A Weighted Biobjective Transformation Technique for Locating Multiple Optimal Solutions of Nonlinear Equation Systems

Wenyin Gong, Yong Wang, *Member, IEEE*, Zhihua Cai, and Shengxiang Yang, *Senior Member, IEEE*

***Abstract*—Due to the fact that a nonlinear equation system (NES) may contain multiple optimal solutions, solving NESs is one of the most important challenges in numerical computation. When applying evolutionary algorithms to solve NESs, two issues should be considered: 1) how to transform an NES into a kind of optimization problem and 2) how to develop an optimization algorithm to solve the transformed optimization problem. In this paper, we tackle the first issue by transforming an NES into a weighted biobjective optimization problem. By the above transformation, not only do all the optimal solutions of an original NES become the Pareto optimal solutions of the transformed biobjective optimization problem, but also their images are different points on a linear Pareto front in the objective space. In addition, we suggest an adaptive multiobjective differential evolution, the goal of which is to effectively locate the Pareto optimal solutions of the transformed biobjective optimization problem. Once these solutions are found, the optimal solutions of the original NES can also be obtained correspondingly. By combining the weighted biobjective transformation technique with the adaptive multiobjective differential evolution, we propose a generic framework for the simultaneous locating of multiple optimal solutions of NESs. Comprehensive experiments on 38 NESs with various features have demonstrated that our framework provides very competitive overall performance compared with several state-of-the-art methods.**

***Index Terms*—Differential evolution (DE), evolutionary multiobjective optimization, nonlinear equation systems (NESs), transformation technique.**

Manuscript received September 28, 2016; revised December 9, 2016 and February 14, 2017; accepted February 15, 2017. Date of publication March 15, 2017; date of current version September 28, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61673397, Grant 61573324, and Grant 61375066, in part by the Innovation-driven Plan in Central South University under Grant 2015CXS012 and Grant 2015CX007, in part by the EU Horizon 2020 Marie Sklodowska-Curie Individual Fellowships under Project 661327, in part by the Engineering and Physical Sciences Research Council of U.K. under Grant EP/K001310/1, and in part by the Hunan Provincial Natural Science Fund for Distinguished Young Scholars under Grant 2016JJ1018. *(Corresponding author: Yong Wang.)*

W. Gong and Z. Cai are with the School of Computer Science, China University of Geosciences, Wuhan 430074, China (e-mail: wygong@cug.edu.cn; zhcai@cug.edu.cn).

Y. Wang is with the School of Information Science and Engineering, Central South University, Changsha 410083, China, and also with the Center for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, U.K. (e-mail: ywang@csu.edu.cn).

S. Yang is with the Center for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, U.K. (e-mail: syang@dmu.ac.uk).

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.2017.2670779

I. Introduction

A NONLINEAR equation system (NES) can be formulated as follows:

$$\begin{cases} e_1(\mathbf{x}) &= 0 \\ &\vdots \\ e_m(\mathbf{x}) &= 0 \end{cases} \tag{1}$$

where $e_i(\mathbf{x}) = 0$ $(i \in \{1, \dots, m\})$ is the ith equation, m is the number of equations, $\mathbf{x} = (x_1, \dots, x_n)^T \in \mathcal{S}$ is a decision vector containing n decision variables, $\mathcal{S} = \prod_{j=1}^{n} [\underline{x}_j, \overline{x}_j]$ is the decision space, and $\underline{x}_j$ and $\overline{x}_j$ are the lower and upper bounds of x_j, respectively. In general, an NES contains at least one nonlinear equation. $\mathbf{x}^*$ is called an optimal solution of an NES if for every $i \in \{1, \dots, m\}$, $e_i(\mathbf{x}^*) = 0$.

It is common to face a considerable number of NESs in the fields of mathematics, science, and engineering [1]. An NES may frequently contain more than one optimal solution, especially when the number of decision variables is equal to or larger than that of the equations. In order to allow the decision maker to select the most preferred solution, the purpose of solving NESs is to simultaneously locate multiple optimal solutions in a single run, which is perhaps one of the most challenging tasks in numerical computation [2].

During the past two decades, solving NESs by evolutionary algorithms (EAs) has been gaining increasing attention from the evolutionary computation research community, mainly because of the fact that EAs work with a population of candidate solutions and have good potential to find multiple optimal solutions of an NES in a single run. In principle, solving NESs by EAs can be considered as a two-step procedure [3]. In the first step, it is necessary to design a transformation technique, with the aim of recasting an NES as a kind of optimization problem. In the next step, an optimization algorithm should be developed to solve the transformed optimization problem.

The current popular transformation techniques can be classified into three categories: 1) single-objective optimization-based transformation techniques [2], [4]–[11]; 2) constrained optimization-based transformation techniques [12], [13]; and 3) multiobjective optimization-based transformation techniques [3], [14], [15]. The first kind of transformation technique usually converts an NES into the following single-objective optimization problem:

$$\min \quad \sum\nolimits_{i=1}^{m} |e_i(\mathbf{x})| \tag{2}$$

or

$$\min \quad \sum\nolimits_{i=1}^{m} {e_i}^2(\mathbf{x}). \tag{3}$$

In addition, a constrained optimization problem is always constructed in the second kind of transformation technique

$$\begin{cases} \min & \sum_{i=1}^{m} |e_i(\mathbf{x})| \\ \text{subject to} & e_i(\mathbf{x}) \geq 0, \ \ i = 1, \ldots, m. \end{cases} \tag{4}$$

These two kinds of transformation techniques have a common feature: only one objective function is involved. Due to this feature, EAs may merely focus on one of the optimal solutions in one run when solving (2)–(4). In order to locate multiple optimal solutions simultaneously, some extra diversity preservation strategies should be incorporated into EAs, such as the repulsion strategy [16]–[18]. However, such extra diversity preservation strategies would inevitably introduce some complicated operators and/or problem-dependent parameters, which have a side effect on the generalization. More importantly, if an NES contains infinitely many optimal solutions, EAs are not capable of maintaining a set of representative optimal solutions under the single-objective framework.

In contrast, multiobjective optimization-based transformation techniques redefine an NES as a multiobjective optimization problem. Note that multiobjective optimization problems also include a set of optimal solutions known as the Pareto optimal solutions. Moreover, the task of solving multiobjective optimization problems is to find multiple Pareto optimal solutions in a single run. Obviously, the above similarities imply that multiobjectivization is quite promising for NESs. Along this line, several methods have been proposed. In [14], each equation represents an objective function, and thus, an NES is converted into an m-objective optimization problem. In [15], there are $(n + 1)$ objective functions in the transformed optimization problem. It is evident that with regard to these two methods, the number of objective functions is relevant to the number of equations or decision variables, respectively. As a consequence, they may suffer from the curse of dimensionality (i.e., many-objective) with the drastic increase of m and n. Unlike [14] and [15], Song *et al.* [3] presented a method named MONES, which transforms an NES into a biobjective optimization problem. In MONES, the first decision variable (i.e., x_1) is exploited to guarantee the conflict between the two objective functions. However, if several optimal solutions of an NES have the same value in the first decision variable, it is very likely that some of them will be lost during the evolution.

In this paper, we propose a weighted biobjective transformation technique (called WeB) for NESs based on our previous work (i.e., MONES) in [3]. WeB shares the same biobjective structure with MONES. Different from MONES, in WeB all the decision variables, rather than just the first decision variable, are linearly weighted to construct the two objective functions. By doing this, WeB is able to remedy the loss of some optimal solutions with the same value in the first decision variable. Indeed, WeB is a generalization of MONES. Systematic experiments on 38 NESs with a wide range of features have demonstrated that this simple modification to MONES can produce significantly better results. WeB has the following advantages over other transformation techniques.

1) Compared with other multiobjective optimization-based transformation techniques, WeB has the potential to map the optimal solutions of an NES into different points on the linear Pareto front in the objective space under the biobjective structure because of the random weights.
2) Compared with both single-objective optimization-based and constrained optimization-based transformation techniques, WeB has the capability to provide a set of representative optimal solutions for the situation, where an NES contains infinitely many optimal solutions.

Additionally, we suggest an adaptive multiobjective differential evolution (AMODE) to solve the transformed biobjective optimization problem effectively. Furthermore, by combining WeB with AMODE, we propose a generic framework for simultaneously locating multiple optimal solutions of NESs. It is empirically shown that the performance of our framework is highly competitive with a lot of well-established methods for NESs.

The rest of this paper is organized as follows. Section II introduces the related work. Section III presents WeB. Section IV describes AMODE. Meanwhile, Section IV gives the details of our generic framework for solving NESs. Section V provides the experimental results. Section VI discusses some issues in our framework. Section VII concludes this paper and points out some future research directions.

II. Related Work

Utilizing multiobjective optimization-based transformation techniques to deal with NESs is a novel idea that has arisen in recent years and is the main focus of this paper. As pointed out previously, in this kind of transformation technique, an NES is transformed into a multiobjective optimization problem. After this transformation, it is generally expected that the objective functions of the transformed optimization problem are in conflict with each other. This property plays an important role, as it can ensure that the optimal solutions of an NES are the Pareto optimal solutions of the transformed optimization problem. Otherwise, it is very difficult to identify the relationship between the original NES and the transformed optimization problem.

Next, we will briefly introduce five representative approaches. Among them, the last two approaches are originally designed for handling multimodal optimization problems [19]. Multimodal optimization problems have the same formulation as single-objective optimization problems. Note, however, that they involve multiple optimal solutions. One may be interested in why the last two approaches can be used for NESs. It is because if an NES is transformed into (2) or (3), this NES is also a single-objective optimization problem with multiple optimal solutions, which means that an NES with single-objective structure is essentially equivalent to a multimodal optimization problem. Thus, the multiobjective optimization-based transformation techniques, which convert a multimodal optimization problem into a multiobjective optimization problem, can be readily extended for coping with NESs.

A. CA

In 2008, Grosan and Abraham [14] proposed a new approach for NESs, called CA. By treating each equation as an objective function, CA converts an NES into the following m-objective optimization problem:

$$\begin{cases} \min \quad f_1(\mathbf{x}) = |e_1(\mathbf{x})| \\ \qquad\vdots \\ \min \quad f_m(\mathbf{x}) = |e_m(\mathbf{x})|. \end{cases} \tag{5}$$

CA provides a simple and straightforward transformation from an NES to a multiobjective optimization problem. However, the main drawback of CA is that its performance will significantly degrade as the number of equations increases [20].

B. MONES

In 2015, Song et al. [3] proposed MONES, which is a biobjective formulation for NESs. MONES consists of two parts: 1) the location function and 2) the system function. The former can be expressed as

$$\begin{cases} \min \ \alpha_1(\mathbf{x}) = x_1 \\ \min \ \alpha_2(\mathbf{x}) = 1 - x_1 \end{cases} \tag{6}$$

where x_1 is the first decision variable of an NES. It is easily deduced that the Pareto front of (6) is a line segment defined by $y = 1 - x$ in the objective space. In addition, the latter can be formulated as

$$\begin{cases} \min \quad \beta_1(\mathbf{x}) = \sum_{i=1}^{m} |e_i(\mathbf{x})| \\ \min \quad \beta_2(\mathbf{x}) = m \times \max\left(|e_1(\mathbf{x})|, \ldots, |e_m(\mathbf{x})|\right). \end{cases} \tag{7}$$

The transformed biobjective optimization problem can be obtained by combining these two parts

$$\begin{cases} \min \quad f_1(\mathbf{x}) = \alpha_1(\mathbf{x}) + \beta_1(\mathbf{x}) \\ \min \quad f_2(\mathbf{x}) = \alpha_2(\mathbf{x}) + \beta_2(\mathbf{x}). \end{cases} \tag{8}$$

It is interesting to note that for any optimal solution $\mathbf{x}^*$ of an NES, $\beta_1(\mathbf{x}^*) = \beta_2(\mathbf{x}^*) = 0$, which means that under this condition (8) degenerates to (6). As a result, all the optimal solutions of an NES are the Pareto optimal solutions of (8), and their images in the objective space are located on the line segment defined by $y = 1 - x$. In multiobjective optimization, linear Pareto front is the simplest type and, consequently, it enables the current multiobjective EAs to find the Pareto optimal solutions more easily compared with other types of Pareto front, such as nonlinear Pareto front [3]. However, since only the first decision variable is chosen to construct the location function, if several optimal solutions have the same value in the first decision variable, MONES might lose some of them.

C. Qin et al.'s Method

Inspired by MONES [3], Qin et al. [15] presented a $(n+1)$-objective transformation technique in 2015, where n is the number of decision variables. This transformation technique is also composed of two parts: 1) the location function and 2) the system function, which is shown in (9) at the bottom of the page. In (9), $R(\mathbf{x})$ is the system function which is the mean of the absolute values of all equations, and C is a parameter to control the shape of the Pareto front, which increases from 0 to infinity during the evolution.

In [15], all the objective functions in the location function conflict with each other, and thus, the optimal solutions of an NES are the Pareto optimal solutions of (9). Moreover, the location function is able to provide a one-to-one mapping from the Pareto optimal set to the Pareto front, thereby overcoming the drawback of MONES to a certain degree. However, similar to CA, this transformation technique will also suffer from the curse of dimensionality with the increase of n.

D. MOMMOP

In 2015, Wang et al. [21] developed a method named MOMMOP to deal with multimodal optimization problems, by generalizing the idea of MONES. Recognizing the shortcoming of MONES, MOMMOP makes use of each decision variable to construct a biobjective optimization problem like (8) and, therefore, n biobjective optimization problems appear. Because of the similarity between NESs and multimodal optimization problems, we revise MOMMOP to solve NESs as follows:

$$\begin{cases} \mathrm{BOP}_1 \begin{cases} \min \ x_1 + \beta_1(\mathbf{x}) \\ \min \ 1 - x_1 + \beta_2(\mathbf{x}) \end{cases} \\ \qquad\qquad\vdots \\ \mathrm{BOP}_n \begin{cases} \min \ x_n + \beta_1(\mathbf{x}) \\ \min \ 1 - x_n + \beta_2(\mathbf{x}). \end{cases} \end{cases} \tag{10}$$

In MOMMOP, when comparing two individuals (denoted as $\mathbf{x}_u$ and $\mathbf{x}_v$), we say $\mathbf{x}_u$ is better than $\mathbf{x}_v$ if $\mathbf{x}_u$ Pareto dominates $\mathbf{x}_v$ on all the n biobjective optimization problems in (10)

$$(\mathbf{x}_u \prec \mathbf{x}_v \text{ on } \mathrm{BOP}_1) \wedge \ldots \wedge (\mathbf{x}_u \prec \mathbf{x}_v \text{ on } \mathrm{BOP}_n). \tag{11}$$

Compared with MONES, MOMMOP achieves the performance improvement at the expense of higher computational time complexity. Additionally, it is hard to analyze the

$$\begin{cases} \min \quad f_1(\mathbf{x}) &= \frac{x_1}{n} + \frac{x_2}{n-1} + \ldots + \frac{x_{n-1}}{2} + \frac{x_n}{1} + C \times R(\mathbf{x}) \times \ln(n+2) \\ \min \quad f_2(\mathbf{x}) &= \frac{x_1}{n} + \frac{x_2}{n-1} + \ldots + \frac{x_{n-1}}{2} + (1 - x_n) + C \times R(\mathbf{x}) \times \ln(n+1) \\ \min \quad f_3(\mathbf{x}) &= \frac{x_1}{n} + \frac{x_2}{n-1} + \ldots + \frac{x_{n-2}}{3} + (1 - x_{n-1}) + C \times R(\mathbf{x}) \times \ln(n) \\ &\ \vdots \\ \min \quad f_n(\mathbf{x}) &= \frac{x_1}{n} + (1 - x_2) + C \times R(\mathbf{x}) \times \ln(3) \\ \min \quad f_{n+1}(\mathbf{x}) &= (1 - x_1) + C \times R(\mathbf{x}) \times \ln(2) \end{cases} \tag{9}$$

property of the Pareto front theoretically since there exist n 2-D objective spaces in MOMMOP.

E. MOBiDE

In 2013, Basak *et al.* [22] designed a biobjective differential evolution for multimodal optimization problems, abbreviated as MOBiDE. In MOBiDE, the first objective function aims to select the individuals of higher quality, and the purpose of the second objective function is to maintain the diversity to prevent the population from converging toward a single optimal solution. These two objective functions can be easily borrowed to solve NESs. For example, each individual $\mathbf{x}_i$ $(i \in \{1, \ldots, NP\})$ in the population is associated with the following two objective functions for NESs, where NP is the population size

$$\begin{cases} \min & f_1(\mathbf{x}_i) = \sum_{k=1}^{m} |e_k(\mathbf{x}_i)| \\ \min & f_2(\mathbf{x}_i) = -\frac{\Omega_i}{NP} \end{cases} \tag{12}$$

where $\Omega_i = \sum_{j=1}^{NP} \| \mathbf{x}_i - \mathbf{x}_j \|$, $\| \mathbf{x}_i - \mathbf{x}_j \|$ denotes the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ in the decision space, and $f_2(\mathbf{x}_i)$ represents the average Euclidean distance from $\mathbf{x}_i$ to all other members in the population.

However, the limitation of MOBiDE is that the two objective functions are not totally in conflict with each other, which leads to an unclear relationship between the optimal solutions of an NES and the Pareto optimal solutions of (12).

Remark 1: Apart from MOMMOP [21] and MOBiDE [22], many other biobjective transformation techniques for multimodal optimization problems have been proposed (see [23]–[25]). We can take advantage of these transformation techniques to solve NESs. Note, however, that they have a similar disadvantage as MOBiDE. Consequently, their capabilities for finding multiple optimal solutions of an NES in a single run are limited. Due to the space limitation, we omit them in this paper.

III. A Weighted Biobjective Transformation Technique

A. Motivation

Based on the above introduction, it is clear that multiobjective optimization-based transformation techniques, due to the similarity between NESs and multiobjective optimization problems, provide a natural advantage over other kinds of transformation techniques for NESs. As analyzed in Section II, in order to obtain competitive performance, the following four properties deserve much attention in this kind of transformation technique.

1) Biobjective structure, which has the least number of objective functions in multiobjective optimization problems, i.e., two.
2) Conflicting objective functions, which enable the optimal solutions of an NES to become the Pareto optimal solutions of the transformed multiobjective optimization problem.
3) Linear Pareto front, which is the simplest type of Pareto front for a multiobjective EA to approximate.
4) One-to-one mapping, which assures that all the optimal solutions of an NES are mapped into different points on the Pareto front in the objective space.

Interestingly, MONES possesses the first three properties. In this paper, we present a simple yet effective *we*ighted *b*iobjective transformation technique (WeB) that preserves the essential details of MONES while eliminating the effect of the fourth property.

B. WeB

In WeB, the location function is defined as

$$\begin{cases} \min & \gamma_1(\mathbf{x}) = \frac{\sum_{i=1}^{n} w_i \times x_i}{\sum_{i=1}^{n} w_i} \\ \min & \gamma_2(\mathbf{x}) = 1 - \frac{\sum_{i=1}^{n} w_i \times x_i}{\sum_{i=1}^{n} w_i} \end{cases} \tag{13}$$

where $\mathbf{w} = (w_1, \ldots, w_n)$ is the weight vector, and w_i is the ith weight randomly chosen from 0 and 1. With respect to the location function, we can give the following comments.

1) Equation (13) produces a weighted linear combination on all the decision variables.
2) The two objective functions totally conflict with each other.
3) The Pareto front is a line segment defined by $y = 1 - x$.
4) If $w_1 \neq 0$ and $(w_2, \ldots, w_n) = (0, \ldots, 0)$, the location function of WeB is equivalent to that of MONES, which means MONES is just a special case of WeB.

In addition, the system function is the same with (2) and $\beta_1(\mathbf{x})$ in (7) of MONES. It is because we would like to make the implementation and formulation as simple as possible.

The weighted biobjective optimization problem can be obtained by combining the location function with the system function

$$\begin{cases} \min & f_1(\mathbf{x}) = \frac{\sum_{i=1}^{n} w_i \times x_i}{\sum_{i=1}^{n} w_i} + \sum_{i=1}^{m} |e_i(\mathbf{x})| \\ \min & f_2(\mathbf{x}) = 1 - \frac{\sum_{i=1}^{n} w_i \times x_i}{\sum_{i=1}^{n} w_i} + \sum_{i=1}^{m} |e_i(\mathbf{x})|. \end{cases} \tag{14}$$

For an optimal solution $\mathbf{x}^*$ of an NES, (14) will degenerate to (13) since $\sum_{i=1}^{m} |e_i(\mathbf{x}^*)| = 0$, which signifies that all the optimal solutions of an NES are the Pareto optimal solutions of WeB and that the Pareto front of WeB is linear.

From the previous description, one can conclude that the implementation of WeB is very simple and it does not introduce any problem-dependent parameters. Moreover, WeB keeps the main properties of MONES, i.e., biobjective structure, conflicting objective functions, and linear Pareto front.

The major difference between WeB and its predecessor MONES is that in WeB, all the decision variables are utilized to design the location function in a linearly weighted fashion. As pointed out previously, MONES might fail to achieve a one-to-one mapping from the optimal set of an NES to the Pareto front, in the case of some optimal solutions having the same value in the first decision variable. That is, the optimal solutions with the same value in the first decision variable will be mapped into the same point on the Pareto front. However, regarding WeB, the probability that the optimal solutions with the same values in certain decision variables or completely different values in all the decision variables are mapped into

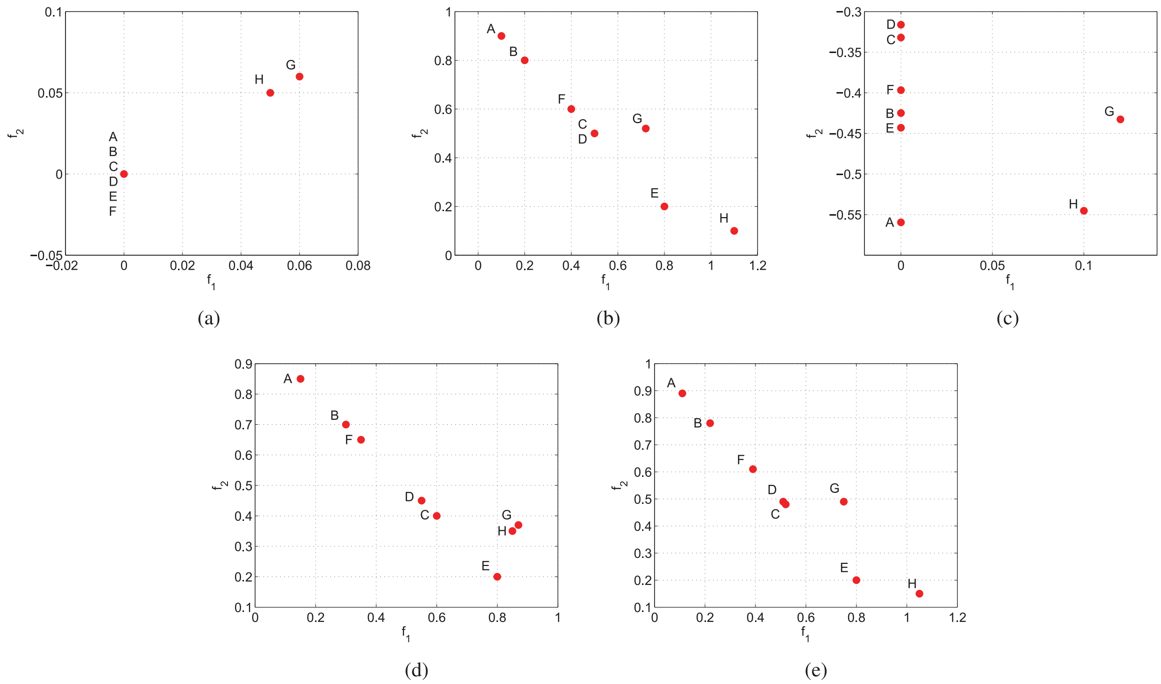

Fig. 1. Images of the eight individuals in the objective spaces defined by five different multiobjective optimization-based transformation techniques for the example in Table I. (a) CA [14]. (b) MONES [3]. (c) MOBiDE [22]. (d) WeB with $\mathbf{w} = (0.5, 0.5)$. (e) WeB with $\mathbf{w} = (0.9, 0.1)$.

TABLE I
EIGHT INDIVIDUALS IN THE EXAMPLE

Solution	x_1	x_2	e_1	e_2
A	0.1	0.2	0	0
B	0.2	0.4	0	0
C	0.5	0.7	0	0
D	0.5	0.6	0	0
E	0.8	0.8	0	0
F	0.4	0.3	0	0
G	0.6	0.9	0.06	0.06
H	1.0	0.5	-0.05	-0.05

the same point on the Pareto front is very low because of the random weights. Therefore, WeB also has the fourth property mentioned in Section III-A.

C. Analysis of the Principle

An example with two decision variables is considered to illustrate the working principles of five different multiobjective optimization-based transformation techniques, i.e., CA [14], MONES [3], MOMMOP [21], MOBiDE [22], and WeB. As introduced in Section II, the Qin *et al.*'s method [15] is also a multiobjective optimization-based transformation technique. However, this method depends on a dynamic control parameter C, and it is not trivial to analyze its performance.

As shown in Table I, this example involves two equations (e_1 and e_2). Suppose that there are eight individuals (denoted as A, B, C, D, E, F, G, and H) in the population. Among them, six individuals (i.e., A, B, C, D, E, and F) are the optimal solutions since all the values of the two equations are equal to zero, and the remaining two individuals (i.e., G and H) are not the optimal solutions since the absolute values of the two equations are greater than zero. Fig. 1 depicts the images of these eight individuals in the objective spaces defined by CA, MONES, MOBiDE, and WeB. Note that there are n biobjective optimization problems in MOMMOP, so we cannot provide a visualized result for it in the 2-D objective space.

Suppose also that the task is to select six individuals from the population for the next generation based on nondominated sorting [26]. Next, we are interested in what happens to these five multiobjective optimization-based transformation techniques.

1) In CA, all the optimal solutions of an NES are mapped into the origin [i.e., $(0, \ldots, 0)$] in the objective space since the objective function values of all the optimal solutions are equal to zero based on (5). As shown in Fig. 1(a), the images of the six optimal solutions lie in the origin. These six optimal solutions are the nondominated solutions in the population and will survive into the next generation.
2) With respect to MONES, C and D are mapped into the same point in the objective space, and seven individuals (six optimal solutions and H) are the nondominated solutions as shown in Fig. 1(b). Among these seven individuals, C and D have the smallest crowding distance [26]. Thus, A, B, C (or D), E, F, and H will be selected into the next generation. Clearly, one of the optimal solutions (i.e., C or D) is missed.
3) For MOMMOP, after a careful analysis, all the individuals in the population are the nondominated solutions according to (10). Subsequently, we compute their crowding distances in the decision space [21], and find that two of the optimal solutions (C and D) have the

smallest crowding distance and will be removed during the selection.

4) In terms of MOBiDE, all the optimal solutions are mapped into different points in the objective space and the population can be divided into six levels of nondominated set as shown in Fig. 1(c): {A}, {E, H}, {B, G}, {F}, {C}, and {D}. As a result, A, B, E, F, G, and H are the individuals with the most potential to be selected and two optimal solutions (C and D) are lost.
5) To implement WeB, all the weights in the weight vector $\mathbf{w}$ are randomly generated from 0 and 1. In order to analyze the effect of the weight vector $\mathbf{w}$, we vary the weights with a step-size equal to 0.1 and obtain the following 11 weight vectors: (0.0, 1.0), (0.1, 0.9), (0.2, 0.8), (0.3, 0.7), (0.4, 0.6), (0.5, 0.5), (0.6, 0.4), (0.7, 0.3), (0.8, 0.2), (0.9, 0.1), and (1.0, 0.0). According to our observation, all the optimal solutions correspond to different points in the objective space for all the weight vectors with the exception of $\mathbf{w} = (1.0, 0.0)$. It is because WeB with $\mathbf{w} = (1.0, 0.0)$ is roughly equivalent to MONES, thereby exhibiting the same drawback. In addition, for all the weight vectors except $\mathbf{w} = (1.0, 0.0)$ and $\mathbf{w} = (0.9, 0.1)$, the nondominated solutions are just the six optimal solutions, which will be chosen for the next generation. For example, the experimental results of WeB with $\mathbf{w} = (0.5, 0.5)$ are given in Fig. 1(d). However, WeB with both $\mathbf{w} = (1.0, 0.0)$ and $\mathbf{w} = (0.9, 0.1)$ tend to lose one of the optimal solutions (i.e., C or D). This can be attributed to the fact that in these two cases, the nondominated solutions consist of seven individuals (the six optimal solutions and H), and one of C and D will be eliminated due to their having the smallest crowding distance. Fig. 1(e) shows the experimental results of WeB with $\mathbf{w} = (0.9, 0.1)$.

We now summarize the above discussions from two aspects.

1) MOBiDE can provide a one-to-one mapping from the set of the optimal solutions to the Pareto front. For ten out of the 11 weight vectors, WeB is also able to achieve that. Unfortunately, it is a fact that CA will map all the optimal solutions into one point in the objective space and MONES will map the optimal solutions with the same value in the first decision variable into one point in the objective space.
2) CA succeeds in selecting the six optimal solutions into the next generation.[1] For nine out of the 11 weight vectors, WeB can also do that. However, MONES, MOBiDE, and MOMMOP definitely lose some of the optimal solutions.

Overall, WeB seems to be the best choice: it offers a one-to-one mapping while maintaining the optimal solutions in a vast majority of cases. It is necessary to emphasize that in the practical implementation of WeB, all the weights in the weight vector $\mathbf{w}$ are randomly generated. Thus, the probability that $\mathbf{w} = (1.0, 0.0)$ is very low and the performance of WeB will be further enhanced.

IV. An Adaptive Multiobjective Differential Evolution

As pointed out previously, when solving NESs by EAs, both the transformation technique and the optimization algorithm are vital. After transforming an NES into a biobjective optimization problem in Section III, the next issue is how to design an optimization algorithm to effectively solve the transformed biobjective optimization problem. To address this issue, we propose an adaptive multiobjective differential evolution, referred to as AMODE, which is an improved version of DEMO proposed in [27]. DEMO is mainly based on NSGA-II—a well-known multiobjective EA [26], whereas the search engine is replaced with differential evolution (DE)—a very popular EA paradigm [28]. Due to its simple structure, ease of implementation, and better performance than NSGA-II, DEMO[2] serves as the baseline optimization algorithm in AMODE. Moreover, two simple improvements are integrated within AMODE to make it more suitable for NESs as follows.

A. Parameter Adaptation

The performance of DE is significantly influenced by its parameter settings, such as the scaling factor F and the crossover control parameter CR [29], [30]. In this paper, the parameter adaptation of AMODE originates from a very competitive DE variant, namely SHADE [31], in which a historical memory of successful parameter settings has been used to produce future parameter values.

In SHADE, each individual $\mathbf{x}_i$ ($i \in \{1, \ldots, NP\}$) in the population has its own F and CR, denoted as F_i and CR_i. At each generation, the successful F_i and CR_i are stored into $\mathbf{S_F}$ and $\mathbf{S_{CR}}$, respectively. The main characteristic of SHADE is that it maintains a historical memory with H entries for $\mathbf{M_F} = \{M_{F,1}, \ldots, M_{F,H}\}$ and $\mathbf{M_{CR}} = \{M_{CR,1}, \ldots, M_{CR,H}\}$. The contents of $\mathbf{M_F}$ and $\mathbf{M_{CR}}$ are initialized to 0.5 and updated as follows:

$$M_{F,k} = \begin{cases} \text{mean}_{\text{WL}}(\mathbf{S_F}) & \text{if } \mathbf{S_F} \neq \emptyset \\ M_{F,k} & \text{otherwise} \end{cases} \tag{15}$$

$$M_{CR,k} = \begin{cases} \text{mean}_{\text{WA}}(\mathbf{S_{CR}}) & \text{if } \mathbf{S_{CR}} \neq \emptyset \\ M_{CR,k} & \text{otherwise} \end{cases} \tag{16}$$

where $k \in \{1, \ldots, H\}$ determines the position to update, $\text{mean}_{\text{WL}}(\mathbf{S_F})$ is the weighted Lehmer mean of $\mathbf{S_F}$, and $\text{mean}_{\text{WA}}(\mathbf{S_{CR}})$ is the weighted arithmetic mean of $\mathbf{S_{CR}}$.[3] During the evolution, the value of k increases generation by generation. If $k > H$, then it is reset to be 1. In SHADE, the contents of $\mathbf{M_F}$ and $\mathbf{M_{CR}}$ are utilized to produce F_i and CR_i for the next generation.

[1] As mentioned in Section II-A, CA will suffer from the curse of dimensionality with the increase of the number of equations.

[2] Note that three versions of DEMO are presented in [27], i.e., "DEMO/parent," "DEMO/closest/dec," and "DEMO/closest/obj." In this paper, DEMO/closest/dec is chosen because of its power to maintain high diversity of the population.

[3] Note that in (15) the weighted Lehmer mean is used since it can generate larger scaling factors than the weighted arithmetic mean [32]. In this way, the diversity of the population can be promoted.

Algorithm 1: Procedure of A-WeB

Input: *NP*: the population size;
H: the entries of the historical memory
Output: The nondominated solutions in the final population
1 Randomly generate a weight vector $\mathbf{w} = (w_1, \ldots, w_n)$, where w_i $(i \in \{1, \ldots, n\})$ is randomly chosen from 0 and 1;
2 Generate the initial population $\mathcal{P} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{NP}\}$;
3 Evaluate each individual in $\mathcal{P}$ according to the weighted biobjective optimization problem in Equation (14);
4 Set all values in the historical memory $\mathbf{M_F} = \{M_{F,1}, \ldots, M_{F,H}\}$ and $\mathbf{M_{CR}} = \{M_{CR,1}, \ldots, M_{CR,H}\}$ to 0.5;
5 $k = 1$;
6 **while** *the termination criterion is not met* **do**
7   Set $\mathcal{A} = \emptyset$, $\mathbf{S_F} = \emptyset$, and $\mathbf{S_{CR}} = \emptyset$;
8   **for** $i = 1$ *to NP* **do**
9     $j = \text{randint}(1, H)$;
10     $F_i = \text{randc}(M_{F,j}, 0.1)$;
11     $CR_i = \text{randn}(M_{CR,j}, 0.1)$;
12   **for** $i = 1$ *to NP* **do**
13   Generate the mutant vector $\mathbf{v}_i$ for $\mathbf{x}_i$ by Equation (17);
14   Generate the offspring $\mathbf{u}_i$ by implementing the binomial crossover of DE on $\mathbf{v}_i$ and $\mathbf{x}_i$;
15   Evaluate $\mathbf{u}_i$ according to the weighted biobjective optimization problem in Equation (14);
16   **for** $i = 1$ *to NP* **do**
17   Find the nearest individual in $\mathcal{P}$ to $\mathbf{u}_i$ according to the Euclidean distance in the decision space, denoted as $\mathbf{x}_s$;
18   **if** $\mathbf{x}_s$ *is Pareto dominated by* $\mathbf{u}_i$ **then**
19       $\mathbf{x}_s = \mathbf{u}_i$;
20   **else if** $\mathbf{x}_s$ *Pareto dominates* $\mathbf{u}_i$ **then**
21       $\mathbf{u}_i$ is discarded;
22   **else**
23       $\mathbf{u}_i$ is added into the archive $\mathcal{A}$;
24   **if** $\sum_{j=1}^{m} |e_j(\mathbf{u}_i)| \leq \sum_{j=1}^{m} |e_j(\mathbf{x}_i)|$ **then**
25       $\mathbf{S_F} \leftarrow \mathbf{S_F} \cup F_i$ and $\mathbf{S_{CR}} \leftarrow \mathbf{S_{CR}} \cup CR_i$;
26   **if** $\mathcal{A} \neq \emptyset$ **then**
27     $\mathcal{P} \leftarrow \text{NSGA-II}(\mathcal{A} \cup \mathcal{P})$;
28   **if** $\mathbf{S_F} \neq \emptyset$ *and* $\mathbf{S_{CR}} \neq \emptyset$ **then**
29   Update $M_{F,k}$ and $M_{CR,k}$ based on Equation (15) and Equation (16), respectively;
30     $k = k + 1$;
31   **if** $k > H$ **then**
32       $k = 1$;

B. Mutation Operator

Several classical mutation operators are known in the DE research community. In this paper, we employ the mutation operator "DE/current/1" to create a mutant vector for each individual $\mathbf{x}_i$ $(i \in \{1, \ldots, NP\})$ in the population

$$\mathbf{v}_i = \mathbf{x}_i + F_i \times \left(\mathbf{x}_{r_1} - \mathbf{x}_{r_2}\right) \tag{17}$$

where $\mathbf{v}_i$ is the mutant vector, the indices r_1 and r_2 are two mutually distinct integers randomly selected from $\{1, NP\}$, and F_i is a scaling factor between 0 and 1.

DE/current/1 is similar to a local search since the scaled difference of two individuals is directly added into the current individual $\mathbf{x}_i$. Thus, the search is carried out around the neighborhood of each individual, which facilitates the diversity of the population and provides an advantage to locate multiple optimal solutions. Note that DE/current/1 has also attracted a lot of attention in multiobjective optimization [33], [34].

C. Combining AMODE With WeB for Solving NESs

By combining AMODE with WeB, we propose a generic framework called A-WeB to simultaneously locate multiple optimal solutions of NESs, which is shown in Algorithm 1.

In each run of A-WeB, all the elements of the weight vector $\mathbf{w}$ are randomly generated and kept unchanged during the evolution, which means that each run has its own weight vector $\mathbf{w}$. Additionally, all the individuals are evaluated according to the weighted biobjective optimization problem in (14). During the evolution, firstly F_i and CR_i are produced for each individual $\mathbf{x}_i$ in the population $\mathcal{P}$ by lines 8–11, where randint$(1, H)$ is a randomly integer from 1 to H, randc$(\cdot, \cdot)$ is a random number obeying a Cauchy distribution, and randn$(\cdot, \cdot)$ is a random number obeying a Gaussian distribution.[4] Subsequently, the mutation operator DE/current/1 and the binomial crossover of DE are used to yield an offspring $\mathbf{u}_i$ for each individual $\mathbf{x}_i$ in $\mathcal{P}$. Afterward, the nearest individual $\mathbf{x}_s$ in $\mathcal{P}$ to $\mathbf{u}_i$ is identified and compared with $\mathbf{u}_i$ based on Pareto dominance. If $\sum_{j=1}^{m} |e_j(\mathbf{u}_i)| \leq \sum_{j=1}^{m} |e_j(\mathbf{x}_i)|$ which suggests that F_i and CR_i are the successful parameter settings, then they are stored into $\mathbf{S_F}$ and $\mathbf{S_{CR}}$, respectively. After the above update, NSGA-II [26] are utilized to choose *NP* individuals from $\mathcal{P}$ and the archive $\mathcal{A}$. Finally, the contents of the historical memory $\mathbf{M_F}$ and $\mathbf{M_{CR}}$ are updated based on (15) and (16), respectively. The above procedure is repeated until the termination criterion is met.

From the above explanations, it can be seen that the implementation of A-WeB is simple and its computational time complexity is the same with NSGA-II. In addition, it only contains two user-defined parameters, i.e., *NP* and *H*.

V. Empirical Studies

Thirty-eight test instances with a broad range of characteristics, denoted as F01–F38, are used for our empirical studies. These 38 test instances are chosen from [3], [14], and [35]–[37], and can be divided into three classes.

1) NESs with known optimal solutions (F01–F21), which include a number of optimal solutions.
2) NESs with unknown optimal solutions (F22–F25), which include infinitely many optimal solutions.
3) Ill-scaled NESs (F26–F38), in which the decision variables have different search ranges.

Table II summarizes the information of these 38 test instances and the details of them can be found in the supplementary material. For 15 test instances (i.e., F04, F07, F09, F12, F14, F15, F16, F18, F19, F20, F23, F24, F27, F30, and F38), some optimal solutions contain the same values in certain decision variables.

A. Performance Metrics

Based on [3] and [38], two performance metrics, i.e., peak ratio (*PR*) and success rate (*SR*), are used to assess the

[4]In this paper, F_i is generated obeying a Cauchy distribution and CR_i is generated obeying a Gaussian distribution. The reasons are twofold. On one hand, recognizing the outstanding performance of SHADE [31], our parameter adaptation follows SHADE. On the other hand, the Cauchy distribution is more helpful to diversify F_i than the Gaussian distribution [32]. Moreover, the Cauchy distribution coupled with the weighted Lehmer mean is more likely to produce larger values of F_i. Hence, under this condition the diversity of the population can be maintained, which is beneficial to find multiple optimal solutions of NESs simultaneously.

TABLE II
CHARACTERISTICS OF 38 TEST INSTANCES, WHERE *n* IS THE NUMBER OF DECISION VARIABLES, *S* IS THE DECISION SPACE, *LI* IS THE NUMBER OF LINEAR EQUATIONS, *NE* IS THE NUMBER OF NONLINEAR EQUATIONS, *NOS* IS THE NUMBER OF KNOWN OPTIMAL SOLUTIONS OF AN NES, *Max_FEs* IS THE MAXIMAL NUMBER OF FUNCTION EVALUATIONS, AND "ACTIVE OPTIMAL SOLUTIONS" INDICATES THAT SOME OPTIMAL SOLUTIONS HAVE THE SAME VALUES IN CERTAIN DECISION VARIABLES

Instance	n	$\mathcal{S}$	LE	NE	NOS	Max_FEs	Active Optimal Solutions
F01	2	$[-1,1]^n$	1	1	2	50,000	no
F02	20	$[-1,1]^n$	0	2	2	50,000	no
F03	2	$[-1,1]^n$	1	1	11	50,000	no
F04	2	$[-1,1]^n$	0	2	15	50,000	yes
F05	2	$[-10,10]^n$	0	2	13	50,000	no
F06	10	$[-2,2]^n$	0	10	1	50,000	no
F07	2	$[-1,1]^n$	1	1	8	50,000	yes
F08	4	$[0,5]^n$	0	4	1	50,000	no
F09	2	$[0,1]^n$	0	2	7	50,000	yes
F10	5	$[-10,10]^n$	4	1	3	100,000	no
F11	6	$[-1,1]^n$	0	6	1	50,000	no
F12	2	$[-2,2]^n$	0	2	10	50,000	yes
F13	2	$[-5,5]^n$	0	2	9	50,000	no
F14	2	$[0,2\pi]^n$	0	2	13	50,000	yes
F15	8	$[-1,1]^n$	0	8	16	100,000	yes
F16	2	$[-2,2]^n$	0	2	6	50,000	yes
F17	20	$[-2,2]^n$	19	1	2	200,000	no
F18	3	$[-1,1]^n$	0	3	7	50,000	yes
F19	2	$[-2,2]^n$	0	2	4	50,000	yes
F20	2	$[-2,2]^n$	0	2	6	50,000	yes
F21	3	$[0,1]^n$	0	3	8	100,000	no
F22	3	$[-1,1]^n$	1	1	infinite	50,000	no
F23	6	$[-1,1]^n$	0	6	infinite	50,000	yes
F24	20	$[-1,1]^n$	1	19	infinite	50,000	yes
F25	10	$[-10,10]^n$	4	6	infinite	50,000	no
F26	3	$[-5,5],[-1,3],[-5,5]$	0	3	2	50,000	no
F27	3	$[-0.6,6],[-0.6,0.6],[-5,5]$	0	3	12	50,000	yes
F28	2	$[0,1],[-10,0]$	0	2	2	50,000	no
F29	2	$[0,2.5],[-4,6]$	0	2	4	50,000	no
F30	2	$[-1,1],[-10,10]$	0	2	4	50,000	yes
F31	2	$[0.25,1],[1.5,2\pi]$	0	2	2	50,000	no
F32	3	$[3,5],[2,4],[0.5,2]$	0	3	1	50,000	no
F33	2	$[-1,-0.1],[-2,2]$	0	2	2	50,000	no
F34	3	$[1,2.5],[0.2,2],[0.1,3]$	0	3	1	50,000	no
F35	2	$[-5,1.5],[0,5]$	0	2	3	50,000	no
F36	2	$[0,2],[10,30]$	0	2	2	50,000	no
F37	3	$[0,2],[-10,10],[-1,1]$	0	3	5	50,000	no
F38	2	$[-2,2],[0,1.1]$	0	2	4	50,000	yes

performance of a method for NESs with known optimal solutions (F01–F21) and ill-scaled NESs (F26–F38) in this paper. Prior to introducing these two performance metrics, we need to explain how to determine the number of the optimal solutions found in a run. Since the optimal solutions of these NESs are known *a priori*, first, we choose one of the optimal solutions. If the minimum Euclidean distance between this optimal solution and all the solutions in the population obtained at the end of a run is less than a predefined accuracy level ϵ, then we consider that an optimal solution is found. Afterward, the same process will be executed for the remaining optimal solutions one by one, and finally the number of the optimal solutions found can be calculated. If all the optimal solutions can be found in a run, then it is called a successful run. *PR* denotes the average percentage of the optimal solutions found over all the runs and *SR* denotes the percentage of the successful runs. In this paper, $\epsilon = 0.01$ if $n \leq 5$, otherwise, $\epsilon = 0.1$.

For NESs with unknown optimal solutions (F22–F25), these two performance metrics cannot be directly applied. Under this condition, the hypervolume metric [39] is used.

Note that for all the three performance metrics (*PR*, *SR*, and hypervolume), the larger the value, the better the performance of a method.

B. Methods in Comparison and Experimental Setup

We compare A-WeB with the following nine methods.

1) *A-MONES, A-MOMMOP, and A-MOBiDE:* These three methods are obtained by combining AMODE with MONES [3], MOMMOP [21], and MOBiDE [22], respectively. MONES, MOMMOP, and MOBiDE have been introduced in Section II. Note that CA [14] is not chosen for comparison since its performance is outperformed by MONES as shown in [3]. In this paper, when implementing MONES, we randomly select a decision variable to construct the location function in each run.
2) *NCDE [40], NSDE [40], LIPS [41], and R3PSO [42]:* These are four state-of-the-art niching methods designed for multimodal optimization problems. We pointed out in Section II that the methods for multimodal optimization problems can be easily generalized to handle NESs. When making use of NCDE, NSDE, LIPS, and R3PSO to solve NESs, an NES is transformed into the single-objective optimization problem in (2).
3) *Rep-SHADE and Rep-CLPSO:* The repulsion strategy presented in [16] is combined with two powerful EAs, i.e., SHADE [31] and CLPSO [43], to solve NESs. The two resultant methods are referred to as Rep-SHADE

TABLE III

RESULTS OBTAINED BY THE MULTIPLE-PROBLEM WILCOXON TEST FOR THE TEN COMPARED METHODS ON F01–F21. R^+ MEANS THE SUM OF RANKS THAT A-WEB PERFORMS BETTER THAN ITS COMPETITOR, AND R^- IS THE SUM OF RANKS FOR THE OPPOSITE

A-WeB VS	PR			SR		
	R^+	R^-	p-value	R^+	R^-	p-value
A-MONES	220.0	11.0	**5.25E-05**	188.0	43.0	**5.22E-03**
A-MOMMOP	169.5	61.5	**4.41E-02**	163.0	68.0	1.03E-01
A-MOBiDE	229.5	1.5	**2.38E-06**	219.0	12.0	**4.77E-06**
NCDE	143.0	88.0	≥ 0.2	138.0	93.0	≥ 0.2
NSDE	145.0	86.0	≥ 0.2	143.0	88.0	≥ 0.2
LIPS	220.5	10.5	**1.91E-06**	229.5	1.5	**2.38E-06**
R3PSO	231.0	0.0	**9.54E-07**	220.5	10.5	**1.91E-06**
Rep-SHADE	141.0	90.0	≥ 0.2	149.0	82.0	≥ 0.2
Rep-CLPSO	192.0	39.0	**6.28E-03**	172.0	59.0	**3.45E-02**

TABLE IV

RANKINGS OBTAINED BY THE FRIEDMAN ALIGNED TEST FOR THE TEN COMPARED METHODS ON F01–F21. THE LOWER THE RANKING, THE BETTER THE PERFORMANCE OF A METHOD. THE BEST AND THE SECOND BEST RESULTS ARE HIGHLIGHTED IN BOLDFACE AND ITALIC, RESPECTIVELY

Method	Ranking (PR)	Ranking (SR)
A-WeB	**50.6667**	**56.0000**
A-MONES	98.3810	94.6905
A-MOMMOP	76.4762	81.1905
A-MOBiDE	161.2381	157.0000
NCDE	72.0238	80.2381
NSDE	71.6905	78.4048
LIPS	174.4524	165.8333
R3PSO	189.8095	172.0000
Rep-SHADE	*70.4286*	*76.5000*
Rep-CLPSO	89.8333	93.1429

and Rep-CLPSO. For Rep-SHADE and Rep-CLPSO, an NES is transformed into the following repulsion function once one optimal solution has been found:

$$\min \quad \sum_{i=1}^{m} |e_i(\mathbf{x})| + \beta \sum_{j=1}^{K} \mathrm{e}^{-\delta_j} \chi_\rho(\delta_j) \tag{18}$$

where

$$\delta_j = \| \mathbf{x} - \mathbf{x}_j^* \| \tag{19}$$

$$\chi_\rho(\delta_j) = \begin{cases} 1, & \text{if } \delta_j \leq \rho \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

K is the number of optimal solutions that have been found, $\mathbf{x}_j^*$ is the jth optimal solution, δ_j is the Euclidean distance between $\mathbf{x}$ and $\mathbf{x}_j^*$, ρ is a small constant to adjust the radius of repulsion areas, and β is a large penalty constant. As suggested in [16], $\rho = 0.01$ and $\beta = 1000$. From (18), we can see that the repulsion strategy creates the repulsion areas around the found optimal solutions and that an individual lies within one of the repulsion areas will be penalized. By doing this, the repulsion strategy has the potential to make the search algorithm find new optimal solutions.

It is evident that to achieve the simultaneous locating of multiple optimal solutions of an NES, among the aforementioned ten methods, the first four methods (i.e., A-WeB, A-MONES, A-MOMMOP, and A-MOBiDE) integrate the multiobjective optimization-based transformation techniques with a multiobjective EA (AMODE), while the remaining six methods integrate the single-objective optimization-based transformation techniques with either the niching strategy or the repulsion strategy.

In our experiments, the parameter settings of A-WeB were: $NP = 100$ and $H = NP$. For the other nine methods, NP was also fixed to 100 and the other parameter settings were the same as in their original papers. Since AMODE is also considered as the optimization algorithm in A-MONES, A-MOMMOP, and A-MOBiDE, the parameter H was fixed to NP for them. The detailed parameter settings of the ten compared methods were given in Table S-R-I of the supplementary material. Fifty independent runs were performed for each test instance with the maximal number of function evaluations (*Max_FEs*) as the termination criterion. Note that *Max_FEs* was set according to the difficulty of an NES as shown in Table II. To have a fair comparison, all the ten methods started with the same initial population in each of 50 runs.

C. Comparison on NESs With Known Optimal Solutions (F01–F21)

The *PR* and *SR* values resulting from the ten compared methods are summarized in Tables S-R-II and S-R-III of the supplementary material, respectively. Next, we will discuss the experimental results from the following four aspects.

1) A-WeB performs the best in comparison with the other nine methods since it obtains both the highest average *PR* value (0.8839) and the highest average *SR* value (0.64). In addition, A-WeB provides both the best *PR* values and the best *SR* values on ten test instances (i.e., F01, F03, F05, F06, F08, F10, F11, F14, F18, and F20). A-WeB also achieves 100% *PR* and 100% *SR* on seven test instances (i.e., F01, F03, F05, F06, F11, F14, and F20), which means that it succeeds in locating all the optimal solutions over all 50 runs. Moreover, the *PR* and *SR* values derived from A-WeB are larger than zero for all the test instances except F15.
2) A-MOMMOP, NCDE, NSDE, and Rep-SHADE show similar and competitive performance in terms of the average *PR* and *SR*. They have the capability to successfully solve five, five, five, and seven NESs, respectively. Note that NCDE and Rep-SHADE fail to find any optimal solution on one (F10) and two (F02 and F08) test instances, respectively.
3) Again, A-MONES and Rep-CLPSO present similar overall performance. However, they do not perform as well as the above five methods. They are able to locate all the optimal solutions of four and six test instances, respectively. Rep-CLPSO cannot find any optimal solution on three test instances (F02, F08, and F10).
4) The performance of A-MOBiDE, LIPS, and R3PSO is found to decrease remarkably. A-MOBiDE and LIPS can achieve 100% successful runs on only two (F06 and F11) and one (F01) test instance, respectively. R3PSO cannot solve any test instance consistently in all runs and does not yield good performance in a vast majority of test instances. As mentioned in Section II, for A-MOBiDE, the relationship between the optimal solutions of an original NES and the Pareto optimal solutions of the transformed biobjective optimization problem cannot be described explicitly. The poor performance of LIPS and

TABLE V

AVERAGE AND STANDARD DEVIATION OF THE HYPERVOLUME VALUES OF THE SEVEN COMPARED METHODS ON F22–F25

	F22	F23	F24	F25	$+/=/-$
A-WeB	0.296675 ± 0.035995	0.686763 ± 0.007750	0.039690 ± 0.001414	4.046544 ± 0.359629	/
A-MONES	0.288931 ± 0.035115 +	0.606079 ± 0.114344 +	0.027886 ± 0.002659 +	3.140386 ± 0.551425 +	4/0/0
A-MOMMOP	0.298077 ± 0.036335 −	0.662113 ± 0.008899 +	0.039333 ± 0.000078 =	3.898406 ± 0.244397 +	2/1/1
NCDE	0.291552 ± 0.033778 +	0.678183 ± 0.006296 +	0.031114 ± 0.001102 +	2.611568 ± 0.240187 +	4/0/0
NSDE	0.291554 ± 0.033673 +	0.680569 ± 0.006895 +	0.031185 ± 0.001429 +	2.676923 ± 0.219315 +	4/0/0
Rep-SHADE	0.296955 ± 0.035146 =	0.683148 ± 0.008029 +	0.034823 ± 0.001041 +	2.531974 ± 0.379190 +	3/1/0
Rep-CLPSO	0.300833 ± 0.036073 −	0.677834 ± 0.008074 +	0.034317 ± 0.001290 +	NA ± NA +	3/0/1
ref. point	(1,1)	(0.6,1.6)	(0.1,1.2)	(1.5,2.5)	/

"+", "=", and "−" indicate that A-WeB is better than, similar to, and worse than its competitor according to the Wilcoxon signed-rank test at $\alpha = 0.05$, respectively.

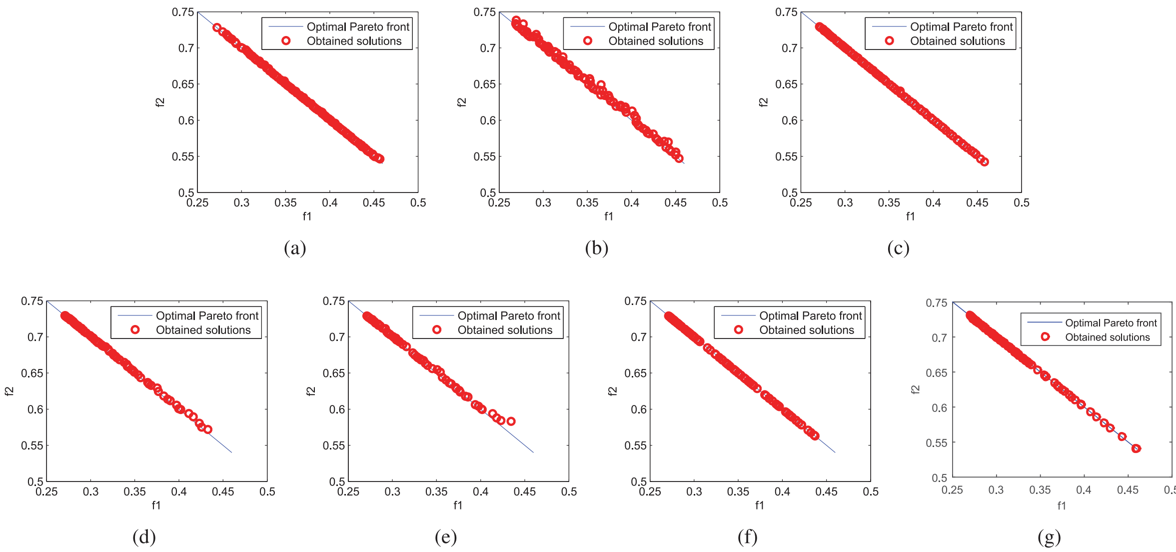

Fig. 2. Images of the obtained nondominated solutions of different methods in the objective space defined by (14) for F22 in a representative run. The solid line is the theoretical Pareto front with $y = 1 - x$. (a) A-WeB. (b) A-MONES. (c) A-MOMMOP. (d) NCDE. (e) NSDE. (f) Rep-SHADE. (g) Rep-CLPSO.

R3PSO could be explained by the fact that they employ simple velocity updating equation of PSO, and as a result, their search ability is limited.

We also tested the statistical differences of the ten compared methods by making use of the multiple-problem Wilcoxon test and the Friedman Aligned test [44]. It is noteworthy that the statistical tests were implemented via the KEEL software [45]. Moreover, we chose the Bonferroni–Dunn method as the *post-hoc* test for the Friedman Aligned test. The statistical test results are given in Tables III and IV.

As shown in Table III, in terms of the multiple-problem Wilcoxon test, A-WeB provides higher R^+ values than R^- values in all the cases for both the *PR* and *SR* metrics. Especially, A-WeB significantly outperforms A-MONES, A-MOBiDE, LIPS, R3PSO, and Rep-CLPSO in that all the p-values are less than 0.05. In addition, it can be seen from Table IV that with respect to both the *PR* and *SR* metrics, A-WeB has the best ranking, followed by Rep-SHADE.

D. Comparison on NESs With Unknown Optimal Solutions (F22–F25)

F22–F25 contain infinitely many optimal solutions. The *PR* and *SR* metrics are not suitable for evaluating the performance of a method on these four test instances. To complete the performance comparison, the hypervolume metric [39] was used. To make the comparison fair, the nondominated individuals in the final populations provided by different methods need to be mapped into the same objective space. In this paper, with the termination of each run, the nondominated individuals in the final populations provided by different methods are mapped into the 2-D objective space defined by (14). Since the Pareto front in this objective space is a line segment defined by $y = 1 - x$, we need to measure how close the nondominated individuals converge toward the Pareto front and how uniformly the nondominated individuals distribute along the Pareto front, which are the two essential goals in multiobjective optimization. Fortunately, the hypervolume metric is effective in measuring both the convergence and uniformness.

Table V records the average and standard deviation of the hypervolume values derived from different methods on F22–F25. In Table V, "NA" denotes that the experimental results of Rep-CLPSO are not available on F25 since Rep-CLPSO cannot find any optimal solution with the prespecified accuracy level (i.e., 0.01) and the repulsion strategy is not triggered under this condition. Due to the fact that A-MOBiDE, LIPS, and R3PSO do not yield good performance in Section V-C, their experimental results are not included for convenience. To test the statistical significance

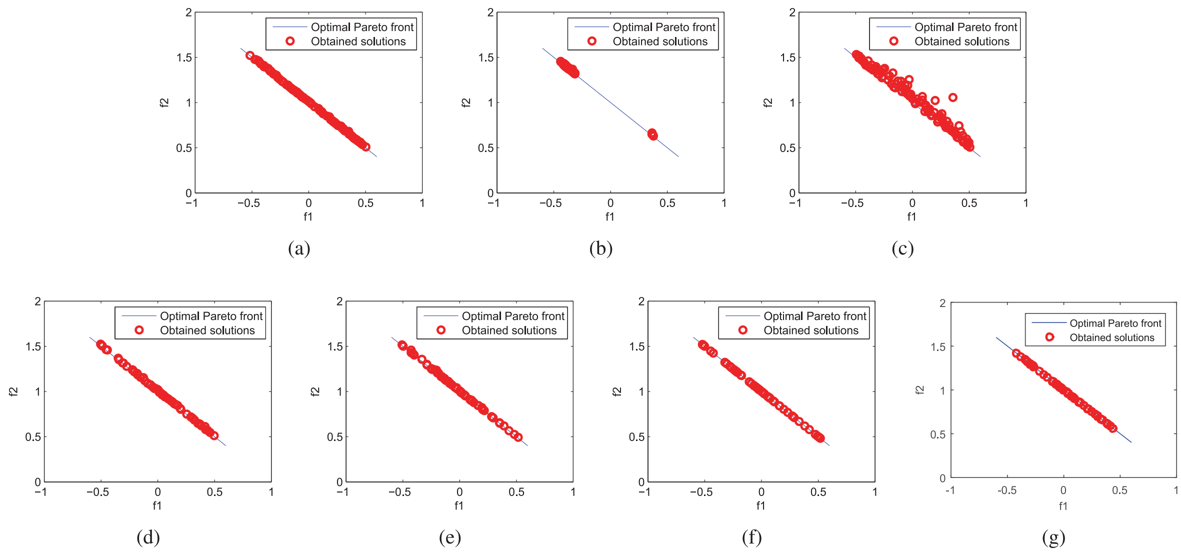

Fig. 3. Images of the obtained nondominated solutions of different methods in the objective space defined by (14) for F23 in a representative run. The solid line is the theoretical Pareto front with $y = 1 - x$. (a) A-WeB. (b) A-MONES. (c) A-MOMMOP. (d) NCDE. (e) NSDE. (f) Rep-SHADE. (g) Rep-CLPSO.

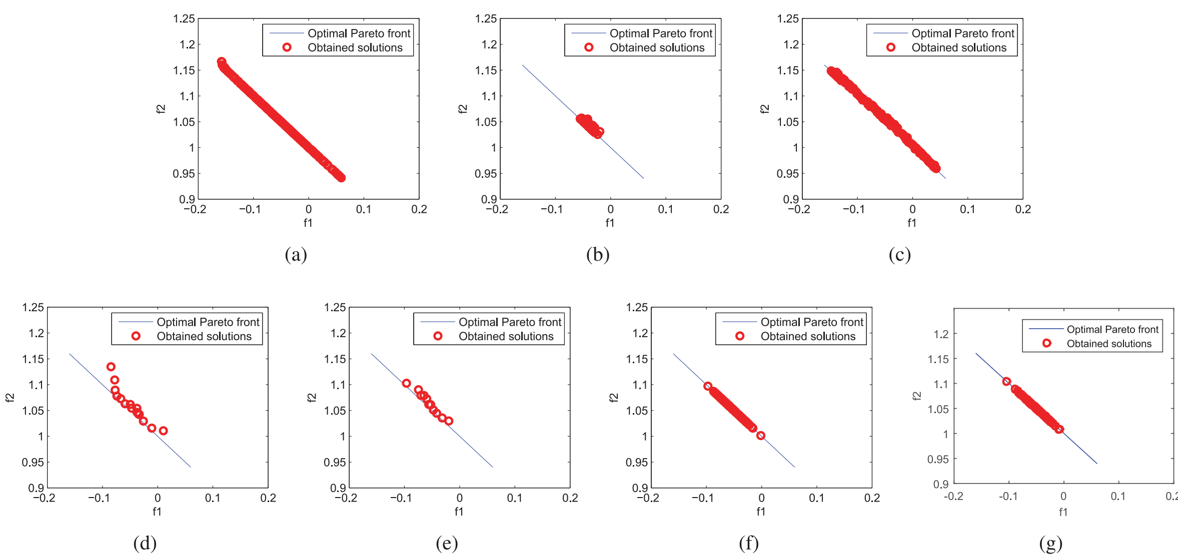

Fig. 4. Images of the obtained nondominated solutions of different methods in the objective space defined by (14) for F24 in a representative run. The solid line is the theoretical Pareto front with $y = 1 - x$. (a) A-WeB. (b) A-MONES. (c) A-MOMMOP. (d) NCDE. (e) NSDE. (f) Rep-SHADE. (g) Rep-CLPSO.

between A-WeB and each competitor, Wilcoxon signed-rank test at a 0.05 significance level was applied. As shown in Table V, A-WeB performs significantly better than A-MONES, A-MOMMOP, NCDE, NSDE, Rep-SHADE, and Rep-CLPSO on four, two, four, four, three, and three test instances, respectively. However, A-MOMMOP and Rep-CLPSO have an edge over A-WeB on only one test instance (F22), and A-MONES, NCDE, NSDE, and Rep-SHADE cannot surpass A-WeB on any test instance. The above comparison confirms that, on the whole, A-WeB presents the best performance among the seven compared methods on four test instances with infinitely many optimal solutions.

Next, we further study the performance differences by observing the nondominated solutions in the final population. Figs. 2–5 provide the experimental results of the seven compared methods in a representative run. In this paper, the run in which the hypervolume value of the nondominated solutions in the final population ranks 26th (from worst to best) is termed as a representative run among 50 runs. It is easy to see that A-WeB is able to consistently produce a set of representative nondominated solutions with good convergence and uniformness on F22–F25. Although the performance of A-MOMMOP is better than that of A-Web on F22, A-MOMMOP is not as effective as A-WeB for approximating the Pareto front of F23 and F25. The reason could be that when comparing two individuals in MOMMOP, the Pareto dominance should be checked on all the n biobjective optimization problems. If the Pareto dominance does not hold between them on any of the n biobjective optimization problems, then they are considered to be nondominated. Thus, the population of MOMMOP might contain a lot of nondominated solutions, which leads to low selection pressure and slow convergence

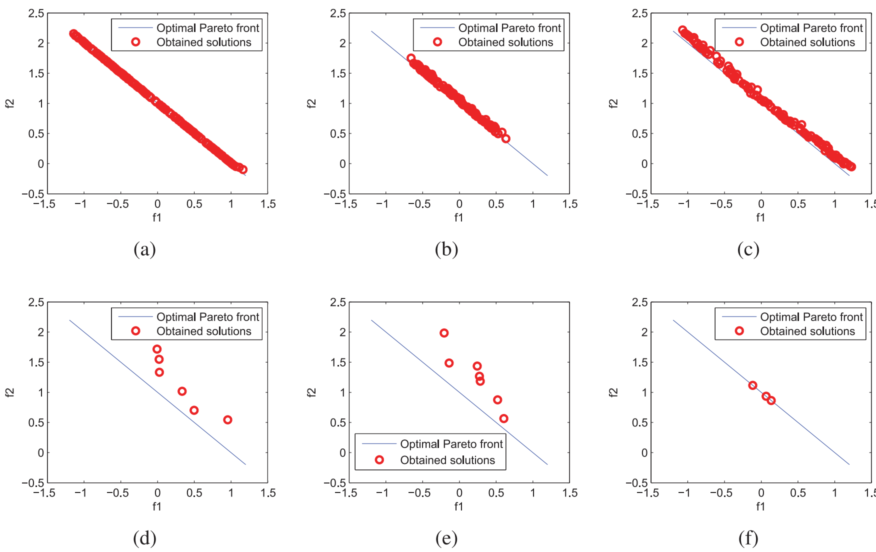

Fig. 5. Images of the obtained nondominated solutions of different methods in the objective space defined by (14) for F25 in a representative run. The solid line is the theoretical Pareto front with $y = 1 - x$. (a) A-WeB. (b) A-MONES. (c) A-MOMMOP. (d) NCDE. (e) NSDE. (f) Rep-SHADE.

TABLE VI
RESULTS OBTAINED BY THE MULTIPLE-PROBLEM WILCOXON TEST FOR THE SEVEN COMPARED METHODS ON F26–F38

A-WeB VS	PR			SR		
	R^+	R^-	p-value	R^+	R^-	p-value
A-MONES	73.0	18.0	5.74E-02	66.5	24.5	1.10E-01
A-MOMMOP	65.0	26.0	1.91E-01	72.5	18.5	**3.42E-02**
NCDE	66.0	25.0	1.68E-01	71.0	20.0	**4.74E-02**
NSDE	48.5	42.5	≥ 0.2	50.5	40.5	≥ 0.2
Rep-SHADE	41.0	50.0	≥ 0.2	45.5	45.5	≥ 0.2
Rep-CLPSO	69.5	21.5	1.02E-01	74.0	17.0	**2.39E-02**

TABLE VII
RANKINGS OBTAINED BY THE FRIEDMAN ALIGNED TEST FOR THE SEVEN COMPARED METHODS ON F26–F38. THE BEST AND THE SECOND BEST RESULTS ARE HIGHLIGHTED IN BOLDFACE AND ITALIC, RESPECTIVELY

Method	Ranking (PR)	Ranking (SR)
A-WeB	*39.5000*	**34.4615**
A-MONES	46.5769	43.5385
A-MOMMOP	53.6538	55.8846
NCDE	52.2692	60.8077
NSDE	**35.9231**	*36.1538*
Rep-SHADE	41.5000	36.8462
Rep-CLPSO	52.5769	54.3077

speed. As depicted in Figs. 3 and 4, A-MONES runs the risk of missing some parts of the Pareto front of F23 and F24, in which some optimal solutions have the same values in certain decision variables. The preformation degradation of A-MONES coincides with our analysis in Section II. Regarding the four single-objective optimization-based methods (NCDE, NSDE, Rep-SHADE, and Rep-CLPSO), they tend to yield decent performance when the number of decision variables is small (i.e., F22 and F23). But, as the number of decision variable increases (i.e., F24 and F25), they cannot obtain promising results in terms of the convergence and uniformness as shown in Figs. 4 and 5.

E. Comparison on Ill-Scaled NESs (F26–F38)

As for the previous test instances, the decision variables have the same search region. A question which arises naturally is how the performance of a method is influenced by the ill-scaled NESs, in which the search ranges of decision variables are different. To this end, we collect 13 ill-scaled NESs (F26–F38) in Table II and the performance of different methods is compared on them. Like Section V-D, the experimental results of A-MOBiDE, LIPS, and R3PSO are omitted.

Tables S-R-IV and S-R-V in the supplementary material summarize the *PR* and *SR* values provided by the seven compared methods, respectively. From Tables S-R-IV and S-R-V, it can be observed that A-WeB achieves the second best average *PR* value, and the same best average *SR* value with NSDE. Additionally, A-WeB provides the best results on nine out of 13 test instances for both the *PR* and *SR* metrics. It can successfully solve nine test instances over all 50 runs.

Tables VI and VII report the statistical test results based on the multiple-problem Wilcoxon test and the Friedman Aligned test, respectively. As shown in Table VI, A-WeB provides higher R^+ values than R^- values when comparing with all the competitors except Rep-SHADE. Rep-SHADE provides higher R^- value than R^+ value for *PR*, and the same R^+ and R^- value for *SR*. As far as the multiple-problem Wilcoxon test at $\alpha = 0.05$ is concerned, the significant differences can be observed in three cases (i.e., A-WeB versus A-MOMMOP, A-WeB versus NCDE, and A-WeB versus Rep-CLPSO) for *SR*, which suggests that under this condition A-WeB is significantly better than A-MOMMOP, NCDE, and Rep-CLPSO. In addition, Table VII indicates that A-WeB ranks the second best and the best in terms of the *PR* and *SR* metrics, respectively. Therefore, on the whole, we can conclude that the performance of A-WeB is highly competitive on the 13 ill-scaled NESs.

Based on the discussions in the above three sections, we can give the following remarks.

1) WeB has the capability to alleviate the drawback of MONES when some optimal solutions have the same values in certain decision variables, which verifies the

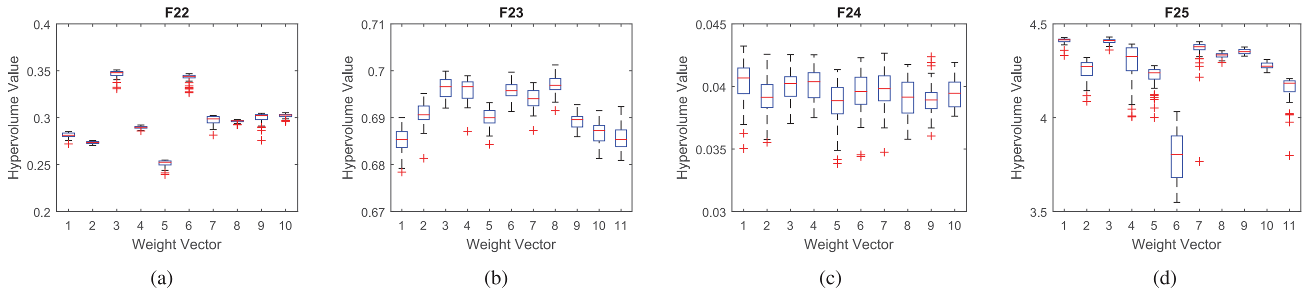

Fig. 6. Box plots of the hypervolume values derived from WeB with different weight vectors for (a)–(d) F22–F25.

main motivation of this paper. We attribute the superiority of WeB to the fact that it uses the information of all the decision variables under the biobjective framework by random weights.

2) The single-objective optimization-based methods with the niching or repulsion strategy (such as NCDE, NSDE, Rep-SHADE, and Rep-CLPSO) can find multiple optimal solutions of an NES in a single run. However, such strategies always introduce some user-defined parameters, which need to be set properly to solve different NESs. More importantly, this kind of method has no specific diversity maintenance mechanism as in multiobjective EAs and, consequently, its performance is poor for the NESs with infinitely many optimal solutions, especially when the number of decision variables is high, which makes WeB more attractive.
3) WeB is the best multiobjective optimization-based transformation technique compared with MONES, MOMMOP, and MOBiDE. A-WeB exhibits the best overall performance when solving different kinds of NESs.

VI. Discussion

The aim of this section is to study the robustness of A-WeB, the scalability of A-WeB, the effectiveness of some mechanisms in A-WeB, and the influence of the parameter settings on the performance of A-WeB.

A. On the Robustness of A-WeB

In A-WeB, the weight vector $\mathbf{w} = (w_1, \ldots, w_n)$ in (14) is randomly generated at the beginning of each run and is kept untouched during the evolution. Then, a straightforward question is how A-WeB performs with a fixed weight vector rather than a random weight vector over all 50 runs. Subsequently, we carried out experiments to answer this question. Specifically, we chose 11 weight vectors in which the first ten weight vectors were randomly produced and the last weight vector contained equal elements for all the decision variables, i.e., $w_1 = \cdots = w_n$. Fifty independent runs were implemented for A-WeB associated with each of the 11 weight vectors and test instances F22–F25 were used to produce the experimental results.

The box plots of the hypervolume values over 50 runs are shown in Fig. 6 for F22–F25. It is noteworthy that if $w_1 = \cdots = w_n$, then for any optimal solution $\mathbf{x}^* = (x_1^*, \ldots, x_n^*)$ of an NES, (14) becomes

$$\begin{cases} \min \quad f_1(\mathbf{x}) = \frac{1}{n} \times \sum_{i=1}^{m} x_i^* \\ \min \quad f_2(\mathbf{x}) = 1 - \frac{1}{n} \times \sum_{i=1}^{m} x_i^*. \end{cases} \tag{21}$$

A unique characteristic of F22 and F24 is that the sum of all the decision variables of an optimal solution is equal to a constant, and thus, the Pareto front is just one point in the 2-D objective space. Consequently, the experimental results of A-WeB with the 11th weight vector are not included in Fig. 6 for F22 and F24. It is necessary to point out that in the practical implementation of A-WeB, the probability of $w_1 = \cdots = w_n$ is extremely low due to the randomization.

As shown in Fig. 6, A-WeB performs similarly for the different fixed weight vectors and the hypervolume values move in a small range. Therefore, we can conclude that the performance of A-WeB is robust with regard to the change of the weight vector.

B. On the Scalability of A-WeB

To better comprehend the performance of A-WeB, its scalability is analyzed in this section. F05 is selected as a test instance because the number of optimal solutions (*NOS*) of F05 is scalable with the search range in each dimension. For example, if $\mathcal{S} = [-60, 60]^2, NOS = 76$; if $\mathcal{S} = [-70, 70]^2, NOS = 89$; if $\mathcal{S} = [-80, 80]^2, NOS = 101$; if $\mathcal{S} = [-90, 90]^2, NOS = 113$; and if $\mathcal{S} = [-100, 100]^2, NOS = 127$. It can be seen that all the above cases contain a large number of optimal solutions. Similar to Section V-D, A-WeB is compared with A-MONES, A-MOMMOP, NCDE, NSDE, Rep-SHADE, and Rep-CLPSO. For these seven compared methods, $NP = 200$ and $Max\_FEs = 300\,000$. All other parameter settings were kept unchanged.

Table VIII reports the experimental results of different methods on F05. From Table VIII, it is clear that A-WeB is able to consistently provide the best *PR* and *SR* values, regardless of the search range in each dimension. Moreover, the *SR* values provided by A-WeB are greater than 0 for all the cases, which means that A-WeB succeeds in finding all the optimal solutions in some independent runs. In contrast, NCDE, NSDE, and Rep-CLPSO cannot achieve any successful run. Therefore, we can conclude that A-WeB has better scalability than other compared methods, and has better potential to be applied to NESs with a large number of optimal solutions.

TABLE VIII
COMPARISON OF DIFFERENT METHODS ON F05 WITH A LARGE NOSS IN DIFFERENT DECISION SPACES. THE BEST RESULT FOR EACH CASE AMONG THE COMPARED METHODS IS HIGHLIGHTED IN BOLDFACE

Method	PR					SR				
	$[-60,60]^2$	$[-70,70]^2$	$[-80,80]^2$	$[-90,90]^2$	$[-100,100]^2$	$[-60,60]^2$	$[-70,70]^2$	$[-80,80]^2$	$[-90,90]^2$	$[-100,100]^2$
A-WeB	**0.9921**	**0.9892**	**0.9743**	**0.9122**	**0.9335**	**0.62**	**0.68**	**0.42**	**0.44**	**0.64**
A-MONES	0.9853	0.9676	0.9343	0.7754	0.8131	0.50	0.24	0.26	0.16	0.30
A-MOMMOP	0.2821	0.3611	0.4671	0.5427	0.7874	0.00	0.00	0.00	0.00	0.30
NCDE	0.4939	0.4335	0.3762	0.3347	0.2885	0.00	0.00	0.00	0.00	0.00
NSDE	0.3897	0.3204	0.2754	0.2276	0.1957	0.00	0.00	0.00	0.00	0.00
Rep-SHADE	0.6700	0.6638	0.6139	0.6234	0.6170	0.34	0.32	0.22	0.24	0.24
Rep-CLPSO	0.1555	0.1342	0.0988	0.0747	0.0688	0.00	0.00	0.00	0.00	0.00

TABLE IX
INFLUENCE OF THE NORMALIZATION OF THE DECISION VARIABLES FOR A-WEB. THE BETTER RESULT FOR EACH TEST INSTANCE BETWEEN THE COMPARED METHODS IS HIGHLIGHTED IN BOLDFACE. IN THE LAST ROW, THE RESULTS IN THE FORM OF (R^+, R^-, p) ARE OBTAINED BY THE MULTIPLE-PROBLEM WILCOXON TEST

Instance	PR		SR	
	A-WeB	A-WeB-1	A-WeB	A-WeB-1
F02	**0.6200**	0.3800	**0.36**	0.10
F04	**0.9573**	0.9413	**0.58**	0.40
F05	**1.0000**	0.8369	**1.00**	0.18
F07	**0.9400**	0.8875	**0.60**	0.34
F08	0.4200	**0.7000**	0.42	**0.70**
F09	0.8371	0.8371	0.12	0.12
F10	**0.8933**	0.7867	**0.68**	0.56
F12	**0.8880**	0.8580	**0.28**	0.24
F13	**0.9733**	0.8867	**0.76**	0.32
F15	**0.6688**	0.6450	0.00	0.00
F16	0.9433	**0.9800**	0.66	**0.88**
F17	**0.6200**	0.5400	**0.24**	0.08
F18	0.9514	**0.9714**	0.70	**0.80**
F19	0.9950	**1.0000**	0.98	**1.00**
F20	**1.0000**	0.9967	**1.00**	0.98
F21	0.8550	0.8550	0.14	0.14
F26	**1.0000**	0.9900	**1.00**	0.98
F27	0.0933	**0.6633**	0.00	**0.04**
F34	**0.8800**	0.8600	**0.88**	0.86
F36	**0.9400**	0.5900	**0.88**	0.18
F37	0.9320	**0.9880**	0.66	**0.94**
F38	**1.0000**	0.9950	**1.00**	0.98
Average	**0.8367**	0.8268	**0.5882**	0.4918
Wilcoxon test	(170.5, 82.5, 1.61E-01)		(168.5, 84.5, 1.52E-01)	

C. On the Normalization of the Decision Variables

A-WeB uses a linearly weighted sum of all the decision variables for its location function as shown in (13). This section studies the effect of normalization of the decision variables. For this purpose, the location function in (13) is modified as

$$\begin{cases} \min \quad \alpha_1(\mathbf{x}) = \frac{\sum_{i=1}^{n} w_i \times x_i'}{\sum_{i=1}^{n} w_i} \\ \min \quad \alpha_2(\mathbf{x}) = 1 - \frac{\sum_{i=1}^{n} w_i \times x_i'}{\sum_{i=1}^{n} w_i} \end{cases} \tag{22}$$

where $x_i' = (x_i - \underline{x}_i / \overline{x}_i - \underline{x}_i)$, $i = 1, \ldots, n$. In this way, x_i' is normalized in [0, 1].

With the modified location function, the resultant A-WeB variant is referred to as A-WeB-1. Table IX shows the performance of A-WeB and A-WeB-1 on test instances F01–F21 and F26–F38, where the experimental results are omitted when both of them achieve $PR = 1.0$ and $SR = 1.0$ for an NES. As can be seen, in contrast to A-WeB-1, A-Web loses on six out of 22 test instances, while gets better results on 14 test instances with respect to *PR*. For *SR*, A-WeB loses on six test instances, yet wins on 13 test instances. In general, A-WeB provides better *PR* and *SR* values on average, and higher R^+ values than R^- values for the *PR* and *SR* metrics.

TABLE X
RESULTS OBTAINED BY THE MULTIPLE-PROBLEM WILCOXON TEST FOR A-WEB WITH DIFFERENT HISTORICAL MEMORY SIZE *H* ON F01–F21

$H = 100$ VS	PR			SR		
	R^+	R^-	p-value	R^+	R^-	p-value
$H = 5$	181.5	49.5	**1.21E-02**	169.5	61.5	6.21E-02
$H = 10$	175.5	55.5	**3.68E-02**	160.5	70.5	9.73E-02
$H = 30$	145.5	85.5	≥ 0.2	135.5	95.5	≥ 0.2
$H = 50$	114.5	116.5	≥ 0.2	125.5	105.5	≥ 0.2
$H = 200$	59.5	171.5	≥ 0.2	73.0	158.0	≥ 0.2
$H = 300$	82.0	149.0	≥ 0.2	89.5	141.5	≥ 0.2
$H = 400$	73.5	157.5	≥ 0.2	77.0	154.0	≥ 0.2
$H = 500$	102.0	129.0	≥ 0.2	98.5	132.5	≥ 0.2

From the above analysis, it seems that the performance of A-WeB cannot be substantially improved by normalizing the decision variables in the location function. This topic still needs a thorough study in the future.

D. Influence of the Historical Memory Size H

The historical memory size H is one of the parameters in A-WeB. In the previous experiments, the default setting $H = NP = 100$ was adopted as in [31]. In this section, its influence on the performance of A-WeB is investigated empirically. The detailed experimental results on test instances F01–F21 are, respectively, given in Tables S-R-VI and S-R-VII of the supplementary material for *PR* and *SR*. In addition, the statistical test results by the multiple-problem Wilcoxon test are reported in Table X.

From Tables S-R-VI, S-R-VII, and X, A-WeB with $H < 100$ shows decreased performance against $H = 100$. On the other hand, for $H > 100$, the performance of A-WeB is improved compared with $H = 100$. The comparison in Table X also indicates that there are no significant differences among A-WeB with the values of H changing from 30 to 500 in that all the p-values are greater than 0.05. Therefore, A-WeB is insensitive to H and the value of H can be chosen from a large range, for example, from 30 to 500.

E. Influence of F and CR in DE

In A-WeB, the scaling factor F and the crossover control parameter CR of DE are tuned in an adaptive way. To study the influence of the parameter settings, A-WeB is compared with its variants with fixed F and CR. For this purpose, four commonly used settings are selected, i.e., $(F, CR) = (0.9, 0.1)$, $(0.9, 0.9)$, $(0.5, 0.3)$, and $(0.5, 0.9)$ [21], [32], [46], and their corresponding methods are referred to as A-WeB-2, A-WeB-3, A-WeB-4, and A-WeB-5, respectively. Tables S-R-VIII and S-R-IX in the supplementary material summarize the *PR* and *SR* values on test instances

TABLE XI
RESULTS OBTAINED BY THE MULTIPLE-PROBLEM WILCOXON TEST FOR A-WEB AND ITS FOUR VARIANTS WITH FIXED PARAMETER SETTINGS ON F01–F21

A-WeB VS	PR			SR		
	R^+	R^-	p-value	R^+	R^-	p-value
A-WeB-2	213.0	18.0	**2.41E-04**	202.0	29.0	**5.34E-04**
A-WeB-3	187.5	43.5	**2.71E-03**	182.5	48.5	**1.85E-02**
A-WeB-4	191.0	40.0	**7.10E-03**	183.5	47.5	**9.44E-03**
A-WeB-5	154.5	76.5	1.54E-01	139.0	92.0	≥ 0.2

TABLE XII
RANKINGS OBTAINED BY THE FRIEDMAN ALIGNED TEST FOR A-WEB AND ITS FOUR VARIANTS WITH FIXED PARAMETER SETTINGS ON F01–F21. THE BEST AND THE SECOND BEST RESULTS ARE HIGHLIGHTED IN BOLDFACE AND ITALIC, RESPECTIVELY

Method	Ranking (PR)	Ranking (SR)
A-WeB	**32.2381**	**30.5000**
A-WeB-2	70.6905	75.5476
A-WeB-3	64.8571	59.5000
A-WeB-4	48.6667	51.6905
A-WeB-5	*48.5476*	*47.7619*

F01–F21, respectively. The statistical test results obtained by the multiple-problem Wilcoxon test and the Friedman Aligned test are given in Tables XI and XII, respectively.

As shown in Tables S-R-VIII and S-R-IX, A-WeB provides the best average *PR* and *SR* values. From Table XI, A-WeB-2, A-WeB-3, and A-WeB-4 suffer from significant performance degradation since all the *p*-values are less than 0.05 when comparing with A-WeB in terms of the *PR* and *SR* metrics. It is evident from Table XII that A-WeB ranks the first.

The above comparison indicates that A-WeB exhibits superior performance against its variants with the fixed parameter settings, while avoiding a trial-and-error process to choose the proper parameter values.

F. Effectiveness of the Parameter Adaptation

A-WeB adapts the parameters *F* and *CR* of DE based on SHADE [31]. Note that adaptive parameter adaptation has been actively studied by DE researchers [47]. In this section, the adaptive parameter adaptation of SHADE is replaced with that of two state-of-the-art DE variants, i.e., jDE [48] and JADE [32], and the resultant methods are called jDE-WeB and JADE-WeB, respectively. We compared the performance of A-WeB with that of jDE-WeB and JADE-WeB. The *PR* and *SR* values resulting from the three compared methods are summarized in Tables S-R-X and S-R-XI of the supplementary material on test instances F01–F21, respectively. In addition, Table XIII reports the statistical test results obtained by the Friedman Aligned test.

As shown in Tables S-R-X, S-R-XI, and XIII, A-WeB provides the best average *PR* and *SR* values and ranks the first. Thus, A-WeB achieves the best overall performance. However, according to our observation, there are no significant differences between A-WeB and the two competitors in terms of the multiple-problem Wilcoxon test in both the *PR* and *SR* metrics. Hence, other adaptive parameter adaptation mechanisms are also effective for handling NESs under our framework.

TABLE XIII
RANKINGS OF THE METHODS FOR PARAMETER ADAPTATION BY THE FRIEDMAN ALIGNED TEST ON F01–F21. THE BEST AND THE SECOND BEST RESULTS ARE HIGHLIGHTED IN BOLDFACE AND ITALIC, RESPECTIVELY

Method	Ranking (PR)	Ranking (SR)
A-WeB	**25.3810**	**26.4524**
jDE-WeB	39.6429	40.3333
JADE-WeB	*30.9762*	*29.2143*

G. Effectiveness of the Mutation Operator

In A-WeB, the mutation operator DE/current/1 in (17) is applied. In this section, we compared it with another classical mutation operator "DE/rand/1." Note that there are also a lot of other mutation operators in the DE literature [49], [50]. We do not conduct comprehensive comparisons between DE/current/1 and them because it is out of the scope of this paper. The experimental results are provided in Table S-R-XII of the supplementary material for test instances F01–F21. When the two compared methods are both capable of achieving $PR = 1.0$ and $SR = 1.0$ for an NES, their results are not reported in Table S-R-XII.

As shown in Table S-R-XII, A-WeB with DE/current/1 provides better results than A-WeB with DE/rand/1 on most test instances both in terms of the *PR* and *SR* metrics. A-WeB with DE/current/1 can also obtain better average *PR* and *SR* values. With respect to the multiple-problem Wilcoxon test, although the differences are not significant at $\alpha = 0.05$, A-WeB with DE/current/1 still gets higher R^+ values than R^- values for both the *PR* and *SR* metrics. Therefore, A-WeB gets great benefit from DE/current/1 to find multiple optimal solutions of an NES simultaneously in a single run.

H. Effect of the Distance Comparison Criterion

For multimodal optimization problems, Wang *et al.* [21] presented a new distance comparison criterion to avoid a large attraction basin containing too many similar individuals and to make the distribution of the population more appropriate. In this section, this distance comparison criterion is also incorporated into A-WeB for solving NESs. The corresponding A-WeB variant is called A-WeB-6. In A-WeB-6, an individual $\mathbf{x}_u$ is said to be better than another individual $\mathbf{x}_v$ if

$$\sum_{i=1}^{m} |e_i(\mathbf{x}_u)| < \sum_{i=1}^{m} |e_i(\mathbf{x}_v)| \wedge norm\_dist(\mathbf{x}_u, \mathbf{x}_v) < \delta \tag{23}$$

where $norm\_dist(\mathbf{x}_u, \mathbf{x}_v)$ denotes the normalized Euclidean distance between $\mathbf{x}_u$ and $\mathbf{x}_v$, and δ is a distance threshold which is set to 0.01 [21]. The experimental results are given in Table S-R-XIII of the supplementary material for test instances F01–F21. Again, the experimental results of those test instances, for which A-WeB and A-WeB-6 can achieve 100% *PR* and 100% *SR*, are omitted.

From Table S-R-XIII, A-WeB-6 is able to obtain better results on 12 and 11 out of 16 test instances in terms of the *PR* and *SR* metrics, respectively. Compared with A-WeB, A-WeB-6 also gets better average *PR* and *SR* values, and higher R^+ values than R^- values for both the *PR* and *SR* metrics. Therefore, this distance comparison criterion is applicable to further improve the performance of A-WeB.

VII. Conclusion

This paper has proposed a weighted biobjective transformation technique named WeB to formulate an NES as a biobjective optimization problem, which extends our previous work [3]. WeB attempts to produce a weighted linear combination of all the decision variables in the two objective functions. Thanks to random weights, WeB is very likely to achieve a one-to-one mapping from the optimal solutions of an NES to different points on the Pareto front of the transformed biobjective optimization problem. Therefore, it can alleviate the risk of losing some optimal solutions with the same values in certain decision variables due to the many-to-one mapping in [3]. Subsequently, we have also suggested an adaptive multiobjective DE named AMODE as the optimization algorithm. By combining WeB with AMODE, a generic framework referred to as A-WeB, has been proposed for dealing with NESs. It is worth noting that the Pareto front of the transformed biobjective optimization problem is linear. We thus expect that A-Web is able to effectively locate multiple Pareto optimal solutions in a single run. As a result, the optimal solutions of an NES can also be obtained correspondingly.

The performance of A-WeB has been extensively tested on 38 test instances which include 21 NESs with known optimal solutions, four NESs with infinitely many optimal solutions, and 13 ill-scaled NESs. Moreover, A-WeB is compared with three multiobjective optimization-based transformation techniques equipped with AMODE as the optimization algorithm and six well-established single-objective optimization-based methods. The empirical studies verify that A-WeB has the best overall performance across these three kinds of NESs.

Since there is no prior knowledge about the importance of each decision variable, the weights in A-WeB were randomly generated. In the future, we plan to design adaptive/self-adaptive weights by online analysis of the importance of the decision variables according to the properties of NESs at hand. Additionally, developing other advanced optimization algorithms (such as the multioperator-based EAs [46], [51]) for NESs will be another part of our future work.

The C++ source code of A-WeB can be downloaded from Y. Wang's homepage: http://ist.csu.edu.cn/YongWang.htm.

Acknowledgment

The authors would like to thank Prof. A. Zhou for his constructive comments and suggestions. They would also like to thank Prof. P. N. Suganthan for providing the source codes of NCDE, NSDE, and LIPS.

References

[1] D. Mehta and C. Grosan, "A collection of challenging optimization problems in science, engineering and economics," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Sendai, Japan, May 2015, pp. 2697–2704.

[2] C. L. Karr, B. Weck, and L. M. Freeman, "Solutions to systems of nonlinear equations via a genetic algorithm," *Eng. Appl. Artif. Intell.*, vol. 11, no. 3, pp. 369–375, 1998.

[3] W. Song, Y. Wang, H.-X. Li, and Z. Cai, "Locating multiple optimal solutions of nonlinear equation systems based on multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 414–431, Jun. 2015.

[4] R. Brits, A. P. Engelbrecht, and F. van den Bergh, "Solving systems of unconstrained equations using particle swarm optimization," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, vol. 3. 2002, pp. 6–9.

[5] Y. Mo, H. Liu, and Q. Wang, "Conjugate direction particle swarm optimization solving systems of nonlinear equations," *Comput. Math. Appl.*, vol. 57, nos. 11–12, pp. 1877–1882, 2009.

[6] A. Noriega, J. L. Cortizo, E. Rodriguez, R. Vijande, and J. M. Sierra, "A new method to approximate the field of movements of 1-DOF linkages with lower-pairs," *Meccanica*, vol. 45, no. 5, pp. 681–692, 2010.

[7] M. Jaberipour, E. Khorram, and B. Karimi, "Particle swarm algorithm for solving systems of nonlinear equations," *Comput. Math. Appl.*, vol. 62, no. 2, pp. 566–576, 2011.

[8] C. A. Voglis, K. E. Parsopoulos, and I. E. Lagaris, "Particle swarm optimization with deliberate loss of information," *Soft Comput.*, vol. 16, no. 8, pp. 1373–1392, 2012.

[9] P. S. Mhetre, "Genetic algorithm for linear and nonlinear equation," *Int. J. Adv. Eng. Technol.*, vol. 3, no. 2, pp. 114–118, 2012.

[10] H. A. E. Oliveira, Jr., and A. Petraglia, "Solving nonlinear systems of functional equations with fuzzy adaptive simulated annealing," *Appl. Soft Comput.*, vol. 13, no. 11, pp. 4349–4357, 2013.

[11] O. E. Turgut, M. S. Turgut, and M. T. Coban, "Chaotic quantum behaved particle swarm optimization algorithm for solving nonlinear system of equations," *Comput. Math. Appl.*, vol. 68, no. 4, pp. 508–530, 2014.

[12] A. F. Kuri-Morales, "Solution of simultaneous non-linear equations using genetic algorithms," *WSEAS Trans. Syst.*, vol. 2, no. 1, pp. 44–51, 2003.

[13] A. Pourrajabian, R. Ebrahimi, M. Mirzaei, and M. Shams, "Applying genetic algorithms for solving nonlinear algebraic equations," *Appl. Math. Comput.*, vol. 219, no. 24, pp. 11483–11494, 2013.

[14] C. Grosan and A. Abraham, "A new approach for solving nonlinear equations systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 3, pp. 698–714, May 2008.

[15] S. Qin, S. Zeng, W. Dong, and X. Li, "Nonlinear equation systems solved by many-objective hype," in *Proc. IEEE Congr. Evol. Comput.*, Sendai, Japan, 2015, pp. 2691–2696.

[16] M. J. Hirsch, P. M. Pardalos, and M. G. C. Resende, "Solving systems of nonlinear equations with continuous GRASP," *Nonlin. Anal. Real World Appl.*, vol. 10, no. 4, pp. 2000–2006, 2009.

[17] E. Pourjafari and H. Mojallali, "Solving nonlinear equations systems with a new approach based on invasive weed optimization algorithm and clustering," *Swarm Evol. Comput.*, vol. 4, pp. 33–43, Jun. 2012.

[18] N. Henderson, W. F. Sacco, and G. M. Platt, "Finding more than one root of nonlinear equations via a polarization technique: An application to double retrograde vaporization," *Chem. Eng. Res. Design*, vol. 88, nos. 5–6, pp. 551–561, 2010.

[19] S. Das, S. Maity, B.-Y. Qu, and P. N. Suganthan, "Real-parameter evolutionary multimodal optimization—A survey of the state-of-the-art," *Swarm Evol. Comput.*, vol. 1, no. 2, pp. 71–88, 2011.

[20] H. Ishibuchi, N. Akedo, and Y. Nojima, "Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 264–283, Apr. 2015.

[21] Y. Wang, H.-X. Li, G. G. Yen, and W. Song, "MOMMOP: Multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 830–843, Apr. 2015.

[22] A. Basak, S. Das, and K. C. Tan, "Multimodal optimization using a biobjective differential evolution algorithm enhanced with mean distance-based selection," *IEEE Trans. Evol. Comput.*, vol. 17, no. 5, pp. 666–685, Oct. 2013.

[23] J. Yao, N. Kharma, and P. Grogono, "Bi-objective multipopulation genetic algorithm for multimodal function optimization," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 80–102, Feb. 2010.

[24] K. Deb and A. Saha, "Multimodal optimization using a bi-objective evolutionary algorithm," *Evol. Comput.*, vol. 20, no. 1, pp. 27–62, 2012.

[25] S. Bandaru and K. Deb, "A parameterless-niching-assisted bi-objective approach to multimodal optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Cancùn, Mexico, 2013, pp. 95–102.

[26] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[27] T. Robič and B. Filipič, "DEMO: Differential evolution for multiobjective optimization," in *Evolutionary Multi-Criterion Optimization* (LNCS 3410), C. A. C. Coello, A. H. Aguirre, and E. Zitzler, Eds. Heidelberg, Germany: Springer, 2005, pp. 520–533.

[28] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, Dec. 1997.

[29] R. Gämperle, S. Müler, and P. Koumoutsakos, "A parameter study for differential evolution," in *Proc. Int. Conf. Adv. Intell. Syst. Fuzzy Syst. Evol. Comput. (WSEAS)*, Interlaken, Switzerland, 2002, pp. 293–298.
[30] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 55–66, Feb. 2011.
[31] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Cancún, Mexico, 2013, pp. 71–78.
[32] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, Oct. 2009.
[33] 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.
[34] K. Li, A. Fialho, S. Kwong, and Q. Zhang, "Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 114–130, Feb. 2014.
[35] R. B. Kearfott, "Some tests of generalized bisection," *ACM Trans. Math. Softw.*, vol. 13, no. 3, pp. 197–220, 1987.
[36] K. Meintjes and A. P. Morgan, "Chemical equilibrium systems as numerical test problems," *ACM Trans. Math. Softw.*, vol. 16, no. 2, pp. 143–151, 1990.
[37] W. F. Sacco and N. Henderson, "Finding all solutions of nonlinear systems using a hybrid metaheuristic with fuzzy clustering means," *Appl. Soft Comput.*, vol. 11, no. 8, pp. 5424–5432, 2011.
[38] X. Li, A. Engelbrecht, and M. G. Epitropakis, "Benchmark functions for CEC'2013 special session and competition on niching methods for multimodal function optimization," Evol. Comput. Mach. Learn. Group, RMIT Univ., Melbourne, VIC, Australia, Tech. Rep., 2013.
[39] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: Wiley, 2001.
[40] B. Y. Qu, P. N. Suganthan, and J. J. Liang, "Differential evolution with neighborhood mutation for multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 5, pp. 601–614, Oct. 2012.
[41] B. Y. Qu, P. N. Suganthan, and S. Das, "A distance-based locally informed particle swarm model for multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 3, pp. 387–402, Jun. 2013.
[42] X. Li, "Niching without niching parameters: Particle swarm optimization using a ring topology," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 150–169, Feb. 2010.
[43] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281–295, Jun. 2006.
[44] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 special session on real parameter optimization," *J. Heuristics*, vol. 15, no. 6, pp. 617–644, 2009.
[45] J. Alcalá-Fdez, L. Sánchez, and S. García. (2015). *KEEL: A Software Tool to Assess Evolutionary Algorithms to Data Mining Problems*. [Online]. Available: http://www.keel.es/
[46] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398–417, Apr. 2009.
[47] M. Drozdik, H. Aguirre, Y. Akimoto, and K. Tanaka, "Comparison of parameter control mechanisms in multi-objective differential evolution," in *Learning and Intelligent Optimization* (LNCS 8994), C. Dhaenens, L. Jourdan, and M.-E. Marmion, Eds. Cham, Switzerland: Springer, 2015.
[48] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Trans. Evol. Comput.*, vol. 10, no. 6, pp. 646–657, Dec. 2006.
[49] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Berlin, Germany: Springer-Verlag, 2005.
[50] J. Zhang and A. C. Sanderson, *Adaptive Differential Evolution: A Robust Approach to Multimodal Problem Optimization*. Berlin, Germany: Springer-Verlag, 2009.
[51] W. Gong, A. Zhou, and Z. Cai, "A multioperator search strategy based on cheap surrogate models for evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 746–758, Oct. 2015.

Wenyin Gong received the B.Eng., M.Eng., and Ph.D. degrees in computer science from the China University of Geosciences, Wuhan, China, in 2004, 2007, and 2010, respectively.

He is a Professor with the School of Computer Science, China University of Geosciences. He has published over 50 research papers in journals and international conferences. His current research interests include evolutionary algorithms, evolutionary optimization, and their applications.

Dr. Gong served as a Reviewer for over 20 international journals, such as the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON CYBERNETICS, the *IEEE Computational Intelligence Magazine*, the ACM TRANSACTIONS ON INTELLIGENT SYSTEMS AND TECHNOLOGY, *Information Sciences*, the *European Journal of Operational Research*, *Applied Soft Computing*, and the *International Journal of Hydrogen Energy*.

Yong Wang (M'08) received the B.S. degree in automation from the Wuhan Institute of Technology, Wuhan, China, in 2003, and the M.S. degree in pattern recognition and intelligent systems and the Ph.D. degree in control science and engineering from Central South University (CSU), Changsha, China, in 2006 and 2011, respectively.

He is currently an Associate Professor with the School of Information Science and Engineering, CSU. His current research interests include theory, algorithm design, and applications of computational intelligence.

Dr. Wang was a recipient of the Hong Kong Scholar by the Mainland—Hong Kong Joint Post-Doctoral Fellows Program, China, in 2013, the Excellent Doctoral Dissertation by Hunan Province, China, in 2013, the New Century Excellent Talents in University by the Ministry of Education, China, in 2013, the 2015 IEEE Computational Intelligence Society Outstanding Ph.D. Dissertation Award, the Hunan Provincial Natural Science Fund for Distinguished Young Scholars, in 2016, and the EU Horizon 2020 Marie Sklodowska-Curie Fellowship, in 2016. He is currently serving as an Associate Editor for *Swarm and Evolutionary Computation*. He was a Reviewer of over 50 international journals and a PC Member of over 40 international conferences. He is a member of the IEEE CIS Task Force on Nature-Inspired Constrained Optimization and the IEEE CIS Task Force on Differential Evolution.

Zhihua Cai received the B.Sc. degree from Wuhan University, Wuhan, China, in 1986, the M.Sc. degree from the Beijing University of Technology, Beijing, China, in 1992, and the Ph.D. degree from the China University of Geosciences, Wuhan, in 2003.

He is currently a Faculty Member with the School of Computer Science, China University of Geosciences. He has published over 100 research papers. His current research interests include data mining, machine learning, evolutionary computation, and their applications.

Shengxiang Yang (M'00–SM'14) received the B.Sc. and M.Sc. degrees in automatic control and the Ph.D. degree in systems engineering from Northeastern University, Shenyang, China, in 1993, 1996, and 1999, respectively.

He is currently a Professor of Computational Intelligence and the Director of the Center for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester, U.K. He has over 220 publications. His current research interests include evolutionary computation, swarm intelligence, computational intelligence in dynamic and uncertain environments, artificial neural networks for scheduling, and relevant real-world applications.

Dr. Yang serves as an Associate Editor or an Editorial Board Member of eight international journals, such as the IEEE TRANSACTIONS ON CYBERNETICS, *Information Sciences*, *Evolutionary Computation*, *Neurocomputing*, and *Soft Computing*.