

# Common Properties of Scalable Multiobjective Problems and a New Framework of Test Problems

Hiroyuki Masuda, Yusuke Nojima, and Hisao Ishibuchi  
Department of Computer Science and Intelligent Systems  
Graduate School of Engineering, Osaka Prefecture University  
Sakai, Osaka 599-8531, Japan  
{hiroyuki.masuda@ci., nojima@, hisaoi@}cs.osakafu-u.ac.jp

**Abstract**—A multiobjective test problem is called “scalable” when the number of its objectives can be arbitrarily specified. Evolutionary many-objective optimization algorithms are usually evaluated using scalable test problems. However, their design is not easy due to the difficulty in formulating a Pareto front and a feasible region in a high-dimensional objective space. As a result, a wide variety of scalable test problems have not been proposed. First, in this paper, existing scalable test problems are examined from some new viewpoints such as the uniqueness of the optimal solution for an objective and the presence of an optimal solution for multiple objectives. It is shown that most existing scalable test problems have some common properties. Next, a new framework for test problem design is proposed. The proposed framework enables us to design the Pareto front in a highly flexible manner. More specifically, we can specify not only its curvature property (e.g., convex, concave and linear) but also its shape (e.g., triangle, rotated triangle, pentagon and hexagon). The feasible region of the objective space can be also designed flexibly. Finally, some scalable test problems are generated by the proposed framework. It is clearly shown that the generated test problems have totally different properties from existing scalable test problems.

**Keywords**—Evolutionary multiobjective optimization; scalable test problem; multiobjective test problems, many-objective test problems, evolutionary many-objective optimization.

## I. INTRODUCTION

Usually evolutionary multiobjective optimization (EMO) algorithms can efficiently solve multiobjective optimization problems when the number of objectives is two or three. However, it has been reported in the literature that the performance of EMO algorithms is significantly degraded by the increase in the number of objectives [1]-[3]. Multiobjective optimization problems with four or more objectives are called many-objective problems. So far, a number of approaches have been proposed to tackle many-objective problems. Solving many-objective problems is a hot topic in the EMO community.

The performance of EMO algorithms is usually evaluated by using multiobjective test problems [4]-[6]. As pointed out in [7], development of EMO algorithms and multiobjective test problems can be seen as a coevolution. That is, development of EMO algorithms is based on (or biased by) the test problems, and new test problems are created so that existing EMO algorithms cannot efficiently solve them. Multiobjective test problems have played and continue to play an important role in both the evaluation and the development of EMO algorithms.

A multiobjective test problem is called scalable when the number of objectives can be specified to an arbitrary number. Frequently used scalable test problems are DTLZ [8] and WFG [9]. The performance of EMO algorithms for many-objective problems is usually evaluated by scalable test problems [10], [11]. Since it is difficult to understand and design the Pareto front and the feasible region in the high-dimensional objective space, formulation of scalable test problems is very difficult.

Test problems should provide a wide variety of different properties to evaluate overall performance of EMO algorithms. With respect to the curvature of the Pareto front, various types such as convex, concave and linear have been used [9], [12]. However, if we focus on the shape instead of the curvature, we can say that the Pareto fronts of most scalable test problems have a similar shape. Fig. 1 shows the Pareto fronts of three-objective scalable test problems. As we can see, they have a similar triangle shape (whereas they were usually said to have different shapes). The test problems in Fig. 1 also share some common properties with respect to the optimal solution for each objective. One is that each objective has a number of different optimal solutions. For example, the optimal solutions of  $f_3$  (i.e., the vertical axis) are all solutions on the thick blue curve on the bottom of each plot. Another common property is that a single solution can be the optimal for multiple objectives. For example, the point A in Fig. 1 is the optimal solution of  $f_2$  and  $f_3$ . That is, we can optimize two objectives simultaneously.

In this paper, we show that most existing scalable test problems have the common properties such as multiple optimal solutions for each objective and a single optimal solution for multiple objectives. That is, existing test problems do not have enough variety with respect to some important aspects, which have not been discussed in the literature. We propose a new framework of scalable test problems to generate a wide variety of multiobjective scalable test problems.

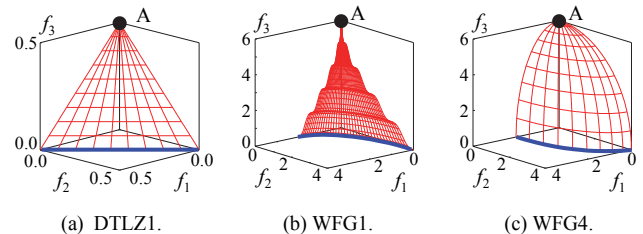


Fig. 1. Pareto fronts of three-objective test problems.

## II. MULTIOBJECTIVE SCALABLE TEST PROBLEMS

This section classifies existing scalable test problems. The structures of those problems are also explained. We focus on the continuous, unconstrained (i.e., box-constrained) scalable test problems with the known Pareto fronts. Here, “scalable” means that the number of objectives can be specified as an arbitrary number. Other classes of test problems such as combinatorial optimization problems or constraint problems are excluded from the scope of this study. Non-scalable multiobjective test problems (i.e., problems with the fixed number of objectives) are also excluded from our discussions. The continuous, unconstrained problems are the most widely studied and frequently used class of test problems in the EMO community [5], [11], [13]. We assume that all objectives are to be minimized. Test problems in this class can be written as

$$\text{Minimize } f(\mathbf{z}) = (f_1(\mathbf{z}) \ f_2(\mathbf{z}) \ \dots \ f_M(\mathbf{z}))^T, \quad (1)$$

$$\text{subject to } z_i^{\text{Min}} \leq z_i \leq z_i^{\text{Max}}, \quad i = 1, 2, \dots, n, \quad (2)$$

where

$M$ : Number of objectives,

$\mathbf{z}$ :  $n$ -dimensional decision vector,

$z_i^{\text{Min}}, z_i^{\text{Max}}$ : Lower and upper bounds for the  $i$ -th variable  $z_i$ .

In [8], three approaches were proposed to create scalable multiobjective test problems. We use these concepts to classify and explain existing test problems. The three approaches are the *multiple single-objective functions approach*, *bottom-up approach* and *constraint surface approach*. The third approach requires constraint conditions. This approach is not discussed because constrained problems are out of the focus of this paper.

### A. Multiple Single-Objective Functions Approach

In this approach, different single objective functions are used for the different objectives. This is simple and intuitive formulation of multiobjective test problems and often used to create two-objective (i.e., non-scalable) problems in the 1990s [5], [7], [9]. SPH- $m$  [14] is a scalable multiobjective test problem created from the well-known Sphere function with different optimal solutions. The  $m$ -th objective of SPH- $m$  can be written as follows:

$$f_m(\mathbf{z}) = z_1^2 + z_2^2 + \dots + z_{m-1}^2 + (z_m - 1)^2 + z_{m+1}^2 + \dots + z_n^2, \quad (3)$$

$$-1000 \leq z_i \leq 1000 \quad \text{for } i = 1, 2, \dots, n.$$

The drawback of this approach is that the optimal solutions and the shape of the Pareto front are not clear. This approach is less popular than the other approach in creating scalable test problems. Distance Minimization Problems (DMP) [15]-[17] and its variants (including SPH- $m$ ) [14], [18] are the only scalable test problems created by the multiple single-objective functions approach. An objective in a DMP is the distance from a solution to a fixed point in the decision space. The number of objectives is the same as the number of points specified in the decision space. The Euclidean distance is often used. Since the distance cannot be a multimodal function, all objectives are unimodal. If we use a multimodal function instead of the distance as a measure of the cost of a solution with respect to a fixed point, the Pareto front of a modified DMP becomes unclear. DMPs have been mainly used as simple problems to visually analyze the behavior of EMO algorithms in the decision space [17], [18].

### B. Bottom-Up Approach

Many existing scalable test problems were created by the bottom-up approach. DTLZ1-7 [8] are the standard scalable test problems created by this approach. WFG [9], P [19] and CEC16 [20] are also created by the same approach and they have similar structures to DTLZ. The main advantage of the bottom-up approach is that the Pareto front and the feasible region in the objective space can be designed separately. Hence, the formulation of the Pareto front is easy and flexible. In this approach, the decision variables are separated into the two groups: Position variables and distance variables. In this paper, we denote them by  $\mathbf{x}$  and  $\mathbf{y}$ , respectively. The Pareto front is designed by using the position variables, and the distance variables are used to create the feasible region.

We explain the concept of the bottom-up approach by using DTLZ. DTLZ1-4 problems with  $M$  objectives can be written as

$$f_i(\mathbf{z}) = (1 + g(\mathbf{y})) \times h_i(\mathbf{x}), \quad i = 1, 2, \dots, M, \quad (4)$$

where  $g(\mathbf{y})$  is a non-negative distance function, and  $h_i(\mathbf{x})$  is a shape function. The first  $(M - 1)$  decision variables are the position variables  $\mathbf{x}$  while the others are the distance variables  $\mathbf{y}$ :  $\mathbf{z} = (z_1, z_2, \dots, z_n) = (\mathbf{x}, \mathbf{y}) = (x_1, x_2, \dots, x_{M-1}, y_1, y_2, \dots, y_{n-M+1})$ . The shape functions  $h_i(\mathbf{x})$  determine the shape of the Pareto front while the distance function  $g(\mathbf{y})$  measures the distance of a solution from the Pareto front. The distance function  $g(\mathbf{y})$  has a solution  $\mathbf{y}^*$  for which  $g(\mathbf{y}^*) = 0$  holds. When  $g(\mathbf{y}) = 0$ , each objective  $f_i(\mathbf{z})$  in (4) is the same as the shape function  $h_i(\mathbf{x})$ :  $f_i(\mathbf{z}) = f_i(\mathbf{x}, \mathbf{y}^*) = h_i(\mathbf{x})$ . In this case, all feasible values of the position variables  $\mathbf{x}$  are Pareto optimal. The Pareto front is constructed by generating objective vectors using all feasible values of  $\mathbf{x}$ . The set of the generated vectors is the Pareto front. When  $g(\mathbf{y}) > 0$ , solutions are not Pareto optimal because all objectives can be simultaneously improved by decreasing  $g(\mathbf{y})$ .

Fig. 2 shows the role of the position and the distance variables for the two-objective DTLZ2. The first decision variable  $x$  is the position variable (since  $M = 2$ ) and the other variables  $\mathbf{y}$  are the distance variables. The first variable  $x$  changes the location of the solution along the Pareto front, and the others  $\mathbf{y}$  change its distance from the Pareto front.

In the WFG test problems, the sum of the distance and shape functions is used instead of their multiplication. The main feature of WFG is that  $\mathbf{x}$  and  $\mathbf{y}$  are calculated from its original decision variables by applying multiple *transformation functions*. The objectives are calculated by using  $\mathbf{x}$  and  $\mathbf{y}$ .

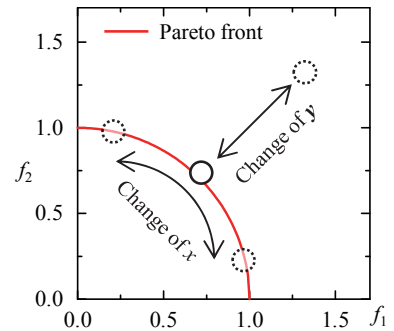


Fig. 2. Structure of the two-objective DTLZ2. The first decision variable  $x$  is the position variable. The other decision variables  $\mathbf{y}$  are the distance variables.

### III. COMMON PROPERTIES OF SCALABLE TEST PROBLEMS

In this section, we point out that almost all existing scalable test problems have common properties, which are not likely to be seen in real-world problems. The next sub-sections explain those common special properties. The existing scalable test problems and their properties are summarized in a table later.

#### A. Multiple Optimal Solutions for Each Objective Function

Test problems in single-objective optimization usually have only a single optimal solution [13]. So, we may expect that each objective in a multiobjective optimization problem has only a single optimal solution. However, each objective in existing scalable test problems usually has multiple different optimal solutions in the objective space.

As explained in Fig. 1, there are multiple Pareto optimal solutions which optimize each objective. Moreover, in DTLZ and P problems, some optimal solutions for a single objective are not Pareto optimal. For example, from the definition of each objective in DTLZ in (4), if the  $i$ -th shape function  $h_i(\mathbf{x})$  is zero,  $g(\mathbf{y})$  does not change the  $i$ -th objective value since  $f_i(\mathbf{z}) = (1 + g(\mathbf{y})) \times h_i(\mathbf{x}) = 0$ . Therefore, by changing the value of  $g(\mathbf{y})$  while keeping  $h_i(\mathbf{x}) = 0$ , other non-Pareto optimal solutions can be generated. All of those solutions are optimal for the  $i$ -th objective since  $f_i(\mathbf{z}) = 0$ .

#### B. Single Optimal Solution for Multiple Objective Functions

In multiobjective optimization, it is assumed that there is no single solution which optimizes all objectives simultaneously. However, almost all objectives in DTLZ and WFG can be optimized simultaneously. In the case of three-objective, any pair of two objectives can be optimized simultaneously as shown in Fig. 1. In many-objective problems, this property looks more strange or unrealistic. For  $M$ -objective DTLZ and WFG problems, if we formulate a  $k$ -objective problem by using  $k$  out of  $M$  objectives ( $k < M$ ), the Pareto front of the  $k$ -objective problem is always a single point for any choice of  $k$  objectives: A single point can optimize all the  $k$  objectives.

This issue was already pointed out in [21]. Since the Pareto front degenerates into only one point if we omit one objective, this property is very problematic in the objective reduction. To evaluate the performance of objective reduction approaches, DTLZ2<sub>BZ</sub> was proposed in [21]. Any two-objectives cannot be optimized in DTLZ2<sub>BZ</sub>. The Pareto front of the three-objective DTLZ2<sub>BZ</sub> is shown in Fig. 3 (a). Another test problem handling this issue is the inverted DTLZ1 [22]. The Pareto front is shown in Fig. 3 (b). As illustrated in [22], the state-of-the-art reference point-based or direction vector-based many-objective algorithms [11], [23] seem to implicitly assume or utilize a similar shape of Pareto fronts of test problems (see Fig. 1).

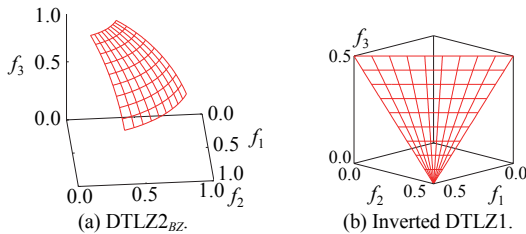


Fig. 3. Pareto fronts different from DTLZ1 and others.

#### C. All Values of Position Variables are Pareto Optimal

In the bottom-up approach, Pareto optimal solutions are obtained by setting the distance function  $g(\mathbf{y})$  as zero. For example, the set of Pareto optimal solutions  $X^*$  for DTLZ2 is written as follows:

$$X^* = \{\mathbf{z} = (\mathbf{x}, \mathbf{y}) \mid 0 \leq x_i \leq 1, y_j = 0.5 \text{ for all } i, j\}. \quad (5)$$

The distance function becomes zero when all distance variables are 0.5. All solutions are Pareto optimal solutions regardless the values of the position variables if the distance variables are 0.5. This means that once we obtain a Pareto optimal solution, other Pareto optimal solutions can be generated by changing the position variables. This simple property was criticized in [24]. As more complicated test problems, the non-scalable test problem suit F [24] and its scalable version P [19] were proposed. Their Pareto optimal sets can be written as

$$X^* = \{\mathbf{z} = (\mathbf{x}, \mathbf{y}) \mid 0 \leq x_i \leq 1, y_j = y_j^*(\mathbf{x}) \text{ for all } i, j\}, \quad (6)$$

where  $y_j^*(\mathbf{x})$  is a function to determine the optimal value of the distance variable. While the optimal values of the distance variables are complicated due to the function  $y_j^*(\mathbf{x})$ , the optimal values of the position variables are still simple. As shown in Table I, almost all problems have this property. Since the multiple single-objective functions approach does not have position variables, they do not have this property.

#### D. Bias Toward the Optimum

DTLZ and WFG have an unintended (or intended) bias toward the optimal value when the number of objectives is increased. For example, we show the first and eighth objectives of the eight-objective DTLZ2:

$$\begin{aligned} f_1(\mathbf{y}) &= (1 + g(\mathbf{y})) \times \cos\left(\frac{\pi}{2}x_1\right) \cos\left(\frac{\pi}{2}x_2\right) \cos\left(\frac{\pi}{2}x_3\right) \\ &\quad \times \cos\left(\frac{\pi}{2}x_4\right) \cos\left(\frac{\pi}{2}x_5\right) \cos\left(\frac{\pi}{2}x_6\right) \cos\left(\frac{\pi}{2}x_7\right), \\ f_8(\mathbf{y}) &= (1 + g(\mathbf{y})) \times \sin\left(\frac{\pi}{2}x_1\right), \\ g(\mathbf{y}) &= \sum_i (y_i - 0.5)^2, \quad 0 \leq x_i \leq 1. \end{aligned} \quad (7)$$

In this formulation,  $(1 + g(\mathbf{y}))$  is common. The difference between these two objectives is the definition of the shape functions (i.e., the cosine and sine terms). Since  $0 \leq \cos(\pi x_i/2) \leq 1$  holds for all position variables  $x_i$ , the first objective  $f_1(\mathbf{y})$  is biased toward zero. In DTLZ and WFG,  $(M - m + 1)$  terms are multiplied to calculate the  $m$ -th objective. When the number of objectives is increased, objectives with small  $m$  such as the first and second objectives are strongly biased while those with large  $m$  such as the last objective are not biased.

Table I shows which test problem has this bias. P and CEC16\_F21 do not have this property. This is because the dimension of their Pareto fronts are one or two regardless the number of objectives. The above mentioned bias is not strong when the dimension of the Pareto front is small even in DTLZ and WFG problems. DTLZ7 and the test problems based on multiple single-objective functions do not have this bias even in the high-dimensional Pareto fronts.

TABLE I. COMMON PROPERTIES OF EXISTING SCALABLE MULTI-OBJECTIVE TEST PROBLEMS.

Category	Problem	A. Multiple Optima	B. Simultaneous Optimization	C. Position Variables	D. Bias to Optimum	E. Distance Variables	F. Same Difficulties
Bottom-Up Approach (Based on DTLZ)	DTLZ1-4 [8]	✓	✓	✓	✓	✓	✓
	DTLZ7 [8]	✓	✓			✓	
	DTLZ2 <sub>BZ</sub> [21]	✓		✓	✓	✓	✓
	Inverted DTLZ1 [22]	✓		✓		✓	✓
Bottom-Up Approach (Based on WFG)	WFG1 [9]	✓	✓	✓	✓	✓	✓
	WFG2 [9]	✓	✓		✓	✓	✓
	WFG4-6 [9]	✓	✓	✓	✓	✓	✓
	WFG7 [9]	✓	✓	✓	✓		✓
	WFG8 [9]	✓	✓	✓	✓	✓	✓
Bottom-Up Approach (Based on P)	WFG9 [9]	✓	✓	✓	✓		✓
	P1-5 [19]	✓		✓		✓	✓
	P6 [19]	✓	✓	✓		✓	✓
	P7 [19]	✓		✓		✓	✓
	CEC16_F21 [20]	✓		✓		✓	✓
Multiple Single-Objective Functions Approach	CEC16_F22 [20]	✓	✓	✓	✓	✓	✓
	DMP [15]-[17]						✓
	SPH- <i>m</i> [14]					✓	✓
	Pareto-Box [18]	✓	✓			✓	✓

### E. Dominance Relation and Distance Variables

In DTLZ and WFG, all objective values can be changed by the same distance function  $g(\mathbf{y})$ . This means that all objectives can be improved by decreasing  $g(\mathbf{y})$ . The improved solution in this manner always dominates the original one. For example, let us assume that we have five solutions with the same position variable  $\mathbf{x}$  and different distance variables  $y_1, \dots, y_5$ , which are improved in this manner. In this case, the five solutions can be ordered using the Pareto dominance relation as shown in (8). Thus multiobjective optimization for the fixed values of the position variables  $\mathbf{x}$  can be handled as single-objective optimization of the distance function  $g(\mathbf{y})$ . Even in the case of many objectives, this is a single-objective problem of  $g(\mathbf{y})$  with only the distance variables  $\mathbf{y}$ .

$$f(\mathbf{x}, y_1) \preceq f(\mathbf{x}, y_2) \preceq f(\mathbf{x}, y_3) \preceq f(\mathbf{x}, y_4) \preceq f(\mathbf{x}, y_5). \quad (8)$$

### F. Same Difficulty for All Objectives

Each objective of a multiobjective problem can have different characteristics. For example, the first objective can be a complicated multimodal function while the second one can be a simple unimodal function. Some algorithms try to allocate different amounts of the available computation resource to different objectives [25], [26]. However, the existing scalable test problems have the same characteristics for all objectives.

It seems easy to define different characteristics for different objectives in the multiple single-objective functions approach because the objectives are independently defined. However, since objectives have to be simple to identify the Pareto front, similar functions are used to all objectives. In DMPs, all objective functions are the Euclidean distance to the given points in the objective space. Thus the difficulty of each objective is the same. As shown in Table I, all test problems based on the multiple single-objective functions approach have this property: All objectives have the same difficulty.

In the bottom-up approach, the difficulties in converging to the Pareto front are controlled by the distance function. DTLZ and WFG use a single distance function for all objectives. If the distance function is a multimodal function, all objective are multimodal. When a solution is local optimal for one objective, it is also local optimal for all the other objectives. In DTLZ7, a distance function is used only for the last objective. That is, only the last objective is difficult and all the other objectives are easy in DTLZ7. P and CEC16 use different distance functions for different objectives but all objectives are unimodal with respect to the distance variables.

All of the above-mentioned properties are summarized for each of the the existing scalable multiobjective test problems in Table I. In this table, DTLZ5-6 [8], [9], [27], DTLZ5(*I*, *M*) [28], WFG3 [29], and DEB3DK [30] are not included because their Pareto fronts are not known due to some unintentional problems in their formulations (e.g., whereas WFG3 was formulated as a degenerated test problem [9], its Pareto front is not degenerated [29]). DTLZ8-9 [8] are not included either because they are constrained problems. The non-scalable problems CEC16\_1-21 [20] are also excluded. DMPs, SPH-*m*, and Pareto-Box are based on the multiple single-objective functions approach. The others are based on the bottom-up approach. Table 1 clearly shows that almost all scalable test problems commonly have the same property.

Real-world problems may have one or two properties in Table 1. However, it is not likely that they have almost all properties. This is because the properties are very special. This may means that the existing scalable test problems can cover only a small part of real-world multiobjective problems. They are not likely to be enough to evaluate EMO algorithms. In the next section, we propose a new framework to create scalable test problems without special features in Table I based on the bottom-up approach.

#### IV. PROPOSED TEST PROBLEM FRAMEWORK

This section proposes a new framework to create scalable multiobjective test problems based on the bottom-up approach. The Pareto front and the feasible region in the objective space of a test problem is separately designed. For visual explanation, two-objective and three-objective problems are mainly shown. However, extension to many-objective problems is trivial.

##### A. Pareto Front Design

As explained in Section III, most existing test problems have multiple optimal solutions for each objective. Moreover, any  $k$  ( $k < M$ ) objectives can be optimized simultaneously. In our framework, the location of the optimal solution for each objective can be directly specified in the objective space. Let us assume that we have six points  $V_1, V_2, \dots, V_6$  in the objective space. They are on the single plane in the objective space. A point  $c$  on the Pareto front is obtained by weighting those points by (9) and (10). The weight vector  $w$  is obtained by the position variables  $x_i$  by (11) where  $n_p$  is the number of points  $V_i$ , which is also the number of the position variables.

$$c = w_1 V_1 + w_2 V_2 + \dots + w_6 V_6, \quad (9)$$

$$0 \leq w_i \leq 1, \quad \sum_{i=1}^{n_p} w_i = 1, \quad (10)$$

$$w_i = x_i / \sum_{j=1}^{n_p} x_j, \quad 0 \leq x_i \leq 1. \quad (11)$$

When all position variables become zero, the solution is infeasible because the weights cannot be defined by (11). The objective values of this infeasible solution are replaced with a large value. However, it is not likely that multiple variables have exactly zero at the same time in numerical experiments. So we can ignore this infeasible solution.

Using an  $M \times n_p$  matrix  $F$  whose  $i$ -th column is  $V_i$ , (9) is rewritten as (12). The Pareto front can be represented by this  $M \times n_p$  matrix  $F$ .

$$c = (V_1 \ V_2 \ \dots \ V_6) w = Fw. \quad (12)$$

The Pareto front is the convex hull (a polygon) of the given points. In Fig. 4,  $V_1, V_2$ , and  $V_3$  are the single optimal solutions for the first, second, and third objectives.  $V_4, V_5$ , and  $V_6$  are the worst solutions in the Pareto front for the these three objectives. Fig. 5 shows some other Pareto fronts generated by this method. The corresponding matrices are also shown. The Pareto front in Fig. 5 (a) is the same as that of DTLZ1. In Fig. 5 (b), the first and second objectives can be optimized simultaneously by  $V_1$ . However, no other pair of objectives can be simultaneously optimized. Fig. 5 (c) has multiple optimal solutions for each objective. However, no pair can be simultaneously optimized.

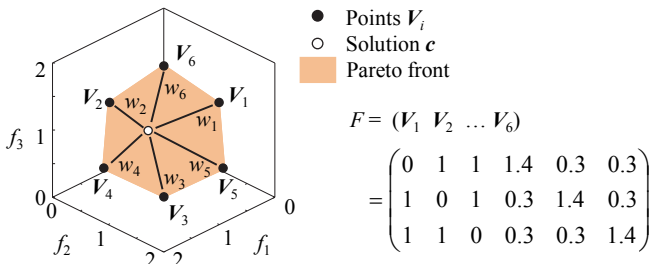


Fig. 4. Calculation of the Pareto front.

$$(a) F = \begin{pmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{pmatrix} \quad (b) F = \begin{pmatrix} 0 & 1 & 1.4 & 0.3 \\ 0 & 1 & 0.3 & 1.4 \\ 2 & 0 & 0.3 & 0.3 \end{pmatrix}$$

$$(c) F = \begin{pmatrix} 0 & 0 & 0.8 & 1.2 & 0.8 & 1.2 & 1.4 & 0.3 & 0.3 \\ 0.8 & 1.2 & 0 & 0 & 1.2 & 0.8 & 0.3 & 1.4 & 0.3 \\ 1.2 & 0.8 & 1.2 & 0.8 & 0 & 0 & 0.3 & 0.3 & 1.4 \end{pmatrix}$$

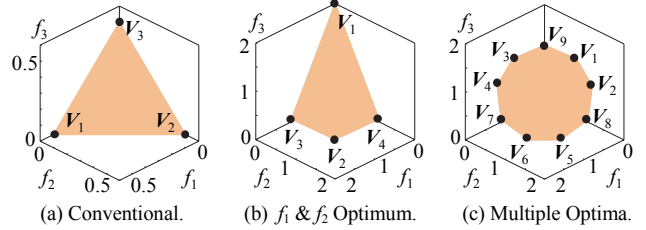


Fig. 5. Various shapes of Pareto fronts by the proposed method.

Next, we change the geometry (i.e., curvature or convexity) of the Pareto front to create a convex/concave Pareto front. First, the Pareto front is normalized into  $[0, 1]^M$  by using the ideal and nadir points of the front. A solution  $c$  on the normalized Pareto front is transformed by a geometry function  $G(c)$  and then de-normalized into the original scale. The geometry function is applied to all objectives. The geometry function should satisfy  $G(0) = 0$ ,  $G(1) = 1$ ,  $0 \leq G(x) \leq 1$  to preserve the ideal and nadir points. If the geometry function is monotonically increasing, the dominance relations among solutions do not change before and after the transformation. If the geometry function is convex (or concave), the resulting Pareto front is also convex (or concave). In this paper, we use the following geometry functions. The linear geometry function does not change the Pareto front. Convex and concave geometry functions make the Pareto front convex and concave, respectively.

$$\text{Linear: } G(c) = c$$

$$\text{Convex: } G(c) = 1 - \sqrt{1-c} \quad (13)$$

$$\text{Concave: } G(c) = \sqrt{c}$$

When the concave function is used for all objectives and all vertices of the Pareto front  $V_1, \dots, V_{n_p}$  satisfy  $\sum_{m=1}^M V_{im} = 1$ , the generated Pareto optimal solutions  $p$  satisfies  $\sum_{m=1}^M p_m^2 = 1$ . The term  $p_i^2$  comes from the inverse function of the concave transformation function. If the front matrix  $F$  is the identity matrix, this Pareto front is the same as the Pareto front of DTLZ2: the Pareto front is a positive part of the sphere surface with radius 1 which is located at the origin. We show some examples of the generated Pareto fronts in Fig. 6.

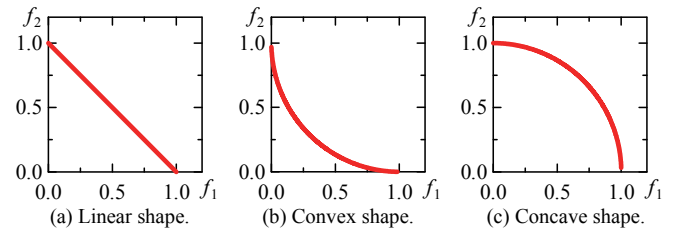


Fig. 6. Pareto fronts and geometry functions. ( $V_1 = (1 \ 0)^T$ ,  $V_2 = (0 \ 1)^T$ )

## B. Feasible Region Design in the Objective Space

Next, we consider the feasible region in the objective space. The existing test problems do not pay much attention to its shape. However, our framework explicitly specifies it. The proposed Pareto front design scheme can specify not only the shape of the Pareto front but also its location. We use multiple fronts to form the outline of the feasible region in the objective space. The feasible region is generated by connecting adjacent fronts by the B-Spline interpolation as illustrated in Fig. 7.

The B-Spline [31] is a widely used technique to generate free-form curves. In this paper, we use the two-degree B-Spline with an open uniform knot vector. Let  $F_k(\mathbf{x})$  be the  $k$ -th front function, which consists of points and shape transformation in the previous sub-section. We assume that the index  $k$  of the fronts starts from 0 (i.e.,  $k = 0, 1, \dots, n_F - 1$ ), where  $n_F$  is the number of the fronts. The  $n_F$  fronts are connected as follows:

$$\begin{aligned} f(\mathbf{x}, \mathbf{y}) &= b_0(t)F_k(\mathbf{x}) + b_1(t)F_{k+1}(\mathbf{x}) + b_2(t)F_{k+2}(\mathbf{x}), \\ b_0(t) &= \begin{cases} (1-t)^2 & k=0 \\ (1-t)^2/2 & \text{otherwise} \end{cases}, \\ b_1(t) &= 1 - b_0(t) - b_2(t), \\ b_2(t) &= \begin{cases} t^2 & k=n_F-2 \\ t^2/2 & \text{otherwise} \end{cases}, \\ k &= \begin{cases} n_F-3 & d=n_F-2 \\ \lfloor d \rfloor & \text{otherwise} \end{cases}, t = d - k, \text{ and } 0 \leq d \leq n_F - 2. \end{aligned} \quad (14)$$

$\lfloor d \rfloor$  is the floor function, which returns the integer part of  $d$ .  $b_0(t)$ ,  $b_1(t)$ , and  $b_2(t)$  are B-Spline basis functions, which determine the weights for each front. Because the B-Spline generates a curve, one parameter  $d$  is required to determine a location on the B-Spline curve. We use the value of a distance function  $g(\mathbf{y})$  as the parameter:  $d = g(\mathbf{y})$ . When  $g(\mathbf{y})$  is small, the solution  $f(\mathbf{x}, \mathbf{y})$  is generated near the first front  $F_0(\mathbf{x})$ .

The feasible region in Fig. 7 is obtained by specifying the front matrices and a one-variable distance function as follows:

$$F_0 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}, F_1 = \begin{pmatrix} 6 & 5 \\ 2 & 3 \end{pmatrix}, F_2 = \begin{pmatrix} 10 & 2 \\ 2 & 10 \end{pmatrix}.$$

Geometry functions: Convex ( $F_1$ ), linear ( $F_2$ ), concave ( $F_3$ ).  
 $g(\mathbf{y}) = (n_F - 2) \times \mathbf{y}$ .

If multiple fronts are arbitrarily specified, the Pareto front of the test problem become unclear. We always design the first front  $F_0$  as the Pareto front. This can be done by specifying multiple fronts in the following manner: The nadir point of the first front dominates the ideal points of the other fronts.

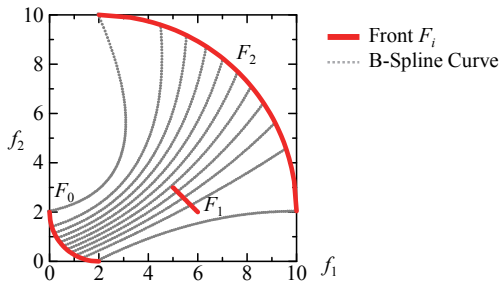


Fig. 7. B-Spline interpolation for fronts.

## C. Distance Function Design

We explain how to specify a distance function. The distance function defines the optimal values of the decision variables and the characteristics of the objective functions.

### 1) Dependency on the Position Variables

As explained in Section II.B, a solution  $\mathbf{z}$  is a Pareto optimal solution if the distance function  $g(\mathbf{y})$  equals zero in the bottom-up approach. This also applies to our test problem. If distance function does not depend on the position variables, the resulting Pareto optimal solution set becomes very simple as explained in Section III.C. To avoid this issue, we assume that the distance function  $g(\mathbf{y})$  becomes zero only when the sum of the position variables is one. Note that the position variables satisfying this condition can generate the entire Pareto front. That is, the use of this condition does not change the shape of the Pareto front of the designed problem. We also assume that the optimal values of the distance variables depend on position variables. Now, the set of the optimal solutions  $X^*$  is written as (15). All decision variables depend on other variables.

$$X^* = \{\mathbf{z} = (\mathbf{x}, \mathbf{y}) \mid 0 \leq x_i, \sum_{i=1}^{n_P} x_i = 1, g(\mathbf{y}; \mathbf{x}) = 0\}. \quad (15)$$

### 2) Use of Multiple Distance Functions

We use different distance functions for different objectives. The B-Spline interpolation is independently calculated for each objective. That is, the B-Spline parameter  $d$  of the  $m$ -th objective is specified by the  $m$ -th distance function  $g_m(\mathbf{y}; \mathbf{x})$ . To keep the Pareto front unchanged, all distance functions must have the same optimal solution:  $g_1(\mathbf{y}^*; \mathbf{x}) = g_2(\mathbf{y}^*; \mathbf{x}) = \dots = 0$ . The shape of the feasible region is understood by the fronts and the correlations of the distance functions as shown in Fig. 8. When the distance functions are similar, the feasible region is outlined by the fronts. When the distance functions are independent (e.g., different variables are used for different distance functions), the feasible region is close to a rectangular shape. The advantage of multiple distance functions is that we can specify a different property for each objective.

We generate multiple distance functions from one base function by applying the following parameterized modification (see the examples in the next section). The base function  $f_B(z)$  defined on  $[z_{\min}, z_{\max}]$  is assumed to have its minimum value 0 at  $z = 0$ . By changing the parameter  $a$ , we can generate a number of different functions.

$$L_{f_B}(z; a) = \begin{cases} \frac{\log(f_B(z) + \varepsilon) - \log(\varepsilon)}{\log(f_1^{\text{Max}} + \varepsilon) - \log(\varepsilon)} & (z \leq 0) \\ \frac{\log(af_B(z) + \varepsilon) - \log(\varepsilon)}{\log(af_2^{\text{Max}} + \varepsilon) - \log(\varepsilon)} & (0 \leq z) \end{cases} \quad (16)$$

$$f_1^{\text{Max}} = \max_{z_{\min} \leq z \leq 0} f_B(z), f_2^{\text{Max}} = \max_{0 \leq z \leq z_{\max}} f_B(z), \varepsilon = 0.01. \quad (17)$$

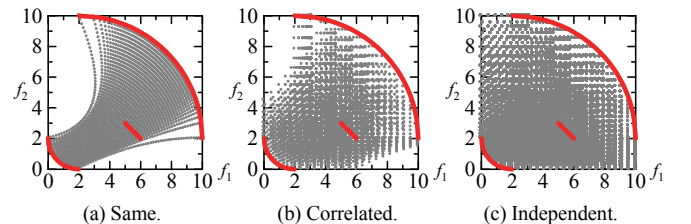


Fig. 8. Correlation of the distance functions and the objective space.

## V. EXAMPLES OF TEST PROBLEM INSTANCES

This section shows some examples of test problems created by the proposed framework. First, the overall formulation of our test problems is summarized below.

$$f_m(\mathbf{x}, \mathbf{y}) = b_0(t_m)F_{k_m,m}(\mathbf{x}) + b_1(t_m)F_{k_m+1,m}(\mathbf{x}) + b_2(t_m)F_{k_m+2,m}(\mathbf{x})$$

for  $m = 1, 2, \dots, M$ .

where

$b_i(t_m)$ : defined in (14),

$F_{k,m}(\mathbf{x}) = \text{norm}^{-1}(S_{k,m}(\text{norm}([F_k \mathbf{w}]_m)))$ ,

$\mathbf{w}$ : calculated from  $\mathbf{x}$  as in (12),

$[\mathbf{v}]_m$ : the  $m$ -th element of a vector  $\mathbf{v}$ ,

$k_m$  and  $t_m$ : calculated with  $d = g_m(\mathbf{y}, \mathbf{x})$ ,

$\mathbf{z} = (\mathbf{x}, \mathbf{y})$  is the decision variables.

$\mathbf{x}$  and  $\mathbf{y}$  are the position and distance variables, respectively.

$$0 \leq x_i \leq 1 \quad (i = 1, \dots, n_p) \quad \text{and} \quad 0 \leq y_j \leq 1 \quad (j = 1, \dots, n_D).$$

$n_p$ : The number of position variables, which is the same as the number of points to define one front.

$n_D$ : The number of distance variables.

The main parameters necessary to create a problem instance are the number of fronts  $n_F$ , the geometry functions  $G_{k,m}(\cdot)$ ,  $M \times n_p$  front matrices  $F_{k,m}$  and distance functions  $g_m(\mathbf{y}, \mathbf{x})$ . The following notations are used to describe the instances.

$E_M$ :  $M \times M$  identity matrix.

$\mathbf{1}_M$ :  $M \times M$  matrix whose elements are all one.

$\|\mathbf{x}\|_1$ : Sum of the position variables.

### [Instance I]

**Front Matrices:** ( $n_F = 3, n_p = 2M$ )

$$F_0 = \frac{1}{M-1} \begin{pmatrix} 0 & 1 & 1 & 1 & N & r & r & r \\ 1 & 0 & 1 & 1 & r & N & r & r \\ 1 & 1 & 0 & 1 & r & r & N & r \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \\ 1 & 1 & 1 & 0 & r & r & r & N \end{pmatrix}$$

$$N = 0.6(M-1), r = (M-1-N)/(M-1)$$

(for two-objective problems,  $F_0 = 0.6(\mathbf{1}_2 - E_2)$  is used.)

$$F_1 = 2F_0 + \mathbf{1}_M, \quad F_2 = F_0 + 4 \cdot \mathbf{1}_M$$

**Geometries:** Convex for all objectives of  $F_0, F_1$  and  $F_2$ .

**Distance function:**

$$g_m(\mathbf{z}) = \frac{n_F - 2}{n_D + 1} \left[ L_{f_b} \left( \frac{\|\mathbf{x}\|_1 - 1}{n_p}; 1 \right) + \sum_{i=1}^{n_D} L_{f_b}(y_i - y^*; ((i+m) \bmod n_D) + 1) \right]$$

$f_B(x) = x^2$ .  $f_1^{\text{Max}}$  and  $f_2^{\text{Max}}$  are  $f_B(-1/n_p)$  and  $f_B(1-1/n_p)$  for the first term. For the other terms,  $f_1^{\text{Max}} = f_B(-y^*)$ ,  $f_2^{\text{Max}} = f_B(1-y^*)$ .

$$y^* = \frac{\sum_{i=1}^M x_i + \sum_{i=M+1}^{2M} (1-x_i)}{4M} + 0.1$$

**Characteristics:**

- Two or more objectives cannot be optimized at once.
- Each objective function has only one optimal solution.
- All objective functions are similar and globally correlated.
- The Pareto front is convex as shown in Fig. 9.

Fig. 9 shows the three-objective Pareto front of this problem. The obtained 100 solutions by NSGA-II are also shown.

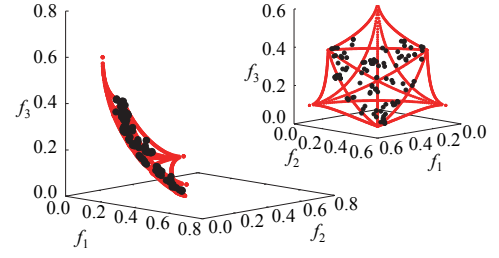


Fig. 9. Pareto front of the problem instance I with three objectives.

### [Instance II]

**Front Matrices:**

The same as the previous three front matrices but the weights  $\mathbf{w}$  is calculated by using  $\mathbf{x}'$  instead of the original position variables  $\mathbf{x}$ . This modification makes the first half objectives relatively easier to optimize.

$$x'_i = \begin{cases} x_i^{0.5} & i \leq M/2 \\ x_i^2 & \text{otherwise} \end{cases}$$

**Geometries:** Concave for all objectives of  $F_0, F_1$  and  $F_2$ .

**Distance function:**

$$g_m(\mathbf{z}) = \frac{n_F - 2}{n_D + 1} \left[ L_{f_b} \left( \frac{\|\mathbf{x}'\|_1 - 1}{n_p}; 1 \right) + \sum_{i=1}^{n_D} L_{f_b}(y_i - y^*; ((i+m) \bmod n_D) + 1) \right]$$

$$f_B(y) = \begin{cases} y^2 & m \leq M/2 \\ 1 + y^2 - \cos((20 + m/2)\pi x) & \text{otherwise} \end{cases}$$

For the first half objectives,  $f_1^{\text{Max}}$  and  $f_2^{\text{Max}}$  are the same as the previous ones. For the other objectives, the constant value 2 is added to  $f_1^{\text{Max}}$  and  $f_2^{\text{Max}}$  which are used to define the first half objectives.  $y^*$  is the same as the previous one.

**Characteristics:**

- Two or more objectives cannot be optimized at once.
- Each objective function has only one optimal solution.
- Objective functions have different difficulties. The first half objective functions are unimodal functions. The other objective functions are multimodal functions.
- Each multimodal function has local optima at different positions.
- The Pareto front is concave as shown in Fig. 10.

Fig. 10 shows the three-objective Pareto front of this problem. The obtained solutions by NSGA-II are also shown. From Fig. 10, we can see that NSGA-II only optimize the easy objective function (i.e., the first objective function).

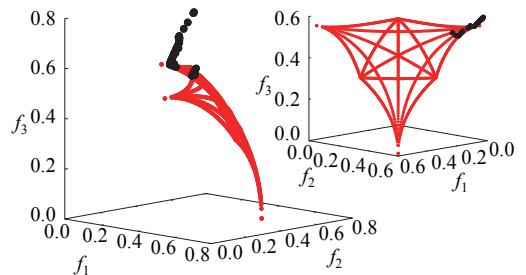


Fig. 10. Pareto front of the problem instance II with three objectives.

## VI. CONCLUSIONS

In this paper, we reviewed existing scalable multiobjective continuous test problems. They were classified into the two groups: The multiple single-objective functions approach and the bottom-up approach. It was shown that the existing test problems are not enough to evaluate EMO algorithms because they have somewhat strange or unusual properties in Table I.

We proposed a new framework for scalable multiobjective test problems. The shapes of the Pareto front and the feasible region can be directly specified by a number of points in an objective space. Using the framework, two test problems were shown. The shapes of their Pareto fronts are different from those of existing scalable test problems. One instance has objectives with different difficulties: The first objective is unimodal and the others are multimodal.

An important future research topic is to evaluate the quality of test problems in a quantitative way. This is a big challenge because the analysis of a lot of real-world problems is needed to decide how to evaluate the quality of test problems.

## REFERENCES

- [1] V. Khare, X. Yao, and K. Deb, "Performance scaling of multiobjective evolutionary algorithms," in *Evolutionary Multi-Criterion Optimization*, vol. 2632, pp. 376-390, April 2003.
- [2] E. J. Hughes, "Evolutionary many-objective optimization: Many once or one many?," *Proc. of 2005 IEEE Congress on Evolutionary Computation*, vol. 1, pp. 222-227, September 2005.
- [3] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," *Proc. of 2008 IEEE Congress on Evolutionary Computation*, pp. 2424-2431, June 1-6, 2008.
- [4] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173-195, 2000.
- [5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002.
- [6] H. Ishibuchi, N. Akedo, and Y. Nojima, "Behavior of multi-objective evolutionary algorithms on many-objective knapsack problems," *IEEE Trans. on Evolutionary Computation*, vol. 19, no. 2, pp. 264-283, 2015.
- [7] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Review of coevolutionary developments of evolutionary multi-objective and many-objective algorithms and test problems," *Proc. of 2014 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*, pp. 178-185, December 9-12, 2014.
- [8] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in A. Abraham, L. Jain, and R. Goldberg (Eds.), *Evolutionary Multiobjective Optimization*, pp. 105-145, Springer-Verlag, London, 2005.
- [9] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. on Evolutionary Computation*, vol. 10, no. 5, pp. 477-506, 2006.
- [10] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Trans. on Evolutionary Computation*, vol. 17, no. 5, pp. 721-736, 2013.
- [11] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. on Evolutionary Computation*, vol. 18, no. 4, pp. 577-601, 2014.
- [12] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "Generalized decomposition and cross entropy methods for many-objective optimization," *Information Sciences*, vol. 282, pp. 363-387, 2014.
- [13] H. Seada and K. Deb, "U-NSGA-III: A unified evolutionary algorithm for single, multiple, and many-objective optimization," *Evolutionary Multi-Criterion Optimization*, vol. 9019, pp. 34-49, March 2015.
- [14] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," Computer Engineering and Networks Laboratory (TIK), Department of Electrical Engineering, Swiss Federal Institute of Technology, *TIK-Report 103*, May 2001.
- [15] M. Köppen and K. Yoshida, "Substitute distance assignments in NSGA-II for handling many-objective optimization problems," *Evolutionary Multi-Criterion Optimization*, vol. 4403, pp. 727-741, 2007.
- [16] H. K. Singh, A. Isaacs, T. Ray, and W. Smith, "A study on the performance of substitute distance based approaches for evolutionary many objective optimization," *Simulated Evolution and Learning*, vol. 5361, pp. 401-410, 2008.
- [17] H. Ishibuchi, Y. Hitotsuyanagi, N. Tsukamoto, and Y. Nojima, "Many-objective test problems to visually examine the behavior of multiobjective evolution in a decision space," *Proc. of 2010 International Conference on Parallel Problem Solving from Nature*, Part II, pp. 91-100, Krakow, Poland, September 11-15, 2010.
- [18] M. Köppen, R. Vicente-Garcia, and B. Nickolay, "The Pareto-Box problem for the modelling of evolutionary multiobjective optimization algorithms," *Adaptive and Natural Computing Algorithms*, pp. 194-197, 2005.
- [19] D. K. Saxena, Q. Zhang, J. A. Duro, and A. Tiwari, "Framework for many-objective test problems with both simple and complicated Pareto-set shapes," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6576, no. 1, pp. 197-211, 2011.
- [20] H. Li, Q. Zhang, P. N. Suganthan, A. Zhou, K. Deb, H. Ishibuchi, and C. A. C. Coello, "The benchmark multiobjective test problems for CEC2016 competition," Technical Report, December 23, 2015.
- [21] D. Brockhoff and E. Zitzler, "Objective reduction in evolutionary multiobjective optimization: Theory and applications," *Evolutionary Computation*, vol. 17, no. 2, pp. 135-166, 2009.
- [22] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. on Evolutionary Computation*, vol. 18, no. 4, pp. 602-622, 2014.
- [23] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation-based evolutionary algorithm for many-objective optimization," *IEEE Trans. on Evolutionary Computation*, vol. 20, no. 1, pp. 16-37, 2016.
- [24] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. on Evolutionary Computation*, vol. 13, no. 2, pp. 284-302, 2009.
- [25] T.-C. Chiang and Y.-P. Lai, "MOEA/D-AMS: Improving MOEA/D by an adaptive mating selection mechanism," *Proc. of 2011 IEEE Congress on Evolutionary Computation*, pp. 1473-1480, 2011.
- [26] A. Zhou and Q. Zhang, "Are all the subproblems equally important? Resource allocation in decomposition-based multiobjective evolutionary algorithms," *IEEE Trans. on Evolutionary Computation*, vol. 20, no. 1, 2016.
- [27] K. Deb and D. K. Saxena, "Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems," *Proc. of 2006 IEEE Congress on Evolutionary Computation*, pp. 3353-3360, July 16-21, 2006.
- [28] D. K. Saxena, A. Duro, A. Tiwari, K. Deb, and Q. Zhang, "Objective reduction in many-objective optimization: Linear and nonlinear algorithms," *IEEE Trans. on Evolutionary Computation*, vol. 17, no. 1, pp. 77-99, 2013.
- [29] H. Ishibuchi, H. Masuda, and Y. Nojima, "Pareto fronts of many-objective degenerate test problems," *IEEE Trans. on Evolutionary Computation* (accepted).
- [30] J. Branke, K. Deb, H. Dierolf, and M. Osswald, "Finding knees in multi-objective optimization," *Parallel Problem Solving from Nature - PPSN VIII*, vol. 3242, pp. 722-731, 2004.
- [31] R. H. Bartels, B. J. C. Beatty, and B. A. Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann, 1987.