H. Ishibuchi, K. Doi, and Y. Nojima, "Characteristics of many-objective test problems and penalty parameter specification in MOEA/D," *Proc. of 2016 IEEE Congress on Evolutionary Computation (Accepted)*.

Characteristics of Many-Objective Test Problems and Penalty Parameter Specification in MOEA/D

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

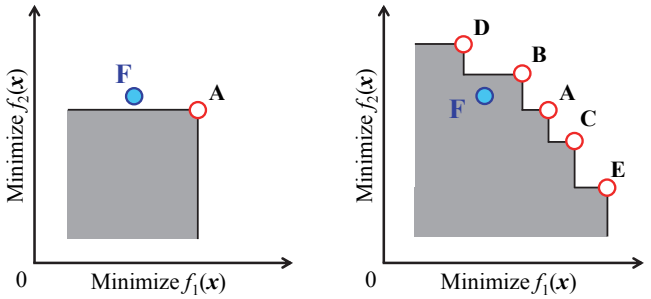
***Abstract*—Recently a number of evolutionary many-objective algorithms have been proposed using uniformly generated weight vectors. Those algorithms can be viewed as improved versions of MOEA/D with the PBI (penalty-based boundary intersection) function. Reference lines are uniformly specified using the weight vectors in the normalized objective space. The basic idea of those algorithms is to find a single solution along each reference line. Whereas a different search mechanism has been devised in each algorithm, solution assignment to reference lines is commonly based on the distance to the nearest reference line. This solution assignment can be interpreted as using a larger penalty value in MOEA/D with PBI. Actually, MOEA/D with PBI works well on frequently-used many-objective test problems DTLZ1-4 when a large penalty value is used. However, the shape of the contour lines of the PBI function suggests the use of a small penalty value for many-objective problems. Moreover, good results have been reported in the literature for many-objective knapsack problems when a small penalty value was used. In this paper, we discuss why good results are obtained from a large penalty value from a viewpoint of characteristics of DTLZ1-4 as test problems. Our discussions on the use of a large penalty value also explain why good results are obtained by recently-proposed weight vector-based evolutionary many-objective algorithms.**

Keywords*—MOEA/D, penalty-based boundary intersection (PBI), evolutionary many-objective optimization, evolutionary multiobjective optimization (EMO).

I. Introduction

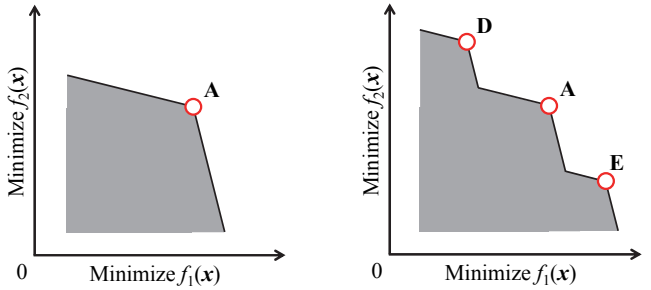
Applications of evolutionary multiobjective optimization (EMO) algorithms to many-objective optimization problems with four or more objectives have been actively studied in the last decade [1]-[3]. Standard well-known Pareto dominance-based EMO algorithms such as NSGA-II [4] and SPEA [5] do not work well on many-objective problems. This is because almost all solutions in a population become non-dominated with each other in a very early stage of evolution when such an EMO algorithm is applied to many-objective problems [1], [6]. As a result, Pareto dominance-based fitness evaluation cannot generate a strong selection pressure toward the Pareto front.

This difficulty can be also explained by the decrease in the relative size of the improving region in the objective space. Let us consider a two-objective minimization problem and solution A in the two-dimensional objective space in Fig. 1 (a). Any solution in the shaded region dominates solution A in the sense of Pareto dominance. As shown in Fig. 1 (a), the relative size of the improving region of solution A is 1/4 (i.e., 25%) of the neighborhood of A in the objective space. Solution F in Fig. 1 (a) is non-dominated. If we use Pareto dominance for solution replacement, A is not replaced with F whereas F looks good. Thanks to population-based search of EMO algorithms, such a solution often dominates some other solutions in the population as shown in Fig. 1 (b). Solution B in Fig. 1 (b) is replaced with solution F when Pareto dominance is used as the solution replacement criterion. However, by increasing the number of objectives from 2 to m, the relative size of the improving region is exponentially decreased from 1/4 to $1/2^m$. Moreover, solution density in the objective space becomes very sparse by increasing the number of objectives. As a result, a new solution is not likely to dominate any solution in the current population. This slows down the evolution of solutions toward the Pareto front. This difficulty can be remedied by increasing the size of the improving region of each solution as shown in Fig. 2 using a modified Pareto dominance relation [7].



(a) Improving region of A. (b) Improving regions of A-E.

Fig. 1. Illustration of Pareto dominance.



(a) Modified region of A. (b) Modified regions of A, D and E.

Fig. 2. Illustration of modified Pareto dominance [7].

This work was supported by JSPS KAKENHI Grant Numbers 24300090 and 26540128.

Since 2007, a number of improved versions have been proposed for MOEA/D (multiobjective evolutionary algorithm based on decomposition [8]). As in MOGLS (multiobjective genetic local search [9]-[11]), a weighted scalarizing function was used in MOEA/D. Whereas a weight vector was randomly updated for parent selection and local search in MOGLS, a set of weight vectors was uniformly pre-specified together with their neighborhood structure in MOEA/D. The same idea of weight vector specification and neighborhood definition was proposed before 2007 (e.g., C-MOGA (cellular multiobjective genetic algorithm [12]) in 2001). However, uniform weight vectors and their neighborhood structure were used only for parent selection in C-MOGA. In MOEA/D, they were also used for solution replacement. A new solution was compared with all neighbors using their weight vectors in MOEA/D. All inferior neighbors were replaced with the new solution.

The characteristic features of MOEA/D can be summarized as the use of (i) a weighted scalarizing function, (ii) uniformly generated weight vectors, (iii) neighborhood-based parent selection, and (iv) neighborhood-based solution replacement. MOGLS has only the first feature while C-MOGA has the first three features. The neighborhood-based solution replacement makes it possible to efficiently drive a population toward the Pareto front. This is because multiple solutions in a population can be replaced with a single good solution.

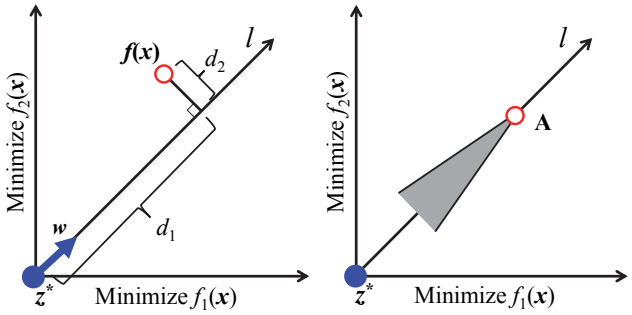
Actually, it was reported that the performance of MOEA/D on many-objective knapsack problems was degraded by decreasing the solution replacement neighborhood size [13]. However, an opposite observation was also reported in the same study [13] for many-objective distance minimization problems with multiple equivalent Pareto regions. That is, the decrease in the solution replacement neighborhood size improved the performance of MOEA/D through the increase in the diversity of solutions. These inconsistent observations can be explained as follows: "Neighborhood-based solution replacement has positive and negative effects on convergence and diversity, respectively." Its usefulness depends on the relation between an offspring and its parents. The original naive neighborhood-based solution replacement in MOEA/D is efficient only when an offspring is generated around its parents (i.e., the distance between an offspring and its parents is small in the objective space).

As we explained, neighborhood-based solution replacement has at least two potential drawbacks: One is the decrease of diversity by replacing multiple solutions with a single good one, and the other is the inefficiency of replacement when a good solution is generated far from its parents in the objective space. These potential drawbacks can be removed by appropriately assigning a newly generated solution to a weight vector [14] or a set of weight vectors where only a single solution is replaced [15]. These ideas have been used in recently proposed weight vector-based evolutionary many-objective algorithms (e.g., NSGA-III [16], I-DBEA [17], MOEA/DD [18] and θ-DEA [19]) where only a single solution is replaced with a newly generated solution. In [16]-[19], a set of reference lines (which can be interpreted as search directions in the objective space) are generated using a set of uniformly pre-specified weight vectors. A new solution is assigned to the nearest reference line. The current solution of the reference line is replaced when it is inferior to the new solution. The distance to the reference line is used together with other criteria for solution comparison.

Solution evaluation in [17]-[19] can be explained using two distances d_1 and d_2 in Fig. 3 (a) when all solutions are non-dominated. The first distance d_1 is measured from the ideal point $\boldsymbol{z}^*$ to the solution $\boldsymbol{f}(\boldsymbol{x})$ along the reference line l while the second distance d_2 is measured from the solution to the reference line. In I-DBEA [17], d_2 and d_1 were used as the primary and secondary criteria, respectively. In MOEA/DD [18] and θ-DEA [19], d_1 and d_2 are combined into a single criterion using a large positive weight value θ for d_2 as

$$\text{Minimize } f^{PBI}(\boldsymbol{x} \mid \boldsymbol{w}, \boldsymbol{z}^*) = d_1 + \theta d_2 . \quad (1)$$

This formulation is exactly the same as the PBI (penalty-based boundary intersection) function in MOEA/D [8]. The parameter θ is a pre-specified positive constant, which is called the penalty parameter. In MOEA/D [8] and MOEA/DD [18], the value of θ was specified as $\theta = 5$. Fig. 3 (b) shows the improving region of solution A with respect to the PBI function in (1) with $\theta = 5$ when A is on the reference line l. In θ-DEA [19], the value of θ was specified as $\theta = 10^6$. This specification may have almost the same meaning as the use of d_1 and d_2 in I-DBEA [17] where d_2 is used as the primary criterion (and d_1 is the secondary criterion). NSGA-III [16] also uses similar solution replacement criteria: the first criterion is Pareto dominance in the merged population of the current and offspring populations, and the second criterion is the distance to the nearest reference line.



(a) Two distances d_1 and d_2 in PBI. (b) Contour lines of PBI with $\theta = 5$.

Fig. 3. Explanation of the PBI function.

As we will show later in this paper (also see [20]), the use of a large value for the penalty parameter θ in the PBI function in (1) makes the improving region very small (see Fig. 3 (b)). The increase in the number of objectives also decreases the relative size of the improving region. Thus we may need to use a small value for θ in (1) in order to realize efficient search by MOEA/D with PBI for many-objective problems as explained in Fig. 2. Actually, it was reported that good results were obtained by MOEA/D with PBI for many-objective knapsack problems when a small value such as 0.1 was used for the penalty parameter θ in [6], [20]. However, in NSGA-III [16], I-DBEA [17], MOEA/DD [18] and θ-DEA [19], the distance to the reference line (i.e., d_2) is mainly used for solution assignment and solution replacement. This is almost the same as the use of a very large value of θ. Moreover, very good

results were reported in [16]-[19] for many-objective DTLZ1-4 problems [21]. Motivated by those results, we apply MOEA/D with PBI to the same test problems. As shown in this paper, good results are obtained even when we use a very large value such as 100 for the penalty parameter θ. Our experimental results are consistent with the reported results by NSGA-III [16], I-DBEA [17], MOEA/DD [18] and θ-DEA [19] in the literature. However, they are inconsistent with the above-mentioned discussion about the difficulty of many-objective optimization and its countermeasure [7] in Fig. 2.

In this paper, we discuss why good results are obtained by MOEA/D with PBI for many-objective DTLZ1-4 problems even when a very large value is used for the penalty parameter θ (whereas experimental results in many-objective knapsack problems [6], [20] and the discussion about the difficulty of many-objective problem suggest the use of a small penalty parameter value). Our discussions on MOEA/D also explain why good results are obtained by NSGA-III [16], I-DBEA [17], MOEA/DD [18] and θ-DEA [19] whereas the distance from each solution to the nearest reference line is used as an important solution replacement criterion.

This paper is organized as follows. First we explain the multiobjective search by MOEA/D with PBI using the contour lines of the PBI function in Section II. In Section III, we report our experimental results on DTLZ1 and DTLZ2 problems with 2, 4, 6 and 8 objectives. Various specifications of the value of the penalty parameter θ are examined between 0.01 and 100. In Section IV, we explain why good results are obtained from a very large value of θ using characteristic features of DTLZ1-4 as test problems. We also discuss the necessity of more general many-objective test problems. For this purpose, we report experimental results on maximization variants of DTLZ where all objectives are maximized (whereas they are minimized in the original DTLZ problems). Totally different observations with respect to the specification of the penalty parameter value are obtained from our two computational experiments: One is on the original DTLZ1 and DTLZ2 problems, and the other is on their maximization variants. Finally, we conclude this paper in Section V.

II. Multiobjective Search by MOEA/D with PBI

A. Multiobjective Optimization Problem

Using the following problem with m objectives, we explain the multiobjective search by MOEA/D with PBI:

$$\text{Minimize } \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), ..., f_m(\boldsymbol{x})) \text{ subject to } \boldsymbol{x} \in \mathbf{X}, \quad (2)$$

where $f_i(\boldsymbol{x})$ is the ith objective to be minimized ($i = 1, 2, ..., m$), $\boldsymbol{x}$ is the decision vector, and $\mathbf{X}$ is the feasible region.

B. MOEA/D with PBI

In MOEA/D [8], the multiobjective problem is decomposed into a number of single-objective problems using a set of uniformly specified weight vectors. Let us denote an m-dimensional weight vector as $\boldsymbol{w} = (w_1, w_2, ..., w_m)$. In MOEA/D, a set of weight vectors is generated using the following formulations (i.e., all weight vectors satisfying the following conditions are generated):

$$\sum_{i=1}^{m} w_i = 1 \text{ and } w_i \geq 0 \text{ for } i = 1, 2, \ldots, m, \quad (3)$$

$$w_i \in \left\{0, \frac{1}{H}, \frac{2}{H}, ..., \frac{H}{H}\right\} \text{ for } i = 1, 2, \ldots, m, \quad (4)$$

where H is a positive integer used for specifying the total number of weight vectors (for details, see [8]).

In this paper, we use the PBI function in MOEA/D. That is, the minimization of the PBI function is the basic structure of single-objective optimization in MOEA/D in this paper. As we have already explained by Fig. 3 in Section I, the PBI function for the minimization problem in (2) is written using the weight vector $\boldsymbol{w}$, the reference point $\boldsymbol{z}^*$ and the penalty parameter θ as

$$\text{Minimize } f^{PBI}(\boldsymbol{x} \mid \boldsymbol{w}, \boldsymbol{z}^*) = d_1 + \theta d_2, \quad (5)$$

where d_1 and d_2 are calculated as follows (see Fig. 3 (a)):

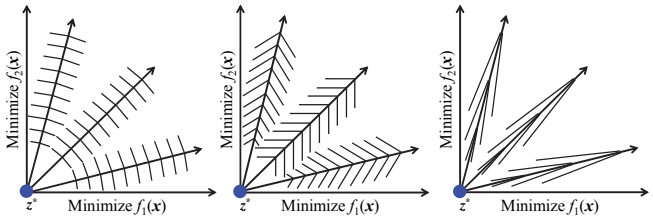
$$d_1 = \left| (\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{z}^*)^T \boldsymbol{w} \right| / \|\boldsymbol{w}\|, \quad (6)$$

$$d_2 = \left\| \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{z}^* - d_1 \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} \right\|. \quad (7)$$

As we can see from (5), the value of the penalty parameter θ shows the importance of the second distance d_2 (i.e., distance to the reference line). In one extreme case where θ is very large, the minimization of the PBI function can be viewed as pushing a solution toward the reference line (i.e., minimizing the second distance d_2). In the other extreme case where θ is very small, the minimization of the PBI function becomes similar to the minimization of the weighted sum.

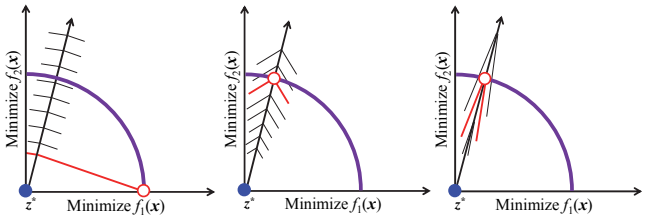
C. Contour Lines of the PBI Function

In Fig. 4, we show the contour lines of the PBI function for three weight vectors ($\boldsymbol{w} = (0.8, 0.2), (0.5, 0.5), (0.2, 0.8)$) and three values of the penalty parameter θ ($\theta = 0.1, 1.0, 10$). The search for a single solution using the PBI function with each weight vector can be explained by the contour lines in Fig. 4. When θ is small (e.g., Fig. 4 (a)), the steepest decent direction is almost the same as the direction toward the reference point $\boldsymbol{z}^*$. A move of a solution away from the reference line can be acceptable if it decreases the distance from the reference point $\boldsymbol{z}^*$. This means that the obtained solution is not always close to the reference line. Actually, the solution can be far away from the corresponding reference line as shown in Fig. 5 (a) where the bold purple curve is the Pareto front of a two-objective minimization problem. The best solution for the weight vector (0.2, 0.8) is obtained at the bottom-right edge of the Pareto front in Fig. 5 (a).



(a) $\theta = 0.1$. (b) $\theta = 1.0$. (c) $\theta = 10$.

Fig. 4. Contour lines of the PBI function for the three weight vectors.



(a) $\theta = 0.1$. (b) $\theta = 1.0$. (c) $\theta = 10$.

Fig. 5. Explanation of the search for the weight vector (0.2, 0.8).

When θ is large (e.g., $\theta = 10$ as in Fig. 4 (c) and Fig. 5 (c)), the steepest decent move of a solution is close to the direction toward the reference line (not toward the reference point z^*). A move of a solution away from the reference line is not likely to be acceptable even when it decreases the distance from the reference point z^*. As a result, it is expected that the obtained solution is very close to the corresponding reference line. This is explained in Fig. 5 (c). When $\theta = 1$, the two distances d_1 and d_2 have the same weight in the PBI function. Thus the contour line has a right angle in Fig. 4 (b) and Fig. 5 (b).

III. Experimental Results on DTLZ1 and DTLZ2

In our computational experiments, MOEA/D with PBI is applied to DTLZ1 and DTLZ2 problems with 2, 4, 6 and 8 objectives under various specifications of the population size as shown in Table 1. The neighborhood size is specified as 10% of the population size. With respect to the penalty parameter value, we examine the following specifications:

$\theta = 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100.$

TABLE I. Various Specifications of Population Size.

Objectives	Specifications of Population Size
$m = 2$	50, 100, 500, 1000, 5000
$m = 4$	56, 84, 455, 969, 4960
$m = 6$	56, 126, 462, 1287, 4368
$m = 8$	36, 120, 330, 792, 6435

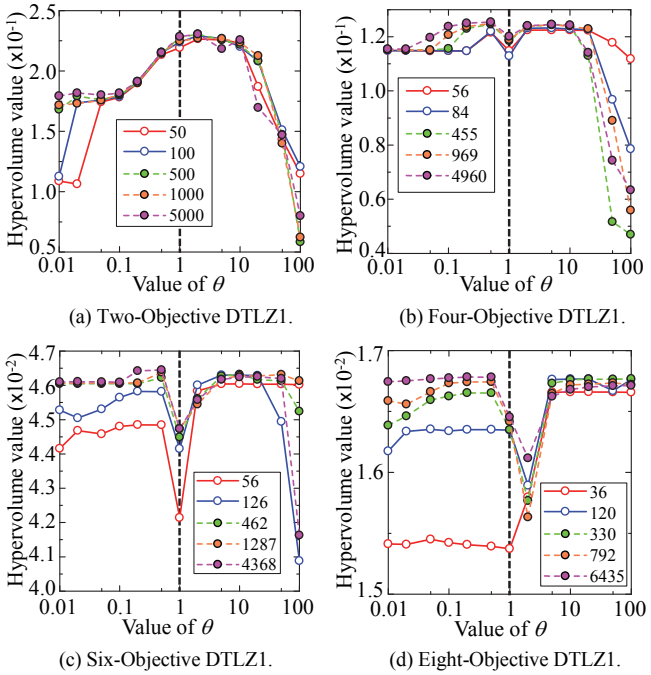
We use the polynomial mutation with the distribution index 20 and the mutation probability $1/n$ where n is the string length, and the SBX crossover with the distribution index 15 and the crossover probability 0.8. The number of decision variables in the DTLZ1 problem with m objectives is specified as $5+m-1$. For DTLZ2, it is specified as $10+m-1$. The total number of examined solutions is used as the termination condition, which is specified as $m \times 10{,}000$ solutions for m-objective problems. As the performance indicator, we use the average hypervolume value over 10 runs. The reference point for the hypervolume calculation is specified as (0.6, 0.6, ..., 0.6) for DTLZ1 and (1.1, 1.1, ..., 1.1) for DTLZ2. We do not use any special setting in our computational experiments (except for the examination of various values of the penalty parameter and the population size). Our setting is similar to commonly-used ones in recent studies on evolutionary many-objective optimizations [16]-[19].

Experimental results are summarized in Fig. 6 and Fig. 7. We can obtain the following observations:

(1) Good results are obtained for the eight-objective DTLZ1 and DTLZ2 problems even when the value of θ is very large (e.g., $\theta = 100$) in Fig. 6 (d) and Fig. 7 (d).

(2) When the value of θ is small (e.g., $\theta \le 0.5$), good results are not obtained in Fig. 7 for all DTLZ2 problems independent of the number of objectives and the population size.

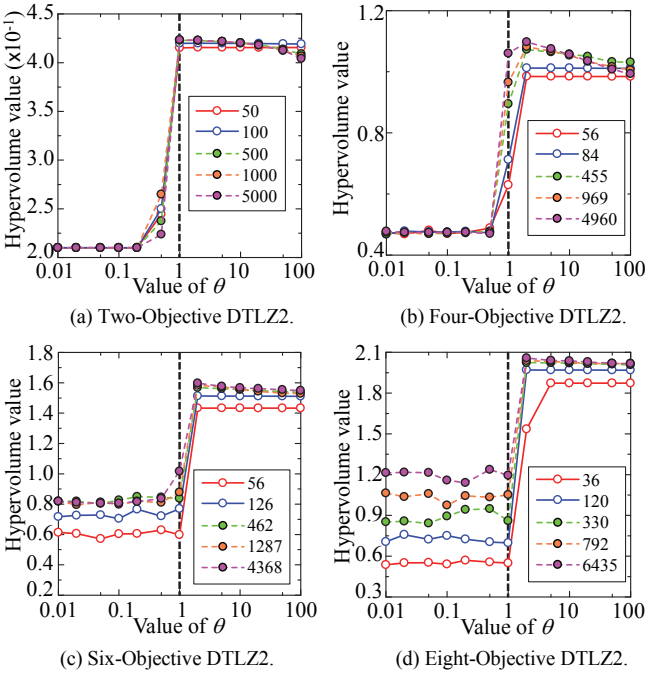
(3) For the six-objective and eight-objective DTLZ1 problems, poor results are obtained from intermediate specifications of θ such as $\theta = 1$ (Fig. 6 (c)) and $\theta = 2$ (Fig. 6 (d)).



(a) Two-Objective DTLZ1. (b) Four-Objective DTLZ1.

(c) Six-Objective DTLZ1. (d) Eight-Objective DTLZ1.

Fig. 6. Average hypervolume values by MOEA/D with PBI on DTLZ1.



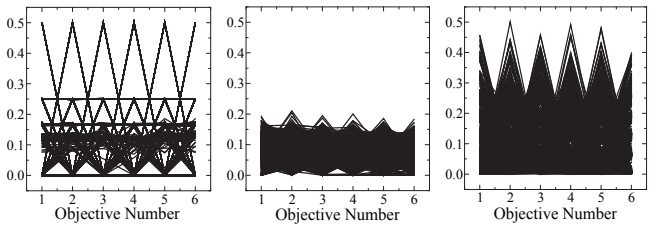
(a) Two-Objective DTLZ2. (b) Four-Objective DTLZ2.

(c) Six-Objective DTLZ2. (d) Eight-Objective DTLZ2.

Fig. 7. Average hypervolume values by MOEA/D with PBI on DTLZ2.

The second observation can be explained using the shape of the Pareto front. Each DTLZ2 problem has a concave Pareto front (see Fig. 9 (b) in the next session). As in the case of the weighted sum, it is difficult for the PBI function with a small value of the penalty parameter θ to find Pareto optimal solutions inside the concave Pareto front (see Fig. 5 (a)). This difficulty explains the second observation: Good results are not obtained from small values of θ in Fig. 7 independent of the number of solutions and the population size.

The explanation of the third observation is not so easy. Each DTLZ1 problem has a linear Pareto front (i.e., line, plane, hyperplane). The contour lines in Fig. 4 and Fig. 5 have a circular cone shape for multiobjective problems with more than two objectives. When the value of θ is large, the top of the contour circular cone is sharp as shown in Fig. 5 (c). Thus the solution for each reference line is obtained around the reference line (i.e., around the top of the contour circular cone). This leads to a large diversity of solutions. In Fig. 8 (c), we show an example of an obtained solution set for the six-objective DTLZ1 problem from the specification of $\theta = 10$. The horizontal axis shows the index of the objective whereas the vertical axis shows the objective value of each objective in Fig. 8. We can see from Fig. 8 (c) that a wide variety of solutions are obtained when the penalty parameter is large.



(a) $\theta = 0.1$. (b) $\theta = 1.0$. (c) $\theta = 10$.

Fig. 8. Obtained solutions for the six-objective DTLZ1 problem.

When the value of θ is small, solutions are likely to be obtained around the boundary of the Pareto front as shown in Fig. 5 (a). However, since the Pareto front is linear and the contour circular cone has a top (whereas it is not sharp), solutions are also obtained inside the Pareto front for some reference lines. That is, some inside solutions are obtained together with other solutions around the boundary of the Pareto front. Fig. 8 (a) shows an example of an obtained solution set from the specification of $\theta = 0.1$. We can see from Fig. 8 (a) that boundary solutions with an objective value 0.5 for a single objective are obtained together with inside solutions with objective values about 0.1 for almost all objectives.

In general, when θ is small, each solution is obtained far from the corresponding reference line as shown in Fig. 5 (a). When θ is large, each solution is obtained around the corresponding reference line as shown in Fig. 5 (c). By increasing θ from a small value to a large one, we can observe a kind of a transition state where the diversity of solutions is very small as shown in Fig. 8 (b). Fig. 8 (b) is an example of an obtained solution set from the specification of $\theta = 1$. Such a solution set is obtained when the sides of many contour circular cones have almost the same tangent as the Pareto front. In this case, many solutions have the same or similar PBI function values. This makes it very difficult for MOEA/D with PBI to find a set of widely distributed solutions as shown in Fig. 8 (b). This difficulty was explained by analyzing the relation between the weight vectors and the obtained solutions in detail in [22].

In the next section, we focus our attention on the first observation. That is, we discuss why good results are obtained for the eight-objective DTLZ1 and DTLZ2 problems even when the value of the penalty parameter θ is very large in MOEA/D with PBI.

IV. Discussions on Experimental Results

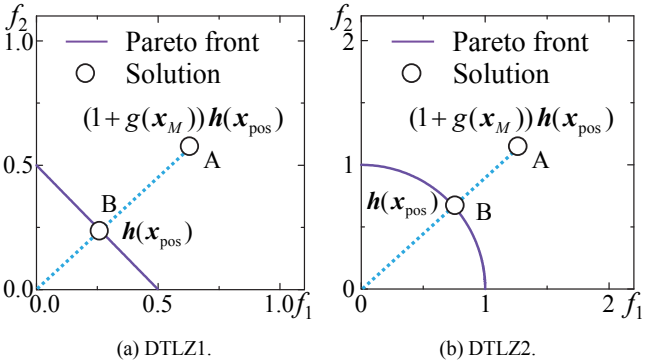
The first observation in the previous section is inconsistent with the discussion about the difficulty of many-objective optimization in Section I. This observation is also inconsistent with the reported results on many-objective knapsack problems in [6] and [20] where the best results were obtained from small values of θ such as $\theta = 0.1$ (poor results were obtained from large values of θ such as $\theta = 10$ in [6] and [20]).

The objective vector $\boldsymbol{f}(\boldsymbol{x})$ in the DTLZ1-4 problems can be written as follows [21]:

$$\boldsymbol{f}(\boldsymbol{x}) = (1 + g(\boldsymbol{x}_M))\boldsymbol{h}(\boldsymbol{x}_{\text{pos}}), \tag{8}$$

where $\boldsymbol{h}(\boldsymbol{x}_{\text{pos}})$ specifies the shape of the Pareto front using the $(m-1)$ decision variables in $\boldsymbol{x}_{\text{pos}}$, and $g(\boldsymbol{x}_M)$ specifies the distance from the Pareto front ($g(\boldsymbol{x}_M) \geq 0$) using k decision variables in $\boldsymbol{x}_M$. Pareto optimal solutions are obtained by minimizing $g(\boldsymbol{x}_M)$ to zero.

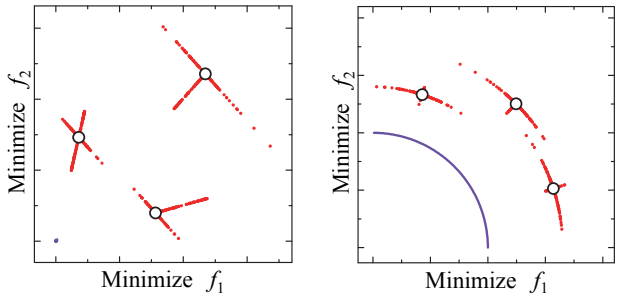
All feasible solutions with $g(\boldsymbol{x}_M) = 0$ are Pareto optimal in DTLZ1-4. The basic problem structure in (8) is illustrated for the two-objective DTLZ1 and DTLZ2 problems in Fig. 9. By changing some values of $\boldsymbol{x}_{\text{pos}}$, Pareto optimal solution B in Fig. 9 (a) and Fig.9 (b) moves along the Pareto front. When some values of $\boldsymbol{x}_M$ are changed, solution B moves toward solution A. Solution A in Fig. 9 (a) and Fig. 9 (b) moves toward solution B or in the opposite direction by changing some values in $\boldsymbol{x}_M$. When we change some values in $\boldsymbol{x}_{\text{pos}}$ of solution A, it moves without changing the distance to the Pareto front (e.g., solution A in Fig. 9 (a) moves in parallel with the Pareto front by changing some values in $\boldsymbol{x}_{\text{pos}}$). That is, possible moves of each solution by changing values in either $\boldsymbol{x}_M$ or $\boldsymbol{x}_{\text{pos}}$ are severely limited by the special problem structure of DTLZ1-4.



(a) DTLZ1. (b) DTLZ2.

Fig. 9. Explanations of the basic problem structure of the DTLZ problems.

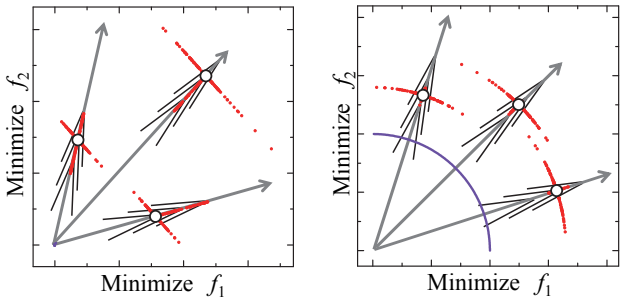
In Fig. 10, we show 100 solutions (red points) generated by mutation from each parent solution denoted by a black circle. The polynomial mutation with the distribution index 20 is applied to a randomly selected single variable of each solution. As we have already explained using Fig. 9, solutions are generated only in the four directions of the parent solution. One direction which is away from the Pareto front clearly degrades the parent solution. Its opposite direction toward the Pareto front clearly improves the parent solution. This means that Pareto dominance relation always holds between the parent and its offspring generated by mutating arbitrary selected variables in $\boldsymbol{x}_M$ of the parent solution. This is totally different from the following implicit assumption in many-objective optimization: Almost all solutions are non-dominated with each other.



(a) Two-objective DTLZ1. (b) Two-objective DTLZ2.

Fig. 10. Generated 100 solutions (red points) from each parent solution (black circle) by a single application of mutation to a randomly selected single variable. The purple point in (a) and the purple curve in (b) shows the Pareto front of each problem.

When the parent solution is on the corresponding reference line and the reference point $\boldsymbol{z}^*$ is the same as the ideal point, all solutions generated by the parent solution by mutating arbitrary variables in $\boldsymbol{x}_M$ are exactly on the reference line as their parent solution as shown in Fig. 11. This means that better solutions are likely to be obtained by applying mutation several times to the parent solution. When the parent solution is not on its reference line, any move toward the Pareto front by mutating values in $\boldsymbol{x}_M$ always improves the distance from the reference line as well as the distance from the Pareto front.



(a) Two-objective DTLZ1. (b) Two-objective DTLZ2.

Fig. 11. Generated 100 solutions from each parent (black circle) by mutation.

In this situation (i.e., when the parent solution is not on its reference line), mutation to arbitrary values in $\boldsymbol{x}_{\text{pos}}$ can improve the parent solution by decreasing the distance to its reference line without degrading the distance to the Pareto front. This is because any changes of values in $\boldsymbol{x}_{\text{pos}}$ never change the distance to the Pareto front. As a result, when the parent is not on the reference line, it is more likely that better solutions will be obtained from mutation than the case where the parent solution is on the reference line.

For removing the above-mentioned search biases based on the special problem structure of the DTLZ1 and DTLZ2 problems, let us consider the following test problem with the same Pareto front as the m-objective DTLZ2 problem:

[Test Problem 1]

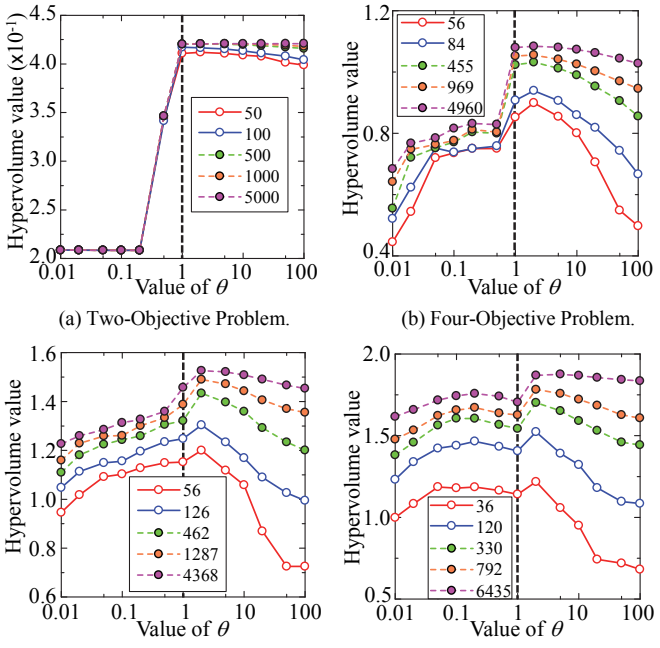
$$\text{Minimize } \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})), \tag{9}$$

$$\text{subject to } f_i(\boldsymbol{x}) = x_i,\ i = 1, 2, \ldots, m, \tag{10}$$

$$1 \le x_1^2 + x_2^2 + \ldots + x_m^2 \le a^2, \tag{11}$$

$$0 \le x_i \le a,\ i = 1, 2, \ldots, m. \tag{12}$$

In this formulation, a is a real number larger than 1. We specify a as $a = 2$. For this problem, we randomly generate $m \times 100{,}000$ feasible solutions. We do not use any genetic operators. We simply generate feasible solutions randomly. Then we choose the best solution for the PBI function with each weight vector and calculate the hypervolume. The average hypervolume values over 10 runs are shown in Fig. 12.



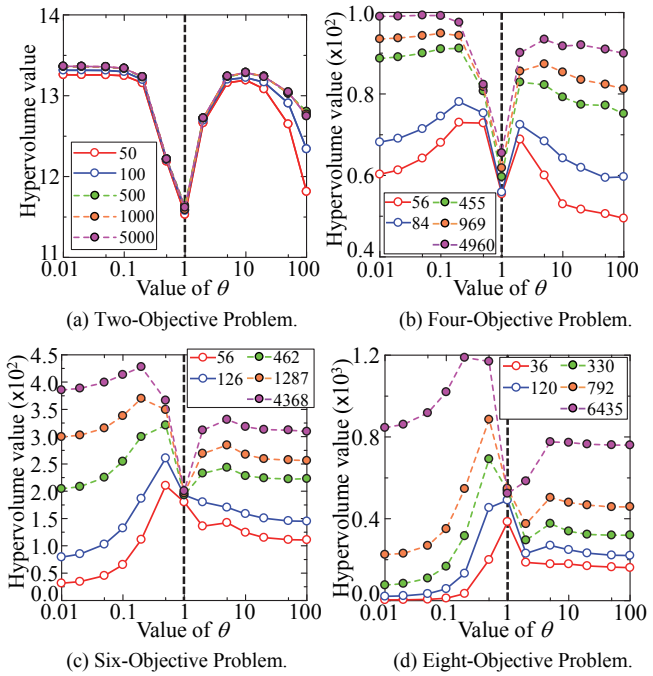
(a) Two-Objective Problem. (b) Four-Objective Problem.

(c) Six-Objective Problem. (d) Eight-Objective Problem.

Fig. 12. Average hypervolume values by PBI on our Test Problem 1.

Except for the case of two objectives in Fig. 12 (a), the increase of the penalty parameter value from $\theta = 2$ to $\theta = 100$ gradually degrades the average hypervolume value for our test problem in Fig. 12. However, the performance deterioration is very slow even for the eight-objective problem in Fig. 12 (d). This may be due to the concave shape of the Pareto front. Since good results are not obtained from small values of the penalty parameter θ, the performance deterioration by the increase of θ looks very slow.

In order to remove the bias based on the concave shape of the Pareto front, we apply MOEA/D with PBI to a variant of the DTLZ2 problem where all objectives are to be maximized (instead of to be minimized in the original formulation of the DTLZ2 problem). Experimental results are summarized in Fig. 13 where $(-0.5, -0.5, ..., -0.5)$ is used as the reference point for hypervolume calculation.



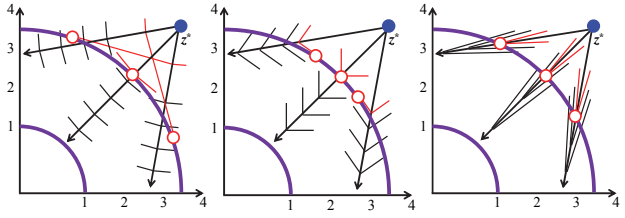
(a) Two-Objective Problem. (b) Four-Objective Problem.

(c) Six-Objective Problem. (d) Eight-Objective Problem.

Fig. 13. Experimental results of MOEA/D with PBI on the maximization variant of the DTLZ2 problem.

In Fig. 13, good results are obtained when the penalty parameter is small. This observation is consistent with the reported results on many-objective knapsack problems in [6], [20]. This observation is also consistent with the discussion on the difficulty of many-objective optimization.

However, the behavior of MOEA/D with PBI cannot be simply explained by the size of the improving region. For example, the clear performance deterioration at $\theta = 1$ in Fig. 13 (and Fig. 6) shows the difficulty in explaining the behavior of the MOEA/D with PBI. As we mentioned in the previous section, the main reason for this performance deterioration is the decrease in the diversity of solutions. In Fig. 14, we show the contour lines of the PBI function for the two-objective maximization DTLZ2 problem. Since this is the maximization problem, the multiobjective search is performed by pulling solutions from the top-right reference point z^*. When $\theta = 1$ in Fig. 14 (b), optimal solutions for many reference lines are obtained around the center of the Pareto front. This is because each optimal solution is obtained inside the corresponding reference line as shown in Fig. 14 (b). When $\theta = 0.1$ in Fig. 13 (a), each optimal solution is far from the corresponding reference line. As a result, the obtained solutions have a large diversity. When $\theta = 10$ in Fig. 14 (c), each optimal solution is close to the corresponding reference line. As a result, the obtained solutions have a large diversity. Only when the penalty parameter is around $\theta = 1$, many solutions are obtained around the center of the Pareto front. Thus good results are not obtained in Fig. 13 from $\theta = 1$.



(a) $\theta = 0.1$. (b) $\theta = 1.0$. (c) $\theta = 10$.

Fig. 14. Contour lines of the PBI function for the maximization variant of the two-objective DTLZ2 problem.

It should be noted that the maximization variants of the DTLZ problems are exactly the same as the original DTLZ problems except for maximizing all objectives instead of minimizing them. This means that new solutions are generated by mutation in the same manner as we explained in Fig. 10 and Fig. 11. However, the maximization variants have the different reference lines as shown in Fig. 14. Thus the optimal solution with respect to each reference line cannot be easily obtained for the maximization variants if compared with the original DTLZ problems.

When we analyze the search behavior of MOEA/D with PBI, we have to consider the relation between the shape of the Pareto front and the shape of the distribution of weight vectors. This issue was pointed out by Jain & Deb [23]. As explained in [23], the shape of the Pareto fronts of the DTLZ1-4 problems is triangular. The shape of the distribution of the weight vectors in MOEA/D (and the other weight vector-based evolutionary many-objective algorithms [16]-[19]) is also triangular. This consistency between the shape of the Pareto front and the shape of the distribution of the weight vectors is another reason for high performance of MOEA/D and the other weight vector-based evolutionary many-objective algorithms.

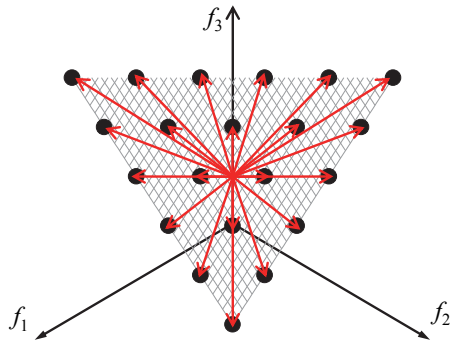


Fig. 15. Rotated triangular shape distribution of the weight vectors for a three-objective maximization problem.

The maximization variants of the DTLZ1-4 problems also have triangular Pareto fronts. However, the shape of the distribution of the weight vectors for maximization problems is rotated triangular as shown in Fig. 15. This is because the

reference point is located at the top-right corner of the objective space (see Fig. 14) instead of the bottom-left corner for minimization problems. That is, the shape of the Pareto fronts of the maximization variants of DTLZ1-4 is not consistent with the shape of the distribution of the weight vectors. This is the same situation as the inverted DTLZ1 problem in Jain & Deb [23].

V. Concluding Remarks

In this paper, we discussed why MOEA/D with PBI works very well on many-objective DTLZ1-4 problems when a large value is used as the penalty parameter (e.g., $\theta = 10$ and $\theta = 100$). Through computational experiments on DTLZ1 and DTLZ2, we explained the following reasons:

(1) Special problem structures of DTLZ1-4, which make mutation-based search along the reference lines very efficient for many-objective problems.

(2) Consistency between the shape of the Pareto front and the shape of the distribution of the weight vectors, which leads to high quality of a solution set when each solution is close to the corresponding reference line.

We removed positive effects of these biases by using the maximization variants of DTLZ1-4 where all objectives were maximized. When we used the maximization variants of DTLZ2 as test problems, better results were obtained from small values of the penalty parameter. That is, our results on the maximization variants were consistent with the reported results on many-objective knapsack problems, which were also consistent with the discussions on the difficulty of many-objective optimization. Our discussions in this paper also suggested that very good results by recently-proposed weight vector-based evolutionary many-objective algorithms would be at least partially biased by the above-mentioned two reasons based on the special characteristic features of the DTLZ1-4 test problems.

References

[1] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," *Proc. of 2008 IEEE Congress on Evolutionary Computation*, pp. 2419-2426, Hong Kong, China, June 1-6, 2008.

[2] C. von Lücken, B. Barán, and C. Brizuela, "A survey on multi-objective evolutionary algorithms for many-objective problems," *Computational Optimization and Applications*, vol. 58, no. 3, pp. 707-756, July 2014.

[3] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Computing Surveys*, vol. 48, no. 1, Article 13, pp. 1-35, September 2015.

[4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, April 2002.

[5] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. on Evolutionary Computation*, vol. 3, no. 4, pp. 257-271, 1999.

[6] H. Ishibuchi, N. Akedo, and Y. Nojima, "Behavior of multi-objective evolutionary algorithms on many-objective knapsack problems," *IEEE Trans. on Evolutionary Computation*, vol. 19, no. 2, pp. 264-283, April 2015.

[7] H. Sato, H. E. Aguirre, and K. Tanaka, "Controlling dominance area of solutions and its impact on the performance of MOEAs," *Lecture Notes in Computer Science 4403: Evolutionary Multi-Criterion Optimization - EMO 2007*, pp. 5-20, Springer, Berlin, March 2007.

[8] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. on Evolutionary Computation*, vol. 11, no. 6, pp. 712-731, December 2007.

[9] H. Ishibuchi and T. Murata, "Multi-objective genetic local search algorithm," *Proc. of 1996 IEEE International Conference on Evolutionary Computation*, pp. 119-124, Nagoya, Japan, May 1996.

[10] H. Ishibuchi and T. Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Trans. on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, vol. 28, no. 3, pp. 392-403, August 1998.

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

[12] T. Murata, H. Ishibuchi, and M. Gen, "Specification of genetic search directions in cellular multi-objective genetic algorithm," *Proc. of 1st International Conference on Evolutionary Multi-Criterion Optimization: EMO 2001* (E. Zitzler et al. (eds.), Lecture Notes in Computer Science 1993, *Evolutionary Multi-Criterion Optimization*, Springer, Berlin), pp. 82-95, March 7-9, 2001.

[13] H. Ishibuchi, N. Akedo, and Y. Nojima, "Relation between neighborhood size and MOEA/D performance on many-objective problems," *Proc. of 7th International Conference on Evolutionary Multi-Criterion Optimization,* pp. 459-474, Sheffield, UK, March 19-22, 2013.

[14] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, "Stable matching-based selection in evolutionary multiobjective optimization," vol. 18, no. 6, pp. 909-923, December 2014.

[15] Y. Yuan, H. Xu, B. Wang, B. Zhang, and X. Yao, "Balancing convergence and diversity in decomposition-based many-objective optimizers," *IEEE Trans. on Evolutionary Computation* (Available from IEEE Xplore as an Early Access Paper).

[16] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, Part I: Solving problems with box constraints," *IEEE Trans. on Evolutionary Computation*, vol. 18, no. 4, pp. 577-601, August 2014.

[17] M. Asafuddoula, T. Ray, and R. Sarker, "A decomposition-based evolutionary algorithm for many objective optimization," *IEEE Trans. on Evolutionary Computation*, vol. 19, no. 3, pp. 445-460, June 2015.

[18] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Trans. on Evolutionary Computation*, vol. 19, no. 5, pp. 694-716, October 2015.

[19] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation based evolutionary algorithm for many-objective optimization," *IEEE Trans. on Evolutionary Computation*, vol. 20, no. 1, pp. 16-37, February 2016.

[20] H. Ishibuchi, N. Akedo, and Y. Nojima, "A study on the specification of a scalarizing function in MOEA/D for many-objective knapsack problems," *Lecture Notes in Computer Science Volume 7997: Learning and Intelligent Optimization – LION 7*, pp. 231-246, Springer, January 2013.

[21] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," *Proc. of 2002 IEEE Congress on Evolutionary Computation*, pp. 825-830, May 12-17, 2002.

[22] H. Ishibuchi, K. Doi, H. Masuda, and Y. Nojima, "Relation between weight vectors and solutions in MOEA/D," *Proc. of 2015 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*, pp. 861-868, Cape Town, December 8-10, 2015.

[23] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. on Evolutionary Computation*, vol. 18, no. 4, pp. 602-622, August 2014.