D-AWA are all based on decomposition and perform better than a number of popular algorithms. SMEA is a newly proposed competitive MOEA. It is based on the self-organizing mapping method (SOM) and the neighborhood relationship concept. SMEA has been compared with some advanced multiobjective evolutionary methods and has shown its advantages over competitive approaches. MOCell and SMPSO are cellular-based and particle swarm optimization-based multiobjective solvers, respectively. And they both can obtain competitive results on ZDT test suites.

We use the implementation of MOEA/D, MOEA/D-DRA, MOCell, and SMPSO provided by the jMetal framework [56].[1] Besides, DMOEA-εC[2] and SMEA are implemented in MATLAB, while MOEA/D-AWA is implemented in C++.[3] All of them are executed on the same computer.

[1] Download from the website https://github.com/jMetal/jMetal.

B. Performance Metrics

Three commonly used performance metrics, i.e., inverted generational distance (IGD) [57], hypervolume (HV) [10], and additive ϵ-indicator ($I_{\epsilon+}$) [58] are employed to evaluate the performance of all compared algorithms.

The IGD metric measures the average distance from a set of uniformly distributed Pareto optimal points over the PF P^* to the approximation set P. It can be formulated as

$$\text{IGD}(P^*, P) = \frac{\sum_{x^* \in P^*} d(x^*, P)}{|P^*|}$$

where $d(x^*, P)$ is the minimal Euclidean distance between x^* and any point in P, and $|P^*|$ is the cardinality of P^*. If $|P^*|$ is large enough to represent the PF very well, IGD(P^*, P) could measure both diversity and convergence of P in a sense. A smaller IGD value indicates a better P.

The HV metric measures the size of the objective space dominated by the solutions in P and bounded by the reference point $\mathbf{r}$. It is defined as

$$\text{HV}(P, \mathbf{r}) = \text{VOL}\left(\bigcup_{x \in P} \left[f_1(x), r_1\right] \times \cdots \times \left[f_m(x), r_m\right]\right)$$

where $\mathbf{r} = (r_1, \ldots, r_m)$ is a reference point in the objective space dominated by any Pareto optimal point, and VOL$(\cdot)$ is the Lebesgue measure. A larger HV value implies a better P.

The additive ϵ-indicator ($I_{\epsilon+}$) gives the factor by which an approximation set P is worse than the PF P^* with respect to all objectives. It is formulated as

$$I_{\epsilon+}\left(P, P^*\right) = \inf_{\epsilon \in R^+}\left\{\forall z^2 \in P^*, \exists z^1 \in P : z^1 \succeq_{\epsilon+} z^2\right\}$$

where $z^1 \succeq_{\epsilon+} z^2$ if and only if $\forall i \in \{1, 2, \ldots, m\} : z_i^1 \leq \epsilon + z_i^2$. It measures the convergence of the approximation set P. A smaller $I_{\epsilon+}$ value indicates a better P.

C. Multiobjective Continuous Test Instances

The ZDT test instances, tri-objective DTLZ test instances [59], UF test suites [60] which are part of the CEC2009 MOP test instances, LZ test suites [16] with complicated PS shapes, and bi-objective WFG test suites [61] with complicated PF shapes are adopted for comparing DMOEA-εC with other six MOEAs.

D. Parameter Settings

1) Public Parameter Settings: For a fair comparison, the choice of parameters are the same as the comparison algorithms. Specifically, the population size is set to $N = 100$ for the ZDT and bi-objective WFG problems. Due to the differences in algorithmic frameworks, N is set to 351, 306, 324,

[2] The source codes of DMOEA-εC can be downloaded from the website http://pris.bit.edu.cn/home/people/OtherStaff/xinbin.htm.

[3] The source codes of MOEA/D-AWA and SMEA are obtained from their authors.

and 300 for the tri-objective DTLZ problem for MOEA/D and its variants, MOCell, DMOEA-εC, and the remaining algorithms, respectively.[4] As to the UF problems, population size is set to $N = 600$ for bi-objective and $N = 1000$ for tri-objective. Since four out of nine LZ test problems are included in the UF test suites, for the remain LZ problems the population size is set to $N = 300$. For a fair comparison, an EP with the size of $S = \lfloor 1.5N \rfloor$ is added to the comparison algorithms. Besides, the DE operator and Gaussian mutation are used in solving ZDT, DTLZ, and WFG test problems. The DE operator and polynomial mutation are adopted in solving UF and LZ test problems. Moreover, control parameters for these reproduction operators are the same as those claimed in comparative MOEAs.[5] All compared algorithms stop when the number of function evaluations reaches the maximum number. For a fair comparison, in accordance with the parameter settings in comparison algorithms, the maximum number of function evaluations is set to 50 000 for the ZDT problems, 75 000 for the tri-objective DTLZ problems, 300 000 for the UF problems, 150 000 for the remain LZ problems, and 45 000 for the bi-objective WFG problems. Finally, each algorithm is executed 30 times independently on each instance.

2) Parameter Settings in MOEA/D, MOEA/D-DRA, and MOEA/D-AWA: Parameter settings adopted here are the same as those claimed in [15], [17], and [20].

- Neighborhood size: $T = \lfloor 0.1N \rfloor$.
- Probability of selecting mate solutions from neighborhood: $\delta = 0.9$.
- Maximal number of replacement: $n_r = \lfloor 0.01N \rfloor$.

In addition to the above mentioned common parameters, the iteration interval of utilizing the dynamic resource allocation strategy is set as *DRA_interval* $= 50$ for MOEA/D-DRA and MOEA/D-AWA. For MOEA/D-AWA, the maximal number of subproblems adjusted is set as nus $= \lfloor 0.05N \rfloor$. The parameter *rate_evol* is set to 0.8.

3) Parameter Settings in SMEA: Parameter settings in SMEA adopted here are the same as those claimed in [53].

- SOM structures: 1-D structure 1×100 for bi-objective MOPs.
- Initial learning rate: $\tau_0 = 0.9$.
- Neighborhood size: $T = 5$.
- Probability of selecting mate solutions from the neighborhood $\delta = 0.7$.
- Maximal number of replacement: $n_r = 1$.

4) Parameter Settings in MOCell: Parameter settings in MOCell adopted here are the same as those claimed in [54].

- Neighborhood: one-hop neighbors (eight surrounding solutions).
- Selection of parents: binary tournament + binary tournament.
- Feedback: 20 individuals.

5) Parameter Settings in SMPSO: Parameter settings in SMPSO adopted here are the same as those claimed in [55].

- The inertia weight: $w = 0.1$.
- The range of C_1 and C_2: $[1.5, 2.5]$.

6) Parameter Settings in DMOEA-εC: When compared with each algorithm, parameter settings in DMOEA-εC are set the same as each competitor. Besides, the setting of IN_m varies with different test problems. IN_m is set to $\lfloor 50\% \cdot$(number of iterations)$\rfloor$ for ZDT problems, $\lfloor 20\% \cdot$(number of iterations)$\rfloor$ for UF, LZ, and WFG problems and $\lfloor 10\% \cdot$(number of iterations)$\rfloor$ for DTLZ problems based on the parameter sensitivity analysis in Section VI-A.

TABLE II
REFERENCE POINTS OF TEST INSTANCES

Instance	Reference point
ZDT1-ZDT4, ZDT6, UF1-UF7, LZ1, LZ3, LZ4, LZ7, LZ9	(1.1, 1.1)
DLTZ1-DTLZ4, UF8-UF10	(1.1, 1.1, 1.1)
DTLZ6	(1.1, 1.1, 6.6)
WFG1-WFG9	(2.2, 4.4)

E. Experimental Results

This part of experiments are designed to study the effectiveness of DMOEA-εC on continuous MOPs. At first, the classical ZDT and DTLZ test suites are investigated. Performance of DMOEA-εC on more complicated test instances will be studied later.

In calculating the performance metrics, 100 nondominated solutions selected from the combination of the evolving population and EP using the farthest-candidate approach (Algorithm 5) are used in the case of ZDT and WFG problems, and 300 in the case of DTLZ problems. Similarly, for UF and LZ test problems 100 and 150 nondominated solutions are selected and used for the performance metrics calculation for two-objective and tri-objective problems, respectively.

With the purpose of calculating the IGD metric value, P^* is chosen to be a set of 500 uniformly distributed points along the true PF for ZDT problems, and 1024 points for DTLZ instances. As to two-objective UF and LZ problems, a set of 1000 uniformly distributed points along the true PF are chosen as P^* except that 21 uniformly distributed points are chosen as P^* for UF5. And for tri-objective UF test problems, P^* is chosen to be a set of 10 000 uniformly distributed points along the true PF. P^* that used for computing IGD metrics for WFG problems is the same as in [53].

Besides, In order to compute the HV metric value, the reference point is set as 1.1 times the true nadir point. Specifically, reference points of different test instances are illustrated in Table II.

The means and standard deviations of IGD, HV, and $I_{\epsilon+}$ metric values over 30 independent runs of each algorithm on 34 test instances are shown in Tables III–V, respectively. The mean HV (IGD/$I_{\epsilon+}$) values for each instance are sorted in descending (ascending) order, and the numbers in the square

[4]Since the algorithmic frameworks of the proposed DMOEA-εC and comparison algorithms are all different, the calculations of the population size N are conducted differently. The population size of MOEA/D is determined by the number of weight vectors $N = C_{H+m-1}^{m-1}$ (m is the number of objectives and H is a controlled parameter). In order to have a comparable population size, for the tri-objective problems we set $H = 25$, thus $N = 351$. The population size of MOCell is determined by the number of cellular grids. For the tri-objective problems we set $N = 17 \times 18 = 306$. The population size of DMOEA-εC is determined by $N = q^{m-1}$ (q is a controlled parameter). For the tri-objective problems we set $q = 18$, thus $N = 324$. For the remaining algorithms, we set $N = 300$.

[5]CR is changed to 0.9 only for ZDT problems.

TABLE III
STATISTICAL RESULTS OF SEVEN ALGORITHMS OVER 30 INDEPENDENT RUNS ON THE 34 INSTANCES IN TERMS OF IGD METRICS

Instance	MOEA/D	MOEA/D-DRA	MOEA/D-AWA	SMEA	MOCell	SMPSO	DMOEA-εC
ZDT1	3.800E-03† [4](0.000E-07)	4.903E-03† [6](8.829E-04)	4.430E-03† [5](7.940E-05)	2.218E-02† [7](7.060E-03)	3.700E-03≈ [2](2.631E-05)	**3.653E-03**§ [1](**5.07E-05**)	3.763E-03[3](6.690E-05)
ZDT2	4.000E-03† [4](1.830E-05)	5.760E-03† [6](1.388E-03)	4.463E-03† [5](6.690E-05)	3.799E-02† [7](1.393E-02)	3.800E-03≈ [2](3.710E-05)	**3.787E-03**§ [1](**3.460E-05**)	3.800E-03[3](7.120E-05)
ZDT3	1.787E-02† [6](1.255E-02)	1.101E-02† [5](5.683E-04)	5.867E-03† [3](1.422E-04)	3.587E-02† [7](2.713E-02)	7.293E-03† [4](9.122E-03)	**4.313E-03**§ [1](**3.457E-05**)	5.141E-03[2](1.696E-04)
ZDT4	3.890E-03§ [2](7.588E-05)	3.722E-02† [6](5.545E-02)	5.867E-03† [5](1.432E-05)	3.110E-00† [7](1.745E-00)	3.953E-03≈ [3](2.932E-04)	**3.707E-03**§ [1](**2.537E-05**)	4.060E-03[4](2.799E-05)
ZDT6	3.330E-03† [6](0.000E-07)	3.100E-03† [5](8.820E-05)	4.173E-03† [7](2.518E-04)	3.003E-03≈ [4](1.825E-05)	3.000E-03≈ [2](1.323E-05)	**3.000E-03**≈ [1](**1.322E-05**)	3.000E-03[3](8.050E-05)
DTLZ1	5.312E-02† [6](1.111E-03)	**2.376E-02**§ [1](**2.596E-04**)	3.056E-02† [3](8.764E-04)	8.434E-01† [7](1.004E-00)	3.819E-02† [5](5.539E-03)	3.396E-02† [4](8.261E-04)	2.476E-02[2](9.966E-04)
DTLZ2	6.684E-02† [6](2.845E-04)	3.075E-02≈ [2](3.893E-04)	**2.998E-02**§ [1](**4.145E-04**)	7.012E-02† [7](7.213E-04)	3.935E-02† [4](1.218E-03)	4.014E-02† [5](1.299E-03)	3.160E-02[3](8.819E-04)
DTLZ3	6.697E-02† [5](5.616E-04)	5.655E-02† [4](1.089E-01)	3.131E-02† [2](1.494E-03)	3.467E-00† [7](5.957E-00)	8.563E-02† [6](2.776E-02)	5.336E-02† [3](5.332E-02)	**3.068E-02[1](7.981E-04)**
DTLZ4	6.703E-02† [6](8.167E-04)	4.963E-02† [4](5.899E-02)	**2.997E-02**≈ [1](**2.578E-04**)	2.154E-01† [7](8.791E-02)	3.670E-02† [3](8.424E-04)	6.267E-02† [5](3.097E-02)	3.020E-02[2](1.021E-03)
DTLZ6	3.875E-01† [7](1.834E-01)	1.578E-01† [5](1.987E-01)	**3.334E-02**≈ [1](**6.906E-04**)	7.852E-02† [4](5.004E-02)	2.044E-01† [6](2.369E-01)	4.541E-02† [3](2.876E-03)	3.360E-02[2](5.777E-04)
UF1(LZ2)	9.397E-03† [5](1.623E-03)	4.423E-03≈ [2](9.610E-05)	5.700E-03≈ [4](1.907E-04)	5.123E-03† [3](9.497E-04)	1.059E-01† [7](2.650E-02)	6.175E-02† [6](9.723E-03)	**4.407E-03[1](9.870E-05)**
UF2(LZ5)	1.436E-02† [5](3.333E-03)	**6.227E-03**§ [1](**1.205E-03**)	8.660E-03† [3](1.322E-03)	1.151E-02† [4](2.263E-03)	4.937E-02† [7](2.582E-02)	2.378E-02† [6](1.930E-03)	6.503E-03[2](5.229E-04)
UF3(LZ8)	2.107E-02† [5](1.990E-02)	7.951E-03† [2](5.675E-03)	1.046E-02† [3](6.849E-03)	1.458E-02† [4](9.682E-03)	3.032E-01† [7](2.079E-02)	1.118E-01† [6](3.447E-02)	**7.209E-03[1](1.355E-02)**
UF4	6.907E-02† [6](7.412E-03)	6.078E-02≈ [4](4.158E-03)	6.469E-02† [5](3.061E-03)	9.769E-02† [7](8.374E-03)	**4.503E-02**§ [1](**1.756E-03**)	5.142E-02§ [2](2.829E-03)	6.017E-02[3](5.017E-03)
UF5	3.355E-01§ [3](1.409E-01)	3.190E-01§ [2](1.228E-01)	**1.224E-01**§ [1](**2.519E-01**)	3.941E-01† [6](8.814E-02)	3.452E-01≈ [4](9.993E-02)	1.872E-00† [7](5.442E-01)	3.845E-01[5](1.251E-01)
UF6	3.096E-01† [5](2.215E-01)	1.765E-01§ [3](1.124E-01)	1.560E-01§ [2](2.930E-03)	**1.024E-01**§ [1](**5.955E-02**)	4.309E-01† [6](1.990E-01)	4.484E-01† [7](9.853E-02)	2.662E-01[4](2.012E-01)
UF7	7.633E-03† [5](3.333E-03)	4.435E-03≈ [2](1.537E-04)	6.430E-03† [4](3.831E-04)	6.413E-03† [3](6.056E-04)	2.954E-01† [7](1.627E-01)	2.253E-02† [6](2.439E-03)	**4.079E-03[1](1.269E-03)**
UF8(LZ6)	7.988E-02† [3](1.389E-02)	6.359E-02† [2](1.144E-02)	1.257E-01† [4](3.783E-02)	2.004E-01† [6](8.619E-02)	2.079E-01† [7](5.437E-02)	1.911E-01† [5](2.907E-02)	**5.284E-02[1](1.058E-02)**
UF9	1.353E-01† [3](4.758E-02)	1.034E-01† [2](5.370E-02)	1.677E-01† [4](2.248E-02)	1.694E-01† [5](8.743E-02)	2.067E-01† [7](6.252E-02)	1.911E-01† [6](2.077E-02)	**4.292E-02[1](1.660E-02)**
UF10	4.490E-01† [5](7.552E-02)	4.249E-01† [4](7.036E-02)	4.833E-01† [6](4.563E-02)	1.039E-00† [7](2.187E-01)	4.151E-01† [3](1.108E-01)	2.756E-01† [2](2.739E-02)	**2.240E-02[1](7.361E-02)**
LZ1	**1.290E-03**≈ [1](**3.724E-05**)	1.664E-03† [5](9.812E-05)	1.486E-03† [4](8.997E-05)	3.613E-03† [7](2.342E-05)	2.367E-03† [6](1.454E-04)	1.431E-03† [3](3.391E-05)	1.395E-03[2](1.230E-03)
LZ3	6.122E-03† [4](3.810E-03)	3.693E-03† [3](6.772E-04)	2.380E-03≈ [2](4.472E-05)	7.129E-03† [5](2.569E-03)	6.570E-02† [7](2.978E-02)	3.627E-02† [6](5.754E-03)	**2.372E-03[1](2.794E-03)**
LZ4	7.265E-03† [5](3.758E-03)	3.075E-03† [3](4.275E-04)	2.733E-03† [2](1.211E-04)	3.979E-03† [4](1.201E-04)	4.570E-02† [7](4.098E-03)	3.740E-02† [6](3.865E-03)	**1.977E-03[1](3.780E-03)**
LZ7	3.056E-02† [5](4.478E-02)	2.599E-03† [3](4.286E-04)	2.040E-03† [2](2.608E-04)	4.065E-03† [4](1.851E-04)	4.070E-01† [7](1.2453E-01)	1.083E-01† [6](5.835E-02)	**1.479E-03[1](2.691E-03)**
LZ9	9.770E-03† [3](4.729E-03)	4.220E-03† [2](6.390E-04)	7.264E-02† [5](5.764E-03)	1.504E-02† [4](3.084E-03)	1.475E-01† [7](6.661E-02)	7.330E-02† [6](1.615E-02)	**3.192E-03[1](4.437E-02)**
WFG1	1.202E-00† [4](2.487E-02)	1.206E-00† [5](2.728E-02)	1.181E-00† [3](3.451E-02)	1.555E-00† [7](1.003E-01)	**9.814E-01**≈ [1](**1.804E-01**)	1.237E-00† [6](5.391E-03)	1.063E-00[2](1.575E-02)
WFG2	7.922E-02† [5](1.301E-02)	7.440E-02† [3](1.660E-02)	1.336E-01† [7](6.518E-02)	**1.492E-02**≈ [1](**9.970E-04**)	7.815E-02† [4](2.234E-02)	8.653E-02† [6](1.651E-02)	1.505E-02[2](8.403E-04)
WFG3	4.259E-02† [5](7.320E-03)	3.449E-02† [3](5.147E-03)	6.601E-02† [6](2.324E-02)	**1.182E-02**§ [1](**1.177E-04**)	3.726E-02† [4](1.399E-02)	6.742E-02† [7](1.566E-02)	1.391E-02[2](1.199E-03)
WFG4	1.063E-01† [7](8.903E-03)	9.827E-02† [6](9.393E-03)	3.355E-02§ [2](9.908E-03)	8.401E-02† [5](5.844E-03)	**1.862E-02**§ [1](**1.942E-03**)	7.037E-02† [4](2.725E-03)	4.008E-02[3](7.401E-03)
WFG5	6.749E-02† [5](3.991E-04)	6.796E-02† [6](8.381E-04)	6.812E-02† [7](2.810E-04)	6.662E-02† [3](1.053E-04)	6.694E-02† [4](1.612E-05)	6.637E-02≈ [2](5.960E-05)	**6.633E-02[1](7.970E-04)**
WFG6	8.576E-02† [7](1.365E-02)	8.278E-02† [6](1.747E-02)	5.481E-02† [5](1.633E-02)	**1.356E-02**§ [1](**3.048E-04**)	5.209E-02† [4](9.408E-03)	2.614E-02≈ [3](1.916E-03)	2.576E-02[2](2.087E-04)
WFG7	2.417E-02† [6](1.276E-03)	2.310E-02† [5](1.297E-03)	1.935E-02† [3](1.705E-03)	1.144E-02† [2](2.596E-04)	1.309E-02† [4](2.303E-04)	3.906E-02† [7](7.782E-03)	**1.082E-02[1](7.213E-04)**
WFG8	1.268E-01† [5](9.854E-03)	1.230E-01† [4](1.360E-02)	1.431E-01† [6](6.412E-02)	**3.323E-02**§ [1](**8.009E-03**)	8.057E-02† [3](7.866E-03)	1.442E-01† [7](1.630E-02)	3.604E-02[2](4.116E-03)
WFG9	8.308E-02† [7](1.802E-02)	7.564E-02† [6](2.411E-02)	5.477E-02† [5](2.333E-02)	2.900E-02≈ [2](4.801E-02)	3.723E-02† [4](2.558E-02)	**1.404E-02**§ [1](**3.439E-04**)	3.054E-02[3](1.664E-03)
† / § / ≈	31/2/1	26/4/4	26/4/4	27/4/3	26/2/6	25/6/3	

brackets are their ranks. The Wilcoxon's rank sum test at a 5% significance level is conducted to test the significance of differences between the mean metric values yielded by DMOEA-εC and comparison algorithms. The numbers in parentheses are the standard deviations. †, §, and ≈ indicate the performance of the DMOEA-εC is better than, worse than, and similar to that of the comparison algorithm according to the Wilcoxon's rank sum test, respectively. The bold data in the table are the best mean metric values for each instance. Besides, Table VI summarizes the overall performance, including the mean ranks and statistical results obtained via the Wilcoxon's rank sum test, of seven algorithms on 34 instances in terms of three metric values.

1) Experimental Results on ZDT and DTLZ Test Suites: The mathematical descriptions of these test problems and true PFs can be found in [59] and [62].

As can be seen in Tables III–V, in terms of IGD metric values, SMPSO shows a significant advantage over DMOEA-εC on ZDT test problems, and the performance of DMOEA-εC is no worse than that of comparison algorithms on all DTLZ problems except the DTLZ1 and DTLZ2 on which MOEA/D and MOEA/D-AWA show better performance, respectively. For HV, DMOEA-εC shows significant superiority over others on both ZDT and DTLZ problems except ZDT3, ZDT4, and DTLZ2. As to $I_{\epsilon+}$, SMPSO outperforms DMOEA-εC significantly on all ZDT problems, and DMOEA-εC shows a clear advantage over others on all DTLZ problems. Besides, both DMOEA-εC and MOEA/D-AWA perform better than any other algorithms on the majority of DTLZ problems in terms of all three metrics. The difference lies in that DMOEA-εC tends to obtain solutions with good convergence, whereas MOEA/D-AWA does well in maintaining good uniformity. In summary, the performance of DMOEA-εC and variants of MOEA/D on ZDT test instances is not as promising as SMPSO. However, the superiority of the proposed DMOEA-εC over comparison algorithms is highlighted on DTLZ problems.

2) Experimental Results on Instances With Complicated PS Shapes (UF and LZ Test Suites): The UF instances come from a set of unconstrained MOP test problems suggested in the CEC2009 contest. The UF and LZ problems involve a strong linkage in variables among the Pareto optimal solutions, thereby posing a great challenge for MOEAs. Their mathematical descriptions and true PFs can be found in [16] and [60].

The experimental results on the bi-objective UF problems in Tables III–V demonstrate that DMOEA-εC shows competitive performance on bi-objective UF test problems except UF4–UF6 in terms of IGD, HV, and $I_{\epsilon+}$ metrics. As to the UF4 problem, MOCell performs best among all algorithms in terms of three metric values. Nevertheless, even the best performer, MOCell, cannot obtain a good approximation of PF. However, as for the tri-objective UF test problems, DMOEA-εC performs similarly or outperforms comparison algorithms significantly in terms of the three metrics.

As to the remaining LZ problems, DMOEA-εC outperforms or performs competitively against competitors in terms of IGD values. For HV, DMOEA-εC shows significant superiority over others on all remaining LZ problems except LZ9. As to $I_{\epsilon+}$, DMOEA-εC has clear advantages over comparison

TABLE IV
STATISTICAL RESULTS OF SEVEN ALGORITHMS OVER 30 INDEPENDENT RUNS ON THE 34 INSTANCES IN TERMS OF HV METRICS

Instance	MOEA/D	MOEA/D-DRA	MOEA/D-AWA	SMEA	MOCell	SMPSO	DMOEA-εC
ZDT1	8.686E-01† [4](1.541E-02)	8.679E-01† [5](8.407E-03)	8.710E-01† [3](1.607E-02)	8.380E-01† [6](1.385E-02)	8.710E-01† [2](6.023E-03)	7.922E-01† [7](6.584E-03)	**8.727E-01[1](1.541E-02)**
ZDT2	5.421E-01≈ [2](1.091E-02)	5.363E-01† [6](7.542E-03)	5.421E-01≈ [3](1.751E-02)	4.783E-01† [7](2.546E-02)	5.393E-01† [4](8.484E-03)	5.389E-01† [5](9.047E-03)	**5.434E-01[1](1.405E-02)**
ZDT3	1.003E-00≈ [4](1.124E-02)	9.988E-01† [6](6.994E-03)	1.006E-00§ [3](1.111E-02)	9.760E-01† [7](2.672E-02)	**1.008E-00**§ [1](**6.811E-03**)	1.006E-00§ [2](7.988E-03)	1.003E-00[5](1.304E-02)
ZDT4	**8.896E-01**§ [1](**1.392E-02**)	8.087E-01† [6](8.906E-03)	8.660E-01≈ [5](1.121E-02)	1.575E-01† [7](1.643E-01)	8.705E-01§ [3](6.904E-03)	8.707E-01§ [2](8.279E-03)	8.668E-01[4](4.691E-02)
ZDT6	5.042E-01† [5](1.635E-02)	5.045E-01† [4](7.675E-03)	4.974E-01≈ [7](1.919E-02)	5.055E-01† [2](9.793E-03)	5.037E-01§ [6](8.735E-03)	5.054E-01† [3](7.660E-03)	**5.086E-01[1](1.469E-02)**
DTLZ1	7.062E-01† [7](1.742E-02)	1.306E-00≈ [2](2.233E-03)	1.122E-00† [3](5.211E-02)	9.318E-01† [6](3.779E-01)	1.111E-00† [5](1.341E-02)	1.120E-00† [4](6.384E-03)	**1.307E-00[1](4.458E-03)**
DTLZ2	7.124E-01† [7](1.232E-02)	**7.717E-01**§ [1](**9.340E-03**)	7.773E-01§ [2](1.373E-02)	7.591E-01§ [3](7.929E-03)	7.406E-01† [5](1.041E-02)	7.351E-01† [6](1.226E-02)	7.435E-01[4](1.528E-02)
DTLZ3	7.006E-01† [4](1.572E-02)	7.295E-01† [3](1.434E-01)	7.596E-01† [2](1.906E-02)	6.628E-01† [6](1.511E-03)	6.745E-01† [5](1.185E-01)	6.515E-01† [7](7.736E-02)	**7.696E-01[1](1.372E-02)**
DTLZ4	7.109E-01† [6](1.528E-02)	7.618E-01† [3](3.022E-02)	7.701E-01† [2](4.222E-02)	6.830E-01† [7](4.605E-02)	7.295E-01† [5](1.155E-01)	7.372E-01† [4](1.627E-02)	**7.840E-01[1](1.656E-02)**
DTLZ6	2.091E-00† [7](2.924E-02)	2.412E-00† [5](2.683E-01)	2.647E-00† [2](1.561E-02)	2.609E-00† [3](1.131E-01)	2.348E-00† [6](3.287E-01)	2.582E-00† [4](5.797E-02)	**3.118E-00[1](3.344E-02)**
UF1(LZ2)	8.599E-01† [5](9.680E-03)	8.670E-01† [4](1.759E-01)	8.675E-01† [3](6.371E-03)	8.697E-01† [2](5.929E-03)	7.104E-01† [7](3.601E-02)	7.625E-01† [6](2.377E-02)	**8.741E-01[1](1.434E-02)**
UF2(LZ5)	8.554E-01† [5](9.815E-03)	8.675E-01≈ [2](1.720E-02)	8.614E-01† [3](7.108E-03)	8.591E-01† [4](6.946E-03)	8.257E-01† [7](2.319E-02)	8.397E-01† [6](8.459E-02)	**8.700E-01[1](1.382E-02)**
UF3(LZ8)	8.428E-01† [5](2.811E-02)	8.502E-01† [4](4.456E-02)	8.546E-01≈ [3](1.767E-02)	**1.039E-00**§ [1](**8.405E-03**)	4.617E-01† [7](3.578E-02)	7.072E-01† [6](4.944E-02)	8.687E-01[2](1.633E-02)
UF4	4.234E-01† [6](1.298E-02)	4.371E-01† [4](1.611E-02)	4.322E-01† [5](9.799E-03)	3.717E-01† [7](1.666E-02)	**4.698E-01**§ [1](**5.862E-03**)	4.536E-01≈ [2](9.896E-03)	4.436E-01[3](1.965E-02)
UF5	2.206E-01§ [3](9.592E-02)	2.749E-01§ [2](8.351E-02)	**4.961E-01**§ [1](**3.461E-02**)	1.288E-01≈ [5](8.754E-02)	2.949E-01§ [4](7.512E-02)	3.999E-02† [7](6.088E-03)	1.256E-01[6](4.731E-02)
UF6	3.376E-01§ [3](1.055E-01)	3.170E-01≈ [6](1.194E-01)	**5.055E-01**§ [1](**6.352E-03**)	4.263E-01§ [2](5.698E-02)	3.269E-01§ [4](1.585E-01)	1.068E-01† [7](7.217E-02)	3.1704E-01[5](1.061E-01)
UF7	6.939E-01† [5](9.662E-03)	**7.056E-01**≈ [1](**1.520E-02**)	7.024E-01† [3](8.854E-03)	6.983E-01† [4](7.392E-03)	4.227E-01† [7](1.279E-01)	6.693E-01† [6](8.930E-03)	7.054E-01[2](1.701E-02)
UF8(LZ6)	6.558E-01† [3](2.293E-02)	**7.250E-01**§ [1](**1.602E-02**)	5.618E-01† [5](7.330E-02)	6.050E-01† [4](5.928E-02)	4.649E-01† [6](1.407E-01)	4.323E-01† [7](5.121E-02)	7.032E-01[2](2.423E-02)
UF9	8.901E-01† [3](7.054E-02)	9.149E-01† [2](4.842E-02)	8.252E-01† [5](3.913E-02)	8.582E-01† [4](7.610E-02)	7.171E-01† [6](1.135E-01)	5.109E-01† [7](1.118E-01)	**1.019E-00[1](5.058E-02)**
UF10	1.446E-01† [5](4.563E-02)	1.956E-01† [3](3.304E-02)	1.377E-01† [6](2.136E-02)	3.284E-03† [7](5.632E-03)	1.618E-01† [4](5.557E-02)	3.171E-01† [2](4.974E-02)	**6.088E-01[1](9.056E-02)**
LZ1	8.762E-01≈ [2](7.136E-03)	8.732E-01† [6](8.086E-03)	8.747E-01† [4](7.732E-02)	8.709E-01† [7](9.450E-03)	8.746E-01† [5](6.578E-03)	8.749E-01† [3](1.011E-02)	**8.797E-01[1](4.100E-03)**
LZ3	8.664E-01† [4](9.613E-03)	8.701E-01† [3](1.132E-02)	8.738E-01† [2](1.197E-02)	8.661E-01† [5](8.253E-03)	7.982E-01† [7](2.010E-02)	8.250E-01† [6](1.354E-02)	**8.760E-01[1](8.710E-03)**
LZ4	8.647E-01† [5](8.462E-03)	8.697E-01† [4](5.884E-03)	8.763E-01≈ [2](2.063E-03)	8.712E-01† [3](7.096E-03)	8.105E-01† [7](8.746E-03)	8.193E-01† [6](9.111E-03)	**8.794E-01[1](7.621E-03)**
LZ7	7.990E-01† [5](8.856E-02)	8.666E-01† [4](8.494E-03)	8.761E-01† [2](1.045E-02)	8.698E-01† [3](8.372E-03)	4.493E-01† [7](8.731E-03)	6.456E-01† [6](1.022E-01)	**8.806E-01[1](1.350E-02)**
LZ9	5.213E-01† [3](1.818E-02)	**5.340E-01**§ [1](**7.182E-03**)	4.207E-01† [5](5.506E-03)	5.118E-01† [4](1.219E-02)	3.573E-01† [7](4.996E-02)	4.116E-01† [6](1.963E-02)	5.322E-01[2](2.514E-02)
WFG1	1.787E-00† [5](6.072E-02)	1.803E-00† [4](8.322E-02)	0.000E-07† [7](0.000E-07)	6.271E-00† [2](2.483E-01)	3.672E-00† [3](3.882E-01)	1.620E-00† [6](5.651E-02)	**8.616E-00[1](1.512E-02)**
WFG2	5.843E-00† [4](7.759E-02)	5.905E-00† [3](9.227E-02)	5.531E-00† [7](2.657E-01)	6.025E-00≈ [2](1.853E-02)	5.799E-00† [5](1.053E-01)	5.652E-00† [6](8.879E-02)	**6.076E-00[1](1.311E-02)**
WFG3	5.430E-00† [5](7.174E-02)	5.464E-00≈ [3](5.647E-02)	5.253E-00† [7](1.693E-01)	**5.617E-00**§ [1](**1.874E-02**)	5.455E-00† [4](1.207E-01)	5.298E-00† [6](1.178E-01)	5.472E-00[2](1.541E-02)
WFG4	2.858E-00† [7](7.019E-02)	2.907E-00† [6](7.325E-02)	3.242E-00§ [2](8.669E-02)	2.984E-00† [5](2.042E-02)	**3.415E-00**§ [1](**1.564E-01**)	3.007E-00† [4](5.869E-02)	3.105E-00[3](1.645E-02)
WFG5	2.981E-00† [4](5.306E-02)	2.977E-00† [5](7.896E-02)	2.923E-00† [7](6.573E-02)	3.003E-00† [3](1.473E-02)	3.006E-00≈ [2](6.018E-02)	2.971E-00† [6](5.716E-02)	**3.025E-00[1](1.410E-02)**
WFG6	2.929E-00† [7](1.117E-01)	2.947E-00† [6](1.220E-01)	3.033E-00† [5](1.349E-01)	**3.373E-00**§ [1](**6.159E-02**)	3.124E-00† [4](8.839E-02)	3.251E-00† [3](7.279E-02)	3.303E-00[2](2.479E-02)
WFG7	3.285E-00† [6](6.356E-02)	3.293E-00† [4](7.889E-02)	3.285E-00† [5](5.917E-01)	3.334E-00† [3](1.670E-02)	3.351E-00≈ [2](7.057E-02)	3.188E-00† [7](8.228E-02)	**3.351E-00[1](1.622E-02)**
WFG8	2.692E-00† [5](7.609E-02)	2.719E-00† [4](7.545E-02)	2.513E-00† [7](2.350E-01)	3.228E-00† [2](1.672E-02)	2.919E-00† [3](7.276E-02)	2.584E-00† [6](1.027E-01)	**3.340E-00[1](1.605E-02)**
WFG9	2.911E-00† [7](1.347E-01)	2.932E-00† [6](1.460E-01)	3.000E-00† [5](1.299E-01)	3.220E-00≈ [2](2.468E-01)	3.128E-00† [4](1.389E-01)	**3.261E-00**§ [1](**4.866E-02**)	3.138E-00[3](2.067E-02)
† / § / ≈	28/3/3	25/4/5	24/5/5	26/5/3	25/7/2	30/3/1	

TABLE V
STATISTICAL RESULTS OF SEVEN ALGORITHMS OVER 30 INDEPENDENT RUNS ON THE 34 INSTANCES IN TERMS OF $I_{\epsilon+}$ METRICS

Instance	MOEA/D	MOEA/D-DRA	MOEA/D-AWA	SMEA	MOCell	SMPSO	DMOEA-εC
ZDT1	7.087E-03† [4](1.456E-04)	1.061E-02† [6](2.261E-03)	8.453E-03† [5](3.674E-04)	2.452E-02† [7](6.146E-03)	5.457E-03§ [2](5.361E-04)	**5.270E-03**§ [1](**1.643E-04**)	5.700E-03[3](8.120E-04)
ZDT2	6.057E-03† [4](5.683E-05)	1.047E-02† [6](3.972E-03)	9.193E-03† [5](4.770E-04)	7.655E-02† [7](3.144E-02)	5.683E-03≈ [2](1.890E-03)	**5.253E-03**§ [1](**1.224E-04**)	5.900E-03[3](7.784E-04)
ZDT3	9.549E-02† [6](1.349E-01)	1.894E-02≈ [3](3.546E-03)	1.093E-02§ [2](1.344E-03)	1.180E-01† [7](6.644E-02)	3.601E-02† [5](9.464E-02)	**5.090E-03**§ [1](**2.695E-04**)	2.101E-02[4](1.221E-02)
ZDT4	7.897E-03† [5](1.261E-03)	7.667E-02† [4](9.489E-02)	8.273E-03† [6](3.694E-04)	3.397E-00† [7](1.774E-00)	6.153E-03§ [2](6.426E-04)	**5.467E-03**§ [1](**1.988E-04**)	6.600E-03[3](1.994E-03)
ZDT6	4.900E-03† [5](8.822E-05)	5.167E-03† [6](3.315E-04)	8.683E-03† [7](5.025E-04)	4.640E-03† [3](2.027E-04)	**4.477E-03**§ [1](**1.501E-04**)	4.610E-03§ [2](2.123E-04)	4.747E-03[4](2.740E-03)
DTLZ1	1.181E-01† [6](7.448E-02)	4.287E-02† [2](2.819E-03)	5.317E-02† [3](5.187E-03)	9.886E-01† [7](1.162E-04)	9.002E-02† [5](1.904E-02)	7.102E-02† [4](4.549E-03)	**4.198E-02[1](3.655E-03)**
DTLZ2	1.381E-01† [7](3.328E-03)	4.932E-02† [4](1.382E-03)	3.871E-02† [2](3.434E-03)	4.684E-02† [3](1.509E-03)	8.159E-02† [6](1.215E-02)	8.083E-02† [5](1.102E-02)	**2.584E-02[1](6.647E-03)**
DTLZ3	1.361E-01† [4](8.566E-03)	5.635E-01† [6](2.480E-02)	4.253E-02† [2](4.258E-03)	4.369E-00† [7](5.322E-00)	2.095E-01† [5](2.598E-01)	1.129E-01† [3](1.282E-01)	**3.997E-02[1](3.283E-03)**
DTLZ4	1.319E-01† [6](1.014E-02)	6.386E-02† [3](5.045E-02)	3.974E-02≈ [2](5.793E-03)	1.975E-01† [7](9.000E-02)	1.011E-01† [5](1.699E-01)	8.883E-02† [4](2.348E-02)	**3.88E-02[1](2.764E-03)**
DTLZ6	1.198E-00† [7](7.085E-01)	5.076E-01† [5](7.204E-01)	5.926E-02† [2](6.048E-03)	1.033E-01† [4](2.193E-01)	6.816E-01† [6](8.319E-01)	8.884E-02† [3](1.593E-02)	**5.395E-02[1](6.029E-03)**
UF1(LZ2)	2.014E-02† [5](6.884E-03)	5.123E-03† [2](1.209E-03)	1.163E-02† [3](1.165E-03)	1.190E-02† [4](6.871E-03)	2.169E-01† [7](5.096E-02)	1.269E-01† [6](2.582E-02)	**3.680E-03[1](3.326E-04)**
UF2(LZ5)	6.313E-02† [5](1.510E-02)	2.634E-02≈ [2](8.849E-03)	2.756E-02† [3](4.562E-03)	4.924E-02† [4](1.432E-02)	1.464E-01† [7](6.597E-02)	6.684E-02† [6](6.506E-03)	**2.613E-02[1](4.614E-03)**
UF3(LZ8)	7.766E-02† [5](7.086E-02)	**1.737E-02**§ [1](**2.098E-02**)	2.706E-02§ [2](3.182E-02)	6.400E-02† [4](2.425E-02)	5.367E-01† [7](8.431E-02)	1.255E-01† [6](4.086E-02)	4.858E-02[3](7.014E-02)
UF4	8.050E-02† [6](5.829E-03)	7.650E-02† [4](7.211E-03)	7.685E-02† [5](6.577E-03)	1.106E-01† [7](7.107E-03)	**4.951E-02**§ [1](**7.064E-03**)	6.254E-02§ [2](7.117E-03)	7.123E-02[3](6.624E-03)
UF5	5.107E-01§ [3](1.079E-01)	4.754E-01§ [2](1.538E-01)	**2.535E-01**§ [1](**2.657E-02**)	5.795E-01§ [4](1.504E-01)	6.310E-01≈ [5](1.667E-01)	1.335E-00† [7](3.920E-01)	6.389E-01[6](1.932E-01)
UF6	5.275E-01§ [3](2.421E-01)	5.456E-01† [5](3.076E-01)	3.005E-01§ [2](1.534E-04)	**2.919E-01**§ [1](**1.441E-01**)	6.239E-01† [7](2.776E-01)	5.685E-01† [6](1.096E-01)	5.346E-01[4](2.961E-01)
UF7	4.689E-02† [5](3.799E-02)	**1.587E-02**§ [1](**2.044E-02**)	2.506E-02≈ [3](3.988E-03)	4.558E-02† [4](5.989E-03)	5.893E-01† [7](2.294E-01)	7.425E-02† [6](1.749E-02)	2.304E-02[2](2.342E-02)
UF8(LZ6)	2.423E-01† [3](2.493E-02)	**1.526E-02**§ [1](**1.685E-02**)	3.465E-01† [5](1.909E-01)	2.489E-01† [4](1.339E-01)	6.434E-01† [7](1.967E-01)	6.047E-01† [6](1.795E-01)	1.956E-01[2](3.391E-02)
UF9	3.628E-01† [3](1.126E-01)	3.793E-01† [4](1.291E-01)	4.090E-01† [6](7.238E-02)	3.237E-01† [2](1.674E-01)	3.962E-01† [5](1.563E-01)	5.152E-01† [7](7.479E-02)	**1.008E-01[1](1.408E-02)**
UF10	8.337E-01† [5](8.264E-02)	7.768E-01† [3](7.043E-02)	8.167E-01† [4](6.870E-02)	1.134E-00† [7](1.288E-01)	8.737E-01† [6](7.471E-02)	7.322E-01† [2](4.031E-02)	**3.241E-01[1](1.387E-01)**
LZ1	2.783E-03≈ [2](5.533E-04)	5.815E-03† [7](1.475E-03)	3.186E-03† [4](5.178E-04)	5.237E-03† [5](1.509E-04)	5.804E-03† [6](1.333E-03)	**2.596E-03**§ [1](**1.622E-04**)	2.883E-03[3](3.400E-03)
LZ3	4.121E-02† [5](2.267E-02)	1.622E-02† [3](8.724E-03)	**4.581E-03**§ [1](**1.789E-04**)	2.774E-02† [4](1.501E-02)	1.800E-01† [7](6.533E-02)	9.367E-02† [6](1.364E-02)	4.949E-03[2](1.264E-02)
LZ4	5.198E-02† [5](2.660E-02)	1.257E-02† [4](3.712E-03)	6.200E-03† [2](5.215E-04)	7.834E-03† [3](1.008E-03)	1.721E-01† [7](1.806E-02)	1.027E-01† [6](1.312E-02)	**3.194E-03[1](3.714E-03)**
LZ7	1.542E-01† [5](1.388E-01)	1.866E-02† [4](6.258E-03)	5.180E-03† [2](8.927E-04)	8.215E-03† [3](7.769E-04)	6.122E-01† [7](1.402E-01)	3.317E-01† [6](9.734E-02)	**3.274E-03[1](1.400E-03)**
LZ9	4.782E-02† [3](2.364E-02)	1.363E-02† [2](2.788E-03)	1.808E-01† [6](5.384E-04)	6.365E-02† [4](1.641E-02)	3.120E-01† [7](7.756E-02)	1.443E-01† [5](3.930E-02)	**5.366E-03[1](1.611E-03)**
WFG1	1.019E-00† [2](1.391E-02)	1.037E-00† [3](3.199E-02)	1.110E-00† [5](8.787E-02)	1.774E-00† [7](1.114E-01)	1.095E-00† [4](1.669E-01)	1.122E-01† [6](1.759E-02)	**9.076E-01[1](5.067E-02)**
WFG2	9.972E-02† [5](1.780E-02)	9.239E-02† [4](2.657E-02)	7.887E-01† [7](4.710E-01)	**1.405E-02**§ [1](**1.398E-03**)	6.008E-01† [6](2.822E-01)	9.064E-02† [3](1.495E-02)	2.315E-02[2](3.611E-03)
WFG3	7.691E-02† [6](1.511E-02)	7.099E-02† [5](1.532E-02)	1.044E-01† [7](2.709E-02)	**1.739E-02**§ [1](**6.155E-04**)	4.176E-02† [3](1.183E-02)	7.018E-02† [4](1.249E-02)	2.059E-02[2](3.297E-03)
WFG4	1.154E-01† [6](5.885E-03)	1.291E-01† [7](2.557E-02)	7.679E-02† [4](1.753E-02)	8.324E-02† [5](4.337E-03)	**2.658E-02**§ [1](**2.289E-03**)	7.346E-02† [3](5.022E-03)	5.235E-02[2](8.255E-03)
WFG5	7.331E-02† [5](1.130E-03)	7.484E-02† [6](6.406E-03)	7.975E-02† [7](3.784E-03)	6.331E-02† [2](3.303E-03)	6.400E-02† [4](5.813E-04)	6.355E-02≈ [3](1.346E-03)	**6.247E-02[1](4.948E-03)**
WFG6	9.071E-02† [6](8.034E-03)	9.049E-02† [5](7.789E-03)	1.431E-01† [7](4.489E-02)	**1.305E-02**§ [1](**8.537E-04**)	5.334E-02† [4](7.811E-03)	3.158E-02≈ [3](8.751E-03)	3.115E-02[2](1.794E-02)
WFG7	4.048E-02† [5](2.225E-03)	3.848E-02† [4](3.650E-03)	5.519E-02† [7](1.553E-02)	1.695E-02≈ [2](2.417E-03)	1.827E-02† [3](7.747E-04)	4.607E-02† [6](5.963E-03)	**1.623E-02[1](7.818E-04)**
WFG8	1.404E-01† [4](1.408E-02)	1.583E-01† [6](2.644E-02)	2.572E-01† [7](1.042E-01)	3.891E-02† [2](5.397E-03)	1.061E-01† [3](9.049E-03)	1.552E-01† [5](1.377E-02)	**2.020E-02[1](6.239E-03)**
WFG9	8.843E-02† [7](9.919E-03)	8.532E-02† [5](1.668E-02)	8.692E-02† [6](1.697E-02)	3.312E-02≈ [2](3.808E-02)	4.115E-02† [4](2.022E-02)	**1.992E-02**§ [1](**1.600E-03**)	3.470E-02[3](1.354E-03)
† / § / ≈	31/2/1	28/4/2	27/5/2	27/5/2	27/5/2	24/8/2	

algorithms on all remaining LZ problems except LZ1 and LZ3 on which the performance of DMOEA-εC is slightly worse than that of SMPSO and MOEA/D-AWA, respectively.

In conclusion, the experimental results on UF and LZ test suites indicate that the superiority of DMOEA-εC is significant on tri-objective UF problems and LZ problems,

TABLE VI
OVERALL PERFORMANCE OF SEVEN ALGORITHMS ON THE 34 INSTANCES IN TERMS OF IGD, HV, AND $I_{\epsilon+}$ METRICS

Instance	MOEA/D	MOEA/D-DRA	MOEA/D-AWA	SMEA	MOCell	SMPSO	DMOEA-εC
Mean Rank	4.7843	3.8431	3.9608	4.2745	4.6667	4.4804	1.9902
Total †/§/ ≈	90/7/5	79/12/11	77/14/11	80/14/8	78/14/10	79/17/6	-

but not significant on bi-objective UF problems especially on UF4–UF6.

3) Experimental Results on Instances With Complicated PF Shapes (WFG Test Suites): In addition to the above mentioned two suites of test problems, experiments on the WFG test instances are conducted to show the ability of DMOEA-εC on dealing with MOPs with complicated PF shapes. The mathematical descriptions and true PFs of WFG test problems can be found in [61].

The experimental results on WFG test suites in Tables III–V show that DMOEA-εC performs significantly better than competitors on WFG1, WFG2, WFG5, and WFG7 problems in terms of IGD, HV, and $I_{\epsilon+}$ metric values. In spite that DMOEA-εC is not the top among all algorithms on the rest WFG test problems, the rank values of DMOEA-εC on the remaining WFG problems are rightly after the best algorithm. To be specific, SMEA performs better than DMOEA-εC on WFG3, WFG6, and WFG8 problems significantly in terms of three metrics. Besides, MOCell and SMPSO show best performance on WFG4 and WFG9 problems, respectively. The reason for the unsatisfactory performance of DMOEA-εC on WFG4 and WFG9 problems might be that DMOEA-εC is not powerful at tackling MOPs with degenerate or deceptive properties. To sum up, DMOEA-εC has shown competitive performance on solving WFG instances with two objectives.

To summarize, as can be seen in Tables III–V, DMOEA-εC achieves significantly better IGD values than MOEA/D, MOEA/DRA, MOEA/D-AWA, SMEA, MOCell, and SMPSO in 31, 26, 26, 27, 26, and 25 out of the 34 test instances, respectively. For HV, DMOEA-εC outperforms these competitors significantly in 28, 25, 24, 26, 25, and 30 out of the 34 instances, respectively. For $I_{\epsilon+}$ metric values, DMOEA-εC performs significantly better than these competitors in 31, 28, 27, 27, 27, and 24 out of the 34 instances, respectively. Table IV summarizes these statistical results and reveals the overall rank of the seven algorithms, that is DMOEA-εC, MOEA/D-DRA, MOEA/D-AWA, SMEA, SMPSO, MOCell, and MOEA/D according to the mean ranks. It indicates that DMOEA-εC has the best performance on these continuous test problems in terms of the three metrics. The superiority of DMOEA-εC can be attributed to the efficient information sharing among neighboring subproblems under the ε-constraint framework, the main objective alternation strategy and two matching procedures which strike a balance between convergence and diversity. Note that for a few test problems, some paired algorithms obtain different comparison results with respect to IGD and HV, although both indicators measure convergence and diversity. For example, DMOEA-εC has a better HV but worse IGD than SMPSO on ZDT1 and ZDT2. The reason for this occurrence is the different preferences of the two indicators. IGD is based on uniformly distributed points along the PF and prefers the distribution uniformity of the solution set. However, HV is influenced more by boundary solutions and has a bias toward the extensity of the solution set.

For a visual observation, Fig. 3 shows the distribution of the final solutions with the minimum IGD value within 30 runs found by DMOEA-εC. It is visually evident that for each ZDT and DTLZ instance, the final population obtained by DMOEA-εC can cover the whole PFs very well and spread uniformly. DMOEA-εC shows good convergence and obtains solutions with good diversity on UF1–UF3 and UF7. For UF4, UF8–UF10, and LZ problems, final solutions obtained by DMOEA-εC approximate the PFs not very well but spread widely along the PFs. For UF5 and UF6, DMOEA-εC can only find some parts of the PFs. As to the WFG test problems with two objectives, DMOEA-εC achieves good convergence and obtains solutions with good diversity on most of the test instances. In summary, Fig. 3 shows that DMOEA-εC can achieve the approximations with both good convergence and diversity for most of the test instances.

V. COMPARISONS ON MULTIOBJECTIVE 0/1 KNAPSACK PROBLEMS

If Ω in the $P0$ is a finite set, then $P0$ is called a combinatorial MOP. The MOKPs are taken as test instances. Extensive experiments are conducted in this part to study and compare DMOEA-εC with MOEA/D on dealing with combinatorial MOPs.

A. Multiobjective 0/1 Knapsack Problems

This section presents the MOKPs used in the following experiments as benchmarks. Given a set of n items and a set of m knapsacks, the MOKPs can be stated as:

$$\begin{aligned} &\text{maximize} \;\; f_i(\mathbf{x}) = \sum_{j=1}^{n} p_{ij}x_j, \; i = 1, \ldots, m \\ &\text{subject to} \;\; \sum_{j=1}^{n} w_{ij}x_j \le c_i, \; i = 1, \ldots, m \\ &\qquad\qquad\quad \mathbf{x} = (x_1, \ldots, x_n) \in \{0, 1\}^n \end{aligned}$$

where $p_{ij} \ge 0$ is the profit of item j in knapsack i, $w_{ij} \ge 0$ is the weight of item j in knapsack i, and c_i is the capacity of knapsack i. $x_i = 1$ means that item i is selected and put into knapsacks.

The MOKPs are NP-hard and can model a variety of applications in resource allocation. A set of nine instances of the MOKPs proposed in [10] are widely used. MOEA/D outperforms a number of MOEAs without additional local search mechanisms on these test instances. In this paper, these nine instances are also used for comparing the performances of DMOEA-εC and MOEA/D.

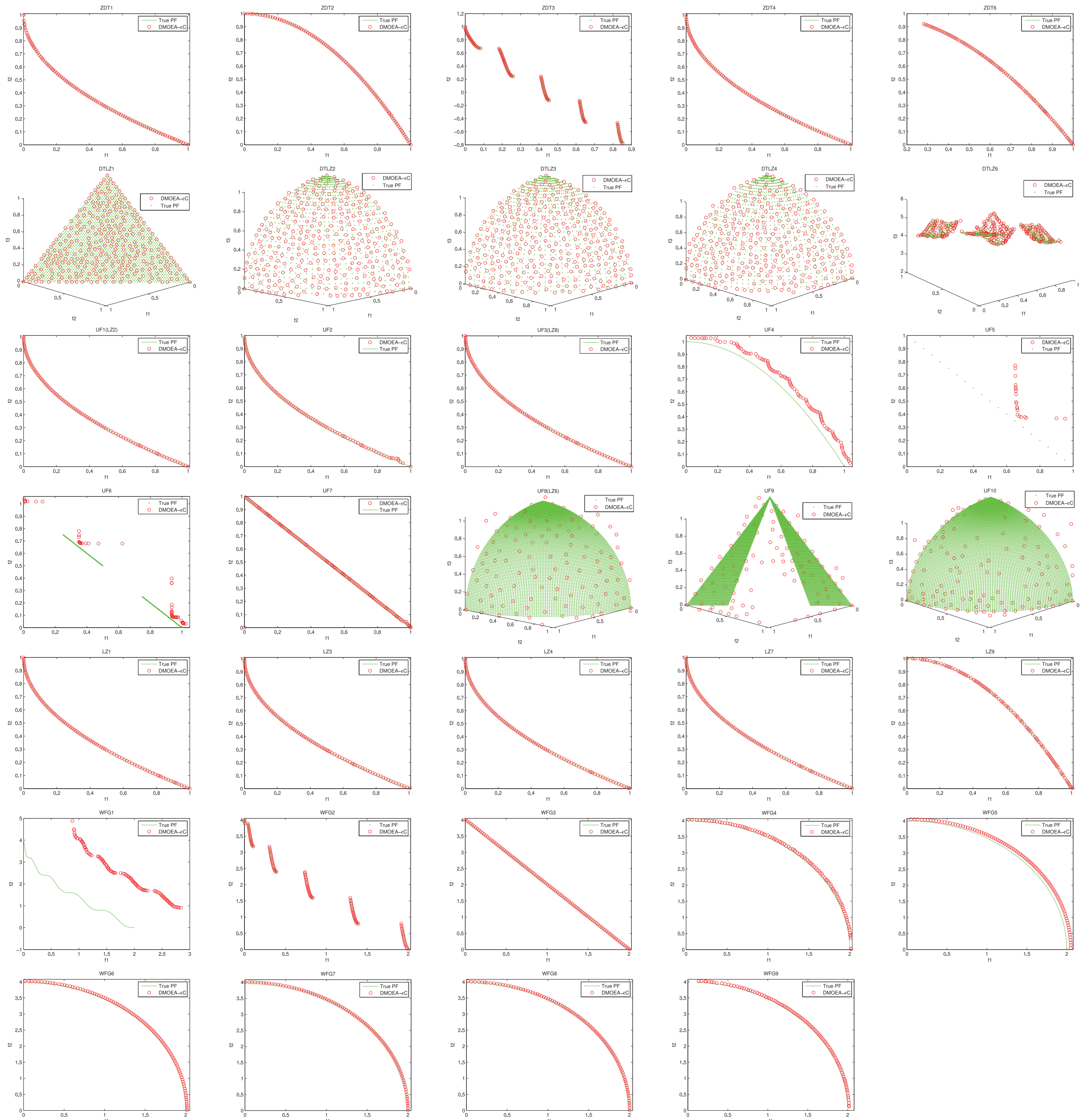

Fig. 3. Final populations in the objective space with the minimum IGD metric value within 30 runs obtained by DMOEA-εC on 34 continuous problems.

The implementation of DMOEA-εC in terms of operators is exactly the same as MOEA/D [15]. Specifically, the one-point crossover and standard mutation operator are used to generate a child solution. The greedy repair method proposed by Jaszkiewicz [63] is adopted after genetic operators. In the greedy repair method, an item with heavy weight in the over-filled knapsacks and little contribution to the single objective function value is more likely to be removed. The initialization of $z_i{}^*$ and $z_i{}^{\mathrm{nad}}$ are realized by taking each f_i and $-f_i$ as the objective function and applying the repair method on a randomly generated point, respectively.

B. Parameter Settings

Due to the differences in algorithmic frameworks, the population size varies in DMOEA-εC and MOEA/D for each MOKP instance, as illustrated in Table VII.[6] The neighborhood size is set as $T = 10$ and the probability of selecting

[6]In order to have a comparable population size, for the tri-objective and four-objective 0/1 knapsack problems, in MOEA/D, we set $H = 25$ and $H = 12$, thus $N = C_{25+3-1}^{3-1} = 351$ and $N = C_{12+4-1}^{4-1} = 455$, respectively. And in DMOEA-εC, we set $q = 19$ and $q = 8$, thus $N = 19^{3-1} = 361$ and $N = 8^{4-1} = 512$, respectively.

TABLE VII
PARAMETER SETTINGS OF DMOEA-εC AND MOEA/D FOR THE TEST INSTANCES OF 0/1 KNAPSACK PROBLEMS

Instance		N in MOEA/D	N in DMOEA-εC	S
m	n			
2	250	150	150	150
2	500	200	200	200
2	750	250	250	200
3	250	351	361	200
3	500	351	361	250
3	750	351	361	300
4	250	455	512	300
4	500	455	512	300
4	750	455	512	300

mate solutions from neighborhood is set as $\delta = 0.9$ for all the instances. And the maximal number of solutions replaced by a new solution is set as $n_r = \lfloor 0.01N \rfloor$. The iteration interval of alternating the main objective index IN_m in DMOEA-εC is set as $\lfloor 10\% \cdot$(number of iterations)$\rfloor$.

Both DMOEA-εC and MOEA/D stop after $500 \times S$ calls of the repair method. Both of them are independently run 30 times for each test instances on an identical computer.

C. Experimental Results

The IGD and HV metrics are used for comparing the performances of different algorithms. In the case where the actual PF is unknown in advance, P^* can be set as an upper approximation of the PF. Jaszkiewicz [63] has produced a good upper approximation to each 0/1 knapsack test instance by solving the linear programming relaxed version of the Tchebycheff formulation of the original MOKP with a number of uniformly distributed weight vectors. The number of the points in the upper approximation is 202 for each of the bi-objective instances, 1326 for the tri-objective instances, and 3276 for the four-objective. In our experiments, P^* is set as such an upper approximation. The reference points for nine MOKP benchmark problems used in calculations of the HV metric values are set to be $\mathbf{r} = \mathbf{0}$.

Table VIII gives the means and standard deviations of the IGD and HV metric values over 30 independent runs of both MOEA/D and DMOEA-εC on the nine MOKP benchmark instances. The Wilcoxon's rank sum test at a 5% significance level is conducted to test the significance of differences between the mean metric values yielded by MOEA/D and the DMOEA-εC. The numbers in parentheses are the standard deviations. †, §, and ≈ mean that the performance of the DMOEA-εC is better than, worse than, and similar to that of MOEA/D according to the Wilcoxon's rank sum test, respectively. The bold data in Table VIII are the best metric values for each instance. Fig. 4 plots the distribution of the final approximation with the minimum IGD metric value among 30 runs of both MOEA/D and DMOEA-εC for each bi-objective test instance. Relaxed PF represents the upper approximation obtained by Jaszkiewicz [63].

With the same number of calls of the repair method, it is clear from Table VIII that DMOEA-εC performs significantly better than MOEA/D in terms of both IGD and HV metric values on all the test instances. For example, the average IGD values obtained by DMOEA-εC are about 46%, 69%, and 48% smaller than those obtained by MOEA/D on instances 250-2, 500-2, and 750-2, respectively. Besides, the larger the number of decision variables and objectives is, the larger differences between DMOEA-εC and MOEA/D are. Fig. 4 demonstrates that the set of final nondominated solutions obtained by MOEA/D is dominated by the set obtained by DMOEA-εC on instances 250-2, 500-2, and 750-2. From these figures, it is also clear that the differences in the final approximations between MOEA/D and DMOEA-εC become greater with the increase of decision variables.

In summary, the statistical results on IGD and HV metric values in Table VIII and the distributions of final approximations on bi-objective test problems in Fig. 4 confirm the superiority of DMOEA-εC over MOEA/D on solving MOKP benchmark problems.

VI. FURTHER DISCUSSION

In this section, the parameter analysis and algorithmic behavior of DMOEA-εC are deeply analyzed. First, the influence of the parameter IN_m on the performance of DMOEA-εC is examined. Then, the algorithmic behavior of DMOEA-εC, including effects of both solution-to-subproblem matching procedure and subproblem-to-solution matching procedure as well as the superiority of the farthest candidate method are further investigated.

A. Parameter Sensitivity Analysis of IN_m

IN_m is a major parameter in DMOEA-εC. It decides how often the algorithm alternates the main objective function. To study how DMOEA-εC is sensitive to this parameter, we take the ZDT1, DTLZ1, DTLZ2, UF1, and WFG2 as examples and test different settings of IN_m in the implementation of DMOEA-εC. Different IN_m values are set as $\lfloor$1%,2%,5%,10%,20%,40%,60%,80%,90%,100%$\cdot$(number of iterations)$\rfloor$. That is to say, the frequency of switching the main objective function becomes smaller and smaller. All the other parameters are kept the same as Section IV. Similarly, 30 independent runs have been conducted for each configuration on these test instances. Fig. 5 shows the variation of means and standard deviations of IGD and HV metrics across all IN_m values on the selected test problems. As shown in Fig. 5, DMOEA-εC performs well with a wide range of IN_m value on UF1 and WFG2. For ZDT1, a large IN_m may be better, while for DTLZ1 and DTLZ2, a small IN_m may be better. Thus it can be claimed that a good setting of IN_m varies with different test instances, such as the settings of IN_m adopted in Section IV. Generally, a larger value of IN_m is good for convergence, while a smaller value of IN_m benefits diversity.

Fig. 5 also reveals that DMOEA-εC with a large IN_m value does not work stably on DTLZ1. A large IN_m value will result in the slow convergence rate and the bad ideal/nadir

TABLE VIII
STATISTICAL RESULTS [MEAN (STD. DEV.)] OF TWO ALGORITHMS OVER 30 INDEPENDENT RUNS ON THE NINE INSTANCES IN TERMS OF IGD AND HV METRICS

Instance	250-2	500-2	750-2	250-3	500-3	750-3	250-4	500-4	750-4
					IGD				
MOEA/D	5.529E+01†	1.773E+02†	4.201E+02†	2.502E+02†	3.792E+02†	7.526E+02†	2.212E+02†	6.151E+02†	1.055E+03†
	(4.948E-00)	(8.968E-00)	(2.109E+01)	(1.014E+01)	(2.792E+01)	(3.474E+01)	(1.665E+01)	(2.847E+01)	(2.823E+01)
DMOEA-εC	**2.970E+01**	**5.456E+01**	**2.167E+02**	**1.899E+02**	**2.007E+02**	**3.005E+02**	**1.485E+02**	**2.070E+02**	**5.671E+02**
	(1.397E-00)	(3.691E-00)	(1.066E+01)	(9.558E-00)	(9.558E-00)	(9.292E-00)	(1.762E+01)	(1.068E+01)	(1.000E+01)
					HV				
MOEA/D	5.239E+07†	6.721E+07†	9.082E+07†	4.837E+11†	1.031E+12†	1.715E+13†	4.833E+15†	3.901E+16†	1.481E+17†
	(3.009E+06)	(5.815E+06)	(1.033E+06)	(3.136E+10)	(1.056E+11)	(2.868E+12)	(1.083E+14)	(1.038E+15)	(3.005E+16)
DMOEA-εC	**4.869E+07**	**6.338E+07**	**1.082E+08**	**8.089E+11**	**4.075E+12**	**3.196E+13**	**6.284E+15**	**8.577E+16**	**4.793E+17**
	(2.967E+06)	(6.235E+06)	(1.140E+07)	(4.204E+11)	(0.747E+11)	(1.962E+12)	(6.186E+13)	(2.350E+15)	(2.648E+16)

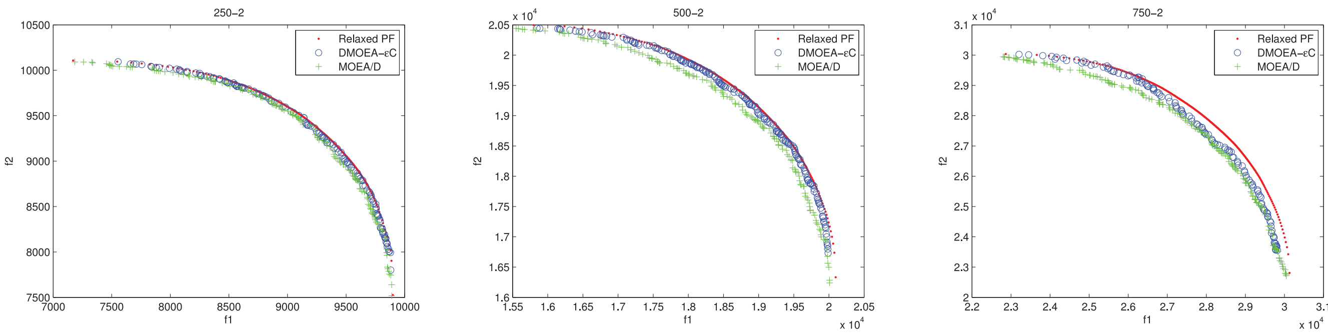

Fig. 4. Final populations in the objective space with the minimum IGD metric value within 30 runs obtained by DMOEA-εC on two-objective MOKP test problems.

approximation of the main objective function, which will affect the diversity of the whole population. However, a small value of IN_m means more times of performing the solution-to-subproblem matching procedure. This will result in that DMOEA-εC consumes more computational resources. Thus, a proper IN_m value strikes a good balance between the performance of DMOEA-εC and the computational cost.

B. Detailed Analysis on Behavior of DMOEA-εC

This section designs experiments to study effects of the solution-to-subproblem matching procedure and the subproblem-to-solution matching procedure, and the superiority of the farthest candidate approach.

1) Effects of the Solution-to-Subproblem Matching Procedure and the Subproblem-to-Solution Matching Procedure: As mentioned above, the solution-to-subproblem matching procedure and the subproblem-to-solution matching procedure strike a balance between convergence and diversity. Then do the above two matching mechanisms indeed play an important role in DMOEA-εC? In order to answer this question, two DMOEA-εC variants, denoted as DMOEA-εC-No_CV and DMOEA-εC-D_No are developed for comparing with the original DMOEA-εC. Detailed descriptions of the two variants will be given in the following.

Furthermore, why is the distance value from a subproblem to a solution selected as the matching criterion for the solution-to-subproblem matching procedure? Why is the constraint violation value of a solution regarding a subproblem adopted as the matching criterion for the subproblem-to-solution matching procedure? To illustrate the effectiveness of the two criteria, another three variants of DMOEA-εC, including DMOEA-εC-D_D, DMOEA-εC-CV_D, and DMOEA-εC-CV_CV, are designed.

- *DMOEA-εC-No_CV:* Different from DMOEA-εC, the solution-to-subproblem matching procedure is removed. And the constraint violation value is still adopted as the matching criterion for the subproblem-to-solution matching procedure.
- *DMOEA-εC-D_No:* In this variant, the subproblem-to-solution matching procedure is removed. And the distance value is still regarded as the matching criterion for the solution-to-subproblem matching procedure.
- *DMOEA-εC-D_D:* In this variant, the distance value is adopted as the matching criterion for the subproblem-to-solution matching procedure.
- *DMOEA-εC-CV_D:* In this variant, the constraint violation value and the distance value are adopted as the matching criterion for the solution-to-subproblem matching procedure and the subproblem-to-solution matching procedure, respectively.
- *DMOEA-εC-CV_CV:* In this variant, the constraint violation value is adopted as the matching criterion for the solution-to-subproblem matching procedure.

All variants are the same as DMOEA-εC except for differences on the two matching procedures. Here we still consider ZDT1, DTLZ1, DTLZ2, UF1, and WFG2 test problems. With the same parameter settings as Section IV, the above

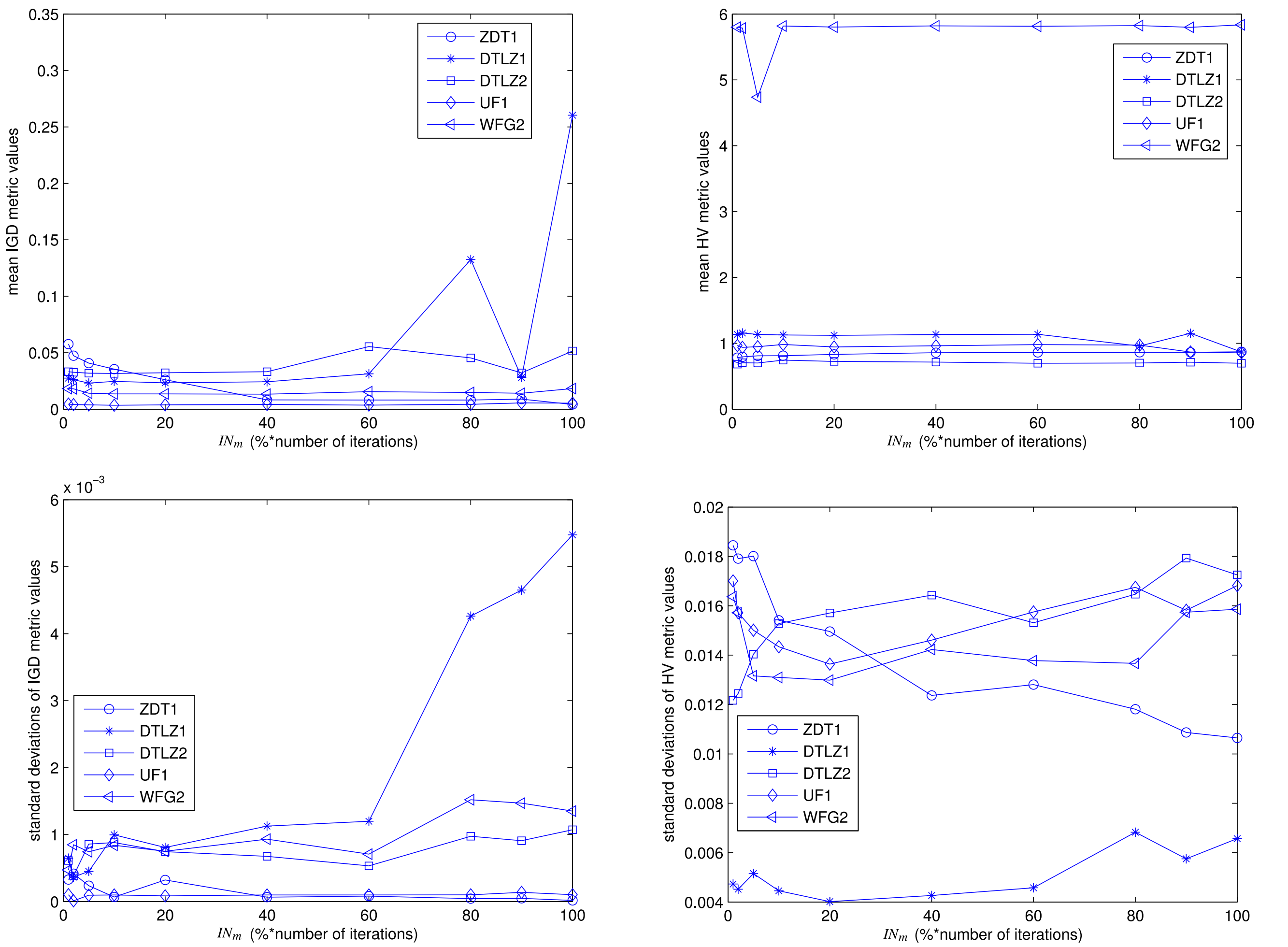

Fig. 5. Means and standard deviations of IGD and HV metric values within 30 runs versus the value of IN_m in DMOEA-εC for ZDT1, DTLZ1, DTLZ2, UF1, and WFG2 test instances.

TABLE IX

STATISTICAL RESULTS [MEAN (STD. DEV.)] OF DMOEA-εC AND ITS FIVE VARIANTS OVER 30 INDEPENDENT RUNS ON THE FIVE SELECTED INSTANCES IN TERMS OF IGD AND HV METRICS

Instance	DMOEA-εC-No_CV	DMOEA-εC-D_No	DMOEA-εC-D_D	DMOEA-εC	DMOEA-εC-CV_D	DMOEA-εC-CV_CV
			IGD			
ZDT1	7.400E-03†(6.050E-05)	5.352E-03†(8.450E-05)	1.961E-01†(2.110E-03)	**3.763E-03(6.690E-05)**	2.075E-01†(1.100E-03)	2.115E-01†(4.743E-03)
DTLZ1	4.385E-02†(7.612E-03)	3.530E-02†(2.724E-03)	5.877E-02†(2.702E-01)	**2.476E-02(9.966E-02)**	3.051E-02†(1.972E-03)	6.213E-02†(2.945E-01)
DTLZ2	5.365E-02†(8.610E-03)	4.030E-02†(4.421E-03)	4.755E-02†(6.055E-03)	**3.160E-02(8.819E-04)**	4.395E-02†(2.765E-03)	4.795E-02†(2.205E-01)
UF1	6.500E-03†(7.263E-04)	5.323E-03†(2.810E-04)	5.500E-03†(1.805E-04)	**4.407E-03(9.870E-05)**	5.752E-03†(6.051E-05)	4.845E-03†(4.465E-4)
WFG2	1.655E-02†(2.451E-04)	1.700E-02†(1.300E-04)	1.545E-02†(4.050E-04)	**1.505E-02(8.403E-04)**	1.58E-02†(3.200E-04)	1.610E-02†(1.825E-04)
			HV			
ZDT1	7.476E-01†(4.239E-02)	7.924E-01†(2.480E-02)	6.304E-01†(1.451E-02)	**8.272E-01(1.541E-02)**	6.038E-01†(1.832E-02)	5.965E-01†(2.930E-02)
DTLZ1	1.107E-00†(1.462E-03)	1.118E-00†(1.558E-03)	1.057E-00†(1.517E-03)	**1.307E-00 (4.458E-03)**	1.117E-00†(3.244E-03)	1.047E-00†(1.663E-03)
DTLZ2	6.588E-01†(1.963E-02)	7.294E-01†(1.758E-02)	6.948E-01†(1.537E-02)	**7.435E-01 (1.528E-02)**	7.094E-01†(1.704E-02)	7.267E-01†(1.694E-02)
UF1	8.637E-01†(1.436E-02)	8.700E-01†(3.891E-02)	8.511E-01†(6.481E-02)	**8.741E-01(1.434E-02)**	8.471E-01†(8.457E-02)	8.628E-01†(6.590E-02)
WFG2	5.926E-00†(1.199E-02)	5.981E-00†(1.693E-02)	5.862E-00†(1.054E-02)	**6.076E-00(1.310E-02)**	5.95E-00†(2.479E-02)	5.905E-00†(5.793E-02)

mentioned five variants are experimentally compared with DMOEA-εC. The experimental results, in terms of the means and standard deviations of the IGD and HV metric values within 30 independent runs obtained by each algorithm for the selected test instances are all shown in Table IX. Similarly, the Wilcoxon's rank sum test at a 5% significance level is conducted to test the significance of differences between the mean metric values yielded by DMOEA-εC and its variants.

Table IX shows that in terms of IGD and HV metrics, the proposed DMOEA-εC is significantly better than its variants on all selected instances. The effectiveness of the solution-to-subproblem matching procedure using the distance value as the matching criterion and the subproblem-to-solution matching procedure adopting the constraint violation value as the matching criterion is confirmed experimentally.

TABLE X
STATISTICAL RESULTS [MEAN (STD. DEV.)] OF DMOEA-εC AND ITS VARIANT DMOEA-εC-CD OVER 30 INDEPENDENT RUNS ON FIVE SELECTED TEST INSTANCES IN TERMS OF IGD AND HV METRICS

	IGD		*HV*	
	DMOEA-εC	DMOEA-εC-CD	DMOEA-εC	DMOEA-εC-CD
ZDT1	**3.763E-03**	8.250E-03†	**8.272E-01**	8.072E-01†
	(6.690E-05)	(2.881E-05)	(1.541E-02)	(8.840E-02)
DTLZ1	**2.476E-02**	3.385E-02†	**1.307E-00**	1.126E-00†
	(9.966E-04)	(1.821E-03)	(4.458E-03)	(3.549E-03)
DTLZ2	**3.160E-02**	3.595E-02†	**7.435E-01**	7.067E-01†
	(8.819E-04)	(6.841E-04)	(1.528E-02)	(1.280E-02)
UF1	**4.407E-03**	6.500E-03†	**8.741E-01**	8.554E-01†
	(9.870E-05)	(2.451e-04)	(1.434E-02)	(1.053E-02)
WFG2	**1.505E-02**	1.62E-02†	**6.076E-00**	5.992E-00†
	(8.403E-04)	(3.216E-04)	(1.310E-02)	(3.793E-02)

2) Superiority of the Farthest Candidate Method: In order to further investigate the superiority of the farthest candidate method when pruning EP, we compare it with the crowding distance-based mechanism used in NSGA-II [9]. Thus, we develop a DMOEA-εC variant, denoted as DMOEA-εC-CD, by replacing the farthest candidate method in DMOEA-εC with the crowding distance-based mechanism.

Take ZDT1, DTLZ1, DTLZ2, UF1, and WFG2 test instances as examples. DMOEA-εC-CD has been experimentally compared with DMOEA-εC with the same parameter settings as Section IV. The experimental results, in terms of the means and standard deviations of the IGD and HV metric values of the final solutions within 30 independent runs obtained by each algorithm for the selected test instances are all shown in Table X. Similarly, the Wilcoxon's rank sum test at a 5% significance level is conducted to test the significance of differences between the mean metric values yielded by DMOEA-εC and its variant DMOEA-εC-CD.

Table X demonstrates the superiority of DMOEA-εC over DMOEA-εC-CD in terms of both IGD and HV metric values on selected five test problems. This also shows the rationality and superiority of the farthest candidate method when pruning the EP.

VII. CONCLUSION

Decomposition and the ε-constraint method are two important strategies in the field of multiobjective optimization. This paper has reformulated MOPs by incorporating the ε-constraint method into the decomposition strategy and proposed a DMOEA-εC framework to deal with MOPs.

DMOEA-εC explicitly decomposes an MOP into a series of scalar constrained optimization subproblems by selecting one of the objectives as the main objective function and assigning each subproblem with an upper bound vector ε. Then these subproblems are optimized simultaneously by evolving a population of solutions. At each generation, each individual solution in the population is associated with a subproblem. The neighborhood relations among these subproblems are defined based on the Euclidean distance between their upper bound vectors. And the assumption that optimal solutions of two neighboring subproblems should be very similar is still valid. Besides, a main objective alternation strategy, a solution-to-subproblem matching procedure and a subproblem-to-solution matching procedure are proposed to strike a balance between convergence and diversity.

DMOEA-εC has been compared with six state-of-the-art MOEAs, i.e., MOEA/D [15], MOEA/D-DRA [17], MOEA/D-AWA [20], SMEA [53], MOCell [54], and SMPOS [55] on 34 continuous test instances and nine MOKPs test problems. A systematical experimental study has demonstrated that DMOEA-εC outperforms or performs competitively against other algorithms on the majority of the test instances.

The sensitivity of the parameter IN_m in DMOEA-εC has been experimentally investigated. Moreover, the algorithmic behavior of DMOEA-εC including the effects of both solution-to-subproblem matching procedure and subproblem-to-solution matching procedure as well as the superiority of the farthest candidate method have been further analyzed. All these experimental results confirm that DMOEA-εC can deal with majority of the continuous benchmark problems and the MOKP test problems successfully.

Future research work includes investigations of adopting alternative methods to solve each constrained subproblem, employing more effective methods for estimating the nadir point and proposing an adjustment strategy for upper bound vectors to further improve the uniformity of the final population. Besides, based on our previous research works [64], [65], the hybridization of different search operators in DMOEA-εC is also worthwhile to be studied.

ACKNOWLEDGMENT

The authors would like to thank Prof. Q. Zhang for his inspiring discussions on MOEA/D, Prof. Y. Qi's constructive discussions on MOEA/D-AWA, and the support of source code of MOEA/D-AWA. They would also like to thank Dr. H. Zhang for providing the source code of SMEA and Dr. L. Chen for checking the source code of DMOEA-εC.

REFERENCES

[1] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston, MA, USA: Kluwer, 1999.
[2] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001.
[3] C. A. C. Coello, D. A. V. Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Norwell, MA, USA: Kluwer, 2002.
[4] A. Zhou *et al.*, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 32–49, 2011.
[5] C. A. C. Coello, "Evolutionary multi-objective optimization: A historical view of the field," *IEEE Comput. Intell. Mag.*, vol. 1, no. 1, pp. 28–36, Feb. 2006.
[6] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization," in *Proc. Int. Conf. Genet. Algorithms*, Champaign, IL, USA, 1993, pp. 416–423.
[7] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, 1994.
[8] J. Horn, N. Nafpliotis, and D. E. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization," in *Proc. IEEE Congr. Evol. Comput.*, Orlando, FL, USA, 1994, pp. 82–87.
[9] K. Deb, A. Pratap, S. Agrawal, 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.

[10] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.

[11] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," Dept. Elect. Eng., Swiss Federal Inst. Technol., Zürich, Switzerland, Tech. Rep. 103, 2001.

[12] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving From Nature—PPSN VIII* (LNCS 3242). Heidelberg, Germany: Springer, 2004, pp. 832–842.

[13] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, 2011.

[14] E. J. Hughes, "Multiple single objective Pareto sampling," in *Proc. IEEE Congr. Evol. Comput.*, vol. 4. Canberra, ACT, Australia, 2003, pp. 2678–2684.

[15] 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.

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

[17] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," School Comput. Sci. Electron. Eng., Univ. Essex, Colchester, U.K., Tech. Rep. CES-491, 2009.

[18] L. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, "Constrained subproblems in a decomposition-based multiobjective evolutionary algorithm," *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 475–480, Jun. 2016.

[19] A. Zhou and Q. Zhang, "Are all the subproblems equally important? Resource allocation in decomposition-based multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 52–64, Feb. 2016.

[20] Y. Qi *et al.*, "MOEA/D with adaptive weight adjustment," *Evol. Comput.*, vol. 22, no. 2, pp. 231–264, 2014.

[21] 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.

[22] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, Aug. 2014.

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

[24] L. Ke, Q. Zhang, and R. Battiti, "MOEA/D-ACO: Multiobjective combinatorial optimization by using decomposition and ant colony," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1845–1859, Jan. 2013.

[25] N. A. Moubayed, A. Petrovski, and J. McCall, "A novel smart multi-objective particle swarm optimisation using decomposition," in *Proc. Int. Conf. Parallel Problem Solving Nature*, Kraków, Poland, 2010, pp. 1–10.

[26] V. A. Shim, K. C. Tan, and K. K. Tan, "A hybrid estimation of distribution algorithm for solving the multi-objective multiple traveling salesman problem," in *Proc. IEEE Congr. Evol. Comput.*, Brisbane, QLD, Australia, 2012, pp. 682–691.

[27] C.-M. Chen, Y.-P. Chen, and Q. Zhang, "Enhancing MOEA/D with guided mutation and priority update for multi-objective optimization," in *Proc. IEEE Congr. Evol. Comput.*, Trondheim, Norway, 2009, pp. 209–216.

[28] K. Sindhya, S. Ruuska, T. Haanpää, and K. Miettinen, "A new hybrid mutation operator for multiobjective optimization with differential evolution," *Soft Comput.*, vol. 15, no. 10, pp. 2041–2055, 2011.

[29] W. Huang and H. Li, "On the differential evolution schemes in MOEA/D," in *Proc. Int. Conf. Nat. Comput.*, Yantai, China, 2010, pp. 2788–2792.

[30] Y. Lai, "Multiobjective optimization using MOEA/D with a new mating selection mechanism," M.S. thesis, Dept. Comput. Sci. Inf., Taiwan Normal Univ., Taipei, Taiwan, 2009.

[31] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Adaptation of scalarizing functions in MOEA/D: An adaptive scalarizing function-based multiobjective evolutionary algorithm," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.*, Nantes, France, 2009, pp. 438–452.

[32] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Simultaneous use of different scalarizing functions in MOEA/D," in *Proc. Genet. Evol. Comput. Conf.*, Portland, OR, USA, 2010, pp. 519–526.

[33] F.-Q. Gu and H.-L. Liu, "A novel weight design in multi-objective evolutionary algorithm," in *Proc. Int. Conf. Comput. Intell. Security*, Wuhan, China, 2010, pp. 137–141.

[34] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive multiobjective optimization by MOEA/D with Gaussian process model," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 456–474, Jun. 2010.

[35] R. Lü, X. Guan, X. Li, and I. Hwang, "A large-scale flight multi-objective assignment approach based on multi-island parallel evolution algorithm with cooperative coevolutionary," *Sci. China Inf. Sci.*, vol. 59, no. 7, pp. 1–17, 2016.

[36] Y. Mei, K. Tang, and X. Yao, "Decomposition-based memetic algorithm for multiobjective capacitated arc routing problem," *IEEE Trans. Evol. Comput.*, vol. 15, no. 2, pp. 151–165, Apr. 2011.

[37] J. Li, J. Chen, B. Xin, and L. H. Dou, "Solving multi-objective multi-stage weapon target assignment problem via adaptive NSGA-II and adaptive MOEA/D: A comparison study," in *Proc. IEEE Congr. Evol. Comput.*, Sendai, Japan, 2015, pp. 3132–3139.

[38] G. H. Wang, J. Chen, T. Cai, and B. Xin, "Decomposition-based multi-objective differential evolution particle swarm optimization for the design of a tubular permanent magnet linear synchronous motor," *Eng. Optim.*, vol. 45, no. 9, pp. 1107–1127, 2013.

[39] Y. Laili, L. Zhang, F. Tao, and P. Ma, "Rotated neighbor learning-based auto-configured evolutionary algorithm," *Sci. China Inf. Sci.*, vol. 53, no. 5, pp. 1–13, 2016.

[40] G. Mavrotas, "Effective implementation of the ε-constraint method in multi-objective mathematical programming problems," *Appl. Math. Comput.*, vol. 213, no. 2, pp. 455–465, 2009.

[41] G. Mavrotas and K. Florios, "An improved version of the augmented ε-constraint method (AUGMECON2) for finding the exact Pareto set in multi-objective integer programming problems," *Appl. Math. Comput.*, vol. 219, no. 18, pp. 9652–9669, 2013.

[42] L. Grandinetti, O. Pisacane, and M. Sheikhalishahi, "An approximate ε-constraint method for a multi-objective job scheduling in the cloud," *Future Gener. Comput. Syst.*, vol. 29, no. 8, pp. 1901–1908, 2013.

[43] W. Zhang and M. Reimann, "A simple augmented ε-constraint method for multi-objective mathematical integer programming problems," *Eur. J. Oper. Res.*, vol. 234, no. 1, pp. 15–24, 2014.

[44] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Methods Appl. Mech. Eng.*, vol. 186, nos. 2–4, pp. 311–338, 2000.

[45] Y. Y. Haimes, L. S. Lasdon, and D. A. Wismer, "On a bicriterion formulation of the problems of integrated system identification and system optimization," *IEEE Trans. Syst., Man, Cybern.*, vol. 1, no. 3, pp. 296–297, Jul. 1971.

[46] H. Isermann and R. E. Steuer, "Computational experience concerning payoff tables and minimum criterion values over the efficient set," *Eur. J. Oper. Res.*, vol. 33, no. 1, pp. 91–97, 1988.

[47] M. J. Alves and J. P. Costa, "An exact method for computing the nadir values in multiple objective linear programming," *Eur. J. Oper. Res.*, vol. 198, no. 2, pp. 637–646, 2009.

[48] B. Metev and V. Vassilev, "A method for nadir point estimation in MOLP problems," *Cybern. Inf. Technol.*, vol. 3, no. 2, pp. 15–24, 2003.

[49] B. Chen, W. Zeng, Y. Lin, and D. Zhang, "A new local search-based multiobjective optimization algorithm," *IEEE Trans. Evol. Comput.*, vol. 19, no. 1, pp. 50–73, Feb. 2015.

[50] S. Kukkonen and K. Deb, "A fast and effective method for pruning of non-dominated solutions in many-objective problems," in *Proc. Int. Conf. Parallel Problem Solving Nat.*, Reykjavík, Iceland, 2006, pp. 553–562.

[51] Y. Wang, Z. Cai, G. Guo, and Y. Zhou, "Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 3, pp. 560–575, Jun. 2007.

[52] D. P. Mitchell, "Spectrally optimal sampling for distribution ray tracing," *ACM SIGGRAPH Comput. Graph.*, vol. 25, no. 4, pp. 157–164, 1991.

[53] H. Zhang *et al.*, "A self-organizing multiobjective evolutionary algorithm," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 792–806, Oct. 2016.

[54] A. J. Nebro, J. J. Durillo, F. Luna, B. Dorronsoro, and E. Alba, "MOCell: A cellular genetic algorithm for multiobjective optimization," *Int. J. Intell. Syst.*, vol. 24, no. 7, pp. 726–746, 2009.

[55] A. J. Nebro *et al.*, "SMPSO: A new PSO-based metaheuristic for multi-objective optimization," in *Proc. IEEE Symp. Comput. Intell. Multi-Criteria Decis. Making*, Nashville, TN, USA, 2009, pp. 66–73.

[56] J. J. Durillo and A. J. Nebro, "jMetal: A JAVA framework for multi-objective optimization," *Adv. Eng. Softw.*, vol. 42, no. 10, pp. 760–771, 2011.

[57] A. Zhou, Q. Zhang, Y. Jin, E. Tsang, and T. Okabe, "A model-based evolutionary algorithm for bi-objective optimization," in *Proc. IEEE Congr. Evol. Comput.*, Edinburgh, U.K., 2005, pp. 2568–2575.

[58] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.

[59] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proc. IEEE Congr. Evol. Comput.*, Honolulu, HI, USA, 2002, pp. 825–830.

[60] Q. Zhang *et al.*, "Multiobjective optimization test instances for the CEC 2009 special session and competition," School Comput. Sci. Electron. Eng., Univ. Essex, Colchester, U.K., Tech. Rep. CES-487, 2009.

[61] S. Huband, L. Barone, L. While, and P. Hingston, "A scalable multi-objective test problem toolkit," in *Proc. 3rd Int. Conf. Evol. Multi-Criterion Optim.*, Guanajuato, Mexico, 2005, pp. 280–295.

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

[63] A. Jaszkiewicz, "On the performance of multiple-objective genetic local search on the 0/1 knapsack problem—A comparative experiment," *IEEE Trans. Evol. Comput.*, vol. 6, no. 4, pp. 402–412, Aug. 2002.

[64] J. Chen, B. Xin, Z. H. Peng, L. H. Dou, and J. Zhang, "Optimal contraction theorem for exploration–exploitation tradeoff in search and optimization," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 39, no. 3, pp. 680–691, May 2009.

[65] B. Xin, J. Chen, J. Zhang, H. Fang, and Z.-H. Peng, "Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: A review and taxonomy," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 5, pp. 744–767, Sep. 2012.

Jie Chen (M'09–SM'12) received the B.S., M.S., and Ph.D. degrees in control theory and control engineering from the Beijing Institute of Technology, Beijing, China, in 1986, 1996, and 2001, respectively.

He was a Visiting Scholar with California State University, Long Beach, CA, USA, from 1989 to 1990. From 1996 to 1997, he was a Research Fellow with the School of Electronic, Electrical and Systems Engineering, University of Birmingham, Birmingham, U.K. He is currently a Professor of Control Science and Engineering with the Beijing Institute of Technology, Beijing, China, where he is also the Head of the State Key Laboratory of Intelligent Control and Decision of Complex Systems. He has co-authored three books and over 200 research papers. His current research interests include multiobjective optimization and decision in complex systems, intelligent control, nonlinear control, and optimization methods.

Dr. Chen serves as a Managing Editor of the *Journal of Systems Science and Complexity* (2014–2017), and an Associate Editor for the IEEE TRANSACTIONS ON CYBERNETICS (2016–2018), as well as several other international journals.

Juan Li received the B.S. degree from the Beijing Institute of Technology, Beijing, China, in 2013, where she is currently pursuing the Ph.D. degree with the School of Automation.

Her research interests include multiobjective evolutionary optimization, combinatorial optimization, and uncertain optimization.

Bin Xin (S'09–M'10) received the B.S. degree in information engineering and the Ph.D. degree in control science and engineering from the Beijing Institute of Technology, Beijing, China, in 2004 and 2012, respectively.

He is currently an Associate Professor with the School of Automation, Beijing Institute of Technology. His current research interests include search and optimization, evolutionary computation, combinatorial optimization, and multiagent systems.