

# Comparing Solution Sets of Different Size in Evolutionary Many-Objective Optimization

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

**Abstract**—When the performance of different evolutionary multiobjective optimization (EMO) algorithms is compared, the same population size is usually used for all EMO algorithms in computer simulations. This setting is to obtain a solution set of the same size from a different algorithm. However, in general, each algorithm may have its own best parameter specifications for each test problem. Thus, it may be difficult to appropriately specify the same population size for all algorithms for their fair comparison. A different algorithm may be evaluated as being the best for a different specification of the population size. An alternative setting is to allow each algorithm to use its own best population size. In this setting, a solution set of different size is obtained from each algorithm. It may be difficult to perform fair comparison using solution sets of different size. In this paper, we discuss the difficulty in comparing EMO algorithms under these two settings of the population size: the same specification for all algorithms and a different specification for each algorithm. First, we discuss the effect of the number of non-dominated solutions on some performance indicators. Next we show the difficulty in the first setting: Performance of each algorithm depends on the population size. Then we discuss the difficulty in the second setting: The size of a solution set obtained by each algorithm is not the same. In this setting, we examine the use of solution selection as a post-processing procedure to choose the same number of solutions from each solution set of different size. The selected solutions are used for performance comparison.

**Keywords**—Evolutionary many-objective optimization, solution selection, multiobjective knapsack problem, performance indicator.

## I. INTRODUCTION

Evolutionary many-objective optimization [1], [2] is a hot topic in the field of evolutionary multiobjective optimization (EMO). Since well-known Pareto dominance-based EMO algorithms such as NSGA-II [3] and SPEA2 [4] usually do not work well on many-objective problems, a number of new EMO algorithms and modifications of existing algorithms have been proposed in the literature [5]-[10]. Many-objective continuous problems such as DTLZ [11] and WFG [12] have been frequently used for performance evaluation of those algorithms [5]-[10]. Many-objective combinatorial problems have been also used for performance evaluation such as TSP [13], job shop scheduling [13], and knapsack problems [14]-[17].

When different EMO algorithms are compared through computer simulations, the total number of examined solutions is usually used as a termination condition of each algorithm.

This is to compare all algorithms under the same computation load. With respect to the specification of the other parameters, there are two manners in the literature. One manner is to use the same parameter values for all algorithms. However, in general, each algorithm has its own best parameter values. Thus it is difficult to appropriately specify the same parameter values for all algorithms for their fair comparison. The other manner is to allow each algorithm to use its own parameter values. Default parameter values in each algorithm, which were suggested by the algorithm's proposer, are often used in computer simulations. Parameter values can be tuned for each algorithm. As a result, totally different parameter values can be used by different algorithms (e.g., the crossover rate can be 1.0 in one algorithm and 0.1 in another algorithm).

However, even in the second case, the same (or similar) value of the population size has almost always been used for performance comparison. Of course, a different population size is used for a different problem. However, for each problem, the same (or similar) population size is used by different algorithms for performance comparison. This is to obtain solution sets of the same (or similar) size by all algorithms. If an EMO algorithm has an archive population, its size is often specified as being equal to the population size of other algorithms with no archive populations for the same reason.

To the best of our knowledge, no studies have compared different EMO algorithms under totally different specifications of the population size (e.g., one algorithm with the population size 100 and another algorithm with the population size 5000). For two-objective and three-objective problems, the use of a very large population with thousands of solutions may be meaningless. This is because thousands of solutions are not likely to be needed to approximate the entire Pareto front of a multiobjective problem with two or three objectives. However, thousands of solutions may be needed to approximate the entire Pareto front of a many-objective problem with four or more objectives. The use of such a large population was examined for many-objective problems in some studies [18]-[21].

Our main focus in this paper is performance comparison of different EMO algorithms with different specifications of the population size. In this setting, performance comparison of EMO algorithms leads to the comparison of solution sets of different size. We discuss the difficulty of fair performance comparison in this setting. We also examine the use of solution selection to remedy the difficulty of fair comparison.

This paper is organized as follows. In Section II, we discuss the relation between the number of solutions and frequently-used performance indicators such as generational distance (GD), inverted generational distance (IGD) and hypervolume. It is shown that large solution sets are beneficial for IGD and hypervolume. In Section III, we examine the sensitivity of the performance of EMO algorithms on the population size. It is shown through computer simulations using HypE [22] and two versions of MOEA/D [23] that the best specification of the population size for each algorithm is totally different. It is also shown that performance comparison results depend on the specification of the population size. In Section IV, we examine the use of solution selection as a post-processing procedure to select a pre-specified number of solutions from an obtained solution set. Solution sets of different size are compared after choosing the same number of solutions from each solution set. Our idea is to compare EMO algorithms with different specifications of the population size using selected solution sets of the same size after the post-processing procedure. Finally, we conclude this paper in Section V.

## II. PERFORMANCE INDICATORS

In this section, we briefly discuss the effect of the number of solutions on three frequently-used performance indicators: GD, IGD and hypervolume.

### A. Generational Distance (GD)

Let us consider the following  $m$ -objective optimization problem with a decision vector  $\mathbf{x}$ , its feasible region  $\mathbf{X}$ , and  $m$  objective functions  $f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})$ :

$$\text{Maximize } f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}) \text{ subject to } \mathbf{x} \in \mathbf{X}. \quad (1)$$

The objective vector corresponding to the decision vector  $\mathbf{x}$  is denoted by  $\mathbf{y}$  as  $\mathbf{y} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))$ . The objective vector  $\mathbf{y}$  is a point in the  $m$ -dimensional objective space.

Let us assume that a set of reference points is given on the Pareto front of the  $m$ -objective problem in the  $m$ -dimensional objective space as  $R = \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_{|R|}\}$  where  $|R|$  is the number of the given reference points in  $R$ . We also assume that a set of non-dominated solutions is obtained in the  $m$ -dimensional objective space as  $Y = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{|Y|}\}$  where  $|Y|$  is the number of the obtained non-dominated solutions in  $Y$ .

The generational distance (GD) of the solution set  $Y$  is defined as the average value of the distance from each solution to the nearest reference point. The following definition is usually used for performance evaluation (see [24] for more formal definitions of GD):

$$GD(Y, R) = \frac{1}{|Y|} \sum_{i=1}^{|Y|} \min\{d(\mathbf{y}_i, \mathbf{r}_j) \mid \mathbf{r}_j \in R\}, \quad (2)$$

where  $d(\mathbf{y}_i, \mathbf{r}_j)$  is the Euclidean distance between  $\mathbf{y}_i$  and  $\mathbf{r}_j$ .

Let us assume that the solutions have already been sorted in ascending order according to the distance to the nearest reference point as follows (see Fig. 1):

$$\min\{d(\mathbf{y}_i, \mathbf{r}_j) \mid \mathbf{r}_j \in R\} < \min\{d(\mathbf{y}_{i+1}, \mathbf{r}_j) \mid \mathbf{r}_j \in R\} \\ \text{for } i = 1, 2, \dots, |Y| - 1. \quad (3)$$

In this case, the best subset of the solution set  $Y$  with respect to GD is  $\{\mathbf{y}_1\}$ , and the worst subset is  $\{\mathbf{y}_{|Y|}\}$ . When the following relation holds, the addition of a new non-dominated solution  $\mathbf{y}^{\text{New}}$  to the solution set  $Y$  degrades GD:

$$GD(Y, R) < \min\{d(\mathbf{y}^{\text{New}}, \mathbf{r}_j) \mid \mathbf{r}_j \in R\}. \quad (4)$$

These discussions show that a large number of solutions are not beneficial for GD. On the contrary, the selection of the single best solution leads to the best GD over all subsets of the given solution set.

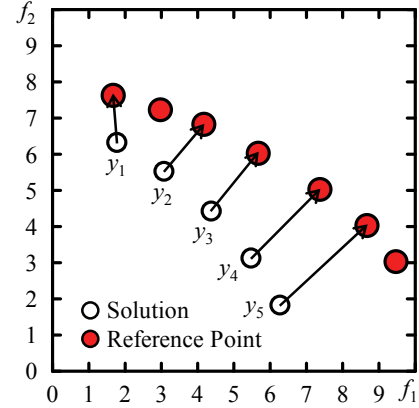


Fig. 1. Illustration of the generational distance (GD).

### B. Inverted Generational Distance (IGD)

The inverted generational distance (IGD) of the solution set  $Y$  is defined as the average value of the distance from each reference point to the nearest solution (see Fig. 2). The following definition is usually used for performance evaluation (see [24] for more formal definitions of IGD):

$$IGD(Y, R) = \frac{1}{|R|} \sum_{j=1}^{|R|} \min\{d(\mathbf{y}_i, \mathbf{r}_j) \mid \mathbf{y}_i \in Y\}, \quad (5)$$

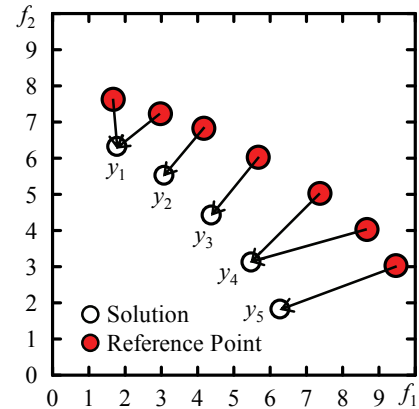


Fig. 2. Illustration of the inverted generational distance (IGD).

Since the following relation always holds, the addition of a new non-dominated solution  $\mathbf{y}^{\text{New}}$  to the solution set  $Y$  never degrades IGD:

$$\min\{d(y_i, r_j) \mid y_i \in Y\} \geq \min\{d(y_i, r_j) \mid y_i \in (Y \cup \{y^{\text{New}}\})\}. \quad (6)$$

Moreover, the removal of any solution from the solution set  $Y$  never improves IGD. In Fig. 2, the removal of any solution degrades IGD. From these discussions, we can see that a large number of solutions are beneficial for IGD.

### C. Hypervolume

The hypervolume (HV) of the solution set  $Y$  is the volume of the subspace dominated by the solutions in  $Y$  in the objective space. A reference point is needed for hypervolume calculation to limit the dominated subspace (see Fig. 3).

The addition of a new non-dominated solution  $y^{\text{New}}$  to the solution set  $Y$  always improves HV (if  $y^{\text{New}}$  dominates the reference point for the hypervolume calculation). Moreover, the removal of any non-dominated solution  $y_i$  from the solution set  $Y$  always degrades HV (if  $y_i$  dominates the reference point). That is, the following relations hold:

$$HV(Y) < HV(Y \cup \{y^{\text{New}}\}), \quad HV(Y - \{y_i\}) < HV(Y). \quad (7)$$

From these discussions, we can see that a large number of solutions are beneficial for the hypervolume.

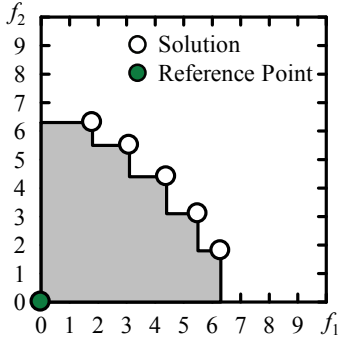


Fig. 3. Illustration of the hypervolume (HV).

### III. SPECIFICATION OF THE POPULATION SIZE

In the previous section, we show that a large number of solutions are beneficial for the inverted generational distance (IGD) and the hypervolume (HV) indicators. However, this does not necessarily mean that good results are obtained for these indicators by using a large population size. This is because a large population size is not necessarily beneficial for the search of EMO algorithms.

In this section, we examine the sensitivity of hypervolume-based performance evaluation results of EMO algorithms on the specification of the population size. As test problems, we use randomly generated 500-item knapsack problems with four and six objectives (for details of the test problems, see [17]).

As EMO algorithms for many-objective problems, we use HypE [22] and two versions of MOEA/D [23]: MOEA/D with the weighted sum (MOEA/D-WS) and MOEA/D with the weighted Tchebycheff (MOEA/D-Tch). Except for the population size (and the neighborhood size in MOEA/D), we use the same parameter specifications as in [17] as follows:

Coding: Binary string of length 500,  
Termination condition: 400,000 solution evaluations,  
Crossover probability: 0.8 (Uniform crossover),  
Mutation probability: 2/500 (Bit-flip mutation),  
Repair: Greedy repair based on the maximum profit rate, which is the same as in [14].

As the population size, the following specifications are examined for the four-objective 500-item problem (4-500) and the six-objective 500-item problem (6-500):

4-500 Problem: 120, 220, 451, 1140, 2024, 4960,  
6-500 Problem: 126, 252, 462, 1287, 2002, 4368.

It should be noted that the population size of MOEA/D cannot be arbitrarily specified due to the combinatorial nature of its generation mechanism of uniform weight vectors. The above-mentioned specifications are selected from possible values of the population size in MOEA/D. Since we use the total number of examined solutions (i.e., 400,000 solutions) as the termination condition, a large population size means the small number of generations. The neighborhood size is specified as 10% of the population size (more specifically, the nearest integer to 10% of the population size).

The average hypervolume value is calculated over 10 runs of each algorithm for the 4-500 problem and 5 runs for the 6-500 problem. We examine two specifications of the reference point for the hypervolume calculation. One reference point is the origin of the objective space: (0, 0, 0, 0) for the 4-500 problem and (0, 0, 0, 0, 0, 0) for the 6-500 problem. Since the origin is far from the Pareto front in the objective space of each test problem, a large diversity of solutions especially around the edge of the Pareto front is needed for good evaluation. The other reference point is much closer to the Pareto front: (15000, ..., 15000). Since this reference point is close to the Pareto front, a good convergence of solutions around the center of the Pareto front is needed for good evaluation.

Simulation results are shown in Figs. 4-7. We can see from these figures that the performance comparison results strongly depend on the population size and the reference point. For example, MOEA/D-Tch is the best when the far reference point and the largest population size are used (see Fig. 4 and Fig. 6). However, MOEA/D-Tch is the worst when the near reference point and the smallest population size are used (see Fig. 5 and Fig. 7).

The best population size with the maximum hypervolume for each algorithm in each figure is as follows:

- The 4-500 problem with the far reference point in Fig. 4:  
**120** (MOEA/D-WS), **4960** (MOEA/D-Tch), **120** (HypE)
- The 4-500 problem with the near reference point in Fig. 5:  
**4960** (MOEA/D-WS), **4960** (MOEA/D-Tch), **451** (HypE)
- The 6-500 problem with the far reference point in Fig. 6:  
**252** (MOEA/D-WS), **4368** (MOEA/D-Tch), **462** (HypE)
- The 6-500 problem with the near reference point in Fig. 7:  
**4368** (MOEA/D-WS), **4368** (MOEA/D-Tch), **462** (HypE)

An interesting observation is that the best population size for MOEA/D-WS strongly depends on the location of the reference point. This observation clearly contrasts to the results

by MOEA/D-Tch where the largest population size is always the best specification for all the four cases in Figs. 4-7.

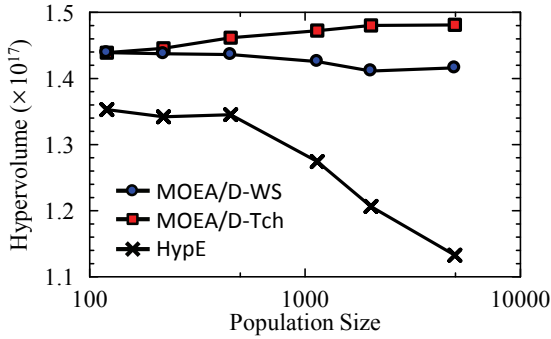


Fig. 4. Results on the 4-500 problem with the far reference point (0, ..., 0).

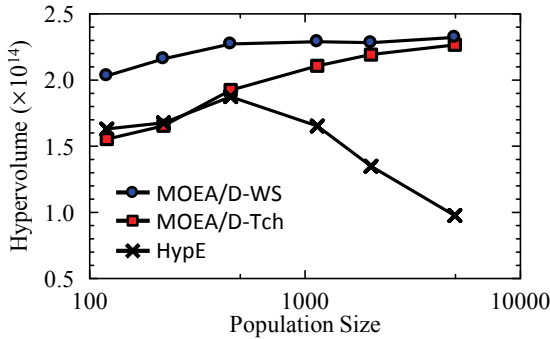


Fig. 5. Results on the 4-500 problem with the near reference point (15000, ..., 15000).

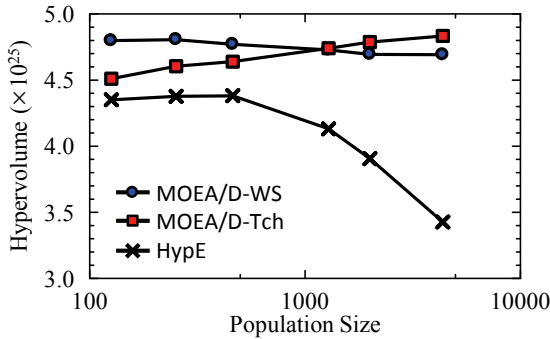


Fig. 6. Results on the 6-500 problem with the far reference point (0, ..., 0).

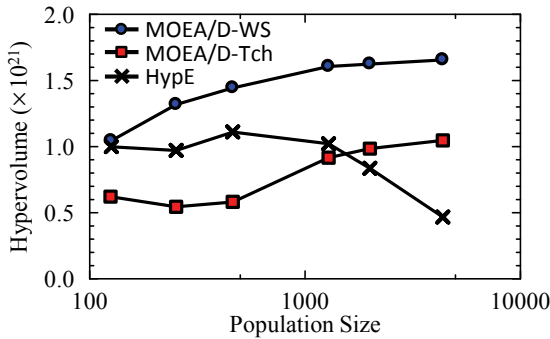


Fig. 7. Results on the 6-500 problem with the near reference point (15000, ..., 15000).

In Fig. 8 and Fig. 9, we show the projection of the obtained solutions by MOEA/D-WS for the 6-500 problem onto the  $f_1$ - $f_2$  subspace of the six-dimensional objective space. In Fig. 8, the population size is 252, which is the best specification for the far reference point (0, 0, 0, 0, 0, 0). Fig. 8 (a) shows the obtained solutions by a single run while Fig. 8 (b) shows all solutions obtained by five runs. In the same manner, Fig. 9 shows the experimental results for the population size 4368, which is the best specification for the near reference point (15000, ..., 15000).

After a large number of generations with a small population, solutions in Fig. 8 are spread out over a larger region than Fig. 9. Since the reference point (0, 0, 0, 0, 0, 0) is far from the Pareto front, a large spread of solutions is needed for good hypervolume values. On the contrary, after a small number of generations with a large population, solutions in Fig. 9 are gathered in a smaller region than Fig. 8. Since the reference point (15000, ..., 15000) is close to the center of the Pareto front, good convergence of solutions is needed for good hypervolume values.

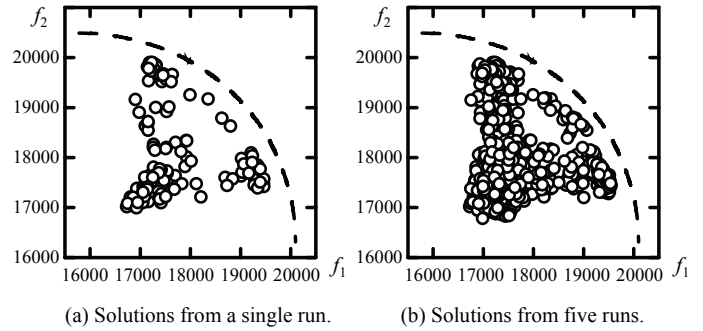


Fig. 8. Obtained solutions of the 6-500 problem by MOEA/D-WS with the best population size 252 for the far reference point (0, ..., 0).

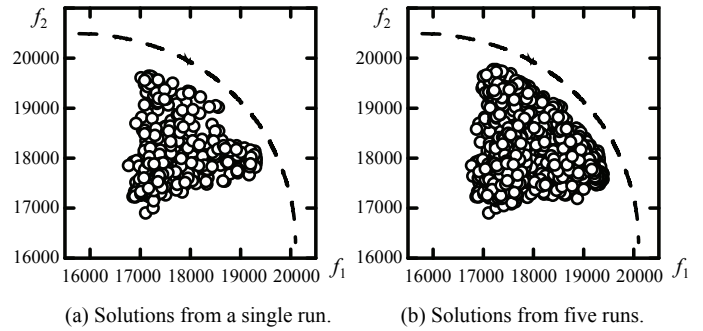


Fig. 9. Obtained solutions of the 6-500 problem by MOEA/D-WS with the best population size 4368 for the near reference point (15000, ..., 15000).

In the same manner as in Fig. 8 and Fig. 9, we show the obtained solutions by MOEA/D-Tch and HypE in Fig. 10 and Fig. 11, respectively. In these figures, the population size is 4368 for MOEA/D-Tch and 462 for HypE, which are the best specifications for the two settings of the reference point.

We can see that the obtained solutions in Fig. 10 have a larger diversity than those in the other figures. This is the reason why good results are obtained by MOEA/D-Tch in Fig. 6 for the far reference point (whereas good results are not obtained by MOEA/D-Tch in Fig. 7 for the near reference point). On the contrary, obtained solutions by HypE are gathered around the center of the Pareto front. Thus good results are not obtained in Fig. 6 for the far reference point.

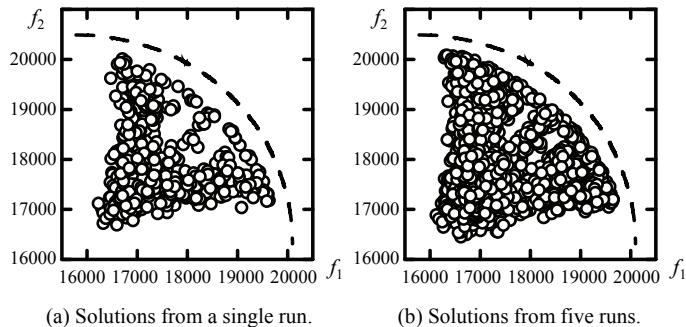


Fig. 10. Obtained solutions of the 6-500 problem by MOEA/D-Tch with the best population size 4368 for the two settings of the reference point.

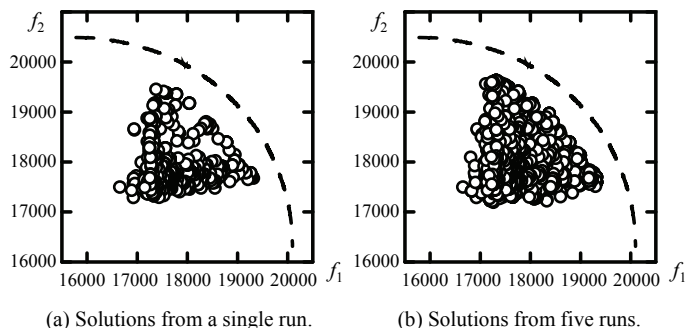


Fig. 11. Obtained solutions of the 6-500 problem by HypE with the best population size 462 for the two settings of the reference point.

If we compare the three EMO algorithms using the best population size for each case, they are evaluated as follows:

The 4-500 problem with the far reference point in Fig. 4:

Best: **MOEA/D-Tch**, Worst: **HypE**

The 4-500 problem with the near reference point in Fig. 5:

Best: **MOEA/D-WS**, Worst: **HypE**

The 6-500 problem with the far reference point in Fig. 6:

Best: **MOEA/D-Tch**, Worst: **HypE**

The 6-500 problem with the near reference point in Fig. 7:

Best: **MOEA/D-WS**, Worst: **MOEA/D-Tch**

However, these comparisons seem to be unfair since the size of the compared solution sets is not the same. For example, MOEA/D-Tch is evaluated using much larger solution sets than HypE. This is because the best specification of the population size for MOEA/D-Tch is larger than 4000 in Figs. 4-7 whereas it is smaller than 500 for HypE. In the next section, we discuss the comparison of solution sets of different size.

## IV. SOLUTION SELECTION

In this section, we compare the following three algorithms under the best specifications of the population size for the far reference point  $(0, \dots, 0)$  of the 6-500 problem in Fig. 6:

MOEA/D-WS with the population size 252,  
MOEA/D-Tch with the population size 4368,  
HypE with the population size 462.

We also compare the three algorithms under the best specifications of the population size for the near reference point  $(15000, \dots, 15000)$  of the 6-500 problem in Fig. 7:

MOEA/D-WS with the population size 4368,  
MOEA/D-Tch with the population size 4368,  
HypE with the population size 462.

Since the best specification of the population size for each algorithm is totally different, the size of the obtained solution set from each algorithm is different. In this section, we examine the use of several solution selection methods in order to compare solution sets of different size after selecting a pre-specified number of solutions from each solution set. In our computer simulations, we select 100 non-dominated solutions from the obtained solution set by each run of each algorithm.

Solution selection has been studied in the EMO community. In [25], [26], exact optimization algorithms were examined to find the best solution subset with respect to the hypervolume. A two-objective solution selection problem was formulated in [27], [28] where the hypervolume is maximized and the number of selected solutions is minimized. In [29], [30], a multiobjective solution selection problem is formulated to maximize the hypervolume for multiple reference points for hypervolume calculation.

Our idea in this paper is to use solution selection as a post-processing procedure for performance comparison of EMO algorithms with different specifications of the population size for many-objective optimization problems. We first remove overlapping solutions in the objective space from the obtained solution set. This is because overlapping solutions have no contributions to hypervolume calculation. If solutions A, B, C and D are overlapping in the objective space, three of them are removed from the solution set (e.g., B, C and D are removed). For the same reason, we remove all dominated solutions from the solution set. After that, solution selection is applied to the remaining non-overlapping non-dominated solution set.

### A. Random Selection

The simplest method for choosing a pre-specified number of solutions is random selection. We randomly select 100 solutions from the solution set obtained by each run of each algorithm. Then we calculate the average hypervolume value. This solution selection method is easy to implement. However, good results are not likely to be obtained since we randomly choose solutions without using any information about the contribution of each solution to hypervolume calculation.

### B. Hypervolume-Based Greedy Inclusion

Another simple method is to incrementally increase the number of solutions included in a solution set in a greedy

manner with respect to the hypervolume. First we choose a single solution with the largest hypervolume value as the first solution. This solution is included in the solution set. Next we choose the second solution which maximizes the hypervolume of the solution set. The set of the first and second solutions is not necessarily the best combination of two solutions since the second solution is selected after the inclusion of the first solution is decided. Now the inclusion of the first and second solutions is decided. Then we choose the third solution which maximizes the hypervolume of the solution set. In this manner, we select 100 solutions from the solution set obtained by each run of each algorithm. A similar idea to the incremental solution inclusion was used in a hypervolume-based EMO algorithm in [31].

### C. Hypervolume-Based Greedy Removal

The hypervolume-based greedy solution selection can be also used for incrementally decreasing the number of solutions in a solution set. First we calculate the contribution of each solution to the hypervolume calculation in the solution set. A single solution with the minimum hypervolume contribution is removed. If the solution set includes multiple solutions with no hypervolume contribution, we can remove all of them. Next we recalculate the contribution of each solution in the remaining solution set. A single solution with the minimum hypervolume contribution is removed from the remaining solution set. In this manner, we select 100 solutions from the obtained solution set by each run of each algorithm. This greedy removal method often needs much more computation time than the greedy inclusion method especially when the original solution set includes much more solutions than the number of solutions to be selected.

### D. Hypervolume-Based Genetic Selection

Solution selection can be viewed as a combinatorial optimization problem. We can apply a genetic algorithm with binary strings to solution selection in the same manner as its application to single-objective knapsack problems. The string length is the same as the number of solutions in the solution set. The inclusion and the exclusion of each solution are denoted by “1” and “0” in each binary string. The fitness value is defined by the hypervolume of the selected solutions in each binary string. The constraint condition is the number of the selected solutions (100 in our computer simulations).

We use the following setting in our genetic algorithm for solution selection.

Coding: Binary string,  
 Population size: 100,  
 Termination condition: 100,000 solution evaluations,  
 Crossover probability: 0.8 (Uniform crossover),  
 Mutation probability: 0.01 from 1 to 0,  
 $1/(\text{String Length})$  from 0 to 1,  
 Repair: Random removal of solutions.

### E. Simulation results

We apply the above-mentioned four methods for solution selection to the obtained solution set of the 6-500 problem by each run of each algorithm with the best population size for

each reference point. In Table I, we show the results when we use the best specification for the far reference point (0, ..., 0). Table I shows the average results over the five solution sets of the 6-500 problem obtained by five runs of each algorithm in Section III. The third row of Table I shows the average number of obtained non-overlapping solutions (i.e., the average size of the solution sets after removing overlapping solutions). The average number of non-overlapping solutions is much smaller than the population size in Table I. That is, a large number of overlapping solutions are included in the obtained solution sets. The fourth row shows the average number of non-dominated solutions (i.e., the average size of the solution sets after removing overlapping solutions and dominated solutions). The obtained solution sets include almost no dominated solutions (less than 3% among the non-overlapping solutions)

The fourth row shows the average size of the solution sets before solution selection. The fifth row shows their average hypervolume value. The last four rows of Table I show the average hypervolume values of the selected solution sets by each solution selection method. For simplicity, “ $\times 10^{25}$ ” is omitted in the last five rows of Table I for the average hypervolume values. The best result in each row is highlighted by bold in each of the last five rows. The average hypervolume value after the genetic solution selection is also shown in Fig. 12 on the vertical axis for each EMO algorithm (since 100 solutions are selected) together with all results in Fig. 6.

If the three EMO algorithms are compared before solution selection, MOEA/D-Tch is evaluated as being the best in Table I under the hypervolume indicator (i.e., 4.83 in the fifth row). However, after solution selection, MOEA/D-WS is evaluated as being the best. These observations show the difficulty of performance comparison using solution sets of different size.

TABLE I. SOLUTION SELECTION ON THE 6-500 PROBLEM WITH THE REFERENCE POINT (0, ..., 0) FOR HYPERVOLUME CALCULATION.

EMO Algorithms	MOEA/D-WS	MOEA/D-Tch	HypE
Population size	252	4368	462
# of non-overlapping solutions	151.2	687.2	272.0
# of non-dominated solutions	151.2	671.0	272.0
HV before solution selection	4.81	<b>4.83</b>	4.38
HV after random selection	<b>4.76</b>	4.62	4.31
HV after greedy inclusion	<b>4.80</b>	4.78	4.38
HV after greedy removal	<b>4.80</b>	4.78	4.38
HV after genetic selection	<b>4.80</b>	4.78	4.38

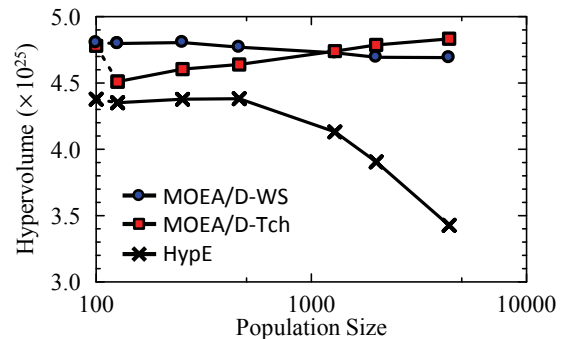


Fig. 12. Results on the 6-500 problem with the near reference point.

Among the four solution selection methods, clearly inferior results are obtained from random selection. As we can see from the last three rows, no clear difference is observed among the other three hypervolume-based solution selection methods. Moreover, the decrease in the average hypervolume value by solution selection is small in Table I. For example, in the case of MOEA/D-Tch, 100 solutions are selected from the obtained non-dominated solution sets of the average size 671. In this case, the average number of removed solutions is 571, which is 85% of the average size of the obtained solution sets. Even in this case, the decrease in the average hypervolume is small (i.e., from 4.83 to 4.78). Moreover, we can see from the results of MOEA/D-Tch in Fig. 12 that much better results are obtained by solution selection than the use of a small population size (e.g., the average hypervolume after the genetic solution selection is 4.78 in Table I whereas it is 4.51 by MOEA/D-Tch with the population size 126).

In Table II, we show the results by the best specification of the population size in each algorithm for the case of the near reference point (15000, ..., 15000) in the same manner as in Table I. We also show the results after the genetic solution selection on the vertical axis in Fig. 13 together with all results in Fig. 7. In Table II, the best results are always obtained by MOEA/D-WS before and after solution selection. This is consistent with Fig. 7 (see Fig. 13) where the best results are always obtained by MOEA/D-WS for all specifications of the population size. However, we can see from Table II and Fig. 13 that the much better average result is obtained by solution selection (i.e., 1.53 by the selected 100 solutions) than MOEA/D-WS with a small population in Fig. 13 (e.g., 1.04 by MOEA/D-WS with the population size 126). The much better average result is also obtained for MOEA/D-Tch by solution selection than the use of a small population size in Fig. 13.

TABLE II. SOLUTION SELECTION ON THE 6-500 PROBLEM WITH THE REFERENCE POINT (15000, ..., 15000) FOR HYPERVOLUME CALCULATION.

EMO Algorithms	MOEA/D-WS	MOEA/D-Tch	HypE
Population size	4368	4368	462
# of non-overlapping solutions	497.4	687.2	272.0
# of non-dominated solutions	483.8	671.0	272.0
HV before solution selection	<b>1.66</b>	1.05	1.11
HV after random selection	<b>1.38</b>	0.74	0.97
HV after greedy inclusion	<b>1.53</b>	0.90	1.09
HV after greedy removal	<b>1.53</b>	0.90	1.09
HV after genetic selection	<b>1.53</b>	0.90	1.10

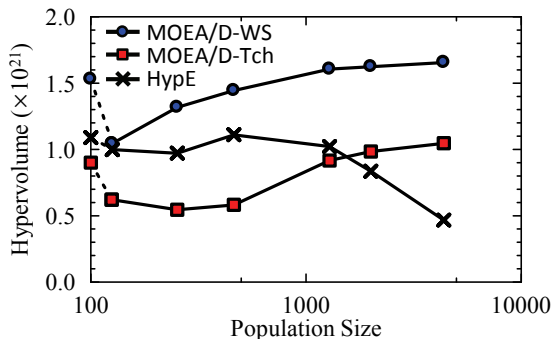


Fig. 13. Results on the 6-500 problem with the near reference point.

As in Table I, almost the same results are obtained by the last three rows in Table II. That is, except for random selection, almost the same results are obtained by the three hypervolume-based solution selection methods in Table II. It is interesting to observe that the better average results are obtained in Table II than MOEA/D-WS with a small population in Fig. 13 even when we use random selection (i.e., 1.38 by randomly selected 100 solutions in Table II and 1.04 by MOEA/D-WS with the population size 126 in Fig. 13).

## V. CONCLUDING REMARKS

In this paper, we obtained the following observations through computer simulations:

1. The best specification of the population size for a different EMO algorithm can be totally different. For example, its best specification for the 6-500 problem with the far reference point (0, 0, 0, 0, 0, 0) was 252 for MOEA/D-WS, 4368 for MOEA/D-Tch, and 462 for HypE.
2. The best specification of the population size for the same EMO algorithm can be totally different for a different setting. For example, the best specification for MOEA/D-WS on the 4-500 problem was 120 for the far reference point (0, 0, 0, 0) for hypervolume calculation while it was 4960 for the near reference point (15000, ..., 15000).
3. Performance comparison results depend on the specification of the population size. For example, with respect to the hypervolume value for the 6-500 problem with the far reference point, MOEA/D-WS was evaluated as being the best for small values of the population size while MOEA/D-Tch was evaluated as being the best for large values of the population size.
4. Much better results can be obtained by solution selection from a large solution set than the execution of an EMO algorithm with a small population size. For example, the average hypervolume value 1.53 was obtained for the 6-500 problem with the near reference point by selecting 100 solutions from the solution sets obtained by MOEA/D-WS with the population size 4368. This value was much larger than the average hypervolume value 1.04 by MOEA/D-WS with the population size 126.
5. The examined solution selection methods worked well except for random selection. No large differences in the average hypervolume of selected solutions were observed among the three hypervolume-based solution selection methods.

These observations show the difficulty in specifying the same population size for all algorithms when their performance is compared through computer simulations. When each algorithm is compared under its best population size, the size of the obtained solution set by each algorithm can be totally different. As a result, fair comparison among different solution sets becomes very difficult. In this paper, we proposed an idea of solution selection to handle this difficulty.

As shown in this paper, some EMO algorithms need a large population for realizing their high search ability for many-objective problems (e.g., MOEA/D-WS and MOEA/D-Tch for

the hypervolume with the near reference point). These algorithms are likely to be unfairly evaluated under the small population size. Solution selection can be used to compare those algorithms with other algorithms that work well with a small population.

In this paper, 100 solutions were selected from each of the obtained solution sets. This specification may need further discussions. For fair comparison, it may be needed to compare different EMO algorithms under some different specifications. This is because performance comparison results of different EMO algorithms may depend on the number of selected solutions. It may be possible that a different EMO algorithm is evaluated as being the best for a different specification of the number of solutions to be used for performance comparison.

#### REFERENCES

- [1] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," *Proc. of 2008 IEEE Congress on Evolutionary Computation*, pp. 2424-2431, June 1-6, 2008.
- [2] C. von Lüken, B. Barán, and C. Brizuela, "A survey on multi-objective evolutionary algorithms for many-objective problems," *Computational Optimization and Applications*, vol. 58, no. 3, pp. 707-756, July 2014.
- [3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002.
- [4] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," *TIK-Report 103*, Computer Engineering and Networks Laboratory (TIK), Department of Electrical Engineering, ETH, Zurich, 2001.
- [5] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired coevolutionary algorithms for many-objective optimization," *IEEE Trans. on Evolutionary Computation*, vol. 17, no. 4, pp. 474-494, August 2013.
- [6] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Trans. on Evolutionary Computation*, vol. 17, no. 5, pp. 721-736, October 2013.
- [7] Z. He, G. G. Yen, and J. Zhang, "Fuzzy-based Pareto optimality for many-objective evolutionary algorithms," *IEEE Trans. on Evolutionary Computation*, vol. 18, no. 2, pp. 269-285, April 2014.
- [8] M. Li, S. Yang, and X. Liu, "Shift-based density estimation for Pareto-based algorithms in many-objective optimization," *IEEE Trans. on Evolutionary Computation*, vol. 18, no. 3, pp. 348-365, June 2014.
- [9] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part I: Solving problems with box constraints," *IEEE Trans. on Evolutionary Computation*, vol. 18, no. 4, pp. 577-601, August 2014.
- [10] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. on Evolutionary Computation*, vol. 18, no. 4, pp. 602-622, August 2014.
- [11] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," *Proc. of 2002 IEEE Congress on Evolutionary Computation*, pp. 825-830, May 12-17, 2002.
- [12] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. on Evolutionary Computation*, vol. 10, no. 5, pp. 477-506, 2006.
- [13] D. Corne and J. Knowles, "Techniques for highly multiobjective optimization: Some nondominated points are better than others," *Proc. of 2007 Genetic and Evolutionary Computation Conference*, pp. 773-780, July 7-11, 2007.
- [14] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. on Evolutionary Computation*, vol. 3, no. 4, pp. 257-271, November 1999.
- [15] A. Jaszkiewicz, "On the performance of multiple-objective genetic local search on the 0/1 knapsack problem - A comparative experiment," *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 4, pp. 402-412, 2002.
- [16] H. Sato, H. E. Aguirre, and K. Tanaka, "Controlling dominance area of solutions and its impact on the performance of MOEAs," *Lecture Notes in Computer Science 4403: Evolutionary Multi-Criterion Optimization - EMO 2007*, pp. 5-20, Springer, Berlin, March 2007.
- [17] H. Ishibuchi, N. Akedo, and Y. Nojima, "Behavior of multi-objective evolutionary algorithms on many-objective knapsack problems," *IEEE Trans. on Evolutionary Computation* (Online Available from Xplore).
- [18] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization by NSGA-II and MOEA/D with large populations," *Proc. of 2009 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1820-1825, October 10-13, 2009.
- [19] N. Kowatari, A. Oyama, H. E. Aguirre, and K. Tanaka, "A study on large population MOEA using adaptive  $\epsilon$ -box dominance and neighborhood recombination for many-objective optimization," *Lecture Notes in Computer Science 7219: Learning and Intelligent Optimization - LION 6*, pp. 86-100, 2012.
- [20] N. Kowatari, A. Oyama, H. Aguirre, and K. Tanaka, "Analysis on population size and neighborhood recombination on many-objective optimization," *Lecture Notes in Computer Science 7492: Parallel Problem Solving from Nature - PPSN XII, Part 2*, pp. 22-31, 2012.
- [21] H. Aguirre, A. Liefooghe, S. Verel, and K. Tanaka, "A study on population size and selection lapse in many-objective optimization," *Proc. of 2013 IEEE Congress on Evolutionary Computation*, pp. 1507-1514, June 20-23, 2013.
- [22] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45-76, Spring 2011.
- [23] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. on Evolutionary Computation*, vol. 11, no. 6, pp. 712-731, December 2007.
- [24] 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. on Evolutionary Computation*, vol. 16, no. 4, pp. 504-522, August 2012.
- [25] K. Bringmann, T. Friedrich, and P. Klitzke, "Generic postprocessing via subset selection for hypervolume and epsilon-indicator," *Lecture Notes in Computer Science 8672: Parallel Problem Solving from Nature - PPSN XIII*, pp. 518-527, 2014.
- [26] T. Glasmachers, "Optimized approximation sets for low-dimensional benchmark Pareto fronts," *Lecture Notes in Computer Science 8672: Parallel Problem Solving from Nature - PPSN XIII*, pp. 569-578, 2014.
- [27] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Single-objective and multi-objective formulations of solution selection for hypervolume maximization," *Proc. of 2009 Genetic and Evolutionary Computation Conference*, pp. 1831-1832, July 8-12, 2009.
- [28] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Selecting a small number of representative non-dominated solutions by a hypervolume-based solution selection approach," *Proc. of 2009 IEEE International Conference on Fuzzy Systems*, pp. 1609-1614, August 20-24, 2009.
- [29] H. Ishibuchi, H. Masuda, and Y. Nojima, "Meta-level multi-objective formulations of set optimization for multi-objective optimization problems: Multi-reference point approach to hypervolume maximization," *Companion of 2014 Genetic and Evolutionary Computation Conference*, pp. 89-90, July 12-16, 2014.
- [30] H. Ishibuchi, H. Masuda, and Y. Nojima, "Selecting a small number of non-dominated solutions to be presented to the decision maker," *Proc. of 2014 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 3850-3855, October 5-8, 2014.
- [31] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Iterative approach to indicator-based multiobjective optimization," *Proc. of 2007 IEEE Congress on Evolutionary Computation*, pp. 3697-3704, September 25-28, 2007.