Behavior of Multiobjective Evolutionary Algorithms on Many-Objective Knapsack Problems

Hisao Ishibuchi, *Fellow, IEEE*, Naoya Akedo, and Yusuke Nojima, *Member, IEEE*

***Abstract*—We examine the behavior of three classes of evolutionary multiobjective optimization (EMO) algorithms on many-objective knapsack problems. They are Pareto dominance-based, scalarizing function-based, and hypervolume-based algorithms. NSGA-II, MOEA/D, SMS-EMOA, and HypE are examined using knapsack problems with 2–10 objectives. Our test problems are generated by randomly specifying coefficients (i.e., profits) in objectives. We also generate other test problems by combining two objectives to create a dependent or correlated objective. Experimental results on randomly generated many-objective knapsack problems are consistent with well-known performance deterioration of Pareto dominance-based algorithms. That is, NSGA-II is outperformed by the other algorithms. However, it is also shown that NSGA-II outperforms the other algorithms when objectives are highly correlated. MOEA/D shows totally different search behavior depending on the choice of a scalarizing function and its parameter value. Some MOEA/D variants work very well only on two-objective problems while others work well on many-objective problems with 4–10 objectives. We also obtain other interesting observations such as the performance improvement by similar parent recombination and the necessity of diversity improvement for many-objective knapsack problems.**

***Index Terms*—Evolutionary many-objective optimization, evolutionary multiobjective optimization (EMO), many-objective problems.**

I. Introduction

SINCE the suggestion by Goldberg [1], Pareto dominance relation has been widely used for fitness evaluation in evolutionary multiobjective optimization (EMO) algorithms [2]–[4]. Whereas Pareto dominance-based algorithms such as NSGA-II [5] and SPEA2 [6] usually work very well on multiobjective problems with two or three objectives, their search ability is often severely degraded by the increase in the number of objectives [7]–[9]. This is because almost all solutions in the current population become nondominated in early generations when EMO algorithms are applied to many-objective problems. In general, the goal of EMO algorithms is to find a wide variety of nondominated solutions that approximate the entire Pareto front of a multiobjective problem. Thus, it is desirable that all solutions are nondominated when the execution of an EMO algorithm is terminated. However, a population of only nondominated solutions is not desirable in early generations. This is because those solutions are not likely to be close to the Pareto front. It is more likely that they are similar to randomly generated initial solutions. Thus, strong selection pressure toward the Pareto front is needed to drive the population to the Pareto front. However, Pareto dominance-based fitness evaluation cannot generate such a strong selection pressure when almost all solutions in the current population are nondominated (even if they are far from the Pareto front).

Various approaches have been proposed for improving the convergence property of Pareto dominance-based algorithms on many-objective problems (see [10]–[15]). However, the convergence improvement often causes a decrease in the diversity of obtained nondominated solutions [16], [17]. Some studies took into account diversity maintenance [18], [19].

Recently it has also been demonstrated that many-objective problems are not always difficult for Pareto dominance-based algorithms [13], [14], [20]–[22]. If most objectives of a many-objective problem are highly correlated (i.e., objective values of different objectives are highly correlated), it is not likely that its Pareto front spreads over a wide range of the high-dimensional objective space. If they are dependent on a few objectives, the dimensionality of the Pareto front in the objective space may be much smaller than the number of objectives. As a result, the search for Pareto optimal solutions of many-objective problems is not always difficult for Pareto dominance-based algorithms.

In this paper, we examine the behavior of three classes of EMO algorithms on many-objective knapsack problems. They are Pareto dominance-based, scalarizing function-based, and hypervolume-based algorithms. As their representatives, we use NSGA-II [5], MOEA/D [23], and SMS-EMOA [24]. This choice is based on their frequent use in the literature. Since the application of SMS-EMOA to many-objective problems is not always possible due to its heavy computation load, we also examine HypE [25] with approximate hypervolume calculation.

As test problems, we generate 500-item knapsack problems with 2–10 objectives. Our test problems are randomly generated by randomly specifying coefficients (i.e., profits) in objectives. Other test problems have dependent or correlated objectives, each of which is generated as a weighted sum of two objectives in the randomly generated test problems. We also use knapsack problems with 100, 1000, and 10 000 items.

This paper is an extended version of our former studies [21], [26], [27] in which we examined the behavior of

Manuscript received July 5, 2013; revised November 21, 2013 and February 25, 2014; accepted February 28, 2014. Date of publication April 3, 2014; date of current version March 27, 2015. This work was supported by the Japan Society for the Promotion of Science under Grant-in-Aid for Scientific Research B 24300090.

The authors are with the Department of Computer Science and Intelligent Systems, Osaka Prefecture University, Osaka 599-8531, Japan (e-mail: hisaoi@cs.osakafu-u.ac.jp; naoya.akedo@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.2014.2315442

NSGA-II, MOEA/D, and SMS-EMOA. Each study addressed only one aspect of many-objective optimization using one or two classes of EMO algorithms. For example, we examined the behavior of NSGA-II, SPEA2, and MOEA/D on many-objective knapsack problems with highly correlated objectives in [21]. NSGA-II and MOEA/D with a large population were examined in [26]. Similar parent mating was examined for NSGA-II and SMS-EMOA in [27]. HypE was not examined in our previous studies. We also use knapsack problems with 100–10 000 items while 500-item problems were always used in our previous studies.

This paper is organized as follows. In Section II, we briefly explain difficulties of many-objective optimization. In Section III, we explain how our many-objective test problems with various characteristics are generated from the two-objective 500-item knapsack problem of Zitzler and Thiele [28]. We also briefly explain NSGA-II, MOEA/D, SMS-EMOA, and HypE in Section III. Experimental results on randomly generated test problems are reported in Section IV, in which the use of a large population and the recombination of similar parents are also examined. In Section V we report experimental results on test problems with correlated or dependent objectives. This paper is concluded in Section VI, in which future research topics on many-objective optimization are also suggested.

II. Difficulties in Many-Objective Optimization

Let us consider the following k-objective problem:

$$\text{Maximize } \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})) \text{ subject to } \boldsymbol{x} \in \boldsymbol{X} \quad (1)$$

where $\boldsymbol{f}(\boldsymbol{x})$ is a k-dimensional objective vector, $f_i(\boldsymbol{x})$ is the ith objective to be maximized ($i = 1, 2, \ldots, k$), $\boldsymbol{x}$ is a decision vector, and $\boldsymbol{X}$ is the set of all feasible decision vectors (i.e., $\boldsymbol{X}$ is the feasible region in the decision space).

A solution $\boldsymbol{y}$ of the maximization problem is said to be dominated by another solution $\boldsymbol{x}$ if the following relations hold:

$$\forall i, f_i(\boldsymbol{y}) \le f_i(\boldsymbol{x}) \text{ and } \exists i, f_i(\boldsymbol{y}) < f_i(\boldsymbol{x}). \quad (2)$$

If $\boldsymbol{y}$ in $\boldsymbol{X}$ is not dominated by any other solution $\boldsymbol{x}$ in $\boldsymbol{X}$, $\boldsymbol{y}$ is called a Pareto optimal solution. A multiobjective problem usually has a large number of Pareto optimal solutions. The Pareto optimal solution set is the set of all Pareto optimal solutions. The projection of the Pareto optimal solution set onto the objective space is called the Pareto front. We also use the concept of nondominated solutions in a population (which is not equal to $\boldsymbol{X}$). If $\boldsymbol{y}$ is not dominated by any other solution $\boldsymbol{x}$ in a population, $\boldsymbol{y}$ is said to be nondominated in the population.

Difficulties in the handling of many-objective problems can be roughly classified into the following five categories.

1) Difficulties in the search for Pareto optimal solutions.
2) Difficulties in the approximation of the entire Pareto front.
3) Difficulties in the presentation of obtained solutions.
4) Difficulties in the choice of a single final solution.
5) Difficulties in the evaluation of search algorithms.

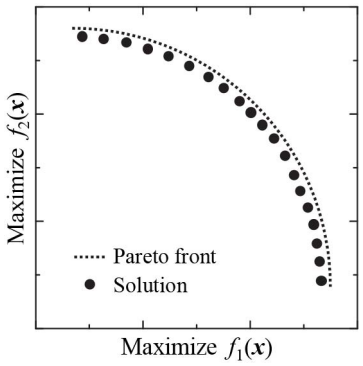

Fig. 1. Approximation of a tradeoff curve using 20 solutions.

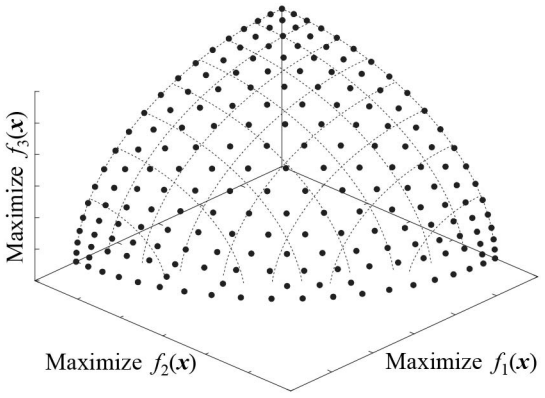

Fig. 2. Approximation of a tradeoff surface using 200 solutions.

Difficulties in 1) have been repeatedly reported. When our k-objective problem in (1) includes many objectives, the Pareto dominance relation in (2) is not likely to hold for many pairs of solutions $\boldsymbol{x}$ and $\boldsymbol{y}$. This is because $f_i(\boldsymbol{y}) \le f_i(\boldsymbol{x})$ is not likely to hold simultaneously for all of the k objectives when k is large. When all solutions in the current population are nondominated, strong selection pressure toward the Pareto front cannot be generated by the Pareto dominance relation in (2). As shown later in this paper, all solutions actually become nondominated in an early generation when EMO algorithms are applied to many-objective problems. As a result, the search ability of Pareto dominance-based algorithms is severely deteriorated.

Difficulties in 2) are related to the dimensionality of the Pareto front. If the Pareto front is a tradeoff curve in a 2-D objective space, less than 100 solutions may be enough for its good approximation as shown in Fig. 1 with 20 solutions. If it is a tradeoff surface in a 3-D objective space, hundreds of solutions may be needed as shown in Fig. 2 with 200 solutions. A huge number of solutions may be needed for a good approximation of the entire Pareto front in the k-dimensional objective space when k is large.

Difficulties in 3) and 4) are related to the interface between an EMO algorithm and a human decision maker. Solution visualization [29] is an important research topic for the handling of these difficulties. In this paper, we do not discuss the handling of the difficulties in 3) and 4) because the behavior of EMO algorithms is our main focus.

In EMO algorithms, the Pareto front is approximated by a fixed number of solutions. However, due to the difficulties in 2), it is almost impossible to obtain a dense distribution of nondominated solutions over the entire Pareto front in the case of many-objective optimization. Such a dense distribution may need an impractically huge number of solutions. Thus, the practical search strategy is somewhere between the following two extremes. One is the focused search for densely distributed nondominated solutions around a small part of the Pareto front. The other one is the global search for sparsely distributed nondominated solutions over the entire Pareto front. Preference-based search (see [30]–[34]) can be viewed as the focused search. Convergence improvement attempts for many-objective problems (see [10]–[15]) often change the behavior of EMO algorithms from the global search toward the focused search. Objective reduction [35] can be also viewed as the focused search. Scalarizing function-based algorithms (see MOEA/D) try to search for solutions over the entire Pareto front.

Difficulties in 5) are related to such a wide variety of search strategies for many-objective optimization. It may be inappropriate to use the same performance measure when we evaluate the performance of different EMO algorithms with different search strategies. In this paper, we examine the global search behavior of EMO algorithms. That is, we examine their search behavior when they search for sparsely distributed nondominated solutions over the entire Pareto front. In this case, the performance of an EMO algorithm is the quality of the obtained solution set to approximate the entire Pareto front. However, it is not easy to evaluate the difference between the obtained solution set and the Pareto front using distance-based performance measures such as the generational distance and the inverted generational distance (IGD) for many-objective optimization. This is because 1) Pareto optimal solutions of many-objective problems are usually unknown and 2) a huge number of Pareto optimal solutions are needed to calculate those measures in a reliable manner. In this paper, we use the hypervolume measure for performance evaluation because we do not know Pareto optimal solutions of our many-objective test problems.

Let us explain the difficulty in measuring the distance between an obtained solution set and the Pareto front using a limited number of Pareto optimal solutions on the Pareto front. For simplicity of visual explanation, we use a two-objective maximization problem whose Pareto front is the line between two points (0, 10) and (10, 0) in a 2-D objective space in Fig. 3 (see [36], [37] for more general discussions on distance-based measures). We also assume the use of the six Pareto optimal solutions (0, 10), (2, 8), (4, 6), (6, 4), (8, 2), and (10, 0) in Fig. 3(d) as a reference set to evaluate three solution sets A, B, and C in Fig. 3(a)–(c). In Fig. 3(a)–(c), solutions in each solution set are denoted by closed circles while open circles show solutions in the reference set in Fig. 3(d). The IGD is the average distance from each solution in the reference set to the nearest solution in a solution set, which is calculated for each of the three solution sets A, B, and C using the reference set D as

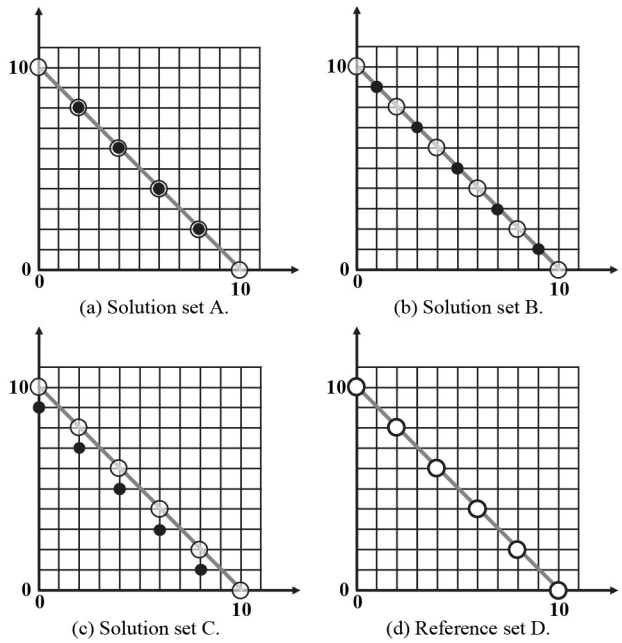

Fig. 3. Three solution sets A, B, and C to be evaluated using a reference set D.

$$IGD(\mathrm{A}) = (\sqrt{8}+0+0+0+0+\sqrt{8})/6 \cong 0.94$$
$$IGD(\mathrm{B}) = (\sqrt{2}+\sqrt{2}+\sqrt{2}+\sqrt{2}+\sqrt{2}+\sqrt{2})/6 \cong 1.41$$
$$IGD(\mathrm{C}) = (1+1+1+1+1+\sqrt{5})/6 \cong 1.21.$$

These calculation results show that B is evaluated as the worst among the three solution sets. However, B may be intuitively viewed as being the best among the three solution sets. From Fig. 3(b) and (c), we can see that the solution set C is dominated by the solution set B. Thus, B should be evaluated as being better than C. However, C has a smaller value of IGD than B. With respect to the comparison between A and B, B looks better than A since B has larger diversity than A on the Pareto front. However, A is evaluated as being better than B by the above-mentioned IGD values. That is, IGD is inconsistent with not only our intuition but also the Pareto dominance relation. These inconsistencies appear since the calculation of IGD is based on the reference set D with only six solutions. For example, if we use 11 Pareto optimal solutions (10, 0), (9, 1), ..., (0, 10) as the reference set D, IGD is calculated as follows for each solution set in Fig. 3(a)–(c)

$$IGD(\mathrm{A}) \cong 1.16, IGD(\mathrm{B}) \cong 0.77, IGD(\mathrm{C}) \cong 1.11.$$

In this case, B is evaluated as the best solution set. However, it is counter-intuitive that C is evaluated as being better than A.

As shown by this simple example, a large number of Pareto optimal solutions are needed in the reference set to obtain more reliable results. The question is how many solutions are needed to obtain reliable results from the distance calculation for many-objective problems. If we need 20 solutions for a single-dimensional Pareto front of a two-objective problem, we may need 20^2 solutions for a 2-D Pareto front of a three-objective problem in order to realize a similar resolution in the 3-D objective space. This discussion can be extended to

the case of a k-objective problem as $20^{(k-1)}$ solutions (see 20^9 for a 10-objective problem is 512 billion). As we have already explained using Figs. 1 and 2, a huge number of nondominated solutions are needed to approximate the entire Pareto front of a many-objective problem. As a result, a huge number of solutions are also needed as the reference set to calculate the distance between a solution set and the Pareto front in a reliable manner.

III. Test Problems and EMO Algorithms

A. Test Problems

As a two-objective test problem, we use the following two-objective 500-item knapsack problem of Zitzler and Thiele [28]:

$$\text{Maximize } \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}),\ f_2(\boldsymbol{x})) \tag{3}$$

$$\text{subject to} \sum_{j=1}^{500} b_{ij}x_j \le c_i,\ i = 1, 2 \tag{4}$$

$$x_j = 0 \text{ or } 1, j = 1, 2, \ldots, 500 \tag{5}$$

$$\text{where } f_i(\boldsymbol{x}) = \sum_{j=1}^{500} a_{ij}x_j,\ i = 1, 2. \tag{6}$$

In this formulation, $\boldsymbol{x}$ is a 500-dimensional binary vector, a_{ij} is the profit of item j according to knapsack i, b_{ij} is the weight of item j according to knapsack i, and c_i is the capacity of knapsack i (i = 1, 2 and j = 1, 2, ..., 500). The values of a_{ij} and b_{ij} were randomly specified as integers in the interval [10, 100] in [28]. The capacity c_i was specified as 50% of the sum of all weights related to each knapsack i in [28]. This two-objective 500-item knapsack problem is referred to as the 2–500 problem.

Test Problems with Randomly Generated Objectives: By randomly specifying the profit a_{ij} of each item j for each objective i as an integer in the interval [10, 100], we generate other eight objectives as

$$f_i(\boldsymbol{x}) = \sum_{j=1}^{500} a_{ij}x_j,\ i = 3, 4, \ldots, 10. \tag{7}$$

Using the randomly generated eight objectives and the two objectives of the 2–500 problem in (6), we generate 500-item knapsack problems with four, six, eight, and ten objectives (i.e., 4–500, 6–500, 8–500, 10–500). A k-objective 500-item problem has the first k objectives in (7): $f_i(\boldsymbol{x})$, i = 1, 2, ..., k. All of our 500-item test problems have the same constraint conditions in (4) and (5) independent of the number of objectives. This means that our 500-item problems have exactly the same feasible solution set. Moreover, their first and second objectives are the same as the two objectives of the 2–500 problem in (3)–(6). Thus, the Pareto optimal solutions of the 2–500 problem are also Pareto optimal in all the other 500-item problems in this paper.

In a similar manner, we generate test problems with 100, 1000, and 10 000 items. Each profit a_{ij} and each weight b_{ij} are randomly specified as integers in [10, 100] to generate ten objectives and two constraint conditions. The capacity c_i of each knapsack is specified as 50% of the total weight.

We also generate other types of k-objective 500-item test problems of the following form for k = 4, 6, 8, 10:

$$\text{Maximize } \boldsymbol{g}(\boldsymbol{x}) = (g_1(\boldsymbol{x}),\ g_2(\boldsymbol{x}),\ \ldots,\ g_k(\boldsymbol{x})) \tag{8}$$

subject to (4) and (5).

This problem has the same constraint conditions as the other 500-item problems. In the following, we explain how $g_i(\boldsymbol{x})$ is specified in different types of test problems.

Test Problems with Correlated Objectives: Using a real number parameter α in [0, 1], correlated objectives $g_i(\boldsymbol{x})$ are specified from the randomly generated objectives $f_i(\boldsymbol{x})$ as

$$g_i(\boldsymbol{x}) = f_i(\boldsymbol{x}),\ i = 1, 2 \tag{9}$$

$$g_i(\boldsymbol{x}) = \alpha \cdot f_1(\boldsymbol{x}) + (1 - \alpha) \cdot f_i(\boldsymbol{x}),\ i = 3, 5, 7, 9 \tag{10}$$

$$g_i(\boldsymbol{x}) = \alpha \cdot f_2(\boldsymbol{x}) + (1 - \alpha) \cdot f_i(\boldsymbol{x}),\ i = 4, 6, 8, 10. \tag{11}$$

The value of α can be viewed as the correlation strength. The minimum correlation strength is 0 (i.e., α = 0) where $g_i(\boldsymbol{x})$ is the same as the randomly generated objective $f_i(\boldsymbol{x})$. The maximum correlation strength is 1 (i.e., α = 1) where $g_i(\boldsymbol{x})$ is the same as $f_1(\boldsymbol{x})$ or $f_2(\boldsymbol{x})$. We examine four specifications of α: α = 0.2, 0.4, 0.6, 0.8.

Test Problems With Dependent Objectives: A k-objective problem with dependent objectives is generated from $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ as follows (k = 4, 6, 8, 10):

$$g_i(\boldsymbol{x}) = f_i(\boldsymbol{x}),\ i = 1, 2 \tag{12}$$

$$g_i(\boldsymbol{x}) = \alpha_{ik} \cdot f_1(\boldsymbol{x}) + (1 - \alpha_{ik}) \cdot f_2(\boldsymbol{x}),\ i = 3, 4, \ldots, k \tag{13}$$

where α_{ik} is specified as follows:

$$\alpha_{ik} = (i - 2)\big/k - 1,\ i = 3, 4, \ldots, k;\ k = 4, 6, 8, 10. \tag{14}$$

For example, the four-objective problem is specified as

$$g_1(\boldsymbol{x}) = f_1(\boldsymbol{x}),\ g_2(\boldsymbol{x}) = f_2(\boldsymbol{x}) \tag{15}$$

$$g_3(\boldsymbol{x}) = 1\big/3 \cdot f_1(\boldsymbol{x}) + 2\big/3 \cdot f_2(\boldsymbol{x}) \tag{16}$$

$$g_4(\boldsymbol{x}) = 2\big/3 \cdot f_1(\boldsymbol{x}) + 1\big/3 \cdot f_2(\boldsymbol{x}). \tag{17}$$

B. Examined EMO Algorithms

We examine the behavior of NSGA-II [5], MOEA/D [23], SMS-EMOA [24], and HypE [25] through computational experiments on the above-mentioned test problems.

NSGA-II is the most frequently-used Pareto dominance-based EMO algorithm in the literature. Its fitness evaluation is based on a rank assignment mechanism called "nondominated sorting" and a secondary measure for diversity maintenance called "crowding distance." An offspring population of size μ is generated by binary tournament selection, crossover, and mutation from a current population of size μ. These two populations are combined into a merged population of size (μ+μ). The next population is constructed by choosing the best μ solutions from the merged population. In NSGA-II, the same fitness evaluation mechanism is used in mating selection and environmental selection.

MOEA/D is an efficient scalarizing function-based EMO algorithm. A multiobjective problem is decomposed into a number of single-objective problems. Each single-objective

problem is defined by the same scalarizing function with a different weight vector. The number of the weight vectors is the same as the number of the single-objective problems, which is also the same as the population size. A single solution is stored for each single-objective problem.

A set of weight vectors $\boldsymbol{w} = (w_1, w_2, \ldots, w_k)$ is specified by the following relations for our k-objective problem:

$$w_1 + w_2 + \cdots + w_k = 1 \tag{18}$$

$$w_i \in \left\{0, \frac{1}{H}, \frac{2}{H}, \ldots, \frac{H}{H}\right\}, \quad i = 1, 2, \ldots, k \tag{19}$$

where H is a user defined positive integer. MOEA/D uses all weight vectors satisfying the relations in (18) and (19). For further discussions on weight vector specifications, see [38].

MOEA/D can be viewed as a cellular algorithm. Each cell has a different weight vector (i.e., a different single-objective problem). For each cell, a prespecified number of the nearest cells are defined as its neighbors. The number of neighbors is a user defined parameter called the neighborhood size. The distance between cells is measured by the Euclidean distance between their weight vectors, which is used to define neighbors.

A new solution at each cell is generated by applying crossover and mutation to a pair of parent solutions, which are randomly selected from its neighbors. Local mating is a feature of MOEA/D. The generated new solution for a cell is compared with the current solution at the cell. Only when the new solution is better, the current solution is replaced with the new one. The new solution is also compared with the current solution at each of its neighbors. The single-objective problem at each neighbor is used for this comparison. Local replacement is another feature of MOEA/D.

An important issue is the choice of a scalarizing function. As in the original paper of MOEA/D [23], we examine the weighted sum, the weighted Tchebycheff and the penalty-based boundary intersection (PBI) function.

The weighted sum is written for our k-objective problem using a weight vector $\boldsymbol{w} = (w_1, w_2, \ldots, w_k)$ as

$$f^{WS}(\boldsymbol{x}|\boldsymbol{w}) = w_1 \cdot f_1(\boldsymbol{x}) + \cdots + w_k \cdot f_k(\boldsymbol{x}) \tag{20}$$

which is to be maximized in MOEA/D.

The weighted Tchebycheff is written using the weight vector $\boldsymbol{w}$ and a reference point $\boldsymbol{z}^* = (z_1^*, z_2^*, \ldots, z_k^*)$ as

$$f^{TE}(\boldsymbol{x}|\boldsymbol{w}, \boldsymbol{z}^*) = \max_{i=1,2,\ldots,k} \{w_i \cdot |z_i^* - f_i(\boldsymbol{x})|\} \tag{21}$$

which is to be minimized in MOEA/D.

Zhang and Li [23] specified the reference point $\boldsymbol{z}^*$ in their computational experiments on knapsack problems as

$$z_i^* = 1.1 \cdot \max\{f_i(\boldsymbol{x}) | \boldsymbol{x} \in \Omega(1) \cup \Omega(2) \cup \cdots \cup \Omega(t)\}, \quad i = 1, 2, \ldots, k \tag{22}$$

where $\Omega(t)$ is the population at the tth generation. In this paper, we mainly use this specification while we also examine other specifications such as a fixed point (30 000, 30 000, ..., 30 000).

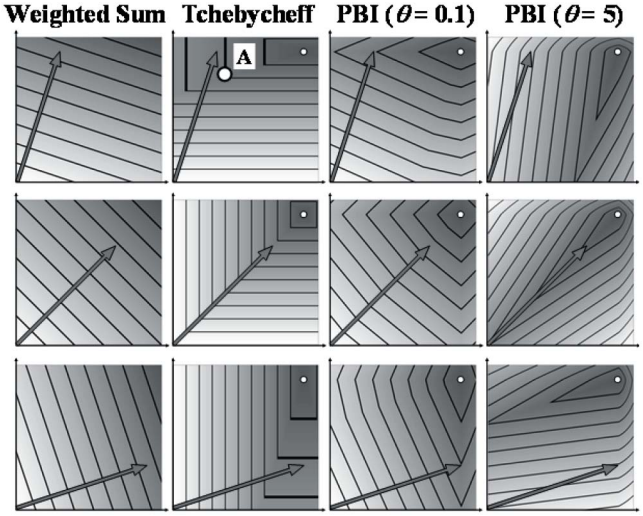

Fig. 4. Contour lines of the three scalarizing functions.

Using the weight vector $\boldsymbol{w}$ and the reference point $\boldsymbol{z}^*$, which are the same as in (21), the PBI function [23] is written as

$$f^{PBI}(\boldsymbol{x}|\boldsymbol{w}, \boldsymbol{z}^*) = d_1 + \theta\, d_2 \tag{23}$$

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

$$d_1 = \frac{\| (\boldsymbol{z}^* - f(\boldsymbol{x}))^T \boldsymbol{w} \|}{||\boldsymbol{w}||} \tag{24}$$

$$d_2 = \left\| f(\boldsymbol{x}) - (\boldsymbol{z}^* - d_1 \frac{\boldsymbol{w}}{||\boldsymbol{w}||}) \right\|. \tag{25}$$

The penalty parameter θ was specified as $\theta = 5$ in Zhang and Li [23]. We also examine the use of smaller penalty parameter values such as 0.01, 0.1, and 1 in addition to $\theta = 5$. The PBI function is to be minimized in MOEA/D. The definition of d_2 in (25) is slightly different from [23] where $\|\boldsymbol{w}\|$ was missing. However, (25) was used in the computational experiments in [23]. In this paper, we use the definition of d_2 in (25).

In Fig. 4, we show the contour lines of each scalarizing function for three weight vectors (0.8, 0.2), (0.5, 0.5), and (0.2, 0.8). The arrow in each plot in Fig. 4 shows the weight vector used in the plot. A scalarizing function with a different weight vector is used as the objective function in a different single-objective problem in MOEA/D. The contour lines in each plot in Fig. 4 show the landscape of the objective function in each single-objective problem. We can see from Fig. 4 that different scalarizing functions have totally different contour lines.

We can also see from Fig. 4 that each contour line of the weighted Tchebycheff has a right angle at its bottom-left corner. Let us assume that a current solution is on the corner of a contour line of the weighted Tchebycheff as shown by the open circle "A" in the top plot under "Tchebycheff" in Fig. 4. In this case, all solutions in the upper-right region of the current solution A are evaluated as being better than A by the weighted Tchebycheff. This is similar to the Pareto dominance-based evaluation. As a result, MOEA/D with the weighted Tchebycheff shows a similar behavior to Pareto dominance-based EMO algorithms as shown later in this paper.

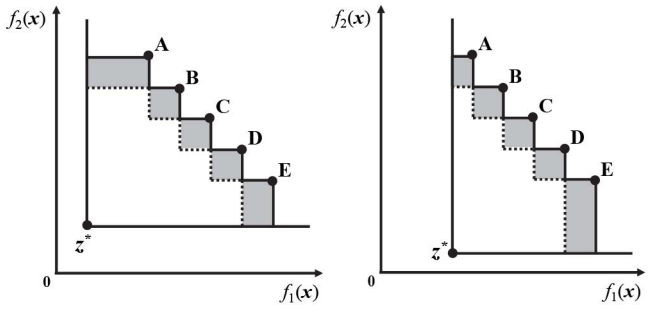

Fig. 5. Hypervolume contribution of each solution for different settings of the reference point z^*.

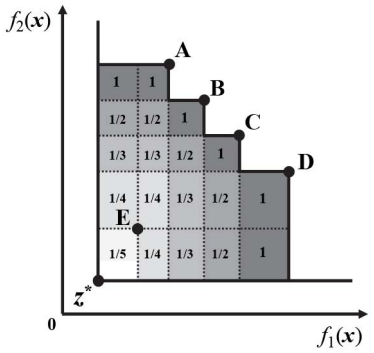

Fig. 6. Hypervolume contribution-based fitness evaluation of each solution in HypE.

Recently, various ideas have been proposed to improve the search ability of MOEA/D. One idea is adaptive weight vector specifications [39]–[41]. Another idea is the decomposition into a number of simpler multiobjective problems [42]. While these ideas look promising for many-objective problems, we use the above-mentioned basic MOEA/D in this paper.

SMS-EMOA is a hypervolume-based EMO algorithm with a (μ+1)ES-style generation update framework. Its high search ability for many-objective problems has been demonstrated in the literature [43]. The basic idea of SMS-EMOA is to search for a solution set with the maximum hypervolume.

In SMS-EMOA, two parents are randomly selected from a current population of size μ to generate a single solution by crossover and mutation. The next population with μ solutions is constructed by removing the worst solution from the merged population with (μ+1) solutions.

In the same manner as in NSGA-II, a rank is assigned to each solution in the merged population. Each solution with the same rank is evaluated by its hypervolume contribution. Let S be a set of solutions with the same rank. We denote the set of all solutions in S excluding a solution $\boldsymbol{x}_i$ in S by $S\backslash\boldsymbol{x}_i$. We also denote the hypervolume of the solution set S by $HV(S)$. The hypervolume contribution of $\boldsymbol{x}_i$ to $HV(S)$ is defined as

$$Contribution(\boldsymbol{x}_i|S) = HV(S) - HV(S\backslash\boldsymbol{x}_i). \tag{26}$$

As shown in Fig. 5, we need a reference point $\boldsymbol{z}^*$ for the calculation of the hypervolume contribution of each solution. In the case of two-objective optimization, the location of the reference point $\boldsymbol{z}^*$ has no effects on the hypervolume contribution of each solution except for the two extreme solutions (i.e., A and E in Fig. 5) if $\boldsymbol{z}^*$ is dominated by all solutions. However, its location has significant effects on more solutions in many-objective optimization. In this paper, we examine some different specifications of the reference point.

If all solutions are nondominated in the merged population, each solution is evaluated by the hypervolume contribution. In this case, the removal of the worst solution from the merged population is the same as the selection of μ solutions with the maximum hypervolume.

We use a fast hypervolume calculation method by While *et al*. [44] in SMS-EMOA (see [44], [45] for fast hypervolume calculation). However, it is still difficult to apply SMS-EMOA to many-objective problems with more than six objectives due to its heavy computation load. Thus, we also use HypE [25] as another hypervolume-based EMO algorithm. HypE and SMS-EMOA have similar but different fitness evaluation mechanisms as explained in the following.

In HypE, Monte Carlo simulation is used for approximate hypervolume calculation. The number of sampling points is a user defined parameter. In computational experiments, we always use the approximate hypervolume calculation in HypE-independent of the number of objectives.

In SMS-EMOA, mating selection is performed randomly. The hypervolume contribution of each solution is calculated as shown in Fig. 5 for environmental selection in SMS-EMOA. In HypE, a different definition of the hypervolume contribution is used for mating selection. When a region is dominated by n solutions, $1/n$ of its hypervolume is assigned to each of the n solutions as shown in Fig. 6. Such a hypervolume sharing mechanism is applied to not only nondominated solutions but also dominated ones (see E in Fig. 6). The nondominated sorting is not used for mating selection. The sum of the shared hypervolume values assigned to each solution is used as its fitness value in binary tournament selection in HypE.

HypE has a (μ+μ) ES-style generation update mechanism. As in NSGA-II, first the nondominated sorting is applied to the (μ+μ) solutions in the merged population. The best μ solutions are selected according to the rank of each solution. When a part of solutions with the same rank should be selected, selection is performed so that the hypervolume of the selected solutions is maximized. That is, the nondominated sorting and the hypervolume maximization are used as the primary and the secondary criterion in environmental selection. Solution selection for the hypervolume maximization is performed in an approximate manner in HypE (see [25] for details).

In Table I, we summarize the characteristics of each EMO algorithm. As shown in Table I, NSGA-II uses the same fitness evaluation mechanism for mating selection and environmental selection. However, MOEA/D and SMS-EMOA use their fitness evaluation mechanisms only for environmental selection. Random mating selection is performed locally in MOEA/D while it is performed globally in SMS-EMOA. In HypE, different fitness evaluation mechanisms are used for mating selection and environmental selection. As shown in Table I, the four algorithms are different in their basic structures, mating selection mechanisms and environmental selection mechanisms.

TABLE I
CHARACTERISTICS OF EACH EMO ALGORITHM

EMO Algorithm	Structure	Mating selection	Environmental selection
NSGA-II	(μ+μ)	Pareto ranking and crowding distance	Pareto ranking and crowding distance
MOEA/D	Cellular	Random (from neighbors)	Scalarizing function at each cell
SMS-EMOA	(μ+1)	Random (from all solutions)	Pareto ranking and Hypervolume contribution
HypE	(μ+μ)	Shared hypervolume contribution	Pareto ranking and hypervolume

For comparison, we examine the performance of multiple runs of the focused search by NSGA-II with a small population. This approach searches for nondominated solutions around the center of the Pareto front and around the best solution with respect to each objective. For our k-objective problem, the population is divided into (k+1) subpopulations of the same size. One subpopulation is used for the search around the center of the Pareto front. Each of the other k subpopulations is used for the search around the best solution of each objective.

For the search around the center of the Pareto front, we apply NSGA-II to a k-objective problem with the following k objectives and the original constraint conditions in (4) and (5):

$$h_i(\boldsymbol{x}) = \sum_{j=1}^{k} f_j(\boldsymbol{x}) + \varepsilon f_i(\boldsymbol{x}), \quad i = 1,\ 2,\ \ldots,\ k \tag{27}$$

where ε is a small positive real number parameter. The ith objective $h_i(\boldsymbol{x})$ is slightly biased toward the original ith objective $f_i(\boldsymbol{x})$ from the sum of all objectives. The value of ε is specified as $\varepsilon = 0.1$ in our computational experiments. For example, $h_i(\boldsymbol{x})$ is specified as $h_1(\boldsymbol{x}) = 1.1f_1(\boldsymbol{x}) + f_2(\boldsymbol{x})$ and $h_2(\boldsymbol{x}) = f_1(\boldsymbol{x}) + 1.1f_2(\boldsymbol{x})$ for two-objective optimization.

The following relation holds from (27), which shows that the focused search is performed toward the best solution with respect to the sum of the original k objectives:

$$\sum_{j=1}^{k} h_j(\boldsymbol{x}) = (k+\varepsilon)\sum_{j=1}^{k} f_j(\boldsymbol{x}). \tag{28}$$

For the search around the best solution of $f_i(\boldsymbol{x})$, we apply NSGA-II to a k-objective problem with the following k objectives ($\varepsilon = 0.1$) and the constraint conditions in (4) and (5):

$$h_i(\boldsymbol{x}) = f_i(\boldsymbol{x}) - \varepsilon \sum_{\substack{j=1 \\ j\neq i}}^{k} f_j(\boldsymbol{x}) \tag{29}$$

$$h_j(\boldsymbol{x}) = f_i(\boldsymbol{x}) + \varepsilon f_j(\boldsymbol{x}),\ j = 1,\ 2,\ \ldots,\ k\ (j \neq i). \tag{30}$$

For example, the four objectives are specified for the search around the best solution of $f_4(\boldsymbol{x})$ as $h_1(\boldsymbol{x}) = f_4(\boldsymbol{x}) + 0.1f_1(\boldsymbol{x})$, $h_2(\boldsymbol{x}) = f_4(\boldsymbol{x}) + 0.1f_2(\boldsymbol{x})$, $h_3(\boldsymbol{x}) = f_4(\boldsymbol{x}) + 0.1f_3(\boldsymbol{x})$, and $h_4(\boldsymbol{x}) = f_4(\boldsymbol{x}) - 0.1f_1(\boldsymbol{x}) - 0.1f_2(\boldsymbol{x}) - 0.1f_3(\boldsymbol{x})$ for the case of $k = 4$.

The following relation holds from (29) and (30), which shows that the focus of the search is the best solution of $f_i(\boldsymbol{x})$, as follows:

$$\sum_{j=1}^{k} h_j(\boldsymbol{x}) = k f_i(\boldsymbol{x}). \tag{31}$$

This approach is referred to as F-NSGA-II (focused NSGA-II) in this paper. We examine its two versions. In one version, NSGA-II is independently executed at each subpopulation. The subpopulations after the execution of NSGA-II are merged into a single population, which is handled as the obtained final population. This version is called F100-NSGA-II since 100% of the computation time is used for the focused search. In the other version, the execution of NSGA-II at each subpopulation is terminated when 90% of the computation time is used. Then NSGA-II is applied to the merged population for the remaining computation time. This version is called F90-NSGA-II.

IV. RESULTS ON RANDOMLY GENERATED TEST PROBLEMS

A. Setting of Computational Experiments

We use the following parameter specifications in all the four EMO algorithms for our 500-item test problems.

1) Coding: Binary string of length 500.
2) Termination condition: 400 000 solution evaluations.
3) Crossover probability: 0.8 (Uniform crossover).
4) Mutation probability: 2/500 (Bit-flip mutation).
5) Population size: 100 (2–500), 120 (4–500), 126 (6–500), 120 (8–500), 220 (10–500).

These specifications of the population size are based on the combinatorial nature of the possible number of weight vectors in MOEA/D in (18) and (19). As a result, a different specification of the population size is used for each test problem.

In MOEA/D, we use the following specifications.

1) Neighborhood size: 10 neighbors.
2) Scalarizing function: Weighted sum, Weighted Tchebycheff, PBI (θ = 0.01, 0.05, 0.1, 0.5, 1, 5).
3) Reference point: $\boldsymbol{z}^*$ in (22).

In SMS-EMOA and HypE, the origin of the objective space is used as the reference point. In HypE, the number of sampling points is specified as 10 000.

During the execution of each EMO algorithm on the 2–500 problem, an infeasible solution is transformed to a feasible one by removing items in an ascending order of the following values until the two constraint conditions in (4) are satisfied:

$$q_j = \max\{a_{ij}/b_{ij} | i = 1,\ 2\}, j = 1,\ 2,\ \ldots,\ 500. \tag{32}$$

This constraint handling is the same as the greedy repair used in Zitzler and Thiele [28] on the 2–500 problem. Exactly the same greedy repair based on (32) is used for all the other 500-item test problems in this paper independent of the number of objectives. That is, a feasible solution is generated from an infeasible one by removing items using the order of items specified by (32). We can use the same greedy repair for all 500-item test problems because they have the same constraint

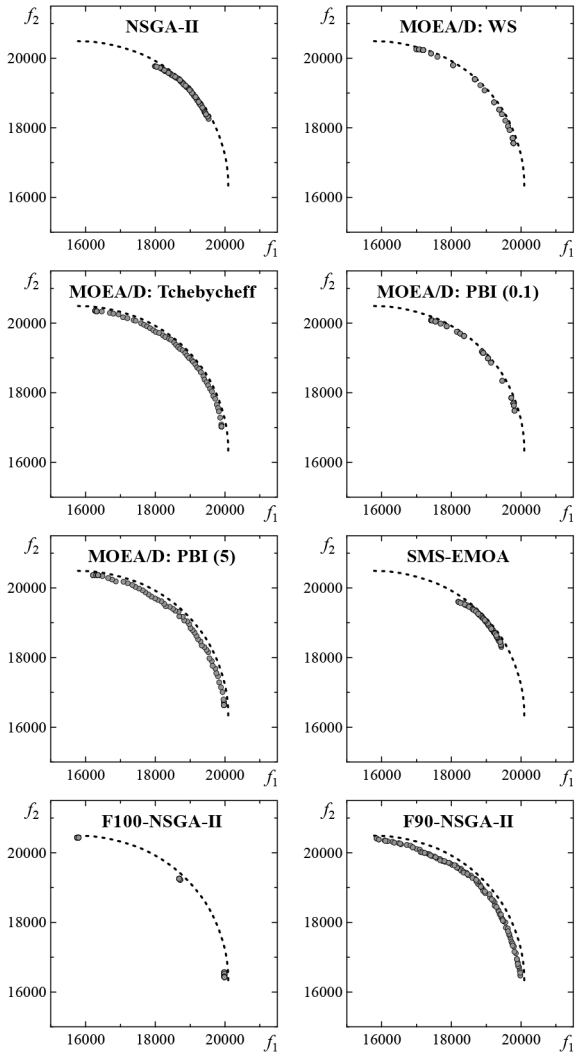

Fig. 7. Results of a single run on the 2–500 problem.

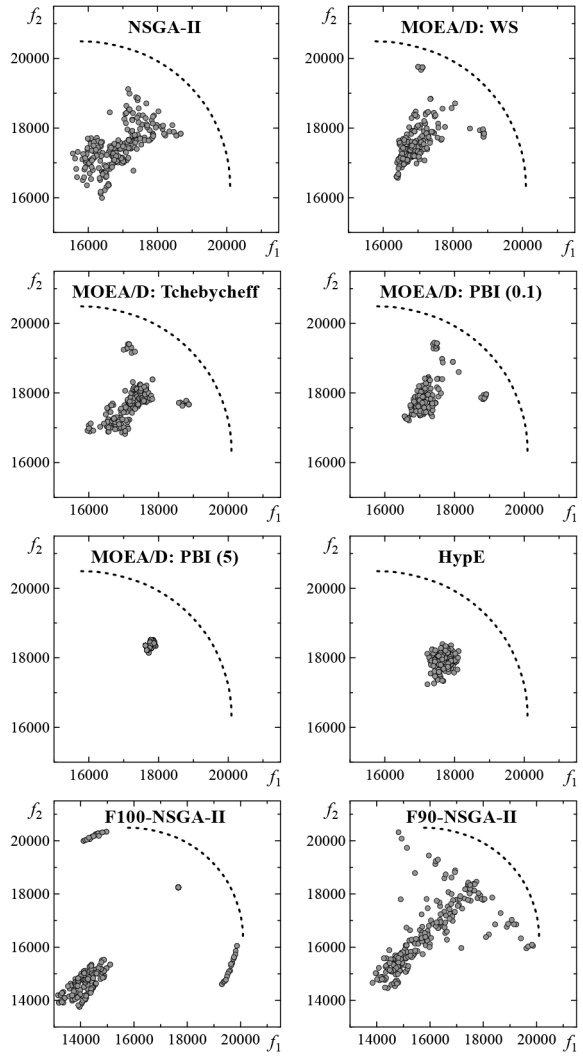

Fig. 8. Results on the 10–500 problem (population size 220).

conditions in (4) and (5). We also use the greedy repair for the other test problems. The removing order of items is specified in the same manner as in (32) independent of the number of items.

B. Visual Examination of Each EMO Algorithm

In Fig. 7, we show all solutions in the final generation in a single run of each EMO algorithm on the 2–500 problem. The dotted line in each plot shows the true Pareto front of the 2–500 problem. A single run in each plot in Fig. 7 is selected from 100 runs of each EMO algorithm in the following manner. First the hypervolume of all solutions in the final generation is calculated for each run using the origin (0, 0) of the objective space as a reference point. Next the average hypervolume value is calculated over 100 runs. Then a single run with the closest hypervolume to the average value is selected for Fig. 7. When we choose a single run from multiple runs in this paper, we always use this hypervolume-based criterion.

With respect to the convergence of solutions toward the Pareto front (dotted line) in Fig. 7, we can observe that MOEA/D with PBI ($\theta = 5$), F-100-NSGA-II, and F-90-NSGA-II are somewhat inferior to the other EMO algorithms. However, extreme nondominated solutions around the two edges of the Pareto front are obtained only by those three EMO algorithms. These observations show the convergence-diversity tradeoff relation in evolutionary multiobjective search by EMO algorithms.

Experimental results of NSGA-II in Fig. 7 are consistent with reported results in the literature (see Sato *et al.* [10]). The search behavior of NSGA-II with strong convergence and weak diversification on the 2–500 problem is due to the dominant effect of the nondominated sorting on the fitness evaluation. Experimental results by F100-NSGA-II show that the focused search works as expected. That is, F100-NSGA-II focuses its multiobjective search on the center and the two edges of the Pareto front. In F90-NSGA-II, the standard NSGA-II is used in the final 10% generations. As a result, many solutions are found by F90-NSGA-II between the center and the two edges of the Pareto front in Fig. 7.

In Fig. 8, we show all solutions in the final generation in a single run of each EMO algorithm on our randomly generated

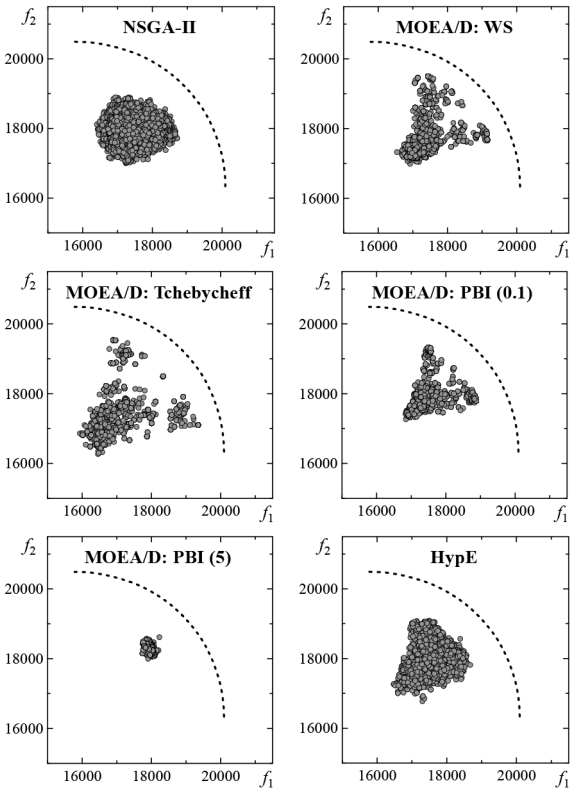

Fig. 9. Results on the 10–500 problem (large population).

10-objective test problem (10–500) in the 2-D subspace with $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$. HypE is used in Fig. 8 instead of SMS-EMOA. Solutions in each plot in Fig. 8 can be viewed as the projection of solutions in the ten-dimensional objective space onto the 2-D subspace.

As we explained in Section III-A, the Pareto front of the 2–500 problem (i.e., the dotted line in Fig. 7) is a part of the Pareto front of the 10–500 problem since the 2–500 and 10–500 problems have the same constraint conditions and the same first and second objectives $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$. However, no solutions close to the dotted line are obtained by the three EMO algorithms in Fig. 8. From this observation, we can see that it is very difficult to search for the entire Pareto front of the 10–500 problem. In Fig. 8, the largest diversity of solutions is obtained by F90-NSGA-II. This observation suggests the potential usefulness of the focused search for many-objective knapsack problems with respect to the diversity of solutions.

One may think that a much larger population is needed for the 10–500 problem. In Fig. 9, the population size is specified as 5005 for MOEA/D and 5000 for the other algorithms (it was 220 in Fig. 8). The neighborhood size in MOEA/D is 100 in Fig. 9. It should be noted that we always use the same termination condition (i.e., 400 000 solution evaluations). Thus, a larger population means a smaller number of generations.

Except for MOEA/D (Tchebycheff) and HypE, we cannot observe a clear increase in the diversity of solutions in Fig. 9 from Fig. 8. Many solutions seem to concentrate around the center of each population in Fig. 9. These observations suggest

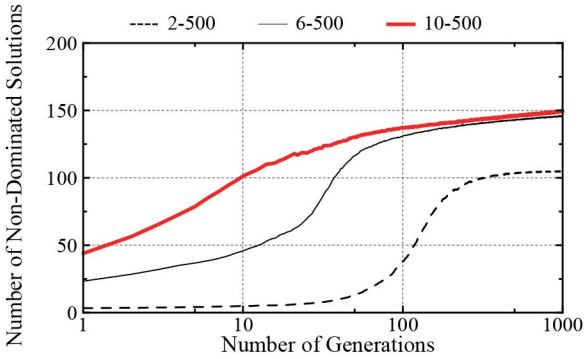

Fig. 10. Average number of nondominated solutions in the merged population in NSGA-II (population size 100).

the necessity of a strong diversification mechanism in EMO algorithms even on many-objective knapsack problems.

For examining the number of nondominated solutions at each generation, we apply NSGA-II with population size 100 to the 2–500, 6–500, and 10–500 problems. In NSGA-II with this setting, an offspring population of size 100 is generated from the current population of size 100. The next population is the best 100 solutions in the merged population with 200 solutions.

As in Sato *et al.* [10], we show the average number of nondominated solutions in the merged population in Fig. 10 over 1000 runs. The number of nondominated solutions becomes more than 100 in very early generations when NSGA-II is applied to the 10–500 problem. In this situation, Rank 1 is assigned to all solutions. As a result, no selection pressure is given toward the Pareto front in mating selection.

It is not easy to visually demonstrate the deterioration of the convergence property of NSGA-II by the increase in the number of objectives in a high-dimensional objective space. Since we do not know the true Pareto fronts of our test problems, the use of distance-based measures is also difficult. We use the following simple measure to evaluate the convergence of solutions toward the center of the Pareto front since its meaning is clear even for many-objective problems:

$$\text{MaxSum}(\Omega) = \max_{\boldsymbol{x}\in\Omega(t)} \sum_{i=1}^{k} f_i(\boldsymbol{x}) \tag{33}$$

where $\Omega(t)$ is a population at the tth generation. MaxSum($\Omega(t)$) is the maximum sum of the k objectives over all solutions at the tth generation, which shows the progress of search toward the Pareto front along the direction $(1, 1, \ldots, 1)$.

We normalize it as $100 \times \text{MaxSum}(\Omega(t))/\text{MaxSum}(\Omega(1))$ where MaxSum($\Omega(1)$) is the value at the first generation. Experimental results corresponding to Fig. 10 are shown in Fig. 11 where the convergence property of NSGA-II seems to be degraded by the increase in the number of objectives.

For comparison, we also apply NSGA-II to the three test problems in Fig. 11 after modifying their objectives as

$$h_i(\boldsymbol{x}) = \sum_{j=1}^{k} f_j(\boldsymbol{x}) + f_i(\boldsymbol{x}), \quad i = 1,\ 2,\ \ldots,\ k. \tag{34}$$

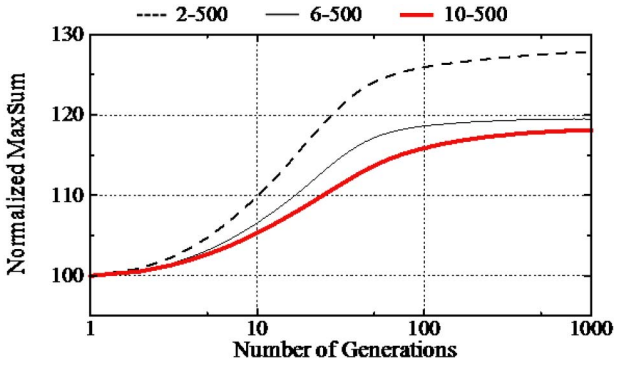

Fig. 11. Normalized MaxSum at each generation of NSGA-II.

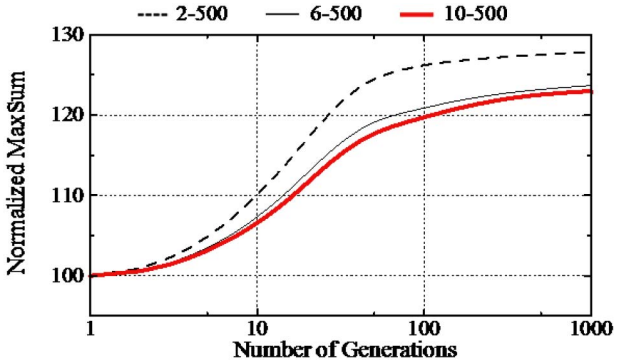

Fig. 12. Results of the focused search using NSGA-II.

This modification is to focus the search around the center of the Pareto front. Experimental results are shown in Fig. 12 where the calculation is performed using the original objectives. Comparison between Figs. 11 and 12 shows clear deterioration of the convergence property of NSGA-II by the increase in the number of the randomly generated objectives.

We compare 100 solutions from each run in Fig. 11 with those in Fig. 12. That is, we examine whether each solution in Fig. 11 is dominated by at least one solution in Fig. 12. Average results over 1000 runs are as follows: 15.23 (2–500), 53.58 (6–500), and 65.43 (10–500) solutions in Fig. 11 are dominated by at least one solution in Fig. 12 on average. We also obtain the following results: 24.70 (2–500), 0.00 (6–500), and 0.00 (10–500) solutions in Fig. 12 are dominated by at least one solution in Fig. 11 on average. These results show that the convergence property of NSGA-II is deteriorated in Fig. 11 by the increase in the number of the randomly generated objectives.

We also examine the number of nondominated solutions in the merged population for a large population of size 5000. Average results over 10 runs are shown in Fig. 13. Whereas we use a very large population, all solutions in the current population become nondominated within 100 generations when NSGA-II is applied to the 6–500 and 10–500 problems.

C. Performance Examination of Each EMO Algorithm

In this subsection, we examine the performance of each EMO algorithm using the hypervolume. As the reference point for hypervolume calculation, we use two specifications. One

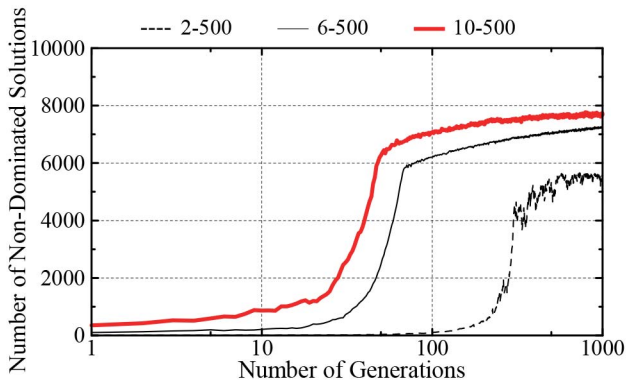

Fig. 13. Average number of nondominated solutions in the merged population in NSGA-II (population size 5000).

is the origin (0, 0, ..., 0) of the objective space, which is far from the Pareto front. Thus, the diversity of solutions has a large effect on hypervolume calculation. The other one is (15 000, 15 000, ..., 15 000), which is much closer to the Pareto front. Thus, both convergence and diversity have large effects.

Tables II and III show average results over 100 runs (except for HypE due to its heavy computation load). The same parameter specifications as in Figs. 7 and 8 are used. In each table, the average hypervolume is normalized using the average result of MOEA/D with the weighted sum. The origin (0, ..., 0) is used as the reference point in Table II while (15 000, ..., 15 000) is used in Table III. It should be noted that the origin is always used as the reference point for fitness evaluation in SMS-EMOA and HypE in both tables. The best result for each problem is shown by bold face.

In Tables II and III, we can observe clear performance deterioration of NSGA-II by the increase in the number of objectives. The deterioration is more clearly observed in Table III where the reference point is close to the Pareto front. This observation shows that the convergence is severely deteriorated. We can also observe performance deterioration of MOEA/D with the weighted Tchebycheff and PBI ($\theta = 5$). It is interesting to note that good results on the 2–500 problem are obtained from these two versions of MOEA/D in Tables II and III.

When we use a very small value for the penalty parameter in MOEA/D with PBI [i.e., MOEA/D: PBI (0.01) in Tables II and III], its experimental results become similar to those of MOEA/D with the weighted sum (i.e., MOEA/D: WS). This is because the PBI function becomes similar to the weighted sum by decreasing the penalty parameter θ to zero [see the formulation of PBI in (23)–(25)]. In Table III, good results are obtained from MOEA/D with PBI when θ is specified as $\theta = 0.1$. In the remaining of this paper, we use $\theta = 0.1$ in addition to the original setting $\theta = 5$ in MOEA/D with PBI.

Fig. 14 shows the distribution (histogram) of the obtained 100 hypervolume values before the normalization in Table II on the 10–500 problem by NSGA-II, MOEA/D with the weighted sum and the weighted Tchebycheff and HypE. Clear differences exist among the four algorithms in Fig. 14. For example, all the 100 hypervolume values by the MOEA/D

TABLE II
RELATIVE AVERAGE HYPERVOLUME. THE REFERENCE POINT (0, 0, . . ., 0) IS FAR FROM THE PARETO FRONT

EMO Algorithm	2-500	4-500	6-500	8-500	10-500
NSGA-II	96.5	86.2	77.8	72.0	65.5
MOEA/D: WS	100.0	**100.0**	**100.0**	**100.0**	**100.0**
MOEA/D: Tchebycheff	100.7	99.7	94.0	90.1	87.7
MOEA/D: PBI (0.01)	99.9	99.7	99.8	**100.0**	99.7
MOEA/D: PBI (0.05)	99.4	98.6	98.5	98.4	98.5
MOEA/D: PBI (0.1)	98.8	96.9	96.6	96.0	95.9
MOEA/D: PBI (0.5)	92.7	82.7	79.6	77.0	74.3
MOEA/D: PBI (1.0)	96.1	78.1	68.0	66.0	63.0
MOEA/D: PBI (5)	100.9	89.3	73.8	67.4	61.9
SMS-EMOA	95.2	91.8	88.4	-	-
HypE	97.2	94.1	92.8[1)]	93.9[2)]	92.5
F100-NSGA-II	100.0	94.7	87.2	79.9	68.2
F90-NSGA-II	**101.0**	98.2	92.4	86.5	82.2

1) 50 runs, 2) 30 runs

TABLE III
RELATIVE AVERAGE HYPERVOLUME. THE REFERENCE POINT (15 000, 15 000, . . ., 15 000) IS CLOSE TO THE PARETO FRONT

EMO Algorithm	2-500	4-500	6-500	8-500	10-500
NSGA-II	91.8	48.7	21.7	8.0	7.1
MOEA/D: WS	100.0	100.0	100.0	100.0	100.0
MOEA/D: Tchebycheff	**102.1**	75.8	60.1	74.1	75.3
MOEA/D: PBI (0.01)	99.7	100.5	103.3	105.9	107.3
MOEA/D: PBI (0.05)	98.8	103.1	112.3	127.1	136.1
MOEA/D: PBI (0.1)	97.2	**104.2**	**121.4**	**150.1**	169.5
MOEA/D: PBI (0.5)	79.1	71.9	105.5	147.0	**173.9**
MOEA/D: PBI (1.0)	90.4	57.0	55.1	76.6	87.5
MOEA/D: PBI (5)	100.9	91.7	80.6	85.3	80.8
SMS-EMOA	87.8	93.5	103.9	-	-
HypE	92.3	85.3	103.0[1)]	140.4[2)]	166.1
F100-NSGA-II	82.9	38.6	31.1	30.8	20.1
F90-NSGA-II	100.5	39.8	19.3	15.6	20.5

1) 50 runs, 2) 30 runs

with the weighted sum are larger than all the 100 values by HypE.

In Table II, the best results are obtained by MOEA/D with the weighed sum for the problems with 4–10 objectives. In Table III, the best results are obtained by MOEA/D with PBI (θ=0.1 and θ=0.5) for those problems. These observations suggest that PBI (θ=0.1 and θ=0.5) has high convergence ability while the weighted sum has high diversification ability. Good results are also obtained by HypE in Table III, which suggests its high convergence ability. In many cases, better results are obtained from the focused search (i.e., F100-NSGA-II and F90-NSGA-II) than the original NSGA-II.

We also examine the performance of each EMO algorithm using large population size: 5000 (2–500), 4960 (4–500), 4368 (6–500), 6435 (8–500), and 5005 (10–500). The neighborhood size is specified as 100 in MOEA/D. Except for these specifications, we use the same setting as in Tables II and III. Average results over 100 runs are summarized in Tables IV and V (except for SMS-EMOA and HypE). Bold face results

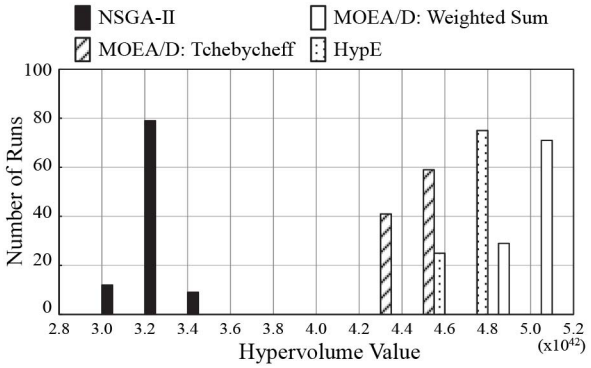

Fig. 14. Distribution of the obtained 100 hypervolume values for the 10–500 problem. The reference point is the origin.

TABLE IV
RELATIVE AVERAGE HYPERVOLUME (LARGE POPULATION). THE REFERENCE POINT IS (0, 0, . . ., 0)

EMO Algorithm	2-500	4-500	6-500	8-500	10-500
NSGA-II	93.4	78.0	70.7	62.3	58.9
MOEA/D: WS	99.4	99.0	99.3	**100.7**	**102.3**
MOEA/D: Tchebycheff	100.5	**103.0**	**102.2**	**101.1**	**93.3**
MOEA/D: PBI (0.1)	98.4	95.3	93.7	93.2	93.5
MOEA/D: PBI (5)	100.6	**95.4**	**78.6**	**71.9**	**65.4**
SMS-EMOA	92.5	-	-	-	-
HypE	94.8	84.4	82.3	75.2	78.7
F100-NSGA-II	98.4	93.7	**90.6**	**82.8**	**74.2**
F90-NSGA-II	98.3	95.6	**96.7**	**89.8**	**87.6**

TABLE V
RELATIVE AVERAGE HYPERVOLUME (LARGE POPULATION). THE REFERENCE POINT IS (15 000, 15 000, . . ., 15 000)

EMO Algorithm	2-500	4-500	6-500	8-500	10-500
NSGA-II	80.3	47.4	**43.9**	**24.0**	**16.2**
MOEA/D: WS	96.7	**114.0**	**160.9**	**232.3**	**255.9**
MOEA/D: Tchebycheff	100.6	**109.6**	**91.4**	56.6	26.6
MOEA/D: PBI (0.1)	94.6	**108.4**	**156.9**	**229.7**	**271.8**
MOEA/D: PBI (5)	100.4	**109.8**	**106.5**	**115.8**	**109.6**
SMS-EMOA	77.2	-	-	-	-
HypE	84.8	72.0	97.8	86.8	123.1
F100-NSGA-II	**88.5**	**39.7**	29.9	17.4	15.2
F90-NSGA-II	92.2	**62.6**	**51.8**	**23.4**	17.0

show the performance improvement by the use of a large population from Tables II and III. In Tables IV and V, the number of runs of SMS-EMOA and HypE is limited due to their heavy computation load with a large population: 2 runs (2–500) of SMS-EMOA, and 100 runs (2–500), 36 runs (4–500), 12 runs (6–500), 4 runs (8–500), and 3 runs (10–500) of HypE.

As in Tables II and III, the performance of NSGA-II is deteriorated by the increase in the number of objectives in Tables IV and V with large populations. Since the total number of solution examinations is prespecified as 400 000 in our computational experiments, a larger population leads to a smaller number of generations, which has negative effects on the search ability of the EMO algorithms. At the same time,

TABLE VI
COMPUTATION TIME FOR SMALL POPULATION (SEC.)

EMO Algorithm	2-500	4-500	6-500	8-500	10-500
NSGA-II	23.7	41.1	61.3	65.6	132.0
MOEA/D: WS	15.9	18.8	20.5	22.3	23.8
MOEA/D: Tchebycheff	16.4	18.8	21.3	22.5	24.0
MOEA/D: PBI (0.1)	16.2	18.6	20.9	22.8	25.0
MOEA/D: PBI (5)	16.2	18.6	20.9	22.2	24.3
SMS-EMOA	136.7	9717	63910	-	-
HypE	570.4	1819	3925	5001	10744
F100-NSGA-II	17.8	18.2	19.2	18.3	20.6
F90-NSGA-II	18.7	20.7	22.7	23.6	30.3

TABLE VII
COMPUTATION TIME FOR LARGE POPULATION (SEC.)

EMO Algorithm	2-500	4-500	6-500	8-500	10-500
NSGA-II	567.8	963.2	2776	5801	7738
MOEA/D: WS	79.1	83.9	67.1	164.5	93.3
MOEA/D: Tchebycheff	78.6	84.9	68.8	167.0	96.8
MOEA/D: PBI (0.1)	83.5	90.5	74.4	174.6	103.9
MOEA/D: PBI (5)	84.0	90.5	74.7	174.3	103.7
SMS-EMOA	227745	-	-	-	-
HypE	1720	6412	19244	57124	76367
F100-NSGA-II	145.3	115.1	92.1	122.0	90.4
F90-NSGA-II	322.8	401.0	523.8	972.9	1140

the increase in the population size has positive effects. As a result, the use of large populations deteriorates some experimental results in Tables IV and V while it improves other results.

In Table VI, we show the average computation time in our computational experiments in Tables II and III with small populations. We also show the average computation time in Table VII for the case of large populations. It should be noted that the same number of solutions were always examined in each algorithm in Tables VI and VII. We can see from Table VI that the increase in the number of objectives severely increases the computation time of SMS-EMOA and HypE. The use of large populations also increases their computation time. The computation time of NSGA-II is also increased by the use of large populations. This is because a larger population needs a longer computation time for the nondominated sorting. Thanks to the use of scalarizing functions, the computation time of MOEA/D is not severely increased by the increase in the number of objectives or the increase in the population size.

To examine the effect of the reference point specification, we examine three settings for MOEA/D: (21 000, ..., 21 000), (30 000, ..., 30 000), and (40 000, ..., 40 000). The first setting is close to the Pareto front, and the third one is far from the Pareto front. For SMS-EMOA and HypE, we also examine three settings: (10 000, ..., 10 000), (0, ..., 0), and $(-10000, \ldots, -10000)$. The first setting is close to the Pareto front, and the third one is far from the Pareto front. Experimental results are summarized in Tables VIII and IX where the best result among the three settings is shown by bold face. It should be noted that the reference point for fitness evaluation is different from that for performance evaluation of each algorithm in each table.

TABLE VIII
RELATIVE AVERAGE HYPERVOLUME BY THE EMO ALGORITHMS WITH VARIOUS SETTINGS OF THE REFERENCE POINT. THE REFERENCE POINT FOR PERFORMANCE EVALUATION IS (0, 0, ..., 0)

EMO Algorithm	2-500	4-500	6-500	8-500	10-500
Tchebycheff (21,000)	**100.6**	**98.9**	**93.2**	**88.3**	**85.9**
Tchebycheff (30,000)	**100.6**	96.5	91.4	86.7	83.1
Tchebycheff (40,000)	100.4	94.7	91.4	86.7	83.1
PBI (0.1) (21,000)	**98.8**	**97.0**	**96.7**	**96.3**	**96.0**
PBI (0.1) (30,000)	**98.8**	**97.0**	96.5	96.2	95.9
PBI (0.1) (40,000)	**98.8**	96.8	**96.7**	96.1	**96.0**
SMS-EMOA (−10,000)	95.1	**92.2**	**88.9**	-	-
SMS-EMOA (0)	**95.2**	91.8	88.4	-	-
SMS-EMOA (10,000)	95.1	91.2	87.3	-	-
HypE (−10,000)	96.9	93.3	91.2[1)]	92.0[2)]	91.5[3)]
HypE (0)	97.2	94.1	92.8[1)]	**93.9**[2)]	**92.5**
HypE (10,000)	**97.3**	**94.9**	**93.4**[1)]	93.7[2)]	92.0[3)]

1) 50 runs, 2) 30 runs, 3) 10 runs

TABLE IX
RELATIVE AVERAGE HYPERVOLUME BY THE EMO ALGORITHMS WITH VARIOUS SPECIFICATIONS OF THE REFERENCE POINT. THE REFERENCE POINT FOR PERFORMANCE EVALUATION IS (15 000, 15 000, ..., 15 000)

EMO Algorithm	2-500	4-500	6-500	8-500	10-500
Tchebycheff (21,000)	**102.1**	**73.1**	**50.2**	**52.7**	**54.7**
Tchebycheff (30,000)	97.8	31.6	23.9	46.8	52.3
Tchebycheff (40,000)	93.4	26.7	23.9	46.8	52.3
PBI (0.1) (21,000)	**97.3**	**104.4**	**121.8**	**150.2**	**169.6**
PBI (0.1) (30,000)	**97.3**	103.6	120.3	148.8	167.4
PBI (0.1) (40,000)	**97.3**	103.0	119.8	149.0	168.3
SMS-EMOA (−10,000)	87.5	93.4	101.2	-	-
SMS-EMOA (0)	**87.8**	93.5	103.9	-	-
SMS-EMOA (10,000)	87.3	**94.1**	**112.5**	-	-
HypE (−10,000)	91.4	78.8	91.5[1)]	124.3[2)]	147.2[3)]
HypE (0)	92.3	85.3	103.0[1)]	140.4[2)]	166.1
HypE (10,000)	**93.2**	**96.0**	**122.1**[1)]	**163.7**[2)]	**201.8**[3)]

1) 50 runs, 2) 30 runs, 3) 10 runs

In Table VIII, we cannot observe any large differences in the experimental results of each algorithm among the three settings of the reference point. Similar observations were reported for other test problems by Judt *et al.* [46].

In Table IX, we can observe large differences in the experimental results by HypE among the three settings. The differences among the three settings in HypE are much larger than those in SMS-EMOA. This is because the use of a reference point close to the Pareto front improves the hypervolume approximation quality in HypE. We can observe this effect in Table VIII where the hypervolume for the performance evaluation is calculated using the origin of the objective space. Even in this case, the best results on the 2–500, 4–500, and 6–500 problems are obtained by HypE (10 000) among the three settings in HypE whereas the worst results on the same three

TABLE X
RELATIVE AVERAGE HYPERVOLUME FOR THE SIX-OBJECTIVE PROBLEMS WITH 100, 500, 1000, AND 10 000 ITEMS. THE REFERENCE POINT IS (0, 0, . . ., 0) FOR HYPERVOLUME CALCULATION

EMO Algorithm	6-100	6-500	6-1,000	6-10,000
NSGA-II	88.0	77.8	75.6	79.6
MOEA/D: WS	**100.0**	**100.0**	**100.0**	100.0
MOEA/D: Tchebycheff	97.5	94.0	93.1	80.6
MOEA/D: PBI (0.1)	96.8	96.6	95.7	98.5
MOEA/D: PBI (5)	78.4	73.8	75.2	92.8
SMS-EMOA	97.7[1)]	88.4	86.0[1)]	96.8[1)]
HypE	95.3[2)]	92.8[2)]	92.7[2)]	89.0[2)]
F100-NSGA-II	82.2	87.2	92.5	98.3
F90-NSGA-II	88.2	92.4	99.3	**100.7**

1) 10 runs, 2) 50 runs

test problems are obtained by SMS-EMOA (10 000) among the three settings in SMS-EMOA.

In HypE, the hypervolume contribution of each solution can be accurately approximated when the reference point is close to the current population. This is because the approximation in HypE is performed by the random sampling in the area between the reference point and the current population. If the reference point is far from the current population, many sampling points are not likely to be fallen near the current population. This deteriorates the approximation quality for each solution, which deteriorates the performance of HypE (-10000) in Tables VIII and IX.

We also examine the effect of the problem size (i.e., the number of items) on each EMO algorithm. Using the same parameter specifications as in Table II, each EMO algorithm is applied to the six-objective test problems with 100, 500, 1000, and 10 000 items. Average experimental results over 100 runs (except for SMS-EMOA and HypE) are summarized in Table X where bold face shows the best result for each problem. From the comparison between Tables II and X, we can observe that the effect of the problem size on the relative average hypervolume in Table X is much smaller than that of the number of objectives in Table II. For example, the relative average hypervolume values of NSGA-II in Table II are decreased by the increase in the number of objectives from 96.5 to 65.5. However, they are between 75.6 and 88.0 in Table X. Of course, it is much more difficult to search for the true Pareto front of the 10 000-item problem than that of the 100-item problem. This is because the size of the search space (i.e., the total number of different strings) is totally different. The size of the search space of the 10 000-item problem is $2^{10\,000}$ while that of the 100-item problem is 2^{100}. However, this does not always mean that the search for good approximate solution sets is much more difficult for the 10 000-item problem.

We can also see from Table X that good results are obtained from F90-NSGA-II for large problems (see the best result for the 10 000-item problem is obtained by F90-NSGA-II). However, good results are not obtained from NSGA-II for large problems. These observations suggest the potential usefulness of the focused search.

TABLE XI
RELATIVE AVERAGE HYPERVOLUME BY MOEA/D WITH VARIOUS SPECIFICATIONS OF THE NEIGHBORHOOD SIZE. THE REFERENCE POINT FOR PERFORMANCE EVALUATION IS (0, 0, . . ., 0)

EMO Algorithm	2-500	4-500	6-500	8-500	10-500
Weighted Sum (5)	99.9	**100.3**	**100.2**	**100.0**	97.1
Weighted Sum (10)	**100.0**	100.0	100.0	**100.0**	**100.0**
Weighted Sum (50)	98.6	92.9	89.2	84.3	83.9
Weighted Sum (Pop)	96.5	85.6	78.8	76.1	70.1
Tchebycheff (5)	**100.7**	99.3	**94.0**	89.4	82.6
Tchebycheff (10)	**100.7**	**99.7**	**94.0**	**90.1**	**87.7**
Tchebycheff (50)	99.8	92.7	85.9	81.1	77.5
Tchebycheff (Pop)	98.2	85.9	78.0	73.4	66.6
PBI (0.1) (5)	98.7	**97.3**	**97.2**	**96.9**	93.4
PBI (0.1) (10)	**98.8**	96.9	96.6	96.0	**95.9**
PBI (0.1) (50)	96.9	89.4	85.3	81.5	80.3
PBI (0.1) (Pop)	94.6	83.0	76.6	74.6	69.3

TABLE XII
RELATIVE AVERAGE HYPERVOLUME BY MOEA/D WITH VARIOUS SPECIFICATIONS OF THE NEIGHBORHOOD SIZE. THE REFERENCE POINT FOR PERFORMANCE EVALUATION IS (15 000, 15 000, . . ., 15 000)

EMO Algorithm	2-500	4-500	6-500	8-500	10-500
Weighted Sum (5)	98.9	**101.0**	94.7	87.9	79.1
Weighted Sum (10)	**100.0**	100.0	100.0	100.0	100.0
Weighted Sum (50)	95.9	89.7	**101.7**	**121.6**	**126.9**
Weighted Sum (Pop)	89.8	73.4	82.1	101.4	100.7
Tchebycheff (5)	101.6	72.1	54.6	62.5	47.4
Tchebycheff (10)	**102.1**	**75.8**	**60.1**	**74.1**	**75.3**
Tchebycheff (50)	97.9	66.4	45.2	69.9	69.3
Tchebycheff (Pop)	92.4	51.0	30.7	53.6	56.3
PBI (0.1) (5)	96.6	**104.5**	118.0	135.5	135.5
PBI (0.1) (10)	**97.2**	104.2	**121.4**	**150.1**	**169.5**
PBI (0.1) (50)	92.2	87.9	109.1	142.2	160.0
PBI (0.1) (Pop)	85.0	70.0	83.3	113.7	116.3

D. Recombination of Similar Parents

In our computational experiments, good results are obtained from MOEA/D. One feature of MOEA/D is local mating where two parents are selected at each cell from its neighborhood.

To check the importance of local mating, we examine four settings of the neighborhood size in MOEA/D: 5, 10, 50, and "Pop" where "Pop" means that the entire population is used as the neighborhood. When the neighborhood size is "Pop," parents are selected randomly from the entire population. Except for the neighborhood size, computational experiments are performed in the same manner as in Tables II and III. Experimental results are summarized in Tables XI and XII where the best result among the four settings is shown by bold face for each test problem. The best results are obtained from the neighborhood size 5 or 10 in many cases in Tables XI and XII. Moreover, the worst results are almost always obtained from the setting of "(Pop)."

Our experimental results in Tables XI and XII suggest the possibility of performance improvement of other algorithms

TABLE XIII
RELATIVE AVERAGE HYPERVOLUME BY EMO ALGORITHMS WITH SIMILAR PARENT RECOMBINATION. THE REFERENCE POINT IS (0, 0, . . ., 0)

EMO Algorithm	2-500	4-500	6-500	8-500	10-500
NSGA-II ($\beta = 1$)	96.5	86.2	77.8	72.0	65.5
NSGA-II ($\beta = 5$)	97.7	89.5	81.3	75.2	69.4
NSGA-II ($\beta = 20$)	**98.7**	**91.5**	**83.6**	**77.6**	**72.1**
SMS-EMOA ($\beta = 1$)	95.2	91.8	88.4	-	-
SMS-EMOA ($\beta = 5$)	96.0	97.2[1)]	96.4[4)]	-	-
SMS-EMOA ($\beta = 20$)	**96.7**	**99.0**[1)]	**98.6**[4)]	-	-
HypE ($\beta = 1$)	97.2	94.1	92.8[1)]	93.9[2)]	92.5
HypE ($\beta = 5$)	98.0	96.1	95.1[3)]	97.2[3)]	98.3[4)]
HypE ($\beta = 20$)	**98.2**	**96.3**	**96.9**[3)]	**100.3**[3)]	**99.0**[4)]

1) 50 runs, 2) 30 runs, 3) 10 runs, 4) 5 runs

TABLE XIV
RELATIVE AVERAGE HYPERVOLUME BY EMO ALGORITHMS WITH SIMILAR PARENT RECOMBINATION. THE REFERENCE POINT IS (15 000, 15 000, . . ., 15 000)

EMO Algorithm	2-500	4-500	6-500	8-500	10-500
NSGA-II ($\beta = 1$)	91.8	48.7	21.7	8.0	7.1
NSGA-II ($\beta = 5$)	95.6	53.5	22.7	8.8	8.2
NSGA-II ($\beta = 20$)	**97.9**	**55.9**	**23.1**	**9.4**	**11.0**
SMS-EMOA ($\beta = 1$)	87.8	93.5	103.9	-	-
SMS-EMOA ($\beta = 5$)	90.3	106.9[1)]	125.4[4)]	-	-
SMS-EMOA ($\beta = 20$)	**92.5**	**109.1**[1)]	**127.2**[4)]	-	-
HypE ($\beta = 1$)	92.3	85.3	103.0[1)]	140.4[2)]	166.1
HypE ($\beta = 5$)	**94.1**	**89.4**	**109.2**[3)]	**150.5**[3)]	**193.6**[4)]
HypE ($\beta = 20$)	92.6	83.7	105.7[3)]	143.5[3)]	162.4[4)]

1) 50 runs, 2) 30 runs, 3) 10 runs, 4) 5 runs

by local mating. We examine the use of the following simple local mating in each of NSGA-II, SMS-EMOA, and HypE.

Step 1: Parent A is chosen by the parent selection mechanism of the EMO algorithm.
Step 2: The parent selection mechanism of the EMO algorithm is iterated β times to choose β candidates.
Step 3: The most similar candidate to Parent A is chosen from the β candidates as a mate of Parent A. The similarity between Parent A and each candidate is measured by the Euclidean distance between them in the objective space.

This is a simplified version of our similarity-based mating scheme [47]. If $\beta = 1$, this similarity-based mating scheme has no effect. By increasing the value of β, a more similar solution to Parent A is chosen as a mate of Parent A.

We examine three settings of β: 1, 5, 20. Experimental results are shown in Tables XIII and XIV. It should be noted that the experimental results with $\beta = 1$ in Tables XIII and XIV are the same as those in Tables II and III where no mating scheme was used, respectively. The best result among the three settings in each algorithm is shown by bold face for each test problem.

In Table XIII with a distant reference point from the Pareto front, the increase of β always improves experimental results. The best results are always obtained from $\beta = 20$. However, in Table XIV with a close reference point to the Pareto front, the best results of HypE are obtained from $\beta = 5$. This observation suggests that the local mating has a larger positive effect on the diversity of solutions than their convergence toward the Pareto front. The positive effect of the local mating on the diversity of solutions was also suggested by experimental results in our former study [48] where it was demonstrated that the use of crossover decreases the diversity of solutions of multiobjective knapsack problems. The decrease in the diversity of solutions by crossover can be also explained by the fact that binary crossover operators such as one-point crossover and uniform crossover are geometric. Geometric crossover always generates offspring between its two parents [49], [50].

The increase in the number of objectives often increases the diversity of solutions in the current population of each EMO algorithm as pointed out by Sato *et al*. [51]. In Table XV, we show the average Hamming distance between all pairs of solutions in the final population of each EMO algorithm on each test problem. From the experimental results by F100-NSGA-II in Table XV, we can see that the diversity of solutions over the entire Pareto front is clearly increased by the increase in the number of objectives. However, it is not likely that good solutions are obtained from the recombination of totally different solutions. Thus, the search ability of each EMO algorithm is improved by the use of the local mating for our knapsack problems as shown in Tables XIII and XIV.

TABLE XV
AVERAGE HAMMING DISTANCE BETWEEN TWO SOLUTIONS IN THE FINAL POPULATION OF EACH EMO ALGORITHM

EMO Algorithm	2-500	4-500	6-500	8-500	10-500
NSGA-II	27.28	91.80	110.93	122.69	125.73
MOEA/D: WS	61.22	93.11	104.45	105.90	104.11
MOEA/D: Tchebycheff	77.32	119.73	116.91	96.21	92.72
MOEA/D: PBI (0.1)	47.92	75.18	85.37	79.98	78.59
MOEA/D: PBI (5)	83.71	71.58	31.32	9.67	8.37
SMS-EMOA	20.01	76.25	82.91	-	-
HypE	61.56	102.31	106.62	95.70	87.59
F100-NSGA-II	88.02	151.02	181.52	202.83	222.31
F90-NSGA-II	69.53	143.63	168.37	180.85	182.86

The convex shape of the Pareto fronts of our knapsack problems is strongly related to the performance improvement by the local mating. In Fig. 15, we show 100 offspring by uniform crossover from two totally different parents (two bold circles in Fig. 15) which are two extreme solutions in the final population by MOEA/D with PBI (5) in Fig. 7.

In Fig. 15, all offspring are generated by uniform crossover around the center of the two parents. We can see in Fig. 15 that no offspring are close to the Pareto front. That is, good solutions close to the Pareto front are not generated by uniform crossover of the totally different parents whereas they are close to the Pareto front. This observation supports the usefulness of the local mating in our computational experiments.

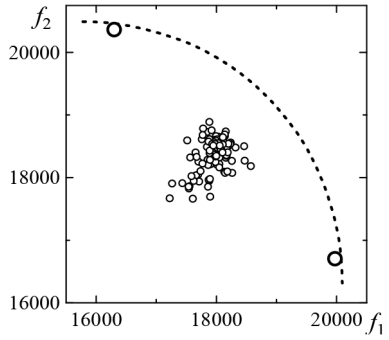

Fig. 15. Two extreme solutions and their 100 offspring.

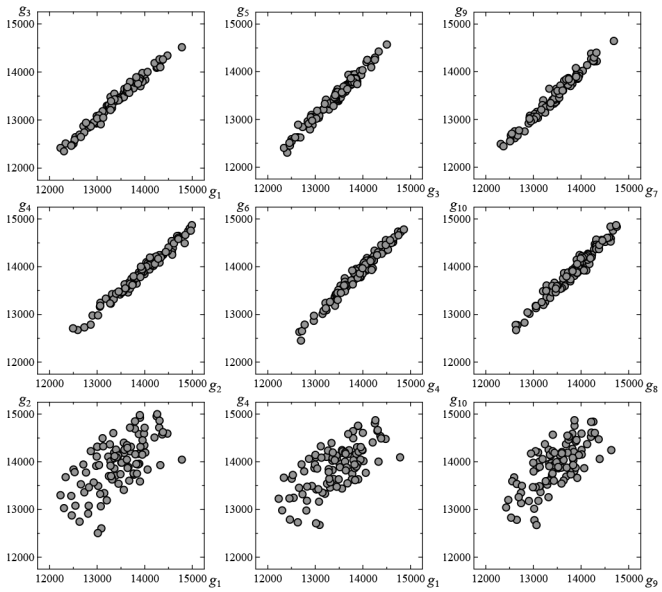

Fig. 16. Randomly generated 100 solutions for 10–500 (0.8).

V. Results on Other Test Problems

A. Results on Test Problems with Correlated Objectives

A correlated objective $g_i(\boldsymbol{x})$ with the correlation strength α is generated from the randomly generated objectives $f_i(\boldsymbol{x})$ in the form of $\alpha \cdot f_1(\boldsymbol{x}) + (1-\alpha) \cdot f_i(\boldsymbol{x})$ for $i = 3, 5, 7, 9$, and $\alpha \cdot f_2(\boldsymbol{x}) + (1-\alpha) \cdot f_i(\boldsymbol{x})$ for $i = 4, 6, 8, 10$. The first objective $g_1(\boldsymbol{x})$ and $g_2(\boldsymbol{x})$ are the same as $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$, respectively. Four values of α are used: $\alpha = 0.2, 0.4, 0.6, 0.8$. The k-objective 500-item problem with the correlation strength α is referred to as the k-500 (α) problem in this paper.

In Fig. 16, we show 100 solutions randomly generated for the 10–500 problem with $\alpha = 0.8$ [i.e., 10–500 (0.8)]. The nine plots in Fig. 16 show the same 100 solutions in different 2-D subspaces. The top three plots show strong correlation among the five objectives $g_i(\boldsymbol{x})$, $i = 1, 3, 5, 7, 9$. The middle three plots show strong correlation among the other five objectives $g_i(\boldsymbol{x})$, $i = 2, 4, 6, 8, 10$. The bottom three plots show that the correlation is not strong between these two groups of objectives.

Computational experiments are performed in the same manner as in Tables II and III. Experimental results are summarized in Tables XVI and XVII. The penalty parameter value θ is specified as $\theta = 0.1$ in MOEA/D with PBI. The best results are shown by bold face.

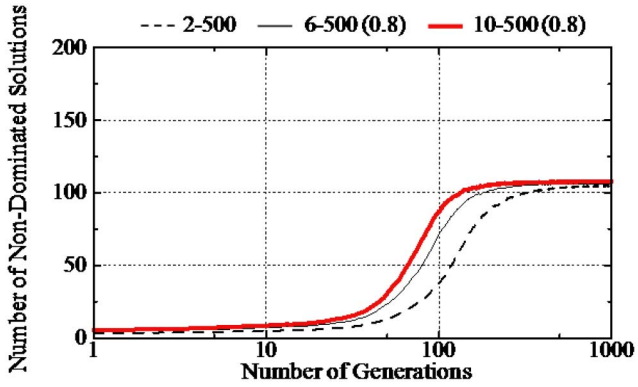

Fig. 17. Average number of nondominated solutions in the merged population when NSGA-II is applied to the 2–500, 6–500 (0.8), and 10–500 (0.8) problems (population size 100).

When the correlation strength α is small in Tables XVI and XVII, we observe severe performance deterioration of NSGA-II by the increase in the number of objectives. However, its performance is not deteriorated when the correlation strength α is large (i.e., $\alpha = 0.8$). For example, the best results are obtained by NSGA-II on the 8–500 and 10–500 problems with $\alpha = 0.8$. An interesting observation is that good results are not obtained by HypE while it showed very good performance in Section IV on randomly generated many-objective problems.

In Fig. 17, we show the average number of nondominated solutions in the merged population when NSGA-II is applied to the 2–500, 6–500 (0.8), and 10–500 (0.8) problems in the same manner as in Fig. 10 in Section IV. When NSGA-II is applied to the randomly generated 10–500 problem in Fig. 10, almost all solutions in the current population become nondominated around the 10th generation (i.e., the number of nondominated solutions becomes more than 100). However, in Fig. 17, the average number of nondominated solutions is smaller than 100 even at the 100th generation when NSGA-II is applied to the 10–500 (0.8) problem with high correlation among objectives. This leads to high convergence property of NSGA-II on the 10–500 (0.8) problem, which is demonstrated by high relative average hypervolume values in Tables XVI and XVII.

High convergence performance of NSGA-II on the 6–500 (0.8), 8–500 (0.8), and 10–500 (0.8) test problems in Table XVII is explained by a small number of nondominated solutions in the current population before the 100th generation in Fig. 17. High convergence performance of SMS-EMOA on 4–500 (0.8) and 6–500 (0.8) in Table XVII can be explained by the utilization of the exact hypervolume-based fitness evaluation together with the use of the nondominated sorting.

On the contrary, relatively poor performance of MOEA/D and HypE on 8–500 (0.8) and 10–500 (0.8) in Table XVII is explained by the performance improvement of NSGA-II. When objectives are highly correlated, it is likely that solutions can be compared by Pareto dominance (see Fig. 17). This leads

TABLE XVI
RELATIVE AVERAGE HYPERVOLUME ON TEST PROBLEMS WITH CORRELATED OBJECTIVES. THE REFERENCE POINT IS (0, 0, ..., 0)

Test Problem	NSGA-II	MOEA/D			SMS-EMOA	HypE
		WS	TE	PBI		
4-500 (0.2)	87.6	100.0	**101.5**	95.9	91.7	91.8
4-500 (0.4)	90.6	100.0	**104.0**	96.4	93.8	92.0
4-500 (0.6)	94.4	100.0	**102.7**	97.5	95.9	93.8
4-500 (0.8)	98.1	100.0	**100.9**	98.9	96.9	96.9
6-500 (0.2)	81.3	**100.0**	98.4	95.2	88.1[3]	89.2
6-500 (0.4)	86.2	100.0	**104.5**	95.0	91.9[3]	89.2
6-500 (0.6)	92.6	100.0	**103.5**	97.7	96.8[3]	92.1
6-500 (0.8)	100.4	100.0	**100.6**	99.7	99.6[3]	98.0
8-500 (0.2)	75.7	**100.0**	94.2	94.1	-	87.8[1]
8-500 (0.4)	80.3	**100.0**	98.3	94.1	-	85.0[1]
8-500 (0.6)	88.6	**100.0**	97.6	96.8	-	88.7[1]
8-500 (0.8)	**101.5**	100.0	99.7	100.0	-	97.3[1]
10-500 (0.2)	69.6	**100.0**	91.1	93.3	-	86.3[2]
10-500 (0.4)	74.1	**100.0**	94.8	92.9	-	83.1[2]
10-500 (0.6)	85.9	**100.0**	96.2	96.1	-	88.5[2]
10-500 (0.8)	**103.5**	100.0	100.5	100.2	-	99.3[2]

1) 60 runs, 2) 30 runs, 3) 20 runs

TABLE XVII
RELATIVE AVERAGE HYPERVOLUME ON TEST PROBLEMS WITH CORRELATED OBJECTIVES. THE REFERENCE POINT IS (15 000, 15 000, ..., 15 000)

Test Problem	NSGA-II	MOEA/D			SMS-EMOA	HypE
		WS	TE	PBI		
4-500 (0.2)	52.5	**100.0**	80.8	96.0	85.9	71.6
4-500 (0.4)	59.3	**100.0**	92.5	92.5	86.5	65.7
4-500 (0.6)	72.2	**100.0**	97.4	93.3	92.6	66.0
4-500 (0.8)	103.0	100.0	99.4	102.0	**104.7**	85.1
6-500 (0.2)	28.5	100.0	52.9	**104.7**	82.4[3]	70.5
6-500 (0.4)	36.5	**100.0**	57.6	93.9	79.8[3]	55.2
6-500 (0.6)	59.8	**100.0**	73.5	98.7	97.9[3]	59.6
6-500 (0.8)	134.6	100.0	98.9	115.4	**147.6**[3]	103.3
8-500 (0.2)	11.0	100.0	56.4	**117.4**	-	76.4[1]
8-500 (0.4)	17.0	100.0	54.8	**101.4**	-	50.8[1]
8-500 (0.6)	38.9	100.0	62.4	**101.7**	-	49.9[1]
8-500 (0.8)	**141.2**	100.0	93.1	120.8	-	97.1[1]
10-500 (0.2)	6.9	100.0	54.2	**126.3**	-	77.5[2]
10-500 (0.4)	9.7	100.0	46.9	**104.2**	-	44.3[2]
10-500 (0.6)	31.1	100.0	57.4	**100.4**	-	52.8[2]
10-500 (0.8)	**173.7**	100.0	105.6	126.4	-	121.0[2]

1) 60 runs, 2) 30 runs, 3) 20 runs

to the increase in the effectiveness of the Pareto dominance-based fitness evaluation. In MOEA/D, high correlation is not taken into account in the weight vector generation. In HypE, high correlation is not taken into account in the random sampling. Since there is a large difference in the performance on 6–500 (0.8) in Table XVII between SMS-EMOA and HypE, it is possible that the highly correlated objectives have a negative effect on the hypervolume approximation mechanism in HypE.

In Fig. 18, we show all solutions in the final population of a single run of NSGA-II on the correlated 10–500 problems for α = 0.2, 0.4, 0.6, 0.8 in the 2-D objective space with the first

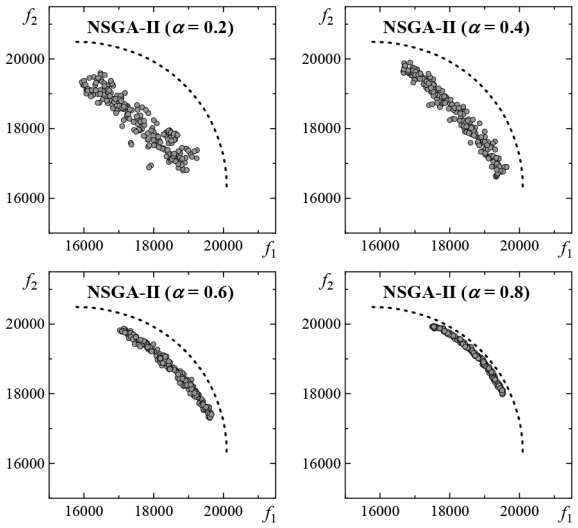

Fig. 18. Results of a single run of NSGA-II on the four correlated 10–500 problems with α = 0.2, 0.4, 0.6, 0.8.

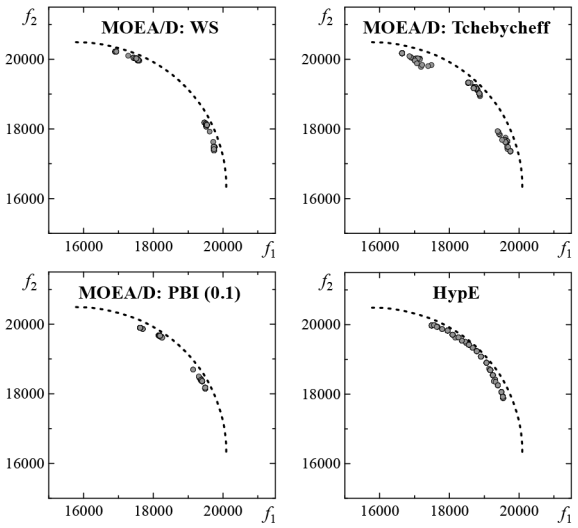

Fig. 19. Results of a single run of each algorithm on the correlated 10–500 problem with a = 0.8.

two objectives. In Fig. 18, we can see that good nondominated solutions close to the Pareto front of the 2–500 problem are obtained from the application of NSGA-II to the correlated 10–500 problem with α = 0.8. As shown in Tables XVI, XVII and Fig. 18, many-objective problems with highly correlated objectives are not difficult for NSGA-II.

For comparison, we also show experimental results by a single run of each of the other algorithms on the 10–500 problem with α = 0.8 in Fig. 19. We can see in Fig. 19 that clustered solutions with wide gaps are obtained by all the three variants of MOEA/D. This is because the uniformly distributed 220 weight vectors are very sparse in the ten-dimensional objective space. As a result, solutions look clustered when they are projected into the 2-D subspace in Fig. 19.

From the comparison between the bottom-right plot in Fig. 18 by NSGA-II and the results by HypE in Fig. 19 on the 10–500 (0.8) problem, we can see that more solutions with

TABLE XVIII
RELATIVE AVERAGE HYPERVOLUME ON TEST PROBLEMS WITH DEPENDENT OBJECTIVES. THE REFERENCE POINT IS (0, 0, ..., 0)

Test Problem	NSGA-II	MOEA/D			SMS-EMOA	HypE
		WS	TE	PBI		
4-500	99.1	100.0	**101.6**	98.5	97.6	98.2
6-500	100.7	100.0	**102.2**	98.1	99.5	98.6
8-500	102.0	100.0	**102.5**	97.9	-	99.3
10-500	**102.6**	100.0	102.3	97.3	-	99.6

TABLE XIX
RELATIVE AVERAGE HYPERVOLUME ON TEST PROBLEMS WITH DEPENDENT OBJECTIVES. THE REFERENCE POINT IS (15 000, 15 000, ..., 15 000)

Test Problem	NSGA-II	MOEA/D			SMS-EMOA	HypE
		WS	TE	PBI		
4-500	**113.7**	100.0	100.5	100.6	109.1	95.5
6-500	130.1	100.0	108.1	97.9	**132.4**	95.0
8-500	**141.7**	100.0	120.9	95.2	-	104.2
10-500	**158.5**	100.0	121.0	90.9	-	104.4

better convergence are obtained by NSGA-II than HypE. This observation is consistent with their results in Tables XVI and XVII on the 10–500 (0.8) problem.

B. Results on Test Problems with Dependent Objectives

A dependent objective $g_i(\boldsymbol{x})$ is generated in the form of $\alpha_{ik} \cdot f_1(\boldsymbol{x}) + (1 - \alpha_{ik}) \cdot f_2(\boldsymbol{x})$ for $k = 3, 4, \ldots, 10$. In the same manner as in Tables II and III, we perform computational experiments on the dependent 4–500, 6–500, 8–500, and 10–500 problems. Experimental results are summarized in Tables XVIII and XIX. In MOEA/D with PBI, the penalty parameter value is specified as $\theta = 0.1$. These tables show that the inclusion of the dependent objectives increases the relative advantages of NSGA-II (while the inclusion of the randomly generated objectives severely deteriorates its performance in Section IV).

Among the three versions of MOEA/D in Tables XVIII and XIX, the best results are obtained by the weighted Tchebycheff for almost all cases. When all objectives are randomly generated in Section IV, MOEA/D with the weighted Tchebycheff works well only on the 2–500 problem in Tables II and III. These observations are consistent because all objectives in our dependent problems are generated from the two objectives of the 2–500 problem. Actually, all of the dependent 4–500, 6–500, 8–500, and 10–500 test problems have the same Pareto optimal solution set as the 2–500 test problem. However, all EMO algorithms solve the dependent 4–500, 6–500, 8–500, and 10–500 test problems as many-objective test problems. Experimental results in Tables XVIII and XIX can be viewed as showing bad effects of the inclusion of dependent objectives on the search behavior of each EMO algorithm. In all EMO algorithms (except for NSGA-II), dependent objectives have bad effects. The search ability of MOEA/D and HypE is severely degraded by the inclusion of dependent objectives. Whereas the search ability of SMS-EMOA is not severely degraded, its computation load is heavily increased by the inclusion of dependent objectives.

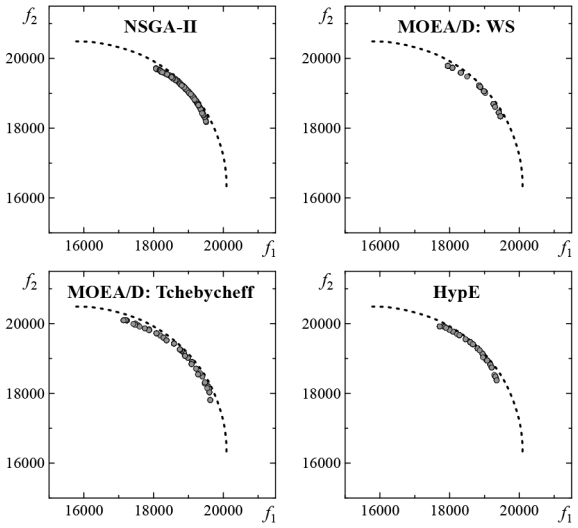

Fig. 20. Results of a single run of each algorithm on the dependent 10–500 problem.

Fig. 20 shows all solutions in the final generation of a single run of each algorithm. We can observe that the distribution of obtained solutions by the two versions of MOEA/D is sparse in comparison with the results by NSGA-II. This is due to the use of the coarsely distributed ten-dimensional weight vectors. From the comparison between the results by NSGA-II and HypE in Fig. 20, more solutions are obtained by NSGA-II along the Pareto front of the 2–500 problem.

VI. CONCLUDING REMARKS AND FUTURE RESEARCH

In this paper, we examined the search behavior of well-known EMO algorithms (NSGA-II, MOEA/D, SMS-EMOA, and HypE) on many-objective knapsack problems with 2–10 objectives. Our experimental results on many-objective knapsack problems with randomly generated objectives were consistent with frequently-reported results: The performance of Pareto dominance-based algorithms was severely deteriorated by the increase in the number of objectives. However, we also demonstrated that NSGA-II worked well on many-objective knapsack problems with highly correlated objectives. It was also shown that the inclusion of dependent objectives clearly increased the relative advantage of NSGA-II over the other algorithms. Whereas HypE showed its high search ability on many-objective knapsack problems with randomly generated objectives, it did not work well on many-objective knapsack problems with highly correlated objectives. MOEA/D worked well on a wide range of test problems. However, as we demonstrated in this paper, its performance heavily depended on the choice of a scalarizing function.

With respect to the choice of an appropriate scalarizing function in MOEA/D, we obtained the following interesting observations: 1) MOEA/D with good specifications for the two-objective knapsack problem such as the weighted Tchebycheff and PBI with $\theta = 5$ did not work well on many-objective knapsack problems; 2) MOEA/D with good

specifications for the 10-objective knapsack problem with randomly generated objectives such as the weighted sum and PBI with $\theta = 0.1$ did not work well on the two-objective knapsack problem; and 3) the best choice of a scalarizing function for hypervolume maximization depended on the specification of a reference point for hypervolume calculation in performance evaluation. The last observation suggests that some scalarizing functions in MOEA/D are beneficial for convergence improvement while others are beneficial for diversity improvement.

We also demonstrated that the performance of NSGA-II, SMS-EMOA, and HypE on many-objective knapsack problems was improved by recombining similar parents. This observation suggests that the choice of parents from the entire population is not a good strategy for many-objective knapsack problems. This was also supported by our experimental results on MOEA/D where the use of a large neighborhood for mating selection degraded the performance of MOEA/D on many-objective knapsack problems.

While NSGA-II did not work well on many-objective knapsack problems with randomly generated test problems, it worked very well when objectives were highly correlated. Using this feature of NSGA-II, we also examined the use of multiple NSGA-II subpopulations to search for different regions of the Pareto front. That is, each NSGA-II subpopulation with highly correlated objectives was used for the focused search toward a small region of the Pareto front. The best results were obtained from the focused search by NSGA-II on the six-objective 10 000-item knapsack problem. This observation suggests the potential usefulness of multiple runs of the focused search for many-objective optimization.

Through computational experiments, we also showed that the performance evaluation results of the same solution sets by the hypervolume measure were totally different depending on the specification of the reference point. For example, when the origin (0, 0, ..., 0) of the objective space was used, the relative average hypervolume values were calculated over 100 solution sets for the 10–500 problem as follows (see Table II):

Weighted Sum: 100.0, PBI (0.l): 95.9, HypE: 92.5.

However, the relative average hypervolume values for the reference point (15 000, 15 000, ..., 15 000) were calculated over the same 100 solution sets as follows (see Table III):

Weighted Sum: 100.0, PBI (0.l): 169.5, HypE: 166.1.

These results suggest the differences in the distribution of solutions obtained by the three algorithms. The solution sets obtained by MOEA/D with the weighed sum may have a larger diversity over the Pareto front than the solution sets by the other algorithms (since MOEA/D with the weighted sum had better hypervolume values than the other algorithms when the reference point was far from the Pareto front). However, more solutions with better convergence toward the Pareto front may be obtained by MOEA/D with PBI (0.1) and HypE around the center of the Pareto front than the MOEA/D with the weighted sum [since MOEA/D with PBI (0.1) and HypE had better hypervolume values when the reference point was close to the Pareto front]. These discussions are consistent with the visual examination of the obtained solution sets in the 2-D objective subspace in Fig. 8 in Section IV. From these discussions, we can see that the hypervolume measure for different reference points can be used for examining the behavior of EMO algorithms on many-objective problems.

All experimental results in this paper showed that solution sets with totally different characteristics were obtained by different EMO algorithms. This may suggest the potential usefulness of ensemble approaches of different EMO algorithms. Experimental results in this paper suggest the ensemble of the weighted sum and PBI (0.1). Another promising combination is PBI with different penalty parameter values (see $\theta = 5$ with the best results on the 2–500 problem and $\theta = 0.1$ with the best results on the 10–500 problem). As a similar idea, ensemble of the weighed sum and the weighted Tchebycheff in MOEA/D was used in [52]. Ensemble of MOEA/D with different neighborhood structures was also proposed in [53].

In this paper, we observed high search ability of HypE on many-objective problems with randomly generated objectives. However, it did not work well on many-objective problems with highly correlated objectives. Its experimental results on 6–500 (0.8) were clearly inferior to those by SMS-EMOA. These observations suggest the necessity of improving its hypervolume approximation mechanism. Utilization of the characteristics of each objective and the relation among different objectives may be an interesting future research topic with respect to the improvement of the hypervolume approximation mechanism in HypE.

Utilization of the characteristics of each objective and the relation among different objectives is also an interesting future research topic with respect to the specification of the weight vectors in MOEA/D for many-objective problems (see [38]–[41] for further discussions on the specification and the adaptation of the weight vectors in MOEA/D). More studies are needed for appropriate weight vector specifications in MOEA/D for many-objective problems. Of course, the choice of an appropriate scalarizing function or an appropriate scalarizing function ensemble is also an interesting future research topic.

The best results were obtained by the focused search of NSGA-II on the six-objective 10 000-item problem. That is, NSGA-II worked very well on many-objective problems with modified objectives. An interesting future research topic is to develop a more sophisticated mechanism to efficiently use multiple NSGA-II subpopulations for the focused search to different regions of the Pareto front. It would be interesting if we could achieve high search ability for many-objective problems by using standard NSGA-II together with modified objectives instead of implementing highly-tailored complicated EMO algorithms.

Our focus in this paper was the search behavior of EMO algorithms when they search for a solution set to approximate the entire Pareto front (i.e., global search). A promising approach to the improvement of approximation quality of a solution set is the hybridization of single-objective optimization methods such as local search into EMO algorithms [54]–[57]. Implementation of many-objective hybrid

algorithms is another interesting research direction. However, the approximation of the entire Pareto front using a limited number of nondominated solutions is impractical in many cases of many-objective optimization. Thus, the hybridization of the focused search toward a part of the Pareto front and the global search for the entire Pareto front seems to be a practically useful research direction. In this sense, NSGA-III [58], [59] can be viewed as such a hybrid search algorithm.

REFERENCES

[1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA, USA: Addison-Wesley, 1989.

[2] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001.

[3] C. A. C. Coello and G. B. Lamont, *Applications of Multi-Objective Evolutionary Algorithms.* Singapore: World Scientific, 2004.

[4] K. C. Tan, E. F. Khor, and T. H. Lee, *Multiobjective Evolutionary Algorithms and Applications.* Berlin, Germany: Springer, 2005.

[5] 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.

[6] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," Comput. Eng. Netw. Lab. (TIK), Dept. Electric. Eng., ETH, Zurich, Switzerland, TIK-Rep. 103, 2001.

[7] R. C. Purshouse and P. J. Fleming, "Evolutionary many-objective optimization: An exploratory analysis," in *Proc. 2003 IEEE Congr. Evol. Comput.*, Canberra, ACT, Australia, pp. 2066–2073.

[8] V. Khare, X. Yao, and K. Deb, "Performance scaling of multi-objective evolutionary algorithms," in *Evolutionary Multi-Criterion Optimization.* Berlin, Germany: Springer, Apr. 2003, pp. 376–390.

[9] E. J. Hughes, "Evolutionary many-objective optimization: Many once or one many?" in *Proc. 2005 IEEE Congr. Evol. Comput.*, Edinburgh, U.K., pp. 222–227.

[10] 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.* Berlin, Germany: Springer, Mar. 2007, pp. 5–20.

[11] D. Corne and J. Knowles, "Techniques for highly multiobjective optimization: Some non-dominated points are better than others," in *Proc. 2007 GECCO.*, London, U.K., pp. 773–780.

[12] S. Kukkonen and J. Lampinen, "Ranking-dominance and many-objective optimization," in *Proc. 2007 IEEE Congr. Evol. Comput.*, Singapore, pp. 3983–3990.

[13] M. Köppen and K. Yoshida, "Substitute distance assignments in NSGA-II for handling many-objective optimization problems," in *Evolutionary Multi-Criterion Optimization*. Berlin, Germany: Springer, Mar. 2007, pp. 727–741.

[14] H. Singh, A. Isaacs, T. Ray, and W. Smith, "A study on the performance of substitute distance based approaches for evolutionary many objective optimization," in *Simulated Evolution and Learning.* Berlin, Germany: Springer, Dec. 2008, pp. 401–410.

[15] Z. He, G. G. Yen, and J. Zhang, "Fuzzy-based Pareto optimality for many-objective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 269–285, Apr. 2014.

[16] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Proc. 2008 IEEE Congr. Evol. Comput.*, Hong Kong, pp. 2424–2431.

[17] H. Ishibuchi, N. Tsukamoto, Y. Hitotsuyanagi, and Y. Nojima, "Effectiveness of scalability improvement attempts on the performance of NSGA-II for many-objective problems," in *Proc. 2008 GECCO*, Atlanta, GA, USA, pp. 649–656.

[18] M. Li, S. Yang, and X. Liu, "Shift-based density estimation for Pareto-based algorithms in many-objective optimization," *IEEE Trans. Evol. Comput.,* vol. 18, no. 3, pp. 348–365, June 2014.

[19] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 5, pp. 721–736, Oct. 2013.

[20] O. Schutze, A. Lara, and C. A. C. Coello, "On the influence of the number of objectives on the hardness of a multiobjective optimization problem," *IEEE Trans. Evol. Comput.*, vol. 15, no. 4, pp. 444–455, Aug. 2011.

[21] H. Ishibuchi, N. Akedo, H. Ohyanagi, and Y. Nojima, "Behavior of EMO algorithms on many-objective optimization problems with correlated objectives," in *Proc. 2011 IEEE Congr. Evol. Comput.*, New Orleans, LA, USA, pp. 1465–1472.

[22] H. Ishibuchi, N. Akedo, and Y. Nojima, "A many-objective test problem for visually examining diversity maintenance behavior in a decision space," in *Proc. 2011 GECCO*, Dublin, Ireland, pp. 649–656.

[23] 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.

[24] 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, Sep. 2007.

[25] 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.

[26] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization by NSGA-II and MOEA/D with large populations," in *Proc. 2009 IEEE Int. Conf. SMC*, San Antonio, TX, USA, pp. 1820–1825.

[27] H. Ishibuchi, N. Akedo, and Y. Nojima, "Recombination of similar parents in SMS-EMOA on many-objective 0/1 knapsack problems," in *Proc. 12th Int. Conf. PPSN II*, Taormina, Italy, Sep. 2012, pp. 132–142.

[28] 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.

[29] D. J. Walker, R. M. Everson, and J. E. Fieldsend, "Visualizing mutually nondominating solution sets in many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 2, pp. 165–184, Apr. 2013.

[30] L. Thiele, K. Miettinen, P. J. Korhonen, and J. Molina, "A preference-based evolutionary algorithm for multi-objective optimization," *Evol. Comput.*, vol. 17, no. 3, pp. 411–436, 2009.

[31] I. Karahan and M. Koeksalan, "A territory defining multiobjective evolutionary algorithms and preference incorporation," *IEEE Trans. Evol. Comput.*, vol. 14, no. 4, pp. 636–664, Aug. 2010.

[32] T. Wagner and H. Trautmann, "Integration of preferences in hypervolume-based multiobjective evolutionary algorithms by means of desirability functions," *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 688–701, Oct. 2010.

[33] J. H. Kim, J. H. Han, Y. H. Kim, S. H. Choi, and E. S. Kim, "Preference-based solution selection algorithm for evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 1, pp. 20–34, Feb. 2012.

[34] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired coevolutionary algorithms for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 4, pp. 474–494, Aug. 2013.

[35] 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.

[36] O. Schütze, X. Esquivel, A. Lara, and C. A. C. Coello, "Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 4, pp. 504–522, Aug. 2012.

[37] O. Schütze, X. Esquivel, A. Lara, and C. A. C. Coello, "Some comments on GD and IGD and relations to the Hausdorff distance," in *Proc. GECCO*, Portland, OR, USA, Jul. 2010, pp. 1971–1974.

[38] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "Generalized decomposition," in *Evolutionary Multi-Criterion Optimization*. Berlin, Germany: Springer, Mar. 2013, pp. 428–442.

[39] S. Jiang, Z. Cai, J. Zhang, and Y. S. Ong, "Multiobjective optimization by decomposition with Pareto-adaptive weight vectors," in *Proc. 7th ICNC*, Shanghai, China, Jul. 2011, pp. 1260–1264.

[40] F. Gu, H.-L. Liu, and K. C. Tan, "A multiobjective evolutionary algorithm using dynamic weight design method," *Int. J. Innov. Comput., Inf. Control*, vol. 8, no. 5(B), pp. 3677–3688, May 2012.

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

[42] H.-L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 450–455, June 2014.

[43] T. Wagner, N. Beume, and B. Naujoks, "Pareto-, aggregation-, and indicator-based methods in many-objective optimization," in *Evolutionary Multi-Criterion Optimization.* Berlin, Germany: Springer, Mar. 2007, pp. 742–756.

[44] 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.
[45] L. M. S. Russo and A. P. Francisco, "Quick hypervolume," *IEEE Trans. Evol. Comput.,* vol. 18, no. 4, pp. 481–502, Aug. 2014.
[46] L. Judt, O. Mersmann, and B. Naujoks, "Do hypervolume regressions hinder EMOA performance? Surprise and relief," in *Evolutionary Multi-Criterion Optimization.* Berlin, Germany: Springer, Mar. 2013, pp. 96–110.
[47] H. Ishibuchi, K. Narukawa, N. Tsukamoto, and Y. Nojima, "An empirical study on similarity-based mating for evolutionary multiobjective combinatorial optimization," *Eur. J. Oper. Res.*, vol. 188, no. 1, pp. 57–75, Jul. 2008.
[48] H. Ishibuchi and K. Narukawa, "Recombination of similar parents in EMO algorithms," in *Evolutionary Multi-Criterion Optimization.* Berlin, Germany: Springer, Mar. 2005, pp. 265–279.
[49] A. Moraglio and R. Poli, "Topological interpretation of crossover," in *Genetic and Evolutionary Computation*. Berlin, Germany: Springer, Jun. 2004, pp. 1377–1388.
[50] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Diversity improvement by non-geometric binary crossover in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 14, no. 6, pp. 985–998, Dec. 2010.
[51] H. Sato, H. E. Aguirre, and K. Tanaka, "Genetic diversity and effective crossover in evolutionary many-objective optimization," in *Learning and Intelligent Optimization*. Berlin, Germany: Springer, Jan. 2011, pp. 91–105.
[52] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Simultaneous use of different scalarizing functions in MOEA/D," in *Proc. GECCO*, Portland, OR, USA, Jul. 2010, pp. 519–526.
[53] S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes," *IEEE Trans. Evol. Comput.*, vol. 16, no. 3, pp. 442–446, Jun. 2012.
[54] 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.
[55] H. Ishibuchi, T. Yoshida, and T. Murata, "Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 204–223, Apr. 2003.
[56] K. Sindhya, K. Miettinen, and K. Deb, "A hybrid framework for evolutionary multi-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 4, pp. 495–511, Aug. 2013.
[57] L. Tang and X. Wang, "A hybrid multiobjective evolutionary algorithm for multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 17, no. 1, pp. 20–45, Jan. 2013.
[58] 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. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.
[59] 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. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, Aug. 2014.

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, Osaka, 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 research interests include fuzzy rule-based classifiers, evolutionary multiobjective optimization, memetic algorithms, and evolutionary games.

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

Naoya Akedo received the B.S. and M.S. degrees in computer science and intelligent systems from Osaka Prefecture University, Osaka, Japan, in 2011 and 2013, respectively.

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

Yusuke Nojima (M'00) received the B.S. and M.S. degrees in mechanical engineering from 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, Osaka, Japan, where he was a Research Associate and is currently an Associate Professor with the Department of Computer Science and Intelligent Systems. His research interests include multiobjective genetic fuzzy systems, evolutionary multiobjective optimization, and parallel distributed data mining.