

Performance of Decomposition-Based Many-Objective Algorithms Strongly Depends on Pareto Front Shapes

Hisao Ishibuchi, *Fellow, IEEE*, Yu Setoguchi, Hiroyuki Masuda, and Yusuke Nojima, *Member, IEEE*

Abstract—Recently, a number of high performance many-objective evolutionary algorithms with systematically generated weight vectors have been proposed in the literature. Those algorithms often show surprisingly good performance on widely used DTLZ and WFG test problems. The performance of those algorithms has continued to be improved. The aim of this paper is to show our concern that such a performance improvement race may lead to the overspecialization of developed algorithms for the frequently used many-objective test problems. In this paper, we first explain the DTLZ and WFG test problems. Next, we explain many-objective evolutionary algorithms characterized by the use of systematically generated weight vectors. Then we discuss the relation between the features of the test problems and the search mechanisms of weight vector-based algorithms such as multiobjective evolutionary algorithm based on decomposition (MOEA/D), nondominated sorting genetic algorithm III (NSGA-III), MOEA/dominance and decomposition (MOEA/DD), and θ -dominance based evolutionary algorithm (θ -DEA). Through computational experiments, we demonstrate that a slight change in the problem formulations of DTLZ and WFG deteriorates the performance of those algorithms. After explaining the reason for the performance deterioration, we discuss the necessity of more general test problems and more flexible algorithms.

Index Terms—Decomposition-based evolutionary algorithms, many-objective evolutionary algorithms, many-objective optimization, many-objective test problems.

I. INTRODUCTION

RECENTLY the development of evolutionary multiobjective optimization (EMO) algorithms was discussed from the point of view of co-evolution with test problems in [1] where the Pareto front was compared with randomly generated initial solutions to explain the characteristic features of each test problem. For example, Fig. 1 shows a test problem in [2] used for performance evaluations of EMO algorithms in the mid-1990s. As we can see from Fig. 1, some initial solutions are very close to the Pareto front. This observation may explain why nonelitist EMO algorithms such as nondominated sorting

Manuscript received December 17, 2015; revised May 6, 2016; accepted June 20, 2016. Date of publication July 12, 2016; date of current version March 28, 2017. This work was supported by the Japan Society for the Promotion of Science under Grant KAKENHI 24300090, Grant KAKENHI 26540128, and Grant KAKENHI 16H02877.

The authors are with the Department of Computer Science and Intelligent Systems, Osaka Prefecture University, Sakai 599-8531, Japan (e-mail: hisao@cs.osakafu-u.ac.jp; yu.setoguchi@ci.cs.osakafu-u.ac.jp; hiroyuki.masuda@ci.cs.osakafu-u.ac.jp; nojima@cs.osakafu-u.ac.jp).

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.2016.2587749

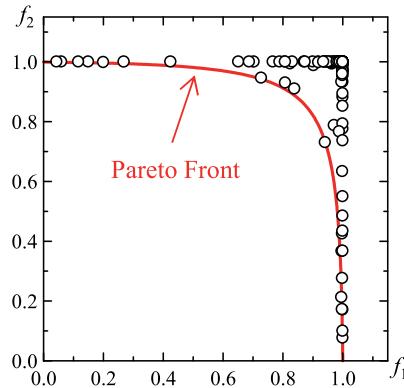


Fig. 1. Randomly generated 200 solutions and the Pareto front of a test problem in [2].

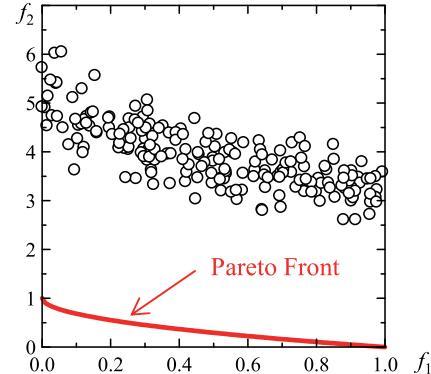


Fig. 2. Randomly generated 200 solutions and the Pareto front of a two-objective ZDT1 problem [5].

genetic algorithm (NSGA) [3] and Niched Pareto genetic algorithm (NPGA) [4] with no strong convergence property were proposed in the mid-1990s. However, initial solutions of a test problem in Fig. 2 [5] are not close to the Pareto front. Thus we need strong convergence whereas no strong diversification is needed in Fig. 2 (since randomly generated initial solutions have a large diversity). These observations explain why elitist EMO algorithms such as strength Pareto evolutionary algorithm (SPEA) [6], NSGA-II [7], and SPEA2 [8] were proposed around 2000. In these algorithms, Pareto dominance was used as the main fitness evaluation criterion together with a secondary criterion for diversity maintenance.

In parallel with the increase in the popularity of the Pareto dominance-based EMO algorithms [6]–[8], many-objective

test problems called DTLZ [9] and WFG [10] were proposed as scalable test problems where the number of objectives can be arbitrarily specified. Multiobjective knapsack problems [6] were also generalized to many-objective test problems with up to 25 objectives [11], [12]. The DTLZ, WFG, and knapsack problems were repeatedly used for demonstrating difficulties of many-objective optimization for the Pareto dominance-based EMO algorithms [13]–[16]. When an EMO algorithm is applied to a many-objective problem, almost all solutions in a population become nondominated with each other in very early generations (e.g., within ten generations) before they converge to the Pareto front. This means that the Pareto dominance-based selection pressure toward the Pareto front becomes very weak. As a result, convergence ability of Pareto dominance-based EMO algorithms is severely degraded by the increase in the number of objectives.

For improving the convergence ability for many-objective problems, various approaches have been proposed such as the modification of the Pareto dominance relation [17] and the introduction of an additional ranking mechanism [18]–[21]. The use of a different fitness evaluation mechanism was also actively studied. Approaches in this direction can be classified into two categories. One is an indicator-based approach such as S metric selection evolutionary multiobjective optimization algorithm (SMS-EMOA) [22] and hypervolume estimation algorithm for multiobjective optimization (HypE) [23]. The advantage of EMO algorithms in this category is a sound theoretical support, which is based on a direct relation between many-objective optimization and indicator optimization. Their main difficulty is heavy computation load for fitness evaluation. The other category is characterized by the use of a scalarizing function for fitness evaluation such as multiple single objective Pareto sampling (MSOPS) [24] and multiobjective evolutionary algorithm based on decomposition (MOEA/D) [25]. These algorithms are referred to as decomposition-based or scalarizing function-based algorithms.

Whereas MOEA/D [25] was not originally proposed for many-objective problems, its high performance as a many-objective optimizer was observed [16], [26]. Recently, a number of new many-objective algorithms have been proposed using the framework of MOEA/D (e.g., improved decomposition-based evolutionary algorithm (I-DBEA) [27], MOEA/D-distance-based updating strategy (MOEA/D-DU) [28], Ensemble Fitness Ranking with a Ranking Restriction scheme (EFR-RR) [28], and θ -DEA [29]). Some other algorithms (e.g., NSGA-III [30], MOEA/DD [31], and U-NSGA-III [32]) can be viewed as using hybrid mechanisms of the Pareto dominance-based fitness evaluation and the MOEA/D framework.

MOEA/D [25] searches for well-distributed solutions using systematically generated weight vectors. As an example, we show a set of 91 weight vectors for a three-objective problem in Fig. 3. In Fig. 4, we show an example of obtained solutions by MOEA/D-PBI with $\theta = 5$ [25] for a three-objective DTLZ2 problem [9]. Well-distributed solutions were obtained in Fig. 4 using the uniformly distributed weight vectors in Fig. 3.

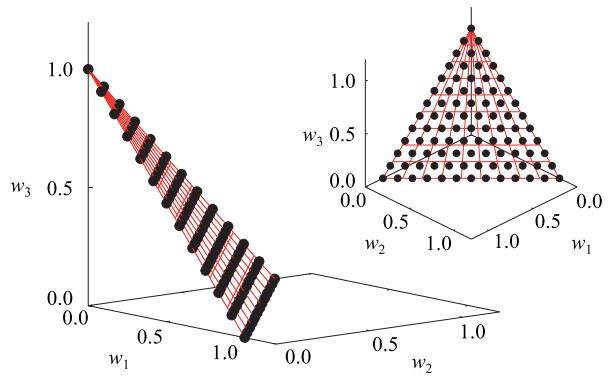


Fig. 3. Example of weight vectors used in MOEA/D (91 weight vectors).

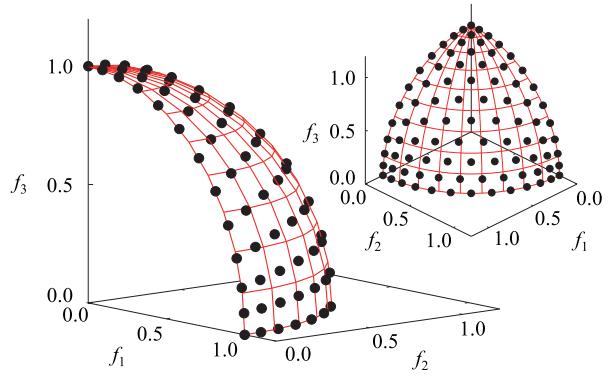


Fig. 4. Example of obtained nondominated solutions by MOEA/D-PBI for a three-objective DTLZ2 problem using the 91 weight vectors in Fig. 3.

In Figs. 3 and 4, a one-to-one mapping was realized between the weight vectors in Fig. 3 and the obtained solutions in Fig. 4. However, this is not always the case. As examined in [33], the number of obtained nondominated solutions by MOEA/D is often much smaller than the number of weight vectors. This is because: 1) a single good solution can be shared by multiple weight vectors and 2) all solutions are not always nondominated. Recently proposed many-objective algorithms [27]–[32] with the MOEA/D framework have mechanisms for improving both the convergence of solutions toward the Pareto front and their uniformity over the entire Pareto front. Their common feature is the use of systematically generated weight vectors. In those algorithms, reference points and/or reference lines are constructed using the weight vectors.

Surprisingly good results were reported by those weight vector-based algorithms on the DTLZ and WFG problems in the literature. For example, the average inverted generational distance (IGD) over 20 runs on a 15-objective DTLZ2 problem was reported in [29] as 1.726×10^{-2} by NSGA-III, 1.133×10^{-2} by θ -DEA, and 6.005×10^{-3} by MOEA/D-PBI. In Fig. 5, we show the reference points used for the IGD calculation in [29] and a set of obtained solutions by a single run of MOEA/D-PBI with $\theta = 5$ (IGD is 5.590×10^{-3} in Fig. 5). The obtained solution set in Fig. 5(b) is almost the same as the reference point set in Fig. 5(a).

Whereas difficulties of many-objective problems were repeatedly pointed out (e.g., see survey papers [12], [34], and [35], Fig. 5 and the reported results

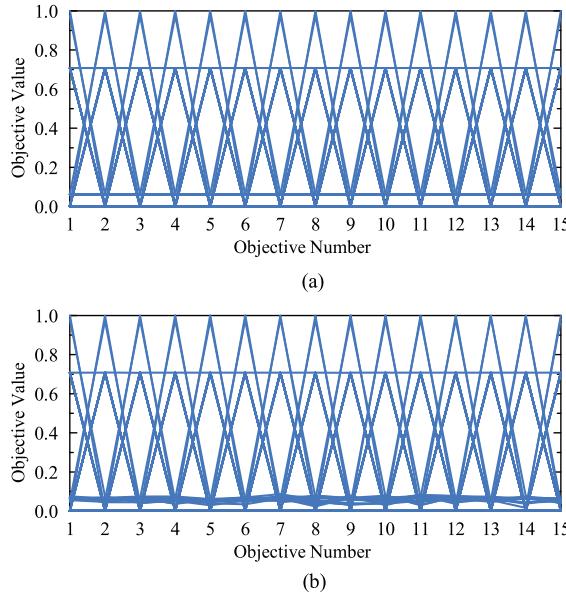


Fig. 5. Reference points for the IGD calculation and a set of obtained solutions by MOEA/D-PBI with $\theta = 5$ for a 15-objective DTLZ2 problem. (a) Reference points. (b) Obtained solutions by a single run of MOEA/D-PBI.

in [27]–[32] may suggest that some DTLZ and WFG problems are not difficult. In this paper, we first examine why surprisingly good results were obtained for some DTLZ and WFG test problems. Then we show that a slight change in the DTLZ and WFG formulations degrades the performance of weight vector-based many-objective algorithms. Our experimental results suggest the overspecialization of those algorithms for the test problems.

This paper is organized as follows. In Section II, we briefly explain the DTLZ [9] and WFG [10] problems. We focus on the shape of the Pareto front of each test problem. In Section III, we explain a common search mechanism of weight vector-based algorithms such as MOEA/D-PBI [25], θ -DEA [29], and NSGA-III [30]. We also discuss the relation between the shape of the Pareto front of each test problem and the common search mechanism of those algorithms. In Section IV, we demonstrate their high performance on the DTLZ and WFG problems. We also demonstrate severe performance deterioration by a slight change in the problem formulations of DTLZ and WFG: all objectives are multiplied by (-1) . In Section V, we explain why the performance of the weight vector-based algorithms is sensitive to such a slight change. We also discuss the necessity of more general test problems and more flexible algorithms. Finally, we conclude this paper in Section VI.

II. MANY-OBJECTIVE TEST PROBLEMS

In general, an M -objective minimization problem with a decision vector \mathbf{x} and its feasible region X is written as

$$\text{Minimize } f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x}) \text{ subject to } \mathbf{x} \in X \quad (1)$$

where $f_i(\mathbf{x})$ is the i th objective to be minimized ($i = 1, 2, \dots, M$). In the EMO community, multiobjective problems with four or more objectives (i.e., $M \geq 4$) are often referred to as many-objective problems [12], [34], [35].

TABLE I
MANY-OBJECTIVE TEST PROBLEMS USED IN EACH STUDY ON WEIGHT VECTOR-BASED MANY-OBJECTIVE EVOLUTIONARY ALGORITHMS

Ref.	Publication Year	Proposed Algorithm	Test Problems	Number of Objectives
[25]	2007	MOEA/D	Knapsack ZDT1-4, 6 DTLZ1-2	2, 3, 4 2 3
[27]	2015	I-DBEA	DTLZ1-4 DTLZ5(I, M) WFG1-9	3, 5, 8, 10, 15 3, 5, 8, 10, 15 3, 5, 10, 15
[28]	2016	MOEA/D-DU EFR-RR	DTLZ1-4, 7 WFG1-9 S-DTLZ1-2	2, 5, 8, 10, 13 2, 5, 8, 10, 13 2, 5, 8, 10, 13
[29]	2016	θ -DEA	DTLZ1-4, 7 S-DTLZ1-2 WFG1-9	3, 5, 8, 10, 15 3, 5, 8, 10, 15 3, 5, 8, 10, 15
[30]	2014	NSGA-III	DTLZ1-4 WFG6-7 S-DTLZ1-2	3, 5, 8, 10, 15 3, 5, 8, 10, 15 3, 5, 8, 10, 15
[31]	2015	MOEA/DD	DTLZ1-4 WFG1-9	3, 5, 8, 10, 15 3, 5, 8, 10
[32]	2016	U-NSGA-III	ZDT1-4, 6 DTLZ1-2 S-DTLZ1-2	2 3, 5, 8, 10 3, 5, 8, 10

Scalable many-objective test problems called DTLZ [9] and WFG [10] have been frequently used to evaluate many-objective algorithms in the literature. Table I summarizes many-objective test problems used for performance evaluation of recently proposed weight vector-based algorithms [27]–[32]. For comparison, we also show multiobjective test problems used for evaluating MOEA/D in [25]. We can see from this table that DTLZ1-4 and WFG1-9 have often been used in the literature. In this section, we briefly explain those test problems.

A. DTLZ Test Problems

The DTLZ test suite was designed by Deb *et al.* [9] as a set of nine scalable test problems (DTLZ1-9). The number of decision variables (say n) of M -objective DTLZ1-7 problems is specified as $n = M + k - 1$, where k is a parameter. The value of k is often specified as $k = 5$ in DTLZ1, $k = 10$ in DTLZ2-6, and $k = 20$ in DTLZ7. In DTLZ8-9, the number of decision variables is specified as $n = 10M$.

Let $\mathbf{y}^* = (y_1^*, y_2^*, \dots, y_M^*)$ be a Pareto optimal solution in the M -dimensional objective space of each test problem. The Pareto front of each of the first four test problems (DTLZ1-4) can be represented by the following formulations [9]:

$$\text{DTLZ1: } \sum_{i=1}^M y_i^* = 0.5 \text{ and } y_i^* \geq 0 \text{ for } i = 1, 2, \dots, M \quad (2)$$

$$\text{DTLZ2-4: } \sum_{i=1}^M (y_i^*)^2 = 1 \text{ and } y_i^* \geq 0 \text{ for } i = 1, 2, \dots, M. \quad (3)$$

As shown in Table I, DTLZ1-4 have been frequently used for performance evaluation of many-objective algorithms. This is because their Pareto fronts are represented by the simple formulations in (2) and (3). DTLZ5-6 were originally proposed as many-objective test problems with degenerate Pareto fronts in [9]. However, their Pareto fronts are not degenerate when they have four or more objectives [10], [36], [37]. Constraint

conditions were introduced to remove the nondegenerate parts of the Pareto fronts [36], [37]. The Pareto fronts of DTLZ7-9 cannot be represented by a simple form as in (2) or (3).

B. WFG Test Problems

The WFG test suite was proposed by Huband *et al.* [10] as a set of nine scalable test problems (WFG1-9). An M -objective WFG problem has k position-related variables and l distance-related variables. Thus the total number of decision variables is $n = k + l$. The suggested values of k and l in [10] were $k = 4$ and $l = 20$ for two-objective problems and $k = 2(M - 1)$ and $l = 20$ for problems with more than two objectives. However, different specifications have been used in the literature. For example, they were specified as $k = M - 1$ and $l = 24 - (M - 1)$ in [29]. These specifications [29] are used in this paper. Since l should be an even number in WFG2 and WFG3, the specification of l is slightly changed as $l = 16$ for $M = 8$ and $l = 14$ for $M = 10$ only in WFG2 and WFG3 in this paper.

WFG1 has a complicated Pareto front, which cannot be represented by a simple form as in (2) or (3). The Pareto front of WFG2 is disconnected. WFG3 was originally designed as a many-objective test problem with a degenerate Pareto front. However, its Pareto front is not degenerate when it has three or more objectives as recently pointed out in [38]. All the other test problems (i.e., WFG4-9) have the following Pareto front:

$$\text{WFG4-9: } \sum_{i=1}^M \left(\frac{y_i^*}{2i} \right)^2 = 1 \text{ and } y_i^* \geq 0 \text{ for } i = 1, 2, \dots, M. \quad (4)$$

One important feature of the Pareto fronts of WFG4-9 in (4) is that the domain of each objective has a different magnitude (e.g., $0 \leq y_1^* \leq 2$ and $0 \leq y_{10}^* \leq 20$). This explains why most of the recently proposed many-objective algorithms [27]–[30], [32] in Table I have normalization mechanisms of the objective space. Since MOEA/D [25] has no normalization mechanism, good results have not been reported for many-objective WFG problems in the literature.

C. Common Feature of DTLZ1-4 and WFG4-9

By normalizing the range of the Pareto front for each objective into the unit interval $[0, 1]$, the Pareto fronts of DTLZ1-4 and WFG4-9 can be rewritten as follows:

$$\text{DTLZ1: } \sum_{i=1}^M y_i^* = 1 \text{ and } y_i^* \geq 0 \text{ for } i = 1, 2, \dots, M \quad (5)$$

DTLZ2-4 and WFG4-9

$$\sum_{i=1}^M (y_i^*)^2 = 1 \text{ and } y_i^* \geq 0 \text{ for } i = 1, 2, \dots, M. \quad (6)$$

These formulations show that DTLZ1-4 and WFG4-9 are similar test problems with respect to the shape of their Pareto fronts. This means that most test problems used for evaluating many-objective algorithms in [27]–[32] in Table I are similar with respect to the shape of their Pareto fronts whereas they are different with respect to other aspects such as the curvature property of the Pareto fronts (e.g., linear and concave) and the type of the objective functions (e.g., multimodal and deceptive). In this paper, we explain our concern

that the development of weight vector-based many-objective evolutionary algorithms seems to be overspecialized for the above-mentioned similarity of the Pareto fronts of DTLZ1-4 and WFG4-9.

III. WEIGHT VECTOR-BASED MANY-OBJECTIVE ALGORITHMS

In MOEA/D [25], a multiobjective problem is decomposed into single-objective problems, each of which is generated by a scalarizing function with a different weight vector. Thus the number of single-objective problems is the same as the number of weight vectors. Since a single best solution is stored for each single-objective problem, the population size is also the same as the number of weight vectors. MOEA/D can be viewed as an improved version of a cellular multiobjective genetic algorithm [39]. MOEA/D is also similar to multi-objective genetic local search (MOGLS) [40]–[43]: both of them optimize scalarizing functions. Whereas weight vectors are systematically generated and fixed in MOEA/D, they are randomly updated in each generation in MOGLS.

A. Weight Vector Specification

In the original MOEA/D [25], all weight vectors $\mathbf{w} = (w_1, w_2, \dots, w_M)$ satisfying the following formulations are generated:

$$\sum_{i=1}^M w_i = 1 \text{ and } w_i \geq 0 \text{ for } i = 1, 2, \dots, M \quad (7)$$

$$w_i \in \left\{ 0, \frac{1}{H}, \frac{2}{H}, \dots, \frac{H}{H} \right\} \text{ for } i = 1, 2, \dots, M \quad (8)$$

where H is a positive integer. The total number of the weight vectors (say N , which is the same as the population size) is calculated as $N = C_{M-1}^{H+M-1}$ (i.e., $N = H+M-1 C_{M-1}$ [25]).

Weight vectors in (7) and (8) are on the hyper-plane specified by (7) in an M -dimensional space. For example, all weight vectors for $M = 3$ are on the triangular shape plane specified by $w_1 + w_2 + w_3 = 1$ and $w_i \geq 0$ for $i = 1, 2, 3$. When $H < M$, at least one element is zero (i.e., $w_i = 0$) in all weight vectors satisfying (7) and (8). That is, no weight vector is inside the hyper-plane specified by (7). However, the use of a large value for H satisfying $H \geq M$ leads to an impractically large number of weight vectors for many-objective problems. For example, if we specify H as $H = 10$ for a ten-objective problem, 92 378 weight vectors are generated. In this case, only a single weight vector $(0.1, 0.1, \dots, 0.1)$ is inside the hyper-plane. All the other weight vectors are on its boundary.

For handling this difficulty, a two-layered approach was proposed in NSGA-III [30] and used in recently proposed weight vector-based many-objective algorithms. In the two-layered approach, two sets of weight vectors $\mathbf{w} = (w_1, w_2, \dots, w_M)$ are generated using (7) and (8). One set of weight vectors is generated from an integer H_1 and used with no modification as the boundary layer weight vectors. The other set of weight vectors is generated from another integer H_2 and used as the inside layer weight vectors after the following modification:

$$w_i = (w_i + 1/M)/2 \text{ for } i = 1, 2, \dots, M. \quad (9)$$

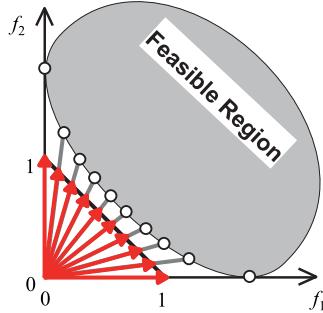
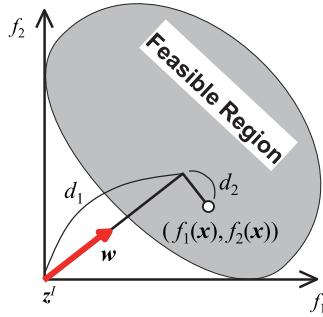


Fig. 6. Basic idea of MOEA/D and weight vector-based algorithms.

Fig. 7. Search for a solution using the weight vector w in MOEA/D.

The two-layered approach is used for many-objective problems with eight or more objectives in this paper. It should be noted that all weight vectors generated by the two-layered approach are on the same hyper-plane specified by (7), which is the same as all weight vectors in the original MOEA/D.

Another frequently used modification is the use of a small number 10^{-6} when $w_i = 0$ in the weighted Tchebycheff function. We use this modification in our computational experiments.

B. Basic Idea of MOEA/D

The basic idea of MOEA/D (and its variants) is to find a set of well-distributed nondominated solutions along the Pareto front using the systematically generated weight vectors as shown in Fig. 6. The realization of this idea is often explained by two distances in Fig. 7: 1) d_1 and 2) d_2 .

In Fig. 7, d_1 is the distance from the ideal point z^I to the solution $(f_1(\mathbf{x}), f_2(\mathbf{x}))$ along the reference line specified by a weight vector, and d_2 is the distance from the reference line to the solution $(f_1(\mathbf{x}), f_2(\mathbf{x}))$. These distances should be minimized for each reference line (i.e., in each single-objective problem).

It is also important to assign a single solution to each reference line. Weight vector-based algorithms [27]–[32] have their own mechanisms to minimize d_1 and d_2 , and to assign a single solution to each reference line. While some algorithms [30]–[32] use Pareto dominance for minimizing d_1 , the basic idea is the same: the search for well-distributed nondominated solutions along the Pareto front using a set of reference lines.

The ideal point z^I is defined by the best value of each objective as shown in Fig. 7. In general, the ideal point z^I

is unknown. So, its approximation is usually used as a reference point z^* in MOEA/D and other weight vector-based algorithms as explained in the next section.

C. Scalarizing Functions

Three scalarizing functions were examined in the original MOEA/D [25]: 1) the weighted sum; 2) the weighted Tchebycheff function; and 3) the penalty-based boundary intersection (PBI) function. For the minimization problem in (1), the weighted sum with a weight vector $w = (w_1, w_2, \dots, w_M)$ is written as

$$\text{Minimize } f^{\text{WS}}(\mathbf{x}|w) = w_1 f_1(\mathbf{x}) + \dots + w_M f_M(\mathbf{x}). \quad (10)$$

The weighted Tchebycheff function is written using a reference point $z^* = (z_1^*, z_2^*, \dots, z_M^*)$ as

$$\text{Minimize } f^{\text{Tch}}(\mathbf{x}|w, z^*) = \max_{i=1,2,\dots,M} \{w_i \cdot |z_i^* - f_i(\mathbf{x})|\}. \quad (11)$$

Each element z_i^* of the reference point z^* is specified by the minimum (i.e., best) value of each objective $f_i(\mathbf{x})$ among the examined solutions during the execution of MOEA/D.

Using a penalty parameter θ ($\theta = 5$ in [25]) and the reference point z^* , the PBI function is written as

$$\text{Minimize } f^{\text{PBI}}(\mathbf{x}|w, z^*) = d_1 + \theta d_2 \quad (12)$$

where d_1 and d_2 are defined as follows (see Fig. 7):

$$d_1 = |(\mathbf{f}(\mathbf{x}) - z^*)^T w| / \|w\| \quad (13)$$

$$d_2 = \left\| \mathbf{f}(\mathbf{x}) - z^* - d_1 \frac{w}{\|w\|} \right\|. \quad (14)$$

D. Neighborhood Structure

In the original MOEA/D [25], each weight vector has its neighbors. A prespecified number of similar weight vectors are defined as neighbors for each weight vector. A weight vector itself is included in its own neighbors. When a solution is to be generated for a weight vector, parents are selected from its neighbors. The generated new solution is compared with the solution of each neighbor. If the new solution is better, the current solution is replaced. The comparison for replacement is performed against all solutions of the neighbors.

This replacement strategy has two potential difficulties. One is explained by the following case. A new solution, which is generated far from the corresponding reference line, is very good for a different reference line while its evaluation is poor for the corresponding reference line. This case is explained in Fig. 8. Let us assume in Fig. 8 that the set of the seven open circles is a current population. We also assume that solution A is generated for a reference line l_2 with two neighbors l_1 and l_3 . The generated solution A, which is very good for a reference line l_2 , is not good for any of l_1 , l_2 , and l_3 . Thus no solution is replaced with A. While such a situation is not likely to happen very often, its appropriate handling may improve the efficiency of MOEA/D. This difficulty can be partially remedied by using a large number of neighbors. However, this remedy may worsen another potential difficulty explained by the following case: many neighboring solutions are replaced with a single good solution. This leads to the decrease in the

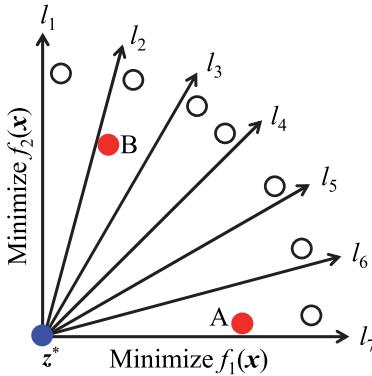


Fig. 8. Illustration of reference lines and generated solutions A and B.

diversity of solutions. For example, if solution B is generated for the reference line l_2 in Fig. 8, all the current solutions of l_1 , l_2 , and l_3 will be replaced with solution B.

E. Performance Improvement Mechanisms

These two difficulties can be remedied by the following replacement policy. A new solution is compared with its similar solutions, and only a single solution is replaced with the new solution. Instead of using the prespecified neighbors, a set of similar solutions is selected for the new solution. First the nearest reference line to the new solution is identified in the objective space (e.g., line l_2 for solution B in Fig. 8). Next all solutions with the same nearest reference line as the new solution are selected as its similar solutions. Then the new solution is compared with each similar solution. Only a single solution can be replaced with the new solution. If the new solution is not better than any similar solutions, no solution is replaced. This policy is implemented in [27]–[32] using different mechanisms for similar solution selection, solution comparison, and solution replacement.

In NSGA-III [30], reference points are specified in the normalized objective space in a similar manner to the weight vector specification in MOEA/D. Using the reference points and the origin of the normalized objective space, reference lines are generated. Each solution is assigned to its nearest reference line. Solution comparison in NSGA-III is performed using nondominated sorting, the number of solutions assigned to each reference line, and the distance from each solution to its nearest reference line. Solutions are updated by a $(\mu + \mu)$ ES-style model. When multiple solutions with the best nondominated sorting rank are assigned to the same reference line, the best solution with the shortest distance to the reference line is selected from them as a member in the next generation. The other solutions are randomly ordered. After the generation update, most reference lines have a single solution. However, it is possible that some reference lines have multiple solutions. It is also possible that other reference lines have no solutions.

In θ -DEA [29], each solution is assigned to its nearest reference line in the same manner as NSGA-III. The PBI function is used for the ranking of solutions assigned to the same reference line. Solutions are updated by a $(\mu + \mu)$ ES-style

model based on the rank of each solution. Some reference lines may have multiple solutions, and others may have no solutions.

In MOEA/DD [31], a $(\mu + 1)$ ES-style model is used as in the original MOEA/D. Each solution is assigned to the nearest weight vector as in NSGA-III and θ -DEA. Each solution is evaluated by its nondominated sorting rank, its PBI value, and the number of solutions assigned to the same weight vector. When all solutions in the current population are non-dominated, one solution with the largest PBI value is deleted from the most crowded weight vector with the largest number of solutions. When some solutions are dominated, one solution to be deleted is selected from the worst rank solutions using the number of solutions assigned to each weight vector and the PBI value of each solution. However, when a weight vector has a single solution, the solution is not deleted for diversity maintenance even if it has the worst nondominated sorting rank.

As we have already explained, weight vector-based algorithms in [27]–[30] and [32] have normalization mechanisms. The two-layered approach in NSGA-III [30] are used in [27]–[32]. In addition to these two common features, each algorithm has its own mechanisms for efficiently realizing the search for a set of well-distributed solutions along the Pareto front of a many-objective problem.

F. Relation Between Test Problems and Algorithms

As shown in Section II, the DTLZ1-4 and WFG4-9 test problems have the following Pareto front in the normalized M -dimensional objective space:

$$\text{Pareto Front: } \sum_{i=1}^M (y_i^*)^k = 1 \text{ and } y_i^* \geq 0 \text{ for } i = 1, 2, \dots, M \quad (15)$$

where $k = 1$ (DTLZ1) and $k = 2$ (DTLZ2-4 and WFG4-9).

Weight vectors in the weight vector-based algorithms [27]–[32] are generated in the M -dimensional weight space as

$$\text{Weight Vectors: } \sum_{i=1}^M w_i = 1 \text{ and } w_i \geq 0 \text{ for } i = 1, 2, \dots, M. \quad (16)$$

The shape of the Pareto front of each test problem in (15) is the same as or similar to the shape of the hyper-plane in (16) on which the weight vectors are generated. This explains why the search for a single best solution for each weight vector leads to a set of well-distributed solutions over the entire Pareto front. From the similarity between the shape of the Pareto front in (15) and the shape of the distribution of the weight vectors in (16), the following questions may arise.

- Q1: How general is the high performance of weight vector-based algorithms on the DTLZ1-4 and WFG4-9 problems?
- Q2: How general are the triangular shape Pareto fronts of the DTLZ1-4 and WFG4-9 test problems?

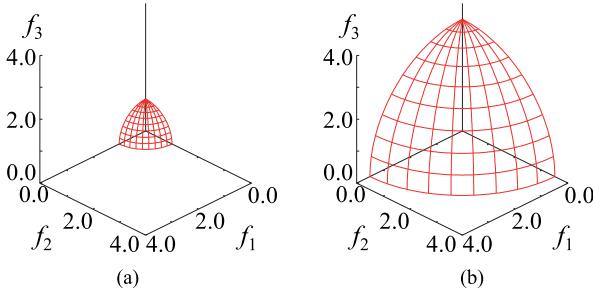


Fig. 9. Pareto fronts of three-objective DTLZ2 and Max-DTLZ2 problems. (a) Original three-objective DTLZ2. (b) Maximization version of DTLZ2.

In the next section, we discuss the first question through computational experiments. The second question is briefly discussed in Section V as a future research topic.

IV. COMPUTATIONAL EXPERIMENTS

In this section, first we explain our test problems generated by slightly changing the DTLZ and WFG formulations. Then we demonstrate how high performance of weight vector-based algorithms on the DTLZ and WFG problems is deteriorated by the slight change in the test problem formulations.

A. Our Test Problems: DTLZ $^{-1}$ and WFG $^{-1}$

As we have already explained, the DTLZ and WFG test problems have the following form:

$$\text{Minimize } f_1(\mathbf{x}), \dots, f_M(\mathbf{x}) \text{ subject to } \mathbf{x} \in X. \quad (17)$$

Our idea is to generate a slightly different test problem from each of the DTLZ and WFG problems. More specifically, we change their general form from (17) to

$$\text{Maximize } f_1(\mathbf{x}), \dots, f_M(\mathbf{x}) \text{ subject to } \mathbf{x} \in X. \quad (18)$$

We use exactly the same objective functions and the same constraint conditions except for changing from “Minimize” in (17) to “Maximize” in (18). The generated problems in (18) are referred to as the Max-DTLZ and Max-WFG problems. Those problems are handled as the following minimization problems:

$$\text{Minimize } -f_1(\mathbf{x}), \dots, -f_M(\mathbf{x}) \text{ subject to } \mathbf{x} \in X. \quad (19)$$

All objectives in DTLZ and WFG are multiplied by (-1) in our test problems in (19). That is, a negative sign is added to all objectives in DTLZ and WFG. In this paper, the minus versions in (19) of DTLZ and WFG are referred to as DTLZ $^{-1}$ and WFG $^{-1}$ (Minus-DTLZ and Minus-WFG), respectively. The minimization formulation in (19) with a minus sign for all objectives is the same as the maximization form in (18). In our computational experiments, we use the minimization form to handle all test problems as minimization problems (i.e., to apply each algorithm with no modification to DTLZ, WFG, DTLZ $^{-1}$, and WFG $^{-1}$ in exactly the same manner).

In Fig. 9, we show the Pareto fronts of the original three-objective DTLZ2 problem and its maximization version in (18): Max-DTLZ2. The Pareto fronts of the two test problems have the same shape with different size. In Fig. 10,

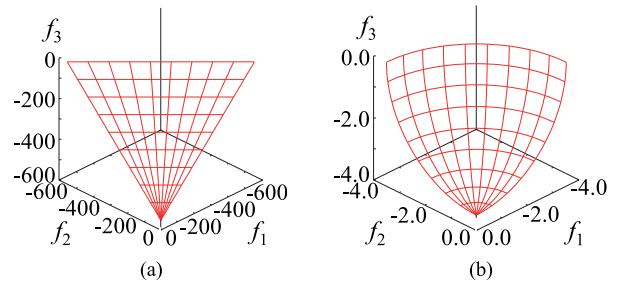


Fig. 10. Pareto fronts of the minus versions of DTLZ1 and DTLZ2. (a) Three-objective DTLZ1 $^{-1}$. (b) Three-objective DTLZ2 $^{-1}$.

we show the Pareto fronts of the minus versions of the three-objective DTLZ1 and DTLZ2 (i.e., DTLZ1 $^{-1}$ and DTLZ2 $^{-1}$). As shown in Fig. 10(a), DTLZ1 $^{-1}$ has a rotated triangular shape Pareto front. The Pareto front of DTLZ2 $^{-1}$ in Fig. 10(b) has a rotated shape of the Pareto front of the maximization version of DTLZ2 in Fig. 9(b). In this section, we examine the performance of weight vector-based algorithms on the DTLZ and WFG test problems and their minus versions in (19).

As shown in Figs. 9 and 10 for DTLZ1 and DTLZ2, the minus versions of DTLZ have much larger Pareto fronts than the original DTLZ problems. When DTLZ has a concave Pareto front, its minus version has a convex Pareto front. Moreover, the difficulty of optimization is not the same. For example, the Pareto optimal solutions of DTLZ2 are obtained when all of its distance variables are 0.5. However, the Pareto optimal solutions of DTLZ2 $^{-1}$ are obtained when all of its distance variables are 0 or 1 (i.e., at the boundaries of the feasible region of each distance variable \$x_i\$ with the constraint condition \$0 \leq x_i \leq 1\$). These discussions clearly show that our idea is not the best way for inverting the Pareto front without changing the other properties of the original DTLZ and WFG test problems. For this purpose, modification of only the shape function may be the best way. In this paper, we use the above-mentioned simple idea for inverting the Pareto front since it is simple and intuitively understandable.

A test problem called the inverted DTLZ1 was formulated in [44] by applying the following transformation to each objective \$f_i(\mathbf{x})\$ of DTLZ1 (\$i = 1, 2, \dots, M\$):

$$f_i(\mathbf{x}) \leftarrow 0.5(1 + g(\mathbf{x})) - f_i(\mathbf{x}) \quad (20)$$

where \$g(\mathbf{x})\$ is a function in the original DTLZ1 formulation [9]. The Pareto front of the inverted DTLZ1 [44] has a rotated shape of the Pareto front of DTLZ1 [9]. The size of their Pareto fronts is the same since the inverted DTLZ1 was formulated by rotating DTLZ1. It should be noted that our DTLZ1 $^{-1}$ has a much larger Pareto front than DTLZ1 and its rotated version as shown in Fig. 10(a).

B. Examined Algorithms and Parameter Specifications

Our computational experiments on DTLZ1-4 and WFG1-9 with 3, 5, 8, and 10 objectives are performed under the same settings as in [29]. We examine the performance of \$\theta\$-DEA [29], NSGA-III [30], and MOEA/DD [31]. We also use NSGA-II [7] and four versions

TABLE II
NUMBER OF WEIGHT VECTORS AND THE POPULATION SIZE FOR EACH TEST PROBLEM (THE SAME SPECIFICATIONS AS IN [29])

Number of objectives	Divisions		Number of weight vectors	Population size of NSGA-II, NSGA-III and θ -DEA
	H_1	H_2		
3	12	-	91	92
5	6	-	210	212
8	3	2	156	156
10	3	2	275	276

of MOEA/D [25]: 1) MOEA/D-WS with the weighted sum; 2) MOEA/D-Tch with the Tchebycheff function; 3) MOEA/D-PBI with the PBI function; and 4) MOEA/D-IPBI with the inverted PBI (IPBI) function [45].

The IPBI function is defined by the distance from the nadir point z^N , which is a vector consisting of the worst value of each objective over the true Pareto front. For minimization problems, the IPBI function [45] is defined as

$$\text{Maximize } f^{\text{IPBI}}(\mathbf{x}|\mathbf{w}, z^N) = d_1 - \theta d_2 \quad (21)$$

where θ is a penalty parameter, and d_1 and d_2 are defined as follows:

$$d_1 = \left\| (\mathbf{z}^N - \mathbf{f}(\mathbf{x}))^T \mathbf{w} \right\| / \|\mathbf{w}\| \quad (22)$$

$$d_2 = \left\| \mathbf{z}^N - \mathbf{f}(\mathbf{x}) - d_1 \frac{\mathbf{w}}{\|\mathbf{w}\|} \right\|. \quad (23)$$

In this paper, we use the same implementation of MOEA/D-IPBI as in [45]: the penalty parameter θ is specified as $\theta = 0.1$, and the nadir point z^N is approximated by the worst value of each objective in the current population.

Multiobjective search is performed in MOEA/D-IPBI by pushing each solution in the direction from the nadir point to the Pareto front by maximizing d_1 and minimizing d_2 . This search mechanism contrasts to MOEA/D-PBI, MOEA/D-Tch and the other weight vector-based algorithms in [27]–[32] where each solution is pulled toward the ideal point.

Exactly the same set of weight vectors is used in all algorithms for each test problem. Table II shows the number of weight vectors. The two-layered approach is used for test problems with eight and ten objectives. The population size is the same as the number of weight vectors except for NSGA-II, NSGA-III, and θ -DEA where the population size is specified as multiples of four (see Table II). The weight value 0 is replaced with 10^{-6} in MOEA/D-Tch.

As a termination condition for each test problem, we use the same prespecified total number of generations as in [29]. The penalty parameter θ in MOEA/D-PBI, MOEA/DD and θ -DEA is specified as $\theta = 5$ ($\theta = 0.1$ in IPBI). The neighborhood size in MOEA/D is 20. The polynomial mutation with the distribution index 20 is used with the mutation probability $1/n$ (n is the string length). The simulated binary crossover with the distribution index 20 (30 in NSGA-III, θ -DEA and MOEA/DD) is used with the crossover probability 1.0. Our parameter specifications are the same as in [29].

We use our own implementations of the four versions of MOEA/D because we have already examined them on various test problems. This is also because we have not observed any clear inconsistency between our results and the reported results in [29]. With respect to NSGA-II, θ -DEA, NSGA-III, and MOEA/DD, we use available codes through the Internet: NSGA-II from jMetal [46], θ -DEA, and NSGA-III from [47] by Yuan *et al.* [29] of the θ -DEA paper, and MOEA/DD from [48] by Li *et al.* [31] of the MOEA/DD paper.

C. Performance Measures

As a performance measure, we use the hypervolume in the same manner as [29] for the original DTLZ and WFG problems (the setting of the reference point for hypervolume calculation is the same as [29]). First, the objective space is normalized using the ideal point z^I and the nadir point z^N so that they are normalized as $z^I = (0, 0, \dots, 0)$ and $z^N = (1, 1, \dots, 1)$. Then the hypervolume is calculated by specifying the reference point as $(1.1, 1.1, \dots, 1.1)$. The same setting for hypervolume calculation is used for DTLZ $^{-1}$ and WFG $^{-1}$. We also use a different reference point $(2, 2, \dots, 2)$ for DTLZ $^{-1}$ and WFG $^{-1}$ to examine whether well distributed solutions are obtained around the boundaries of their Pareto fronts. We use a fast calculation method of the exact hypervolume value proposed in [49] for all test problems with 3, 5, 8, and 10 objectives.

We also calculate the IGD indicator in the normalized objective space for all test problems. The Euclidean distance is used in the normalized objective space of each test problem. A set of reference points for the IGD calculation is generated for each test problem as follows. For DTLZ1-4, DTLZ1-4 $^{-1}$, WFG4-9, and WFG3-9 $^{-1}$, we randomly generate 100 000 reference points on the true Pareto front using the uniform distribution. For the other test problems (i.e., WFG1-3 and WFG1-2 $^{-1}$), we use all nondominated solutions among all of the obtained solutions in Tables III and IV since the specification of the uniform distribution over the true Pareto front is difficult for these test problems.

D. Experimental Results on Original Test Problems

The average hypervolume values over 101 runs on the original DTLZ1-4 and WFG1-9 problems are summarized in Table III. The best average result is highlighted by bold and underlined for each test problem. The worst four average results are shaded. For DTLZ1-4, the best average results are obtained by MOEA/DD for 12 out of the 16 problems (75%). For WFG4-9, the best average results are obtained by θ -DEA for 23 out of the 24 problems (96%). In Table III, these two algorithms are the best. The best average result for each test problem in WFG1-3 is obtained by a different algorithm due to their different features as test problems.

The best average result is not obtained by MOEA/D or MOEA/DD for WFG4-9 due to the lack of an objective space normalization mechanism. The performance of MOEA/DD on WFG4-9 can be improved by a normalization mechanism. Just for comparison, we perform computational experiments on the normalized WFG4-9 test problems after changing all objectives as $f_i(\mathbf{x}) = f_i(\mathbf{x})/2i$, $i = 1, 2, \dots, M$. For the

TABLE III
AVERAGE HYPERVOLUME VALUES FOR THE REFERENCE POINT (1.1, 1.1, ..., 1.1) OVER 101 RUNS ON THE ORIGINAL TEST PROBLEMS. THE BEST AVERAGE RESULT FOR EACH TEST PROBLEM IS HIGHLIGHTED BY BOLD AND UNDERLINED. THE WORST FOUR AVERAGE RESULTS FOR EACH TEST PROBLEM ARE SHADED

Problem	M	NSGA-III	θ -DEA	MOEA/DD	PBI	Tch	WS	IPBI	NSGA-II
DTLZ1	3	1.11508	1.11767	<u>1.11913</u>	1.11711	1.06842	0.39572	0.48149	1.07411
	5	1.57677	1.57767	<u>1.57794</u>	1.57768	1.51186	0.50052	0.02284	0.00000
	8	2.13770	<u>2.13788</u>	2.13730	2.13620	2.05463	0.96246	1.44289	0.00000
	10	<u>2.59280</u>	2.59272	2.59260	2.59220	2.51973	1.07913	1.90272	0.00000
DTLZ2	3	0.74336	0.74390	<u>0.74445</u>	0.74418	0.70168	0.33187	0.33100	0.69708
	5	1.30317	1.30679	<u>1.30778</u>	1.30728	1.14598	0.61944	0.27191	0.67442
	8	1.96916	1.97785	<u>1.97862</u>	1.97817	1.35469	0.68315	0.54410	0.00004
	10	2.50878	2.51416	<u>2.51509</u>	2.51500	1.69045	0.83883	0.64925	0.00000
DTLZ3	3	0.73300	0.73642	<u>0.73944</u>	0.73654	0.69553	0.33026	0.31397	0.69959
	5	1.29894	1.30376	<u>1.30638</u>	1.30398	1.14475	0.60143	0.00750	0.00000
	8	1.95007	1.96849	<u>1.97162</u>	1.74240	1.33166	0.66684	0.29765	0.00000
	10	2.50727	2.51279	<u>2.51445</u>	2.50933	1.69956	0.80348	0.52362	0.00000
DTLZ4	3	0.73221	0.71077	<u>0.74484</u>	0.48232	0.45889	0.17191	0.23377	0.70481
	5	1.30839	<u>1.30878</u>	1.30876	1.20680	1.00426	0.42941	0.33457	1.00881
	8	1.98025	1.98078	<u>1.98083</u>	1.86439	1.35100	0.71296	0.53303	0.00000
	10	2.51524	<u>2.51539</u>	2.51532	2.43536	1.56890	0.95488	0.64498	0.00000
WFG1	3	0.65088	0.70151	0.69393	0.67291	<u>0.92204</u>	0.73804	0.81622	0.75944
	5	0.85608	1.14844	1.23809	1.34797	<u>1.51824</u>	1.36724	1.36241	1.03120
	8	1.36206	1.88297	1.91925	1.73875	<u>2.05117</u>	1.85604	1.75472	1.51083
	10	2.22078	2.38349	2.37705	1.78435	<u>2.46470</u>	2.27031	2.18237	2.38032
WFG2	3	1.22359	<u>1.22945</u>	1.22193	1.11888	1.12990	1.12266	1.16687	1.20760
	5	<u>1.59770</u>	1.59708	1.55672	1.52205	1.58417	1.42821	1.42081	1.58790
	8	<u>2.13629</u>	2.12442	2.04619	2.01678	2.13569	2.11651	2.11529	2.13214
	10	2.58890	2.57778	2.48332	2.45715	<u>2.58891</u>	2.57478	2.57367	2.58882
WFG3	3	0.81929	0.81556	0.77295	0.75364	0.80041	0.48971	0.74146	<u>0.82967</u>
	5	1.01000	1.02782	0.95386	0.89357	0.88322	0.71619	0.93099	<u>1.06314</u>
	8	1.21146	1.11348	1.15306	0.74674	1.27479	0.92248	1.41331	<u>1.41857</u>
	10	<u>1.55771</u>	1.55919	1.37737	0.55186	1.69917	1.13233	1.72878	<u>1.76576</u>
WFG4	3	0.72867	<u>0.72949</u>	0.72031	0.68710	0.66650	0.34131	0.63483	0.67605
	5	1.28496	<u>1.28736</u>	1.26067	1.15695	1.01300	0.71180	1.04810	1.07969
	8	1.96402	<u>1.96426</u>	1.83751	1.19841	1.33398	0.95883	1.45141	1.40330
	10	2.50322	<u>2.50376</u>	2.22383	1.43393	1.49165	1.20197	1.74551	1.70402
WFG5	3	0.68658	<u>0.68706</u>	0.67698	0.65668	0.61681	0.27764	0.58174	0.65059
	5	1.22187	<u>1.22209</u>	1.18965	1.11627	0.93276	0.58164	0.96542	1.06695
	8	1.84995	<u>1.85027</u>	1.71196	1.27483	1.18970	0.96591	1.33675	1.39529
	10	2.34640	<u>2.34644</u>	2.07711	1.53615	1.35553	1.18471	1.57386	1.61976
WFG6	3	0.68696	<u>0.68698</u>	0.67923	0.65655	0.62307	0.28542	0.58469	0.64111
	5	1.21978	<u>1.22284</u>	1.19424	1.04043	0.93460	0.55026	0.97587	1.01175
	8	<u>1.84625</u>	1.84330	1.69055	0.71742	1.17924	0.63171	1.21597	1.27938
	10	2.32660	<u>2.32759</u>	2.01837	0.82027	1.44519	0.77606	1.48368	1.59677
WFG7	3	0.72894	<u>0.73099</u>	0.72126	0.61145	0.66659	0.33309	0.62859	0.68591
	5	1.29190	<u>1.29548</u>	1.25983	1.07723	1.01449	0.63899	1.04794	0.97811
	8	1.97138	<u>1.97353</u>	1.82024	0.83439	1.30773	0.71170	1.45307	1.22911
	10	2.50754	<u>2.50858</u>	2.25713	0.95972	1.59993	0.97177	1.73385	1.59601
WFG8	3	0.66560	<u>0.66687</u>	0.65741	0.62986	0.61394	0.24450	0.26792	0.61230
	5	1.18225	<u>1.18354</u>	1.15376	0.95660	0.60364	0.46673	0.82273	0.96648
	8	1.75970	<u>1.76647</u>	1.70621	0.30471	1.20786	0.67808	1.24044	1.28486
	10	2.28203	<u>2.28502</u>	2.10729	0.27470	1.60952	0.82704	1.57781	1.69433
WFG9	3	0.67519	<u>0.67978</u>	0.67146	0.57864	0.62177	0.25170	0.51403	0.62199
	5	1.21058	<u>1.22122</u>	1.15493	1.02426	0.78608	0.53143	0.94420	0.92841
	8	1.80911	<u>1.83678</u>	1.60407	0.97800	1.23897	0.72454	1.18318	1.07824
	10	2.34332	<u>2.36516</u>	1.92977	1.15138	1.59168	0.86178	1.49927	1.42611

normalized WFG4-9 test problems, the best average results are obtained by MOEA/DD for 14 out of the 24 test problems (58%). This observation together with the results on DTLZ1-4

in Table III suggests that MOEA/DD with a normalized mechanism may be the best algorithm for DTLZ1-4 and WFG4-9.

TABLE IV
AVERAGE HYPERVOLUME VALUES OVER 101 RUNS ON OUR MINUS TEST PROBLEMS FOR THE REFERENCE POINT (1.1, 1.1, ..., 1.1).
THE BEST AVERAGE RESULT FOR EACH TEST PROBLEM IS HIGHLIGHTED BY BOLD AND UNDERLINED. THE WORST
FOUR AVERAGE RESULTS FOR EACH TEST PROBLEM ARE SHADED

Problem	M	NSGA-III	θ -DEA	MOEA/DD	PBI	Tch	WS	IPBI	NSGA-II
DTLZ1 ⁻¹	3	<u>0.27258</u>	0.25057	0.24887	0.26146	0.27141	0.03935	0.17744	0.26905
	5	0.01265	0.00898	0.00972	<u>0.01739</u>	0.01208	0.00083	0.00671	0.01520
	8	<u>5.227E-05</u>	4.499E-05	0.881E-05	0.598E-05	3.215E-05	0.139E-05	2.855E-05	3.568E-05
	10	<u>1.185E-06</u>	0.451E-06	0.100E-06	0.079E-06	0.620E-06	0.025E-06	0.567E-06	0.765E-06
DTLZ2 ⁻¹	3	0.68986	0.69303	0.68912	0.69439	0.68780	<u>0.70652</u>	0.70650	0.68187
	5	0.13957	0.13496	0.08794	0.15984	0.15556	0.14930	0.14910	<u>0.17147</u>
	8	4.454E-03	3.406E-03	2.690E-03	<u>5.978E-03</u>	0.459E-03	1.560E-03	1.560E-03	4.585E-03
	10	<u>6.308E-04</u>	5.541E-04	1.836E-04	5.199E-04	0.052E-04	0.640E-04	0.639E-04	3.797E-04
DTLZ3 ⁻¹	3	0.69251	0.69468	0.68990	0.69609	0.68667	<u>0.70650</u>	0.70650	0.68267
	5	0.12951	0.13273	0.08190	0.15902	0.15199	0.14891	0.14886	<u>0.16472</u>
	8	0.00414	0.00401	0.00255	<u>0.00596</u>	0.00050	0.00156	0.00156	0.00390
	10	0.00054	<u>0.00059</u>	0.00018	0.00052	0.00001	0.00006	0.00006	0.00033
DTLZ4 ⁻¹	3	0.69397	0.69546	0.68942	0.59319	0.68049	<u>0.70650</u>	0.64625	0.68358
	5	0.12326	0.11428	0.07242	0.12296	0.14878	0.14881	0.13995	<u>0.16970</u>
	8	<u>4.582E-03</u>	3.921E-03	2.198E-03	2.020E-03	0.485E-03	1.563E-03	1.340E-03	3.886E-03
	10	6.065E-04	<u>6.409E-04</u>	2.569E-04	2.333E-04	0.043E-04	0.642E-04	0.649E-04	3.006E-04
WFG1 ⁻¹	3	0.10955	0.08936	0.08475	0.03944	0.07838	0.04427	0.06037	<u>0.12500</u>
	5	0.00221	0.00155	0.00094	0.00033	0.00174	0.00089	0.00113	<u>0.00296</u>
	8	1.835E-06	1.401E-06	1.028E-06	0.126E-06	3.015E-06	1.767E-06	1.798E-06	<u>3.640E-06</u>
	10	1.891E-08	1.524E-08	0.962E-08	0.149E-08	4.755E-08	2.414E-08	2.533E-08	<u>4.974E-08</u>
WFG2 ⁻¹	3	<u>0.38373</u>	0.38347	0.38123	0.37769	0.37505	0.20617	0.31447	0.36889
	5	0.01067	0.00805	0.00611	0.00500	<u>0.01143</u>	0.00398	0.00443	0.01055
	8	0.784E-05	0.638E-05	0.383E-05	0.368E-05	<u>1.585E-05</u>	0.690E-05	0.730E-05	1.290E-05
	10	0.795E-07	0.569E-07	0.441E-07	0.378E-07	<u>2.304E-07</u>	0.885E-07	0.977E-07	1.787E-07
WFG3 ⁻¹	3	<u>0.26507</u>	0.24959	0.23184	0.25481	0.25408	0.03245	0.11691	0.26451
	5	0.01279	0.00912	0.00388	0.00459	0.01082	0.00053	0.00286	<u>0.01312</u>
	8	<u>3.666E-05</u>	1.415E-05	0.262E-05	0.417E-05	1.598E-05	0.083E-05	0.300E-05	2.035E-05
	10	<u>6.673E-07</u>	2.511E-07	0.250E-07	0.483E-07	2.704E-07	0.106E-07	0.499E-07	4.847E-07
WFG4 ⁻¹	3	0.66343	0.68880	0.66140	0.68582	0.66881	0.68655	<u>0.69140</u>	0.66561
	5	0.12711	0.14416	0.10758	0.13711	0.08523	0.10288	0.11997	<u>0.14780</u>
	8	5.007E-03	<u>5.123E-03</u>	0.255E-03	0.602E-03	0.548E-03	2.351E-03	1.914E-03	2.758E-03
	10	<u>5.475E-04</u>	2.537E-04	0.039E-04	0.239E-04	0.171E-04	1.539E-04	1.151E-04	1.951E-04
WFG5 ⁻¹	3	0.66841	0.68748	0.67405	0.68567	0.67011	0.68645	<u>0.69118</u>	0.67184
	5	0.12789	0.12399	0.12320	0.13919	0.08783	0.10558	0.12259	<u>0.16091</u>
	8	0.00421	<u>0.00436</u>	0.00062	0.00080	0.00050	0.00237	0.00195	0.00250
	10	<u>0.00046</u>	0.00025	0.00002	0.00003	0.00001	0.00016	0.00011	0.00015
WFG6 ⁻¹	3	0.68331	<u>0.69235</u>	0.67553	0.68534	0.66845	0.68665	0.69144	0.68281
	5	0.13628	0.12549	0.12332	0.13846	0.08150	0.10292	0.11987	<u>0.16948</u>
	8	<u>0.00450</u>	0.00382	0.00075	0.00076	0.00043	0.00236	0.00194	0.00248
	10	<u>0.00053</u>	0.00022	0.00002	0.00003	0.00001	0.00016	0.00011	0.00020
WFG7 ⁻¹	3	0.65101	0.68135	0.65126	0.67742	0.65881	0.68664	<u>0.69143</u>	0.65047
	5	0.11727	0.11857	0.11268	0.13727	0.08508	0.10297	0.11996	<u>0.14742</u>
	8	<u>0.00441</u>	0.00382	0.00049	0.00054	0.00050	0.00237	0.00192	0.00340
	10	<u>0.00047</u>	0.00023	0.00002	0.00002	0.00001	0.00015	0.00011	0.00032
WFG8 ⁻¹	3	0.68958	<u>0.69311</u>	0.67910	0.68517	0.66818	0.68660	0.69143	0.68535
	5	0.13845	0.12755	0.12962	0.13872	0.08272	0.10293	0.11978	<u>0.17643</u>
	8	<u>0.00446</u>	0.00405	0.00129	0.00090	0.00038	0.00237	0.00195	0.00381
	10	<u>0.00055</u>	0.00023	0.00005	0.00003	0.00001	0.00016	0.00012	0.00034
WFG9 ⁻¹	3	0.67193	0.68446	0.64574	0.66636	0.65325	0.68255	<u>0.68630</u>	0.66060
	5	0.13747	0.12627	0.11905	0.13411	0.09712	0.10808	0.12487	<u>0.15893</u>
	8	<u>0.00478</u>	0.00431	0.00088	0.00073	0.00075	0.00222	0.00181	0.00380
	10	<u>0.00048</u>	0.00026	0.00003	0.00003	0.00003	0.00014	0.00010	0.00040

E. Experimental Results on Modified Test Problems

Table IV shows the average hypervolume values for the reference point (1.1, 1.1, ..., 1.1) over 101 runs on our minus test problems: DTLZ⁻¹ and WFG⁻¹. From the comparison

between Tables III and IV, we can see that totally different results are obtained with respect to the performance comparison among the eight algorithms. For example, the best average results are obtained by θ -DEA for 27 out of the 52 test

problems (52%) in Table III, but only for six test problems (12%) in Table IV. In Table IV, NSGA-III looks the best. The best average results are obtained by NSGA-III for 19 test problems (37%). The relative performance of MOEA/D-WS is much better in Table IV than in Table III. An interesting observation is that the performance of NSGA-II is not always bad for many-objective test problems in Table IV.

The difference in the test problems between Tables III and IV is only the multiplication of (-1) to each objective. Except for this change, the test problems are the same between the two tables. However, totally different results are obtained. Especially, the performance of the best algorithms in Table III (i.e., θ -DEA and MOEA/DD) is deteriorated in Table IV.

In Table V, we show experimental results evaluated by the hypervolume with the reference point $(2, 2, \dots, 2)$. Totally different results are obtained between Tables III (on DTLZ and WFG) and V (on $DTLZ^{-1}$ and WFG^{-1}) with respect to algorithm comparison. That is, good results are obtained by the last four algorithms in Table V for $DTLZ^{-1}$ and WFG^{-1} while good results are obtained by the first four algorithms in Table III for DTLZ and WFG. An interesting observation is that better results in Table V are obtained from NSGA-II than NSGA-III for $WFG4-9^{-1}$ while better results are obtained from NSGA-III than NSGA-II for $WFG4-9$ in Table III.

Performance evaluation results by the IGD indicator are shown in Table VI for DTLZ and WFG and Table VII for $DTLZ^{-1}$ and WFG^{-1} . We can obtain similar observations from the IGD-based comparison results and the hypervolume-based comparison results. For example, the best results for almost all DTLZ and WFG problems are obtained from the first four algorithms in Tables III and VI whereas the best results for almost all $DTLZ^{-1}$ and WFG^{-1} problems are obtained from the last four algorithms in Tables V and VII. We can also observe very bad results (i.e., very large average IGD values) of NSGA-II on $DTLZ1-4$ in Table VI.

V. DISCUSSION AND FUTURE RESEARCH TOPICS

A. Discussion on Experimental Results

For further discussing the experimental results in Tables III–VII, we show a single run result of each algorithm on the three-objective $DTLZ1^{-1}$ problem in Fig. 11. Among 101 runs, we select a single run with the median hypervolume value in Table IV to show a typical result. Fig. 11 shows all solutions in the final generation of the selected run. In Fig. 11, three values of θ are examined for MOEA/D-IPBI ($\theta = 0.1$ was used in Tables III–VII based on the reported results in [45]).

Well-distributed solutions are obtained by MOEA/D-IPBI in Fig. 11(i) and (j). Such a good solution set is not obtained by the other algorithms. In Fig. 11(d), ten solutions are obtained inside the Pareto front together with many solutions on its boundary. This result is explained by the inconsistency between the shape of the Pareto front and the shape of the distribution of the weight vectors in Fig. 12. The shaded region is the projection of the Pareto front, which is the region of weight vectors intersecting with the Pareto front.

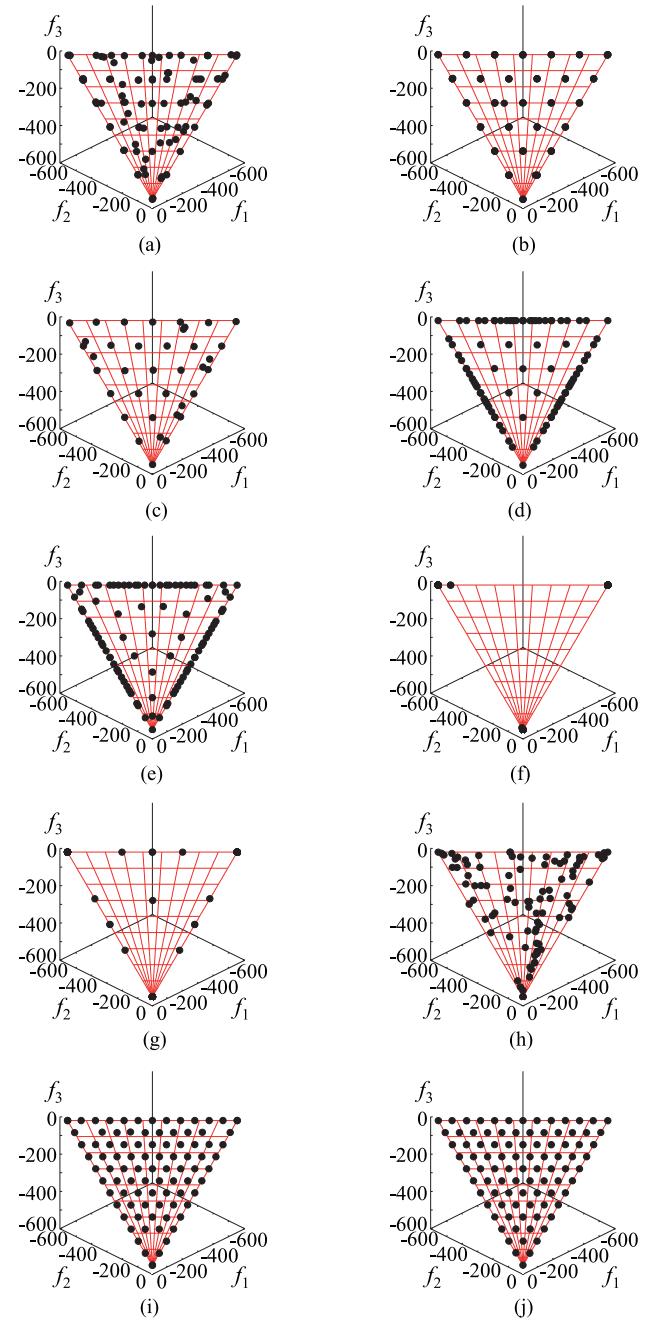


Fig. 11. Experimental results of a single run of each algorithm on the three-objective $DTLZ1^{-1}$ problem. (a) NSGA-III. (b) θ -DEA. (c) MOEA/DD. (d) MOEA/D-PBI. (e) MOEA/D-Tch. (f) MOEA/D-WS. (g) MOEA/D-IPBI ($\theta = 0.1$). (h) NSGA-II. (i) MOEA/D-IPBI ($\theta = 1$). (j) MOEA/D-IPBI ($\theta = 5$).

Almost the same figure as Fig. 12 was used in [44] for explaining experimental results of NSGA-III on the inverted DTLZ1 problem. As shown in Fig. 12 (and in [44]), the weight vectors are uniformly distributed over the triangle whereas the shape of the Pareto front is a rotated triangle. We can see from Fig. 12 that the ten weight vectors are inside the projection of the Pareto front. Those weight vectors correspond to the ten inside solutions in Fig. 11(d). Since each weight vector in MOEA/D-PBI always has a single solution, many solutions are obtained on the boundary of the

TABLE V
AVERAGE HYPERVOLUME VALUES OVER 101 RUNS ON OUR MINUS TEST PROBLEMS FOR THE REFERENCE POINT $(2, 2, \dots, 2)$.
THE BEST AVERAGE RESULT FOR EACH TEST PROBLEM IS HIGHLIGHTED BY BOLD AND UNDERLINED. THE WORST
FOUR AVERAGE RESULTS FOR EACH TEST PROBLEM ARE SHADED

Problem	M	NSGA-III	θ -DEA	MOEA/DD	PBI	Tch	WS	IPBI	NSGA-II
DTLZ1 ⁻¹	3	5.41176	5.34726	5.29282	5.49085	5.51609	4.08046	5.13110	5.35306
	5	9.11119	6.95719	7.49073	8.62126	10.84719	6.31454	8.87033	10.07195
	8	7.37671	6.33654	6.66479	6.50813	21.38202	9.55525	18.07406	16.54193
	10	7.86919	4.18089	6.96420	6.57523	31.87081	12.43649	26.69442	23.85618
DTLZ2 ⁻¹	3	6.62189	6.56659	6.55971	6.62696	6.66773	6.72553	6.72544	6.61258
	5	15.89627	15.08368	14.28746	15.37201	17.44268	17.75316	17.74782	16.60895
	8	27.65750	19.54767	20.73854	19.99574	40.10508	44.32023	44.29348	41.84263
	10	46.12984	33.41141	26.99350	27.95097	64.63141	76.26890	76.23983	78.06037
DTLZ3 ⁻¹	3	6.65425	6.59691	6.59933	6.64415	6.67007	6.72555	6.72553	6.61990
	5	15.40685	15.02431	14.35655	15.36004	17.42006	17.74347	17.74202	16.50106
	8	26.92151	20.99691	20.46621	19.75676	40.17889	44.30160	44.26602	39.90168
	10	43.42177	34.58500	26.59275	27.69662	64.75250	76.25468	76.24625	75.01578
DTLZ4 ⁻¹	3	6.66826	6.62146	6.60293	6.14964	6.64071	6.72555	6.43316	6.62453
	5	16.37175	15.12450	14.32240	13.73034	17.39446	17.74047	17.22725	16.68285
	8	35.59274	22.81013	17.07119	16.02484	39.99005	44.26434	39.83043	41.78905
	10	65.11754	43.23888	24.22737	21.92158	63.38510	76.25646	75.37652	77.34646
WFG1 ⁻¹	3	3.94840	3.49316	3.23797	2.61888	3.65774	3.41223	3.67012	4.48059
	5	5.90557	5.52515	3.28134	2.45346	6.03186	5.07280	5.31238	7.51392
	8	5.41109	5.53844	3.57939	2.00473	9.53704	8.07837	7.81115	10.76195
	10	5.51259	5.25281	3.30902	2.01233	13.10962	10.02642	10.29219	13.68541
WFG2 ⁻¹	3	6.12225	6.12167	6.10910	6.11625	6.13610	5.74837	5.98037	6.07935
	5	10.77797	10.28187	9.64829	7.10282	11.14371	9.43515	9.54685	10.78963
	8	10.30634	7.78976	9.05292	6.83792	17.98575	15.06020	15.20026	16.56406
	10	10.45323	5.90525	9.47472	7.32244	23.59947	18.80022	19.09314	21.44471
WFG3 ⁻¹	3	5.32265	5.34031	5.12519	5.37874	5.47014	4.01439	4.77819	5.32966
	5	9.40589	6.95570	6.43180	6.11943	10.70830	6.02464	7.87696	9.85104
	8	8.28285	4.01439	6.22156	6.09463	18.74847	9.01976	11.19079	14.92069
	10	8.38143	3.84791	6.43680	6.51623	26.57833	11.04830	14.65720	21.15174
WFG4 ⁻¹	3	6.46868	6.53600	6.38145	6.47756	6.62071	6.67954	6.69466	6.56320
	5	15.71312	15.26109	13.06622	13.72311	16.37306	16.74087	17.21992	16.17657
	8	30.42847	24.08956	15.29133	15.46200	36.01661	38.72524	40.05739	38.62815
	10	48.82166	24.32377	17.64685	19.26735	57.13700	64.73160	67.03299	68.40053
WFG5 ⁻¹	3	6.50169	6.55414	6.43149	6.47133	6.61413	6.67041	6.68557	6.58371
	5	16.16456	14.98968	13.91439	13.83229	16.40218	16.74631	17.22362	16.59823
	8	29.21383	22.73550	21.44486	16.25172	36.25025	38.90775	40.25929	39.53740
	10	47.57705	23.77569	29.23595	19.86993	57.69460	65.02107	67.23864	69.08351
WFG6 ⁻¹	3	6.56605	6.58471	6.45193	6.46840	6.62128	6.67989	6.69498	6.61808
	5	16.19334	14.99690	14.20128	13.88373	16.34692	16.74439	17.22435	16.79942
	8	28.23951	21.00836	22.84306	16.15885	35.61979	38.72410	40.08199	38.95506
	10	42.60624	22.50419	30.59149	19.77665	56.07817	64.76615	67.00523	69.90950
WFG7 ⁻¹	3	6.44984	6.55400	6.36083	6.46137	6.58880	6.67973	6.69480	6.52938
	5	16.29530	14.92831	13.76747	13.70136	16.30640	16.74451	17.22413	16.18200
	8	26.93878	20.60801	18.57115	15.12162	35.93195	38.70618	40.07723	39.29581
	10	39.54707	22.90411	23.95598	18.53367	57.36886	64.62800	66.92247	73.20667
WFG8 ⁻¹	3	6.58684	6.58269	6.48000	6.46767	6.62155	6.67943	6.69468	6.63291
	5	16.36596	14.95199	14.76723	13.91851	16.32779	16.73741	17.22083	17.03252
	8	28.25011	21.77714	25.62944	16.55536	35.58745	38.71102	40.07927	41.34084
	10	43.36500	22.87174	35.96638	19.94470	56.04570	64.76123	67.10037	76.53494
WFG9 ⁻¹	3	6.51204	6.55986	6.35162	6.40979	6.54754	6.65257	6.66428	6.54007
	5	16.49599	15.06437	14.25769	13.75819	16.36125	16.75200	17.18507	16.49621
	8	29.77776	22.00054	22.88395	16.04369	37.01047	38.89802	40.06594	39.49413
	10	45.00021	24.22697	31.19761	19.58884	60.81918	65.23011	67.13225	74.06843

Pareto front in Fig. 11(d). Those boundary solutions are the best solutions for the outside weight vectors in Fig. 12. For the same reason, many solutions are obtained on the boundary in Fig. 11(e).

Multiobjective search in weight vector-based algorithms in [27]–[32] as well as MOEA/D-PBI and MOEA/D-Tch can be viewed as pulling all solutions toward the ideal point using the weight vectors. The weight vectors in those algorithms are

TABLE VI
AVERAGE IGD VALUES OVER 101 RUNS ON THE ORIGINAL TEST PROBLEMS. THE BEST AVERAGE RESULT FOR EACH TEST PROBLEM IS HIGHLIGHTED BY BOLD AND UNDERLINED. THE WORST FOUR AVERAGE RESULTS FOR EACH TEST PROBLEM ARE SHADED

Problem	M	NSGA-III	θ -DEA	MOEA/DD	PBI	Tch	WS	IPBI	NSGA-II
DTLZ1	3	0.04362	0.04170	<u>0.04138</u>	0.04175	0.06082	0.50173	0.42397	0.06481
	5	0.11308	0.11125	<u>0.11110</u>	0.11128	0.22189	0.73685	6.52117	19.87954
	8	0.17984	<u>0.17513</u>	0.17541	0.17601	0.23603	0.72480	0.52039	75.18619
	10	0.19094	<u>0.18527</u>	0.18552	0.18611	0.23786	0.78417	0.49928	77.22337
DTLZ2	3	<u>0.05799</u>	0.05804	0.05801	0.05800	0.07318	0.54279	0.54641	0.07182
	5	0.19403	<u>0.19363</u>	0.19368	0.19368	0.32648	0.69062	0.93890	0.31393
	8	0.40062	0.39802	0.39575	<u>0.39572</u>	0.46026	0.94291	0.99204	1.90946
	10	0.46752	0.46462	0.46145	<u>0.46120</u>	0.53319	1.00370	1.05344	2.15108
DTLZ3	3	0.06261	0.05908	<u>0.05824</u>	0.05848	0.07349	0.54419	0.54800	0.07194
	5	0.19601	0.19496	<u>0.19384</u>	0.19400	0.32551	0.70566	40.98681	116.19480
	8	0.41225	0.40224	<u>0.39694</u>	0.46660	0.47438	0.94647	1.23378	348.09573
	10	0.46843	0.46545	<u>0.46165</u>	0.46321	0.53973	1.01331	1.12693	308.79409
DTLZ4	3	0.07550	0.10791	<u>0.05800</u>	0.45495	0.47158	0.83789	0.71489	0.07012
	5	0.19378	0.19373	<u>0.19372</u>	0.33507	0.45264	0.82880	0.89434	0.22875
	8	0.39672	0.39597	<u>0.39534</u>	0.53322	0.64479	0.95178	1.00074	2.11783
	10	0.46302	0.46191	<u>0.46074</u>	0.56608	0.61814	0.99026	1.05641	2.33543
WFG1	3	0.21258	0.18074	0.18377	0.20233	<u>0.07600</u>	0.20087	0.15597	0.16604
	5	0.29117	0.20606	0.17134	0.19663	<u>0.08683</u>	0.18288	0.18297	0.26815
	8	0.16839	0.07692	<u>0.06678</u>	0.08509	0.08045	0.10808	0.12427	0.33417
	10	0.08868	0.09112	<u>0.07619</u>	0.14610	0.10095	0.10556	0.11972	0.23599
WFG2	3	0.04072	<u>0.03577</u>	0.04866	0.08872	0.08739	0.17910	0.12579	0.05805
	5	0.05691	<u>0.05685</u>	0.08325	0.10423	0.15136	0.21243	0.20765	0.12767
	8	<u>0.07015</u>	0.08495	0.09183	0.09860	0.11937	0.13764	0.13030	0.19386
	10	<u>0.05969</u>	0.08920	0.09114	0.09578	0.11840	0.13169	0.12416	0.19704
WFG3	3	0.15399	0.28832	0.05425	<u>0.03745</u>	0.04070	0.20844	0.19232	0.05006
	5	0.09697	0.12176	0.12018	<u>0.08618</u>	0.15235	0.34998	0.28723	0.10195
	8	0.23351	0.56029	<u>0.14305</u>	0.22451	0.33536	0.56095	0.43524	0.15998
	10	0.16754	0.41979	<u>0.15640</u>	0.31725	0.39634	0.57148	0.55067	0.16206
WFG4	3	<u>0.05818</u>	0.05823	0.07217	0.07700	0.09484	0.52334	0.25250	0.07274
	5	0.19213	0.19223	0.26733	0.30864	0.41147	0.63375	0.42761	<u>0.18244</u>
	8	0.39954	0.39905	0.51790	0.72445	0.51843	0.85709	0.59237	<u>0.37909</u>
	10	0.46687	0.46624	0.66822	0.84257	0.58032	0.92412	0.70445	<u>0.45848</u>
WFG5	3	0.06216	<u>0.06212</u>	0.07543	0.07569	0.10004	0.52875	0.24320	0.07718
	5	0.18937	0.18935	0.25529	0.29036	0.40381	0.65914	0.41589	<u>0.18139</u>
	8	0.39141	0.39123	0.51273	0.67067	0.51038	0.81440	0.48871	<u>0.36793</u>
	10	0.45671	<u>0.45638</u>	0.65521	0.80237	0.56802	0.88882	0.55651	0.45670
WFG6	3	0.06237	<u>0.06236</u>	0.07542	0.08158	0.09964	0.53091	0.24512	0.08111
	5	<u>0.18939</u>	0.18942	0.26168	0.32816	0.40693	0.67423	0.41625	0.19635
	8	0.39279	<u>0.39211</u>	0.52623	0.84861	0.52593	0.92164	0.70887	0.40164
	10	0.45856	<u>0.45750</u>	0.66364	0.95099	0.57914	0.97505	0.81882	0.46819
WFG7	3	0.05858	<u>0.05843</u>	0.07272	0.10435	0.09461	0.53919	0.25365	0.07482
	5	<u>0.19302</u>	0.19308	0.26131	0.34346	0.40967	0.67685	0.42667	0.22350
	8	0.39970	<u>0.39841</u>	0.50986	0.81487	0.52613	0.92975	0.61293	0.43800
	10	0.46668	<u>0.46543</u>	0.63276	0.94246	0.59069	0.97643	0.65444	0.49155
WFG8	3	0.06858	<u>0.06826</u>	0.07974	0.08798	0.10758	0.53692	0.50862	0.09200
	5	0.19572	<u>0.19568</u>	0.27004	0.31288	0.51613	0.70712	0.51826	0.21824
	8	0.41691	<u>0.41495</u>	0.49936	0.80811	0.54876	0.92428	0.79070	0.43170
	10	0.50584	0.49280	0.64259	0.92544	0.62707	1.00382	0.86101	<u>0.48245</u>
WFG9	3	0.06403	<u>0.06323</u>	0.07385	0.10025	0.09920	0.50142	0.26204	0.08311
	5	<u>0.18615</u>	0.18634	0.24683	0.29613	0.47733	0.66154	0.44104	0.21086
	8	0.39688	<u>0.39539</u>	0.51814	0.71655	0.53759	0.85700	0.67375	0.45885
	10	0.46273	<u>0.46209</u>	0.66553	0.83358	0.60033	0.92832	0.73585	0.50534

illustrated in Fig. 13(a). Figs. 12 and 13(a) show the same weight vectors. The ten inside solutions in Fig. 11(d) are obtained by the ten inside weight vectors in Fig. 12. Almost the same ten inside solutions are obtained in Fig. 11(a)–(d) since all algorithms in Fig. 11(a)–(d) have the same basic idea of

multiobjective search: pulling all solutions toward the ideal point using the weight vectors in Fig. 13(a).

When IPBI is used, multiobjective search is performed by pushing all solutions from the nadir point to the Pareto front as shown in Fig. 13(b). In this case, the distribution of weight

TABLE VII

AVERAGE IGD VALUES OVER 101 RUNS ON OUR MINUS TEST PROBLEMS. THE BEST AVERAGE RESULT FOR EACH TEST PROBLEM IS HIGHLIGHTED BY BOLD AND UNDERLINED. THE WORST FOUR AVERAGE RESULTS FOR EACH TEST PROBLEM ARE SHADED

Problem	M	NSGA-III	θ -DEA	MOEA/DD	PBI	Tch	WS	IPBI	NSGA-II
DTLZ1 ⁻¹	3	0.06023	0.08080	0.07764	0.07235	0.06726	0.46615	0.15033	0.05772
	5	0.15781	0.21539	0.18317	0.13134	0.17583	0.58701	0.24709	0.12841
	8	0.19939	0.22664	0.28573	0.42514	0.27490	0.67675	0.28280	0.21727
	10	0.19114	0.25461	0.28540	0.42793	0.30308	0.68698	0.30610	0.22753
DTLZ2 ⁻¹	3	0.06849	0.07061	0.07231	0.06733	0.08081	0.05795	0.05797	0.07106
	5	0.20140	0.22591	0.26158	0.20294	0.19062	0.19319	0.19338	0.17835
	8	0.39590	0.44648	0.44169	0.38813	0.46403	0.39536	0.39528	0.34390
	10	0.41607	0.45647	0.50964	0.44616	0.55239	0.46082	0.46084	0.38069
DTLZ3 ⁻¹	3	0.06945	0.06923	0.07147	0.06640	0.08231	0.05799	0.05799	0.07117
	5	0.20451	0.22735	0.26960	0.20317	0.19464	0.19361	0.19366	0.18317
	8	0.39347	0.43321	0.44242	0.38697	0.46253	0.39517	0.39519	0.34945
	10	0.41589	0.45076	0.50678	0.44444	0.55227	0.46063	0.46065	0.38427
DTLZ4 ⁻¹	3	0.06933	0.06795	0.07172	0.14957	0.08734	0.05800	0.10622	0.07001
	5	0.21479	0.24070	0.27921	0.27387	0.19831	0.19371	0.21271	0.17809
	8	0.36310	0.42714	0.45958	0.52122	0.46517	0.39528	0.43285	0.35118
	10	0.39219	0.43337	0.48691	0.52423	0.55814	0.46055	0.46365	0.39096
WFG1 ⁻¹	3	0.05290	0.10325	0.11665	0.29294	0.15693	0.37597	0.29304	0.04386
	5	0.11311	0.19147	0.32509	0.55926	0.17943	0.35539	0.31550	0.06629
	8	0.26898	0.34785	0.39702	0.99344	0.16412	0.32690	0.31295	0.10819
	10	0.29323	0.38999	0.43426	0.96296	0.14229	0.33064	0.31388	0.09226
WFG2 ⁻¹	3	0.04190	0.04306	0.05567	0.05739	0.04694	0.33038	0.22170	0.06632
	5	0.07881	0.14449	0.18688	0.24328	0.07440	0.37314	0.32020	0.09517
	8	0.20953	0.35963	0.33890	0.45393	0.11549	0.42160	0.38028	0.15819
	10	0.25544	0.43200	0.36370	0.50669	0.12588	0.46156	0.40894	0.15103
WFG3 ⁻¹	3	0.06083	0.08114	0.08829	0.08214	0.08449	0.49054	0.24954	0.05986
	5	0.16490	0.21009	0.26876	0.34333	0.19835	0.62850	0.37408	0.13927
	8	0.25089	0.31689	0.44622	0.47588	0.34387	0.72693	0.51896	0.25419
	10	0.25988	0.29982	0.46272	0.49450	0.36499	0.75953	0.50412	0.25219
WFG4 ⁻¹	3	0.07067	0.06850	0.08952	0.08597	0.09036	0.07125	0.06708	0.07162
	5	0.20365	0.21184	0.27358	0.25924	0.27079	0.25575	0.22894	0.18955
	8	0.38615	0.41425	0.63248	0.60677	0.48890	0.42937	0.41881	0.37240
	10	0.43182	0.49779	0.72355	0.66994	0.55916	0.47924	0.47420	0.41505
WFG5 ⁻¹	3	0.07096	0.06901	0.08740	0.08494	0.08793	0.07083	0.06681	0.07152
	5	0.20813	0.23067	0.24814	0.25227	0.26722	0.25228	0.22598	0.18380
	8	0.39476	0.42790	0.52843	0.58467	0.48796	0.42769	0.41669	0.37803
	10	0.44100	0.49854	0.61452	0.65891	0.55862	0.47784	0.47400	0.42550
WFG6 ⁻¹	3	0.07064	0.06940	0.08761	0.08656	0.09107	0.07121	0.06708	0.07144
	5	0.20879	0.23472	0.23969	0.25092	0.27386	0.25563	0.22895	0.18108
	8	0.39207	0.44079	0.51870	0.58752	0.49671	0.42929	0.41820	0.38577
	10	0.43261	0.50390	0.61078	0.66057	0.56964	0.47855	0.47419	0.42065
WFG7 ⁻¹	3	0.07491	0.06984	0.09135	0.08764	0.08919	0.07122	0.06709	0.07665
	5	0.21990	0.24024	0.26035	0.25841	0.26725	0.25555	0.22880	0.19349
	8	0.39812	0.44856	0.57769	0.61427	0.48629	0.42953	0.41873	0.36740
	10	0.43765	0.50225	0.65658	0.68282	0.55764	0.48052	0.47575	0.39785
WFG8 ⁻¹	3	0.07182	0.07039	0.08438	0.08642	0.09138	0.07125	0.06708	0.07267
	5	0.21132	0.23438	0.22655	0.24985	0.27496	0.25585	0.22920	0.18775
	8	0.39399	0.44081	0.49022	0.57774	0.49801	0.42906	0.41764	0.37975
	10	0.43266	0.50071	0.57353	0.65643	0.57134	0.47790	0.47233	0.40987
WFG9 ⁻¹	3	0.06858	0.06769	0.08732	0.08791	0.08518	0.07062	0.06719	0.07407
	5	0.20468	0.23095	0.23795	0.25551	0.25081	0.24740	0.22190	0.19209
	8	0.39243	0.43482	0.51895	0.59280	0.46407	0.42810	0.41925	0.37479
	10	0.44781	0.49704	0.60650	0.66438	0.52325	0.47860	0.47591	0.40130

vectors (i.e., reference lines) is consistent with the shape of the Pareto front. This explains why well-distributed solutions are obtained by MOEA/D-IPBI in Fig. 11(i) and (j).

In NSGA-III, θ -DEA, and MOEA/DD, each solution is assigned to its nearest reference line (whereas each reference

line has its best solution in MOEA/D). If there is no solution close to a reference line, no solution is assigned. In Fig. 12, all weight vectors outside the shaded region have no solution. As a result, many solutions are not obtained on the boundary of the Pareto front by NSGA-III, θ -DEA, and MOEA/DD.

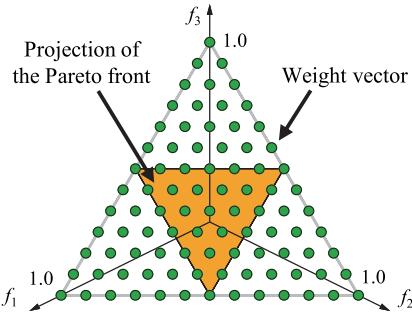


Fig. 12. Relation between a set of weight vectors and the Pareto front.

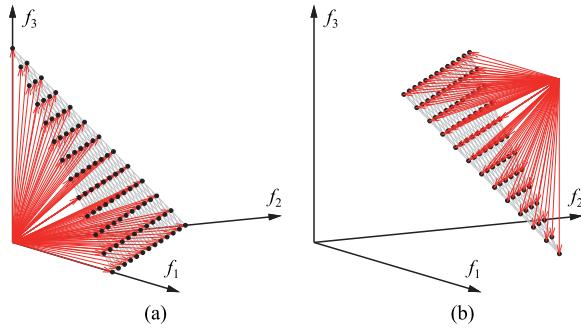


Fig. 13. Weight vectors used for three-objective minimization. (a) Most weight vector-based algorithms. (b) Weighted sum and IPBI.

In these algorithms, a single reference line can have multiple solutions. This is the reason why more inside solutions are obtained in Fig. 11(a)–(c) than in Fig. 11(d). In NSGA-III, the second solution for each reference line is selected randomly from the assigned solutions with the best rank. As a result, solution distribution looks somewhat random in Fig. 11(a). Since the second solution for each reference line in θ -DEA is the second best solution with respect to PBI, all solutions assigned to the same reference line are almost the same as shown in Fig. 11(b). Similar solution sets are obtained in Fig. 11(b) and (c) since θ -DEA and MOEA/DD have similar mechanisms to choose the second best solution for each reference line.

In the same manner as Fig. 11, we show an experimental result of a single run of each algorithm on the three-objective DTLZ2⁻¹ problem in Fig. 14. Good results are obtained by MOEA/D-WS and MOEA/D-IPBI in Fig. 14. One difference between Figs. 11 and 14 is the results by MOEA/D-WS and MOEA/D-IPBI ($\theta = 0.1$). Since the Pareto front of DTLZ1⁻¹ is a plane, well-distributed solutions are not obtained in Fig. 11 by MOEA/D-WS and MOEA/D-IPBI ($\theta = 0.1$). However, good results are obtained by these algorithms in Fig. 14 since the Pareto front of DTLZ2⁻¹ is convex. It should be noted that the weighted sum and the IPBI function with $\theta = 0.1$ have similar contour lines [45].

As in Fig. 11, the distribution of the weight vectors is not consistent with the shape of the Pareto front in Fig. 14 except for MOEA/D-WS and MOEA/D-IPBI. Thus, many solutions are obtained on the boundary of the Pareto front in Fig. 14(d) and (e). However, the distribution of solutions around the center of the Pareto front is similar between two

groups. One is Fig. 14(b)–(d) with the inconsistent weight vector distribution, and the other is Fig. 14(f), (g), (i), and (j) with the consistent weight vector distribution. This can be explained by the convexity of the Pareto front of DTLZ2⁻¹. As shown in Fig. 6, when the Pareto front is convex, more solutions are obtained around its center by pulling solutions toward the ideal point. In this case, less solutions are obtained around the center by pushing solutions from the nadir point as shown in Fig. 14(f), (g), (i), and (j). As a result of these two effects (i.e., the inconsistency between the weight vector distribution and the Pareto front shape, and the Pareto front convexity), the distribution of solutions around the center of the Pareto front is similar between Fig. 14(b)–(d) and Fig. 14(f), (g), (i), and (j).

Thanks to random selection of the second solution for each reference point in NSGA-III, more solutions are obtained around the center of the Pareto front in Fig. 14(a) than the other algorithms in Fig. 14. This observation in Fig. 14(a) explains why good average results are obtained by NSGA-III in Table IV where the hypervolume is measured from the reference point (1.1, 1.1, ..., 1.1).

Hypervolume contributions of solutions around the center of the Pareto front are relatively large when the reference point is close to the Pareto front. By moving a reference point away from the Pareto front, their contributions become relatively small since contributions of solutions on the boundary of the Pareto front increase. This explains why the evaluation results of NSGA-III are not good in Table V with the reference point (2, 2, ..., 2). The difference in the evaluation results of NSGA-III between Tables IV and V suggests that many solutions around the center of the Pareto front are obtained by NSGA-III as shown in Figs. 11(a) and 14(a).

We also show an experimental result of a single run of each algorithm on the ten-objective DTLZ2⁻¹ and WFG9⁻¹ problems in Figs. 15 and 16, respectively. In Fig. 15, a wide variety of solutions are not obtained by the first four algorithms. This observation explains why those algorithms have low evaluation results on the ten-objective DTLZ2⁻¹ problem in Table V with the reference point (2, 2, ..., 2) for hypervolume calculation whereas some of them have high evaluation results in Table IV with the reference point (1.1, 1.1, ..., 1.1). This is also the case in Fig. 16. Thanks to a large diversity of solutions in Figs. 15(h) and 16(h), NSGA-II has the best results on the ten-objective DTLZ2⁻¹ and WFG9⁻¹ problems in Table V.

A large diversity of obtained solutions by NSGA-II in Figs. 15(h) and 16(h) also explains good evaluation results by the IGD indicator in Table VII. Fig. 16(h) also shows low convergence ability of NSGA-II. In Fig. 16, the Pareto front satisfies $-2i - 1 \leq f_i(x) \leq -1$ for $i = 1, 2, \dots, 10$. However, some solutions in Fig. 16(h) have objective values close to 0.

B. Future Research Topics: Test Problems

As reported in [27]–[32] on evolutionary many-objective optimization, very good results are obtained by weight vector-based algorithms such as NSGA-III, θ -DEA, and MOEA/DD on the DTLZ1-4 and WFG4-9 problems in Table III. This is because the shapes of the Pareto fronts of these test problems are consistent with the shape of the distribution of

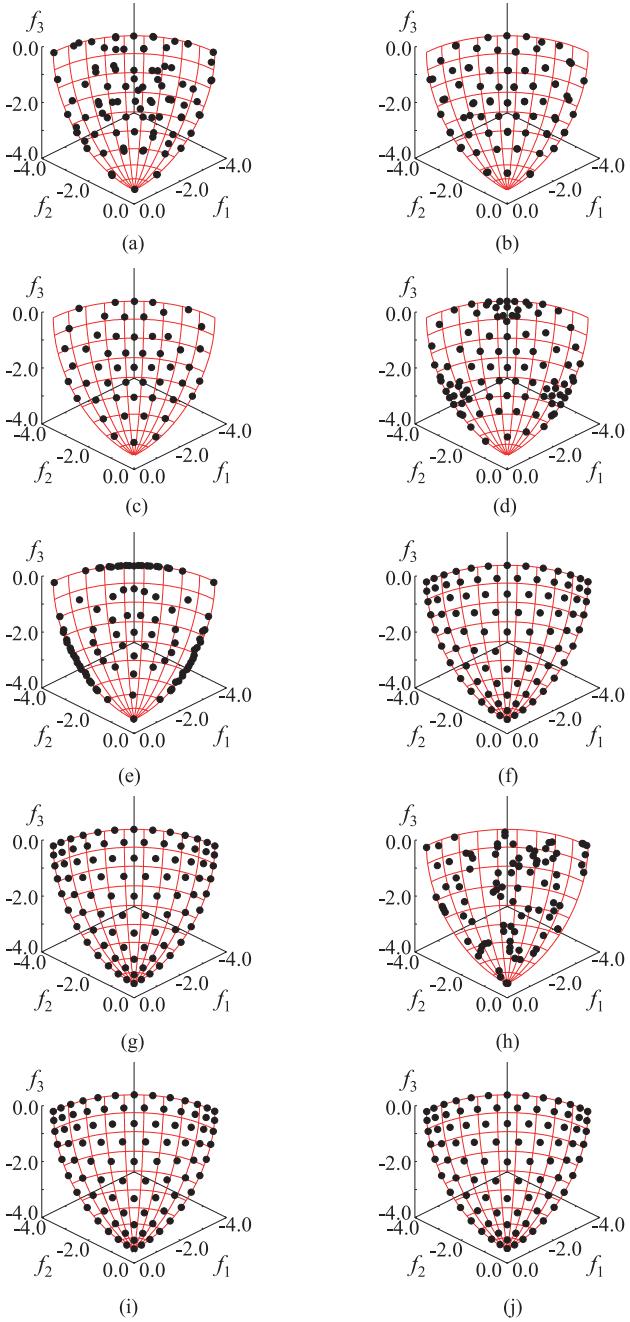


Fig. 14. Experimental results on the three-objective DTLZ2⁻¹ test problem. (a) NSGA-III. (b) θ -DEA. (c) MOEA/DD. (d) MOEA/D-PBI. (e) MOEA/D-Tch. (f) MOEA/D-WS. (g) MOEA/D-IPBI ($\theta = 0.1$). (h) NSGA-II. (i) MOEA/D-IPBI ($\theta = 1$). (j) MOEA/D-IPBI ($\theta = 5$).

the weight vectors in these algorithms. The DTLZ1-4 and WFG4-9 problems have the following Pareto fronts in the normalized objective space:

$$\sum_{i=1}^M (y_i^*)^k = 1 \text{ and } y_i^* \geq 0 \text{ for } i = 1, 2, \dots, M \quad (24)$$

where $k = 1$ (DTLZ1) and $k = 2$ (DTLZ2-4 and WFG4-9).

One question for future research is the similarity of these test problems to real-world many-objective problems. In other

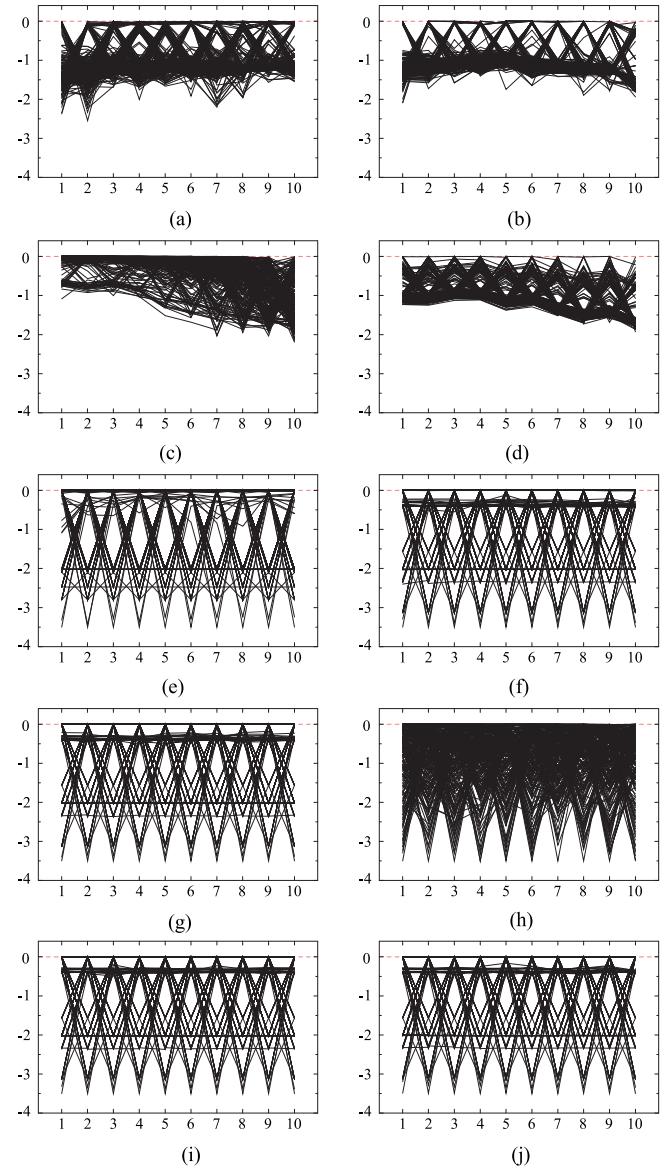


Fig. 15. Experimental results on the ten-objective DTLZ2⁻¹ test problem. (a) NSGA-III. (b) θ -DEA. (c) MOEA/DD. (d) MOEA/D-PBI. (e) MOEA/D-Tch. (f) MOEA/D-WS. (g) MOEA/D-IPBI ($\theta = 0.1$). (h) NSGA-II. (i) MOEA/D-IPBI ($\theta = 1$). (j) MOEA/D-IPBI ($\theta = 5$).

words, the question is their generality as test problems. We need much more research to answer this question. However, we can say that the Pareto front in (24) is very special in the following sense: an arbitrarily selected $(M-1)$ objectives can be simultaneously optimized. For example, the Pareto front in (24) includes a Pareto optimal solution $(1, 0, 0, \dots, 0)$ where all objectives except for the first one simultaneously have their individual optimal values. This means that an $(M-1)$ -objective problem created by removing an arbitrarily selected single objective from any of the DTLZ1-4 and WFG4-9 problems always has a single absolutely optimal solution $(0, 0, \dots, 0)$. That is, the ideal point of the $(M-1)$ -objective problem is a feasible solution, which dominates all the other feasible solutions. This feature sounds very strange since it is not likely that $(M-1)$ objectives

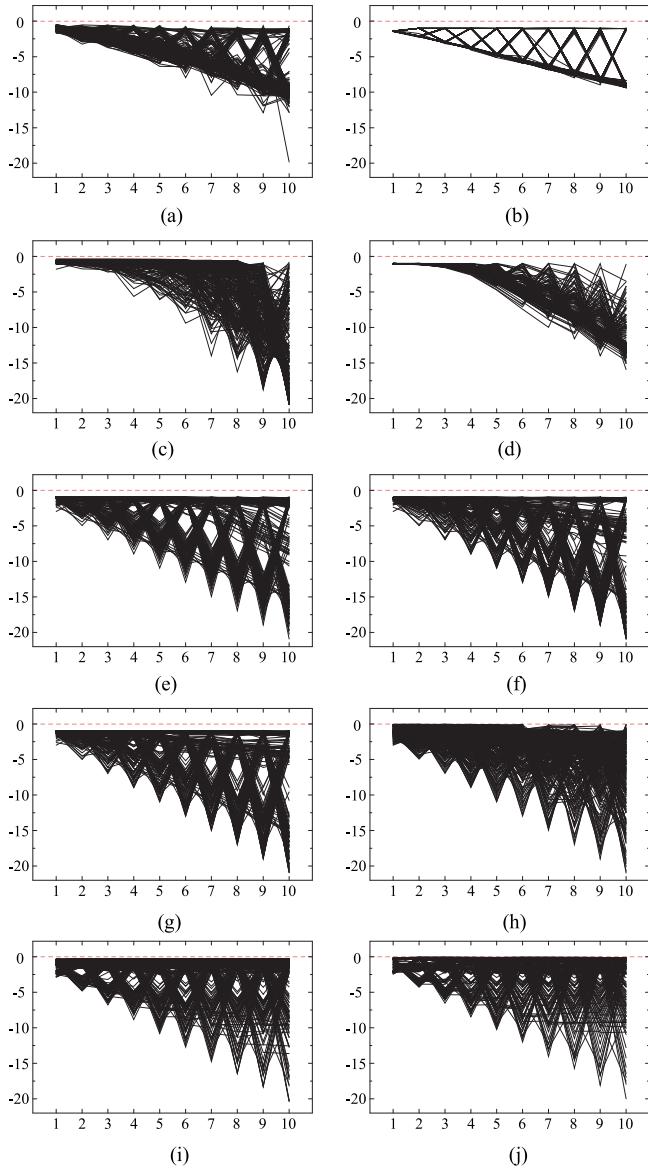


Fig. 16. Experimental results on the ten-objective WFG9⁻¹ test problem. (a) NSGA-III. (b) θ -DEA. (c) MOEA/DD. (d) MOEA/D-PBI. (e) MOEA/D-Tch. (f) MOEA/D-WS. (g) MOEA/D-IPBI ($\theta = 0.1$). (h) NSGA-II. (i) MOEA/D-IPBI ($\theta = 1$). (j) MOEA/D-IPBI ($\theta = 5$).

of a real-world M -objective problem can be simultaneously optimized with no conflict among them.

The minus versions of DTLZ1-4 and WFG4-9 do not have this strange feature. However, they may have some other strange features because their original versions have the above-mentioned feature. For example, in the DTLZ1-4⁻¹ problems, optimization of a single objective always leads to the worst values of all the other objectives. The Pareto optimal solution $(-3.5, 0, 0, \dots, 0)$ of DTLZ2⁻¹ is an example of such a single-objective optimization result where the optimization of the first objective leads to the worst values for all the other objectives.

Our experimental results show that the consistency between the shape of the Pareto front and the shape of the distribution of the weight vectors has a large effect on the performance of weight vector-based algorithms. This was demonstrated by

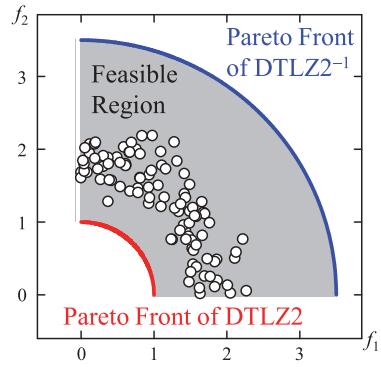


Fig. 17. Pareto fronts of the two-objective DTLZ2 and its maximization version Max-DTLZ2 which is equivalent to the two-objective DTLZ2⁻¹.

Jain and Deb [44] for NSGA-III on the three-objective and five-objective inverted DTLZ1 problems. Our experimental results also suggest that the size of the Pareto front may have a large effect on the performance of EMO algorithms on many-objective problems. In Fig. 17, we show the Pareto fronts of the two-objective DTLZ2 and DTLZ2⁻¹ problems together with their feasible region and randomly generated 100 initial solutions. In Fig. 17, DTLZ2⁻¹ is shown as the maximization problem of DTLZ2: Max-DTLZ2. Thus the two problems (DTLZ2 and its maximization version) have exactly the same feasible region. As we can see from Fig. 17, DTLZ2 has a small Pareto front in comparison with the spread of the randomly generated initial solutions. It is likely that a good solution set can be obtained for DTLZ2 by pulling those initial solutions toward the ideal point. That is, the convergence of solutions toward the Pareto front is important for DTLZ2.

However, in DTLZ2⁻¹ in Fig. 17, not only the convergence of solutions but also the diversification of solutions is very important for finding a set of well-distributed solutions over the entire Pareto front. Moreover, the improvement in the diversity of solutions in Fig. 17 has a positive effect on the convergence of solutions toward the Pareto front of DTLZ2⁻¹ (whereas it has a negative effect for DTLZ2). For example, the crowding distance-based fitness evaluation tends to increase the range of objective values for each objective in the population. In Fig. 17, the maximum range for each objective is obtained by the two extreme solutions on the edge of the Pareto front of DTLZ2⁻¹ [i.e., one extreme solution with the maximum value of $f_1(\mathbf{x})$ and the other with the maximum value of $f_2(\mathbf{x})$ of Max-DTLZ2]. These two extreme solutions will help the convergence of other solutions toward the Pareto front of DTLZ2⁻¹. However, they have a negative effect on the convergence of solutions toward the Pareto front of the original DTLZ2. The difference in the effects of the diversity improvement efforts on the convergence of solutions toward the Pareto front between DTLZ2 and DTLZ2⁻¹ in Fig. 17 explains the difference of the performance of NSGA-II between DTLZ1-4 and DTLZ1-4⁻¹ in Tables III–VII. NSGA-II is evaluated as the worst among the examined eight algorithms for DTLZ1-4 by the hypervolume in Table III and also by the IGD indicator in Table VI. However, it is evaluated as the best algorithm for DTLZ1-4⁻¹ by IGD in Table VII.

Let us briefly explain why good solutions can be obtained for many-objective DTLZ, WFG, DTLZ^{-1} , and WFG^{-1} test problems as shown in Figs. 5, 15, and 16. In all of the DTLZ1-4 and DTLZ1-4 $^{-1}$ problems and some of the WFG1-9 and WFG1-9 $^{-1}$ problems, decision variables are separable into distance variables and position variables. In those problems, any changes only in the distance variables simply modify only the distance of a solution from the Pareto front. When only position variables are changed, only the position is changed without changing the value of the distance function. For illustrating these properties of most test problems based on their separable decision variables, first we randomly generate five initial solutions. The generated five initial solutions are shown by open circles in Fig. 18. Then we generate a new solution by randomly changing the values of all distance variables of each initial solution (i.e., by replacing the current values of the distance variables with randomly generated values within their domains). This is iterated 100 times to generate 100 solutions for each initial solution. In this manner, we generate 500 solutions. Fig. 18 shows the generated solutions for the two-objective DTLZ1-4 and WFG1-9 problems [and DTLZ1 with five and ten objectives in Fig. 18(a2) and (a3)]. We also perform the same experiment by changing the values of all position variables (see Fig. 19).

In Fig. 18, the generated solutions from the same parent are on the same line except for WFG7 and WFG9. Since this also holds for many-objective problems from the DTLZ and WFG problem formulations, the generated solutions by changing distance variables can be compared by the Pareto dominance relation. This explains why good results are obtained by NSGA-II for some many-objective test problems. However, this is totally different from a general case of many-objective optimization where almost all solutions are nondominated with each other. Moreover, each test problem in DTLZ and WFG has only a single distance function independent of the number of objectives. By optimizing the single distance function, we can obtain Pareto optimal solutions. This means that the search for Pareto optimal solutions is single-objective optimization independent of the number of objectives.

Fig. 19 shows the generated solutions by changing position variables. Except for WFG2, generated solutions from the same solution have the same distance from the Pareto front. This also holds for many-objective problems from the DTLZ and WFG problem formulations. If a solution is Pareto optimal (i.e., if it is on the Pareto front), all solutions generated by changing position variables are also Pareto optimal. Thus, we can easily increase the diversity of solutions by changing position variables without deteriorating their convergence. From these discussions on Figs. 18 and 19, we can see that most test problems in DTLZ and WFG are not difficult as many-objective problems. This observation is consistent with the reported results by the weight vector-based algorithms in [27]–[32] where very good solutions were obtained (which are almost the same as a reference point set for the IGD calculation as shown in Fig. 5). This is also consistent with our experimental results by NSGA-II (e.g., the best results are obtained by NSGA-II for some of the WFG problems in Table VI with the IGD indicator).

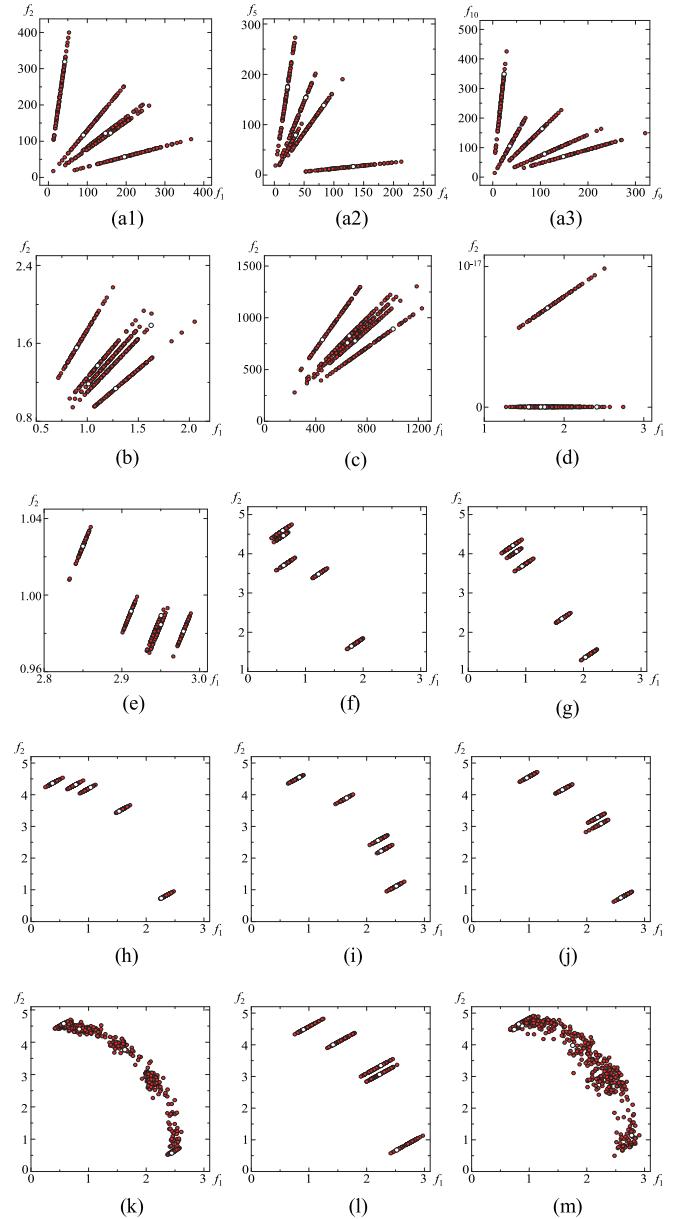


Fig. 18. Generated solutions by changing distance variables. (a1) 2-Obj. DTLZ1. (a2) 5-Obj. DTLZ1. (a3) 10-Obj. DTLZ1. (b) 2-Obj. DTLZ2. (c) 2-Obj. DTLZ3. (d) 2-Obj. DTLZ4. (e) 2-Obj. WFG1. (f) 2-Obj. WFG2. (g) 2-Obj. WFG3. (h) 2-Obj. WFG4. (i) 2-Obj. WFG5. (j) 2-Obj. WFG6. (k) 2-Obj. WFG7. (l) 2-Obj. WFG8. (m) 2-Obj. WFG9.

Our observation from Figs. 18 and 19 also explains the features of some weight vector-based algorithms. Since the Pareto dominance relation holds among solutions generated by changing distance variables, Pareto dominance-based fitness evaluation is used in some weight vector-based algorithms (whereas it does not work well on many-objective problems in general). For the same reason, the convergence improvement is not difficult. Thus a large penalty value (i.e., $\theta = 5$) is used in the PBI function even for many-objective problems (whereas the PBI function with a large penalty value does not work well on many-objective problems in general). These discussions suggest that the recent development of weight vector-based algorithms is overspecialized for DTLZ and WFG with respect

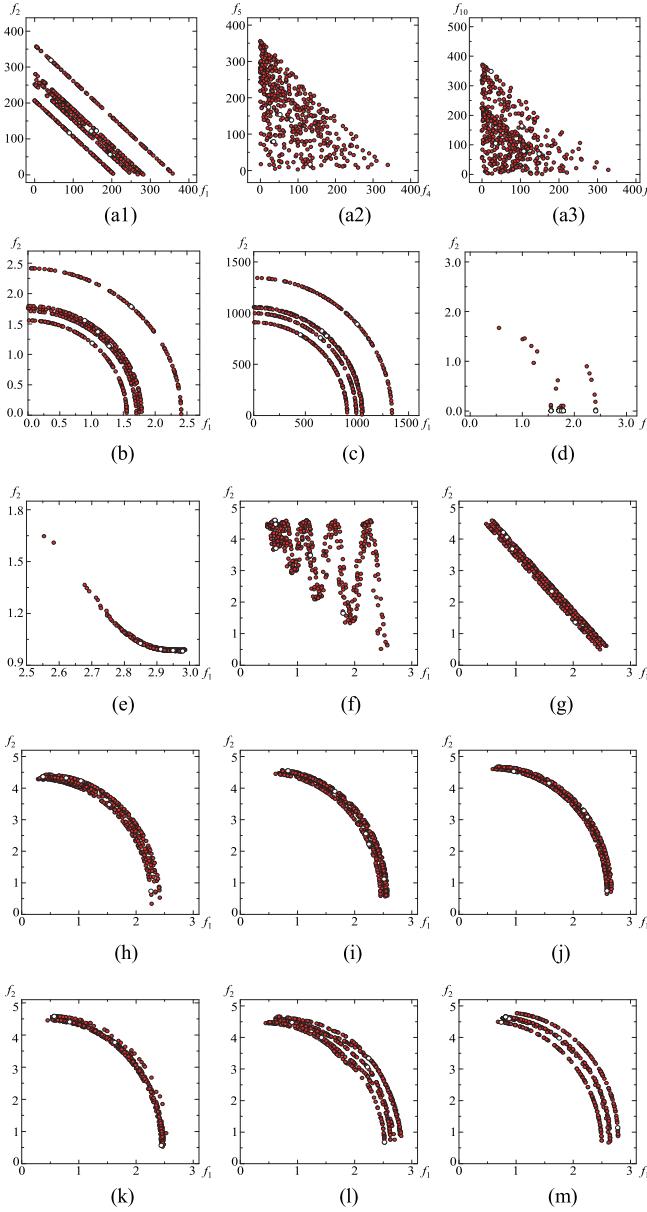


Fig. 19. Generated solutions by changing position variables. (a1) 2-Obj. DTLZ1. (a2) 5-Obj. DTLZ1. (a3) 10-Obj. DTLZ1. (b) 2-Obj. DTLZ2. (c) 2-Obj. DTLZ3. (d) 2-Obj. DTLZ4. (e) 2-Obj. WFG1. (f) 2-Obj. WFG2. (g) 2-Obj. WFG3. (h) 2-Obj. WFG4. (i) 2-Obj. WFG5. (j) 2-Obj. WFG6. (k) 2-Obj. WFG7. (l) 2-Obj. WFG8. (m) 2-Obj. WFG9.

to not only the shape of the distribution of the weight vectors but also their fitness evaluation mechanisms (in other words, their convergence-diversification balance).

As we have explained in this paper, the DTLZ and WFG problems are test problems with special characteristic features. Thus an important future research topic is the creation of a wide variety of test problems with respect to the shape of the Pareto front, the size of the Pareto front, the relation among decision variables, and the relation among objectives. The size of the Pareto front can be rephrased as the shape of the feasible region in the objective space.

C. Future Research Topics: Algorithm Developments

Our experimental results showed that good results were obtained when the shape of the distribution of weight vectors

is the same as or similar to the shape of the Pareto front. In general, we do not know the shape of the Pareto front. It is much more difficult to find the exact shape of the Pareto front of a many-objective problem than the case of two or three objectives. So it may be desirable for many-objective optimizers to have robust search ability with respect to the shape of the Pareto front. A simple idea is to simultaneously use multiple sets of weight vectors with different distributions. For example, it may be a good idea to simultaneously use both the PBI and IPBI functions in a single MOEA/D algorithm. In [50], the simultaneous use of the weighted sum and the weighted Tchebycheff was examined. Then it was shown that better results were obtained from their simultaneous use than their individual use (i.e., MOEA/D-WS and MOEA/D-Tch) on many-objective knapsack problems.

Another idea is the adaptation of the distribution of weight vectors to the shape of the Pareto front. This idea has been suggested for MOEA/D in $p\alpha$ -MOEA/D [51]. Adaptive NSGA-III (A-NSGA-III) [44] can be viewed as a representative algorithm in such a research direction. A-NSGA-III has simple mechanisms for weight vector deletion and creation. Its performance has not been evaluated on many-objective problems with eight or more objectives (i.e., many-objective problems for which the two-layered approach was used to generate weight vectors). Development of adaptive many-objective algorithms seems to be an interesting future research direction. However, before implementing an efficient and effective adaptation mechanism of the weight vector distribution, we may need to further examine the behavior of weight vector-based algorithms on a wide variety of many-objective test problems under various settings such as a large number of weight vectors together with a large population, a large number of weight vectors together with a small population, and a small number of weight vectors together with a large population. Analyzing multiobjective search behavior under these settings may help us to understand the necessity of weight vector deletion and creation.

VI. CONCLUSION

In this paper, we clearly showed the similarity between the shape of the distribution of the weight vectors in weight vector-based many-objective algorithms and the shape of the Pareto fronts of frequently used many-objective test problems (i.e., DTLZ1-4 and WFG4-9). For demonstrating the high sensitivity of their performance to the shape of the Pareto front, we formulated DTLZ⁻¹ and WFG⁻¹ problems by multiplying (-1) to each objective of the DTLZ and WFG problems. Whereas DTLZ1-4 and WFG4-9 have triangular shape Pareto fronts, DTLZ1-4⁻¹ and WFG4-9⁻¹ have rotated triangular shape Pareto fronts. We demonstrated through computational experiments that totally different performance comparison results among different algorithms were obtained from the original test problems and their minus versions. Whereas the best results were obtained by θ -DEA and MOEA/DD on the original DTLZ and WFG problems, they were outperformed by MOEA/D-WS and MOEA/D-IPBI on their minus versions with 3, 5, 8, and 10 objectives. Those high performance weight vector-based algorithms on

the original DTLZ and WFG problems were also outperformed by NSGA-II on some of the minus versions even when they have ten objectives.

One difficulty of recently developed weight vector-based algorithms is the lack of appropriate handling of reference lines outside the Pareto front (i.e., with no intersection with the Pareto front). This difficulty can be rephrased as the lack of appropriate criteria for choosing the second solution for each reference line. This difficulty was demonstrated by our experimental results on DTLZ⁻¹ and WFG⁻¹ where NSGA-III with random selection of the second solution outperformed θ -DEA and MOEA/DD with the selection of the second best solution. This difficulty is not clear when we apply them to frequently used many-objective test problems such as DTLZ1-4 and WFG4-9. This is because all reference lines are inside the Pareto fronts of those test problems thanks to their triangular shape Pareto fronts. In general, the above-mentioned difficulty is severe when many reference lines are outside the Pareto front. In this sense, many-objective test problems with degenerate Pareto fronts are difficult for weight vector-based algorithms. This discussion is consistent with Table III where the best results were obtained by NSGA-II for all test problems of WFG3 and also with Table VI where good results were not obtained by θ -DEA for any test problem of WFG3. Whereas WFG3 is not a degenerate test problem, the nondegenerate part of its Pareto front is very small (see [38] for the shape of the Pareto front of WFG3). Thus many weight vectors have no intersection with the Pareto front. The use of a wide variety of test problems with various Pareto front shapes will prevent the development of weight vector-based algorithms from being overspecialized for a special class of many-objective test problems such as DTLZ1-4 and WFG4-9. The use of a wide variety of test problems will encourage the development of appropriate handling mechanisms of reference lines outside the Pareto front including their online adaptation.

We also suggested that the DTLZ and WFG test problems are not difficult even when they have many objectives due to their special features in the test problem formulations. As a result, weight vector-based algorithms developed for those problems have fitness evaluation mechanisms which are not always suitable for many-objective optimization such as Pareto dominance and the emphasis on the minimization of the distance from the nearest reference line (e.g., the use of a large penalty value for the distance d_2 in the PBI function). These discussions also clearly show the necessity of a wide variety of many-objective test problems.

Our computational experiments explained the following reasons for high performance of recently proposed weight vector-based algorithms on many-objective DTLZ1-4 and WFG4-9 test problems.

- 1) Triangular shape Pareto front of each test problem, which is the same as or similar to the shape of the distribution of the weight vectors.
- 2) Relatively small size Pareto front of each test problem in comparison with the feasible region in the objective space, which is suitable for multiobjective search by pulling solutions toward the ideal point using the weight vectors.

- 3) Easy convergence and easy diversification of solutions due to separable decision variables, which make it possible to focus on the uniformity of obtained solutions over the Pareto front.

Thanks to these special features of frequently used many-objective test problems, uniformly distributed and well converged solutions can be obtained over the entire Pareto front even for the case of many objectives. However, these features are not likely to hold in real-world applications. Our computational experiments showed that high performance of recently proposed weight vector-based algorithms on many-objective test problems was deteriorated by changing the shape and the size of the Pareto front of each test problem. In the design of an MOEA/D-based algorithm for a many-objective problem at hand, weight vectors should be carefully specified based on the shape and the size of the Pareto front. A scalarizing function (i.e., a single-objective optimization problem) should be carefully defined for each weight vector depending on the difficulty of the convergence and the diversification in order to strike a good balance between them. However, these features (i.e., Pareto front shape, Pareto front size, convergence difficulty, and diversification difficulty) are usually unknown in real-world applications. Thus we may need an adaptation mechanism for the weight vectors and the scalarizing function. The framework of MOEA/D continues to be useful for developing such an adaptive many-objective algorithm. This is because MOEA/D has a large flexibility in weight vector specification and scalarizing function definition. We can use any set of weight vectors as well as any scalarizing function in MOEA/D-based many-objective algorithms. In an adaptive many-objective algorithm, the distribution of weight vectors will be adjusted to the shape and the size of the Pareto front, and the search mechanism for each weight vector will be adjusted by changing its scalarizing function depending on the difficulty of the convergence and the diversification.

REFERENCES

- [1] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Review of coevolutionary developments of evolutionary multi-objective and many-objective algorithms and test problems," in *Proc. IEEE Symp. Comput. Intell. Multi Criteria Decision Making*, Orlando, FL, USA, 2014, pp. 178–184.
- [2] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization," in *Proc. 5th Int. Conf. Genetic Algorithms*, 1993, pp. 416–423.
- [3] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, 1994.
- [4] J. Horn, N. Nafpliotis, and D. E. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization," in *Proc. 1st IEEE Int. Conf. Evol. Comput.*, Orlando, FL, USA, 1994, pp. 82–87.
- [5] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.
- [6] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.
- [7] 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.
- [8] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," *Comput. Eng. Netw. Lab. (TIK)*, Dept. Elect. Eng., ETH, Zurich, Switzerland, TIK Tech. Rep. 103, 2001.

- [9] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization*, A. Abraham, L. Jain, and R. Goldberg, Eds. London, U.K.: Springer-Verlag, 2005, pp. 105–145.
- [10] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, Oct. 2006.
- [11] D. Brockhoff and E. Zitzler, "Offline and online objective reduction in evolutionary multiobjective optimization based on objective conflicts," *Comput. Eng. Netw. Lab. (TIK)*, Dept. Elect. Eng., ETH, Zurich, Zürich, Switzerland, TIK Tech. Rep. 269, 2007.
- [12] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Proc. IEEE Congr. Evol. Comput.*, Hong Kong, 2008, pp. 2419–2426.
- [13] V. Khare, X. Yao, and K. Deb, "Performance scaling of multi-objective evolutionary algorithms," in *Evolutionary Multi-Criterion Optimization* (LNCS 2632). Heidelberg, Germany: Springer-Verlag, 2003, pp. 376–390.
- [14] R. C. Purshouse and P. J. Fleming, "On the evolutionary optimization of many conflicting objectives," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 770–784, Dec. 2007.
- [15] T. Wagner, N. Beume, and B. Naujoks, "Pareto-, aggregation-, and indicator-based methods in many-objective optimization," in *Evolutionary Multi-Criterion Optimization* (LNCS 4403). Berlin, Germany: Springer, 2007, pp. 742–756.
- [16] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization by NSGA-II and MOEA/D with large populations," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, San Antonio, TX, USA, 2009, pp. 1758–1763.
- [17] H. Sato, H. E. Aguirre, and K. Tanaka, "Controlling dominance area of solutions and its impact on the performance of MOEAs," in *Evolutionary Multi-Criterion Optimization* (LNCS 4403). Berlin, Germany: Springer, 2007, pp. 5–20.
- [18] H. E. Aguirre and K. Tanaka, "Adaptive ε -ranking on many-objective problems," *Evol. Intell.*, vol. 2, no. 4, pp. 183–206, Dec. 2009.
- [19] D. W. Corne and J. D. Knowles, "Techniques for highly multiobjective optimisation: Some nondominated points are better than others," in *Proc. Genet. Evol. Comput. Conf.*, London, U.K., 2007, pp. 773–780.
- [20] S. Kukkonen and J. Lampinen, "Ranking-dominance and many-objective optimization," in *Proc. IEEE Congr. Evol. Comput.*, Singapore, 2007, pp. 3983–3990.
- [21] M. Köppen and K. Yoshida, "Substitute distance assignments in NSGA-II for handling many-objective optimization problems," in *Evolutionary Multi-Criterion Optimization* (LNCS 4403). Berlin, Germany: Springer, 2007, pp. 727–741.
- [22] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *Eur. J. Oper. Res.*, vol. 181, no. 3, pp. 1653–1669, 2007.
- [23] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, 2011.
- [24] E. J. Hughes, "Evolutionary many-objective optimisation: Many once or one many?" in *Proc. IEEE Congr. Evol. Comput.*, Edinburgh, U.K., Sep. 2005, pp. 222–227.
- [25] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.
- [26] 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.
- [27] M. Asafuddoula, T. Ray, and R. Sarker, "A decomposition-based evolutionary algorithm for many objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 445–460, Jun. 2015.
- [28] Y. Yuan, H. Xu, B. Wang, B. Zhang, and X. Yao, "Balancing convergence and diversity in decomposition-based many-objective optimizers," *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 180–198, Apr. 2016.
- [29] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation-based evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 16–37, Feb. 2016.
- [30] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.
- [31] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 694–716, Oct. 2015.
- [32] H. Seada and K. Deb, "A unified evolutionary optimization procedure for single, multiple, and many objectives," *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 358–369, Jun. 2016.
- [33] H. Ishibuchi, K. Doi, H. Masuda, and Y. Nojima, "Relation between weight vectors and solutions in MOEA/D," in *Proc. IEEE Symp. Comput. Intell. Multi Criteria Decis. Making*, Cape Town, South Africa, Dec. 2015, pp. 861–868.
- [34] C. von Lücken, B. Barán, and C. Brizuela, "A survey on multi-objective evolutionary algorithms for many-objective problems," *Comput. Optim. Appl.*, vol. 58, no. 3, pp. 707–756, Jul. 2014.
- [35] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Comput. Surveys*, vol. 48, no. 1, pp. 1–35, Sep. 2015.
- [36] K. Deb and D. K. Saxena, "Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems," in *Proc. IEEE Congr. Evol. Comput.*, Vancouver, BC, Canada, Jul. 2006, pp. 3353–3360.
- [37] D. K. Saxena, J. A. Duro, A. Tiwari, K. Deb, and Q. Zhang, "Objective reduction in many-objective optimization: Linear and nonlinear algorithms," *IEEE Trans. Evol. Comput.*, vol. 17, no. 1, pp. 77–99, Feb. 2013.
- [38] H. Ishibuchi, H. Masuda, and Y. Nojima, "Pareto fronts of many-objective degenerate test problems," *IEEE Trans. Evol. Comput.*, to be published, doi: 10.1109/TEVC.2015.2505784.
- [39] T. Murata, H. Ishibuchi, and M. Gen, "Specification of genetic search directions in cellular multi-objective genetic algorithms," in *Evolutionary Multi-Criterion Optimization* (LNCS 1993). Berlin, Germany: Springer, 2001, pp. 82–95.
- [40] H. Ishibuchi and T. Murata, "Multi-objective genetic local search algorithm," in *Proc. 3rd IEEE Int. Conf. Evol. Comput.*, Nagoya, Japan, May 1996, pp. 119–124.
- [41] H. Ishibuchi and T. Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Trans. Syst., Man, Cybern. C. Appl. Rev.*, vol. 28, no. 3, pp. 392–403, Aug. 1998.
- [42] A. Jaszkiewicz, "Genetic local search for multi-objective combinatorial optimization," *Eur. J. Oper. Res.*, vol. 137, no. 1, pp. 50–71, Feb. 2002.
- [43] A. Jaszkiewicz, "On the performance of multiple-objective genetic local search on the 0/1 knapsack problem—A comparative experiment," *IEEE Trans. Evol. Comput.*, vol. 6, no. 4, pp. 402–412, Aug. 2002.
- [44] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, Aug. 2014.
- [45] H. Sato, "Inverted PBI in MOEA/D and its impact on the search performance on multi and many-objective optimization," in *Proc. Genet. Evol. Comput. Conf.*, Vancouver, BC, Canada, Jul. 2014, pp. 645–652.
- [46] *jMetal Website*. Accessed on Sep. 29, 2015. [Online]. Available: <http://jmetal.sourceforge.net/>
- [47] *Yuan's Many-Objective EA Codes*. Accessed on Jul. 15, 2016. [Online]. Available: <https://github.com/yyxhdy/ManyEAs>
- [48] *MOEA/DD Code*. Accessed on Jul. 15, 2016. [Online]. Available: <http://www.cs.bham.ac.uk/~likw/publications.html>
- [49] L. While, L. Bradstreet, and L. Barone, "A fast way of calculating exact hypervolumes," *IEEE Trans. Evol. Comput.*, vol. 16, no. 1, pp. 86–95, Feb. 2012.
- [50] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Simultaneous use of different scalarizing functions in MOEA/D," in *Proc. Genet. Evol. Comput. Conf.*, Portland, OR, USA, Jul. 2010, pp. 519–526.
- [51] J. Siwei, C. Zhihua, Z. Jie, and O. Yew-Soon, "Multiobjective optimization by decomposition with Pareto-adaptive weight vectors," in *Proc. 7th Int. Conf. Natural Comput.*, Shanghai, China, Jul. 2011, pp. 1260–1264.



Hisao Ishibuchi (M'93–SM'10–F'14) received the B.S. and M.S. degrees in precision mechanics from Kyoto University, Kyoto, Japan, in 1985 and 1987, respectively, and the Ph.D. degree in computer science from Osaka Prefecture University, Sakai, Japan, in 1992.

Since 1987, he has been with Osaka Prefecture University, where he is currently a Professor with the Department of Computer Science and Intelligent Systems. His current research interests include fuzzy rule-based classifiers, evolutionary multiobjective optimization, memetic algorithms, and evolutionary games.

Dr. Ishibuchi was the IEEE Computational Intelligence Society (CIS) Vice-President for Technical Activities from 2010 to 2013. He is an AdCom Member of the IEEE CIS from 2014 to 2016, an IEEE CIS Distinguished Lecturer from 2015 to 2017, and an Editor-in-Chief of the *IEEE Computational Intelligence Magazine* from 2014 to 2017. He is also an Associate Editor for the *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, the *IEEE TRANSACTIONS ON CYBERNETICS*, and the *IEEE ACCESS*.



Yu Setoguchi received the B.S. degree in computer science and intelligent systems from Osaka Prefecture University, Sakai, Japan, in 2015, where he is currently pursuing the master's degree with the Department of Computer Science and Intelligent systems.

His current research interests include evolutionary multiobjective optimization and many-objective optimization.



Hiroyuki Masuda received the M.S. degree in computer science and intelligent systems from Osaka Prefecture University, Sakai, Japan, in 2016.

His current research interests include evolutionary multiobjective optimization and multiobjective problem design.



Yusuke Nojima (M'00) received the B.S. and M.S. degrees in mechanical engineering from the Osaka Institute of Technology, Osaka, Japan, in 1999 and 2001, respectively, and the Ph.D. degree in system function science from Kobe University, Hyogo, Japan, in 2004.

Since 2004, he has been with Osaka Prefecture University, Sakai, Japan, where he was a Research Associate and where he is currently an Associate Professor with the Department of Computer Science and Intelligent Systems. His current research interests include multiobjective genetic fuzzy systems, evolutionary multiobjective optimization, and parallel distributed data mining.

Prof. Nojima is an Associate Editor of the *IEEE Computational Intelligence Magazine*.