This paper has been published as "H. Ishibuchi, L. M. Pang, and K. Shang, "Difficulties in fair performance comparison of multi-objective evolutionary algorithms [research frontier]," *IEEE Computational Intelligence Magazine*, vol. 17, no. 1, pp. 86-101, Feb. 2022. DOI: 10.1109/MCI.2021.3129961"

Difficulties in Fair Performance Comparison of Multi-Objective Evolutionary Algorithms

Hisao Ishibuchi, Lie Meng Pang, and Ke Shang
Southern University of Science and Technology, China

***Abstract*—The performance of a newly designed evolutionary algorithm is usually evaluated by computational experiments in comparison with existing algorithms. However, comparison results depend on experimental setting; thus, fair comparison is difficult. Fair comparison of multi-objective evolutionary algorithms is even more difficult since solution sets instead of solutions are evaluated. In this paper, the following four issues are discussed for fair comparison of multi-objective evolutionary algorithms: (i) termination condition, (ii) population size, (iii) performance indicators, and (iv) test problems. Whereas many other issues related to computational experiments such as the choice of a crossover operator and the specification of its probability can be discussed for each algorithm separately, all the above four issues should be addressed for all algorithms simultaneously. For each issue, its strong effects on comparison results are first clearly demonstrated. Then, the handling of each issue for fair comparison is discussed. Finally, future research topics related to each issue are suggested.**

Keywords*—Performance comparison, performance indicators, test problems, termination condition, evolutionary multi-objective optimization.

I. Introduction

Evolutionary multi-objective optimization (EMO) is an active research topic. Various EMO algorithms are proposed every year. A newly proposed algorithm is evaluated through computational experiments by comparing it with existing ones using performance indicators. Experimental results are reported as a number of tables together with statistical significance test results. Those results usually show high performance of the new algorithm. In this manner, a number of EMO algorithms were proposed including well-known classic algorithms (e.g., SPEA [1], NSGA-II [2], MOEA/D [3], SMS-EMOA [4]) as well as recently-proposed algorithms for many-objective optimization (HypE [5], PICEA-g [6], GrEA [7], NSGA-III [8], Two_Arch2 [10], MOEA/DD [9], θ-DEA [11], RVEA [12], SPEA/R [13], LMEA [14], AR-MOEA [15]).

In each of these studies, it was shown by computational experiments that a newly proposed algorithm outperformed existing algorithms. However, fair comparison of different EMO algorithms is not easy since the evaluated performance of each algorithm usually depends on experimental setting. In general, each algorithm can be evaluated under its best setting. For example, each algorithm can use an appropriate mutation operator and an appropriate mutation probability. This paper does not discuss the specifications of those parameters which can be specified separately from the other algorithms. However, some items in the setting of the computational experiments should be commonly specified over all algorithms.

This paper discusses how to specify the common setting of computational experiments, which is used by all the compared EMO algorithms. More specifically, the focus of this paper is the setting of the following four issues: (i) the termination condition of each algorithm, (ii) the population size of each algorithm, (iii) performance indicators, and (iv) test problems. Except for the population size, it is clear that the same setting should be used in all the compared EMO algorithms in computational experiments. In single-objective optimization, each algorithm can use a different population size specification. This is because the performance is evaluated using the obtained best solution. However, a solution set (usually the final population) is used to evaluate each EMO algorithm. In many cases, larger solution sets tend to have better indicator values. Thus, the same population size has been used whereas each EMO algorithm has its own appropriate specification.

Our intention is to clearly illustrate various difficulties in fair comparison of EMO algorithms, which are related to the above-mentioned four issues. Our discussions on those difficulties are to encourage the future development of the EMO research field without excessively focusing on the proposal of overly-specialized new algorithms in a specific setting. This is because those algorithms are not likely to work well on various real-world tasks.

In this paper, the dependency of performance comparison results on the termination condition is examined in Section II. Such dependency indicates the importance of performance comparison under the anytime algorithm framework. Next, the effect of the population size is examined in Section III. It is shown that different comparison results are obtained depending on population size specifications. For handling this issue, the use of the solution subset selection framework is suggested. From the final population or from all the evaluated solutions, a solution subset of a pre-specified size is selected for fair comparison. Then, in Section IV, difficulties related to performance indicators [16] are discussed. It is shown that the choice of a performance indicator has large effects on comparison results. It is also shown that comparison results depend on the setting in each indicator. For example, different reference points for hypervolume calculation generate different comparison results. Our experimental results show the necessity of using multiple indicators. In Section V, the choice of test problems is discussed. Comparison results based on newly proposed test problems are totally different from reported results for the DTLZ [17] and WFG [18] test suites. This indicates that the use of various test problems is needed in computational experiments for fair comparison. Finally, in Section VI, this paper is concluded with a list of suggestions.

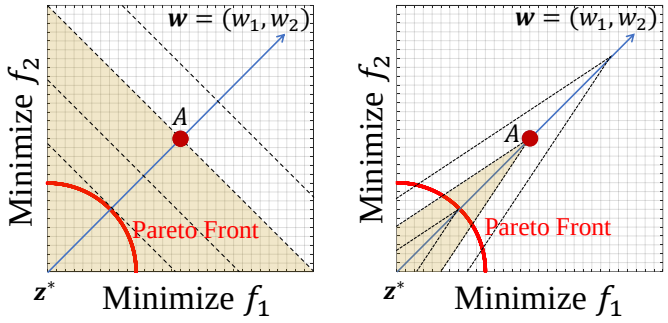
Corresponding Author: Hisao Ishibuchi (hisao@sustech.edu.cn)

II. Specification of Termination Condition

In computational experiments, a pre-specified number of solutions evaluated in each algorithm is used to terminate its execution. That is, all algorithms are compared under the same pre-specified computational effort. The specification of this termination condition is discussed in this section.

In general, the point in EMO algorithm design is to implement a good balancing mechanism between diversity and convergence (exploration and exploitation, global search and local search). Some algorithms are developed to quickly find good non-dominated solutions. Others are designed to find well-distributed solutions after spending long computation time. If they are compared, convergence-oriented algorithms will show better performance in early generations. However, after a large number of generations, diversity-oriented algorithms can be better. It is likely that the specification of the termination condition has large effects on their comparison results.

To demonstrate large effects of the termination condition, three implementations of MOEA/D [3] are compared: MOEA/D-WS with a weighted sum function, MOEA/D-PBI with a PBI function, and MOEA/D-Tch with a Tchebycheff function. As the penalty parameter value (i.e., θ) in MOEA/D-PBI, $\theta = 5$ is used as in many studies including the MOEA/D paper [3]. Contour lines are shown in Fig. 1 for the weighted sum and the PBI function for the weight vector $\boldsymbol{w} = (0.5, 0.5)$. It is assumed that both objectives $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ are to be minimized. The shaded area in each figure in Fig. 1 shows the better region than a current solution A with respect to the corresponding scalarizing function. If a new solution is generated in the shaded region, the current solution is replaced.



(a) Weighted sum. (b) PBI function ($\theta = 5$).

Fig. 1. Contour lines for $\boldsymbol{w} = (0.5, 0.5)$. The shaded area shows the better region than point A by each scalarizing function.

In Fig. 1, if a new solution is generated from the current solution A randomly in its neighborhood in the objective space, the current solution A has a much higher probability to be replaced with the new solution in Fig. 1 (a) than Fig. 1 (b). As a result, the weighted sum in Fig. 1 (a) has stronger convergence ability than the PBI function in Fig. 1 (b). However, the final solution for the weighted sum is not always close to the weight vector. Thus, well-distributed solutions are not always obtained from the weighted sum [19]. In contrast, the final solution for the PBI function is usually close to the weight vector since the better region is narrow and close to the weight vector in Fig. 1 (b). After many generations, it is likely that well-distributed solutions are obtained by the PBI function. When the Tchebycheff function is used, the better region of the current solution A is the same as its dominating region, which is smaller than Fig. 1 (a) but larger than Fig. 1 (b).

Each version of MOEA/D is applied to the 10-objective DTLZ2 [17] and WFG3 [18] test problems. For comparison, the 10-objective HTNY19 [20], [21] test problem is also used. DTLZ and WFG are well-known frequently-used test suites. The Pareto front of DTLZ2 is concave triangular [17]. WFG3 was originally intended to be a degenerate test problem [18]. However, it actually has a partially degenerate Pareto front [22]. HTNY19 is a recently-proposed difficult scalable test problem, which has the following formulation [20], [21]:

$$\text{Minimize} \quad f_i(\boldsymbol{x}) = \max\left\{0,\ x_i - \beta \sum_{\substack{j=1 \\ j \neq i}}^{m} x_j\right\} \quad \text{for} \quad i = 1, 2, ..., m, \quad (1)$$

$$\text{subject to} \quad 0 \le x_i \le 100 \quad \text{for} \quad i = 1, 2, ..., m, \quad (2)$$

$$\text{where} \quad f_i(\boldsymbol{x}) = 10000 \quad \text{for} \quad i = 1, 2, ..., m \quad \text{if} \quad \sum_{i=1}^{m} f_i(\boldsymbol{x}) < 1. \quad (3)$$

The value of β is specified as $\beta = 0.1$ in our experiments.

All objectives in each DTLZ (WFG) test problem share the same distance function. Thus, good convergence is easily realized by minimizing the distance function. If the value of the distance function is zero, the solution is on the Pareto front. The distance from the Pareto front is proportional to the distance function value. Thus, the convergence improvement of solutions is single-objective optimization. As a result, it is easy for many EMO algorithms to find good solutions close to the Pareto fronts of DTLZ, WFG and their minus versions [23]. In contrast, the objectives of HTNY19 have no shared distance function. Moreover, there is an interrelation among the objectives, which is specified by the parameter β in Eq. (1). These features make HTNY19 very difficult.

The following setting is used in each algorithm:

Number of decision variables (n):
$n = 19$ in DTLZ2 and WFG3, $n = 10$ in HTNY19.
Population size: 275 for all 10-objective problems.
Neighborhood size: 20.
Polynomial mutation: Index 20, Probability $1/n$.
SBX crossover: Index 30, Probability 1.0.
Termination condition: 1, 2, 5, 10, 20, 50, ...,10000 generations.
Number of runs of each algorithm: 31 independent runs.

The performance of each algorithm is evaluated for various specifications of the termination condition from 1 to 10000 generations. Except for the termination condition, the above setting is the same as or similar to the setting in many other studies (e.g., in the NSGA-III paper [8]). The hypervolume indicator [24] is used to evaluate each algorithm. Hypervolume values are calculated for the reference point (1.2, ..., 1.2) using the normalized objective space with the nadir point (1, ..., 1) and the ideal point (0, ..., 0). The PlatEMO framework [25] is used in our computational experiments.

In Figs. 2-4, average hypervolume values are summarized for each algorithm on each test problem. The vertical axis of each figure shows average hypervolume (HV) values. In Fig. 2 on DTLZ2, average hypervolume values quickly increase in the first 100 generations. After that, their changes are small. These observations suggest that solutions in each algorithm are quickly converged to the Pareto front. Since DTLZ2 is not a

difficult problem to achieve good convergence, solution distributions on the Pareto front have large effects on comparison results. The best and worst results are obtained by MOEA/D-PBI and MOEA/D-WS, respectively, independent of the termination condition. These observations are consistent with other studies (e.g., [23]) where the best and worst results on DTLZ1-4 were obtained by MOEA/D-PBI and MOEA/D-WS, respectively, among the three MOEA/D implementations.

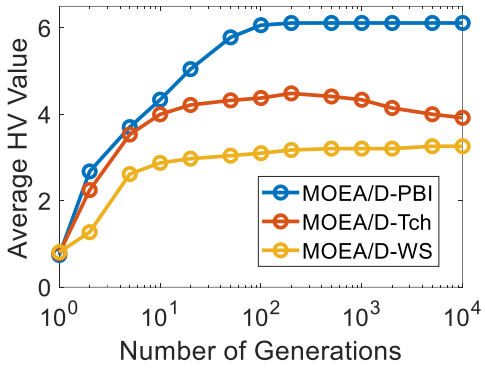


Fig. 2. Average HV results for 10-objective DTLZ2.

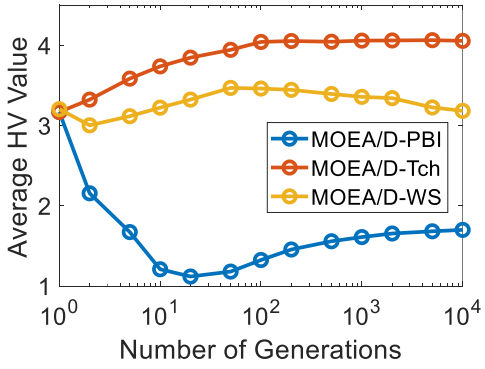


Fig. 3. Average HV results for 10-objective WFG3.

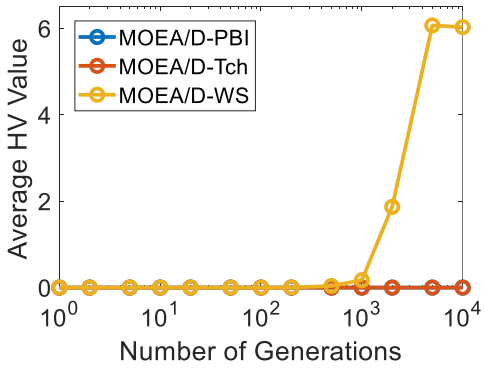


Fig. 4. Average HV results for 10-objective HTNY19.

Figure 3 shows the experimental results on WFG3. WFG3 has a partially degenerate Pareto front [22]. Its shape is totally different from the weight vector distribution. As a result, MOEA/D-PBI shows a strange search behavior in Fig. 3. As in Fig. 2, the change of the average hypervolume values in Fig. 3 is small after the 100th generation of each algorithm. In Fig. 3, MOEA/D-Tch shows the best performance independent of the termination condition.

In Fig. 2 and Fig. 3, performance comparison results are independent of the termination condition (whereas they are strongly problem-dependent). This is because the frequently-used DTLZ and WFG test problems are easy problems for MOEA/D with respect to the convergence of solutions since all objectives in each test problem share a common distance function. That is, it is easy for MOEA/D to find good solutions close to their Pareto fronts. As a result, comparison results do not depend on the convergence ability of each algorithm (they depend on the distribution of solutions in each algorithm).

However, when the three MOEA/D implementations are applied to difficult problems in terms of convergence, their comparison results depend on the termination condition as demonstrated in Fig. 4 for 10-objective HTNY19. If the termination condition is less than or equal to 1000 generations, the average hypervolume values are zero by all the three implementations. However, if the termination condition is 5000 or 10000 generations, MOEA/D-WS clearly shows much better results than the other two MOEA/D implementations.

To clearly show the effect of the termination condition, the 7-objective HTNY19 problem is used in the same manner as in Figs. 2-4. Figure 5 shows experimental results where only the population size is changed from 275 (in Figs. 2-4) to 294 (in Fig. 5). In Fig. 5, MOEA/D-WS with strong convergence ability shows the best performance before the 20th generation and the worst performance after the 50th generation.

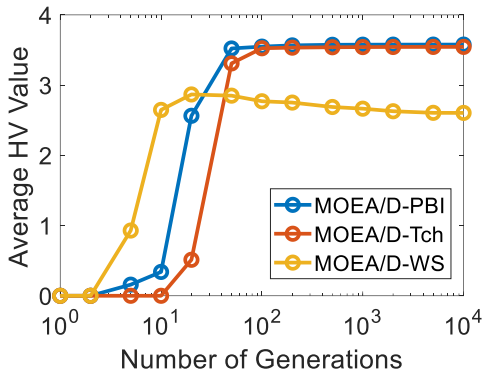


Fig. 5. Average HV results for 7-objective HTNY19.

To further analyze the experimental results in Fig. 4 and Fig. 5, the convergence of each solution and the diversity of each population are examined during the execution of each algorithm. To evaluate the convergence, the sum of all objective values of each solution is used. When the sum is 1, the solution is on the Pareto front. The sum also shows the Manhattan distance of each solution from the ideal point (0, 0, ..., 0). The average value of the sum of all objective values is calculated at each generation over all solutions in 31 runs of each algorithm. Whereas the generational distance (GD) indicator [26] has been often used as a convergence indicator, the sum of all objective values is used here for its simplicity. To evaluate the diversity of the population, the number of different solutions in the population is counted in each generation. In general, MOEA/D has a strong diversification ability due to the use of uniformly-distributed weight vectors. However, when multiple weight vectors share the same solution, the diversity of solutions in a population is not high

since the population includes duplicated solutions. In this case, the number of different solutions is smaller than the number of weight vectors (i.e., the population size). The average value of the number of different solutions is calculated at each generation over 31 runs of each algorithm.

Experimental results on the 7-objective HTNY19 problem are shown in Fig. 6 and Fig. 7, which correspond to Fig. 5. As shown in Fig. 6, the sum of all objective values quickly converges to 1 in each algorithm. This means that all solutions converge to the Pareto front. Among the three algorithms, MOEA/D-WS shows the fastest convergence ability, which explains the quickest increase of the HV value by MOEA/D-WS in Fig. 5. However, in Fig. 7, the number of different solutions decreases throughout the execution of MOEA/D-WS. As a result, the average HV values by MOEA/D-WS in Fig. 5 are smaller than those by the other algorithms except for very early generations.

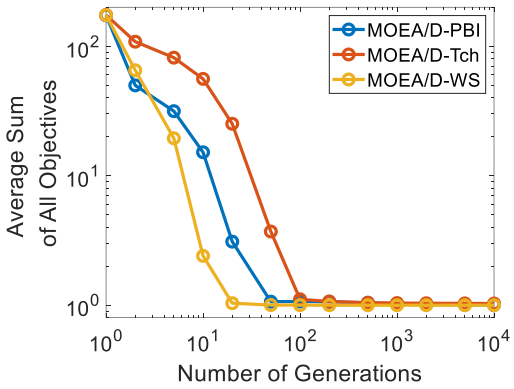


Fig. 6. Average sum of all objectives for 7-objective HTNY19.

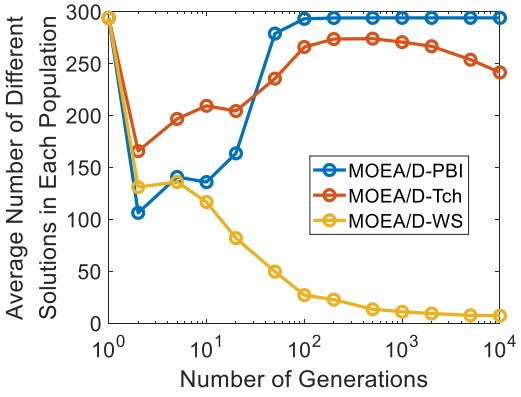


Fig. 7. Number of different solutions for 7-objective HTNY19.

Experimental results on the 10-objective HTNY19 problem are shown in Fig. 8 and Fig. 9, which correspond to Fig. 4. Figure 8 suggests that MOEA/D-Tch and MOEA/D-PBI have some difficulty in pushing the population towards the Pareto front. This observation is consistent with Fig. 4 where these two algorithms always have zero average HV values. In Fig. 8, only MOEA/D-WS can converge the population to the Pareto front. After converging to the Pareto front, the number of different solutions is severely decreased by MOEA/D-WS in Fig. 9. A similar behavior of MOEA/D-WS is also observed in Fig. 7.

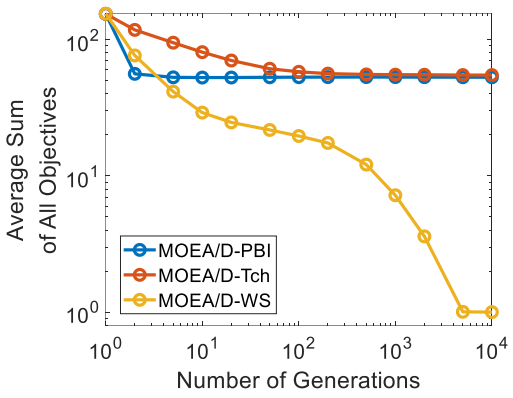


Fig. 8. Average sum of all objectives for 10-objective HTNY19.

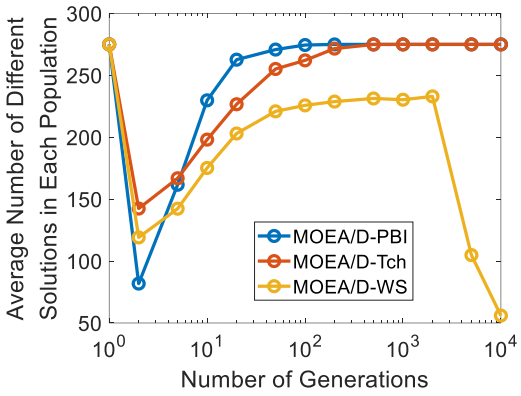


Fig. 9. Number of different solutions for 10-objective HTNY19.

The severe decrease in the number of different solutions at the second generation in Fig. 7 and Fig. 9 (i.e., from the left-most circles to the second left-most circles in these figures) is due to the random solution assignment in the first generation in all the three versions of MOEA/D. In MOEA/D, each weight vector has a randomly generated initial solution. Thus the total number of different solutions at the first generation is exactly the same as the population size. Since a solution is randomly generated and randomly assigned to each weight vector, the assigned solutions are not suitable for many weight vectors. As a result, many solutions are replaced with new good solutions in the second generation. Since multiple solutions are replaced with a single good solution, the number of different solutions is severely decreased in the second generation.

The dependency of performance comparison results on the termination condition is clearly shown in Fig. 4 and Fig. 5. If each algorithm is terminated at the 100th generation in Fig. 4, all algorithms are evaluated as having the same performance (whereas they show different performance after the 5000th generation). In Fig. 5, MOEA/D-WS can be evaluated as being the best and the worst depending on the termination condition. Since any specification is not fair, it is needed to use various termination conditions (at least two or three specifications). In this case, EMO algorithms are compared as anytime algorithms [27]. In some studies (e.g., Tanabe et al. [28]), EMO algorithms are compared as anytime algorithms under various termination conditions. Since it is not likely that a single algorithm is always the best over a wide range of generations, the choice of an appropriate algorithm from a large algorithm

pool depending on the termination condition seems to be a promising research direction. Of course, it is an interesting challenge to try to develop an EMO algorithm with robust performance over a wide range of termination conditions. In real-world applications, it is also important to develop a specialized EMO algorithm which works well under a specific termination condition (e.g., under a limited number of solution evaluations such as 500 solution evaluations).

When an anytime EMO algorithm is designed, an important issue is the performance oscillation of the current population through generation update. Some EMO algorithms with a steady state ($\mu+1$) generation update mechanism such as SMS-EMOA [4] carefully improve the current population by creating only a single new solution at each generation. Thus, usually they do not show any severe performance oscillation of the current population through generation update. However, some other EMO algorithms with a ($\mu+\mu$) generation update mechanism such as NSGA-II [2] replace many solutions with new solutions. As a result, they often show performance oscillation through generation update. That is, the performance of the current population can be worse than that of the previous population. Moreover, the final populations of those algorithms often include solutions that are dominated by old solutions (which were generated and deleted in previous generations) as shown in [29]. For designing a good anytime EMO algorithm with stable performance, it is important to carefully store good solutions in the current or archive population to achieve a stable performance with no severe performance oscillation.

III. Specification of Population Size

In general, performance indicator values are influenced by the solution set size. The hypervolume value is improved by adding any non-dominated solution to a solution set. Thus, larger solution sets are likely to have larger hypervolume values than smaller ones. For example, in NSGA-II, the final population is constructed by removing half solutions from a merged population of parents and offspring. It is likely that a larger (i.e., better) hypervolume value can be obtained by using the merged population instead of the final population. These discussions explain why the size of the final population should be the same for fair comparison (i.e., why the population size should be the same in all the compared EMO algorithms).

However, it is likely that an appropriate population size is different in each EMO algorithm. Some algorithms work well with a large population while others are efficient with a small population. Thus, any specification is not fair. This is because different performance comparison results will be obtained from different specifications. Depending on the population size, different algorithms will be evaluated as good algorithms.

For demonstrating large effects of the population size on performance comparison results, MOEA/D-Tch [3], NSGA-III [8], MOEA/DD [9] and θ-DEA [11] are applied to the five-objective WFG3 problem in a similar manner to [30] under the following setting:

Population size: 210, 1001, 5985.
Neighborhood size: 20.
Polynomial mutation: Index 20, Probability 1/n.
SBX crossover: Index 30, Probability 1.0.
Termination condition: 210,000 solution evaluations.

Each algorithm is evaluated under three population size specifications: standard (210), large (1001), and very large (5985). The termination condition corresponds to 1000 generations with the standard population size 210. Average hypervolume values are calculated using the final population in the normalized objective space over 31 runs.

Figure 10 summarizes experimental results for the three population size specifications in each EMO algorithm. The population size has large effects on comparison results in Fig. 10. While MOEA/D-Tch with the very large population works very well, its performance is not good under the standard setting. The largest average hypervolume value is obtained in Fig. 10 by MOEA/D-Tch with the very large population. However, it cannot be concluded that MOEA/D-Tch is the best algorithm in Fig. 10. This is because the calculated average hypervolume value of 5985 solutions cannot be compared in a fair manner with that of much fewer solutions obtained by the other algorithms (e.g., 210 solutions by NSGA-III).

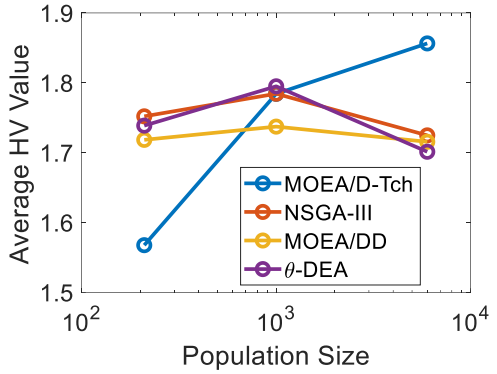


Fig. 10. Average HV results of the final population for five-objective WFG3.

One idea to compare all results in Fig. 10 in a fair manner is to select a solution subset of the same size (e.g., 210 solutions) from the final population. Here, a hypervolume-based greedy forward selection method [31], [32] is used. Solutions are selected one by one in a greedy manner for hypervolume maximization of the selected solution subset. That is, each solution is evaluated in the greedy algorithm by its contribution to the hypervolume of the selected solution subset. From the final population, 210 solutions are selected using the greedy algorithm. After that, their hypervolume is calculated. Then, the average result is calculated over 31 runs. Figure 11 shows experimental results where the largest average hypervolume value is obtained by MOEA/D-Tch when the population size is very large. Note that all solution subsets are compared under the condition of selecting 210 solutions in Fig. 11 whereas the final populations of different size (i.e., from 210 to 5985 solutions) are compared in Fig. 10. Thus, it can be concluded from Fig. 11 that the best solution sets are obtained by MOEA/D-Tch with the very large population.

The comparison between Fig. 10 and Fig. 11 shows that they are almost the same. The right-most four circles in Fig. 10 (and Fig. 11) are the average HV values of the final populations with 5985 solutions (and their subsets with 210 solutions). However, there are no large differences between Fig. 10 and Fig. 11. That is, the average HV values are not severely decreased by selecting 210 solutions from the final populations

with 5985 solutions.

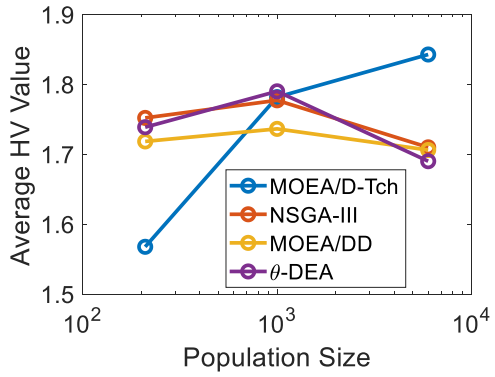


Fig. 11. Average HV results of selected 210 solutions from the final population in each run in Fig. 10.

In order to examine this observation, the number of non-dominated solutions is counted in each final population. The average number of non-dominated solution in each final population of size 5985 is as follows: 2964.33 (49.53%) in MOEA/D-Tch, 5984.50 (99.99%) in NSGA-III, 5710.89 (95.42%) in MOEA/DD, and 5983.33 (99.97%) in θ-DEA. The number of non-dominated solutions is much larger than 210. Next, the average HV values of selected solution subsets with different size are calculated. The following different specifications of the solution subset size are examined: 1, 2, 3, 4, 5, 10, 20, 30, 40, 50, 100, 210, 5985. Experimental results are summarized in Fig. 12. In this figure, NSGA-III results (red circles) are almost the same as MOEA/DD results (yellow circles). This figure shows that the selected solution subsets with 100 and 210 solutions have almost the same HV values as all the 5985 solutions in the final populations.

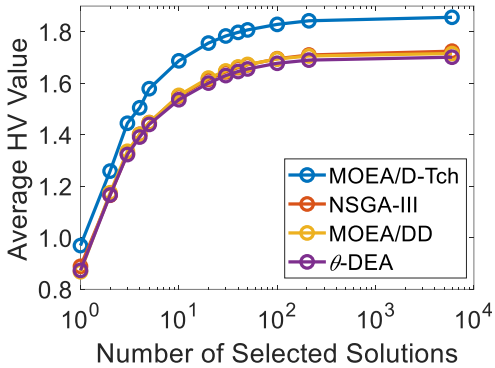


Fig. 12. Average HV results of selected solution subsets of different size (i.e., with 1, 2, 3, 4, 5, 10, 20, 30, 40, 50, 100, 210, 5985 solutions) from the final populations.

Another idea for fair performance comparison is to select a same size solution subset from all the evaluated solutions in each run (instead of the selection from the final population). Using the hypervolume-based greedy method, 210 solutions are selected from the evaluated 210,000 solutions in each run in Fig. 10. Then, their hypervolume value is calculated. Experimental results are summarized in Fig. 13, which are average results over 31 runs. Independent of the population size, MOEA/D-Tch shows high performance in Fig. 13. Note that better results are obtained in Fig. 13 than Fig. 10 and Fig. 11 (e.g., average hypervolume values are larger than 1.8 in many cases in Fig. 13). That is, better solution subsets can be selected by using not only solutions in the final population but also all the other evaluated solutions as candidate solutions.

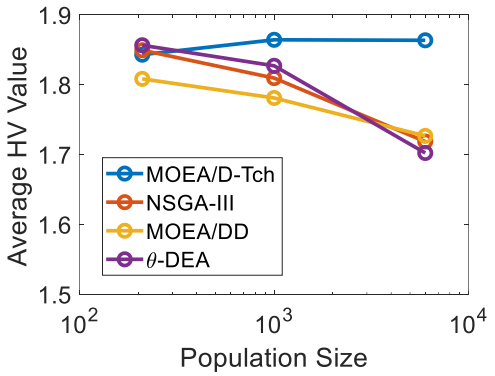


Fig. 13. Average HV results of selected 210 solutions from all the evaluated solutions in each run in Fig. 10.

It is shown in Fig. 13 that MOEA/D-Tch has high search ability to find good solutions during its execution independent of the population size. However, it needs a very large population to keep good solutions in the current population. This is the reason why the performance of the final population of MOEA/D is severely deteriorated by using a regular population size in Fig. 10 (i.e., the left-most blue circle in Fig. 10). It is also shown in Fig. 13 that the search ability of the other three algorithms is deteriorated by increasing the population size. This is explained as follows. Since our computational experiments are performed under the same number of solution evaluations, a larger population size means a smaller number of generations. The decrease in the number of generations degrades the convergence ability of those algorithms.

The use of the final population under the condition of the same population size for fair comparison is the frequently-used standard practice in the EMO community. However, different comparison results are obtained depending on the population size specification. To compare experimental results under different population size specifications, the idea of solution subset selection can be used as shown in Fig. 11 and Fig. 13. Better results than the final population in Fig. 10 are obtained by selecting solutions from all the evaluated solutions in Fig. 13. This means that a good subset of all the evaluated solutions is not always stored as the final population in each EMO algorithm. It has been clearly demonstrated in [29] that the final population often includes dominated solutions by other solutions which are generated and deleted in previous generations (i.e., which are not in the final population).

Figure 10 shows that the population size specification has large effects on final population-based comparison results. Whereas the appropriate specification of the population size for each algorithm is different, it is not fair to compare different algorithms using their final populations of different size (e.g., it is not fair to compare 210 solutions with 5985 solutions). In this situation, each algorithm can be evaluated in a fair manner after selecting a solution subset of the same size from the final population or from all the evaluated solutions.

Performance comparison results based on selected subsets from all the evaluated solutions have been reported in some studies [28], [30], [33]-[35]. By selecting solution subsets of the same size, each EMO algorithm can be evaluated under its best population size specification. However, at least the following two issues need to be further addressed:

(i) How to specify the size of solution subsets for performance comparison (i.e., how to specify the number of solutions to be selected for performance comparison).

(ii) How to select solution subsets efficiently in a realistic computation time, especially for many-objective problems.

For the first issue, it was reported in [36] that the size of solution subsets has only minor effects on comparison results (whereas the population size has large effects as shown in Figs. 10, 11, 13). That is, similar comparison results were obtained from computational experiments with various specifications of the solution subset size from 15 solutions to more than 5000 solutions in [36]. This is also consistent with Fig. 12.

With respect to the second issue, the main difficulty is that the number of evaluated solutions is very large, which can be hundreds of thousands of solutions. From those solutions, first, non-dominated solutions are selected. One research topic is the design of an efficient pre-screening method of non-dominated solutions [37]. The challenge is the handling of such a huge number of solutions. Next, a solution subset of a pre-specified size is selected from the obtained non-dominated solutions. This step includes a number of interesting and important research topics, which will be explained later.

Figure 14 shows the percentage of non-dominated solutions among evaluated solutions until the corresponding generation for the 10-objective DTLZ2, WFG3 and HTNY19 problems during the execution of MOEA/D-Tch with the same setting as in Section II (e.g., the population size is 275). In Fig. 14, the percentage of non-dominated solutions is high in all the three test problems. After 10,000 generations (i.e., among the evaluated 2,750,000 solutions), it is 84.6% (DTLZ2), 41.9% (WFG3), and 64.9% (HTNY19). That is, more than one million non-dominated solutions are obtained by each run on each test problem on average. The percentage is lower for WFG3 than that for the other test problems. This is because WFG3 has a partially degenerate Pareto front [22]. However, even for WFG3, more than 40% of the evaluated solutions (i.e., more than 1,100,000 solutions) are non-dominated.

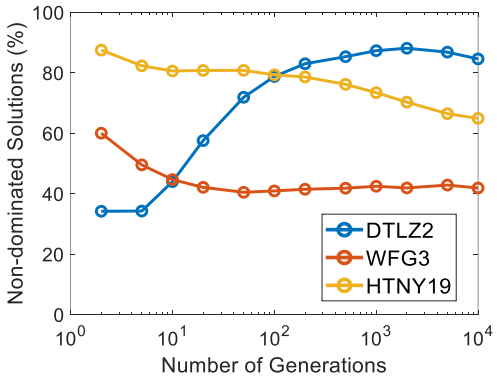


Fig. 14. Percentage of non-dominated solutions in evaluated solutions until the corresponding generation of MOEA/D-Tch (average results over 31 runs). For example, the percentage at the 1000th generation is calculated using the evaluated solutions in the first 1000 generations.

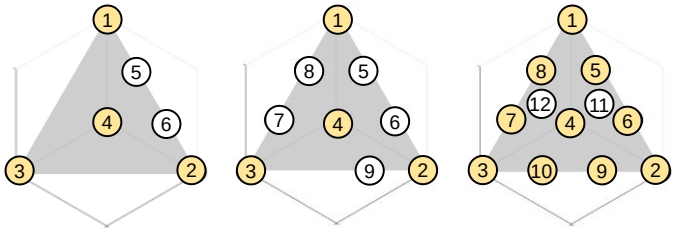
Solution subset selection using the hypervolume indicator has been actively studied. This research topic is often called hypervolume subset selection. Exact optimization methods have been proposed for maximizing the hypervolume of the selected subset [38]-[42]. The focus of those studies is mainly on environmental selection where the next population is selected from current and offspring solutions. It is difficult to apply those exact methods to solution subset selection from a large number of candidate (i.e., non-dominated) solutions for performance comparison due to their high computation complexity. This is the reason why the hypervolume-based greedy method is used in this paper.

Since hypervolume calculation is time-consuming, efficient methods which are not based on hypervolume calculation are often used for subset selection when a large number of candidate solutions are given. For example, in distance-based greedy methods [28], [43], the distance from each remaining solution to the selected solution subset (i.e., to the nearest selected solution) is calculated, and the most distant remaining solution is selected one by one. As initial solutions, the distance-based greedy method in [28] selects m extreme solutions of an m-objective problem. In [43], a randomly selected single extreme solution is used as an initial solution. Those distance-based greedy methods try to maximize the diversity and uniformity of solutions by maximizing the minimum distance between the selected solutions.

In Fig. 15 (a), the distance-based greedy method [43] is illustrated. First, a large number of candidate solutions are generated on the linear Pareto front defined by $f_1 + f_2 + f_3 = 1$ and $0 \leq f_i \leq 1$ for $i = 1, 2, 3$ in Fig. 15. More specifically, 5050 solutions are generated in the same manner as Das & Dennis method [44] (i.e., the generation method of weight vectors in MOEA/D). Next, its subset S is selected by the distance-based greedy method [43] in Fig. 15 (a). For comparison, the corresponding results by the hypervolume-based greedy method are shown in Fig. 15 (b) for the reference point (2, 2, 2). The reference point is specified as (1.2, 1.2, 1.2) in the hypervolume-based greedy method in Fig. 15 (c). The number in each circle in Fig. 15 shows the order of the selected solutions. In Fig. 15 (a), one extreme solution at the top of the Pareto front is first selected whereas one solution in the center region is first selected in Fig. 15 (b) and Fig. 15 (c).

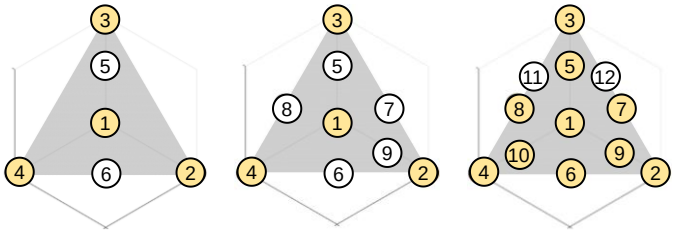
Figure 15 shows that uniformly distributed solution sets are not always obtained in a greedy manner. In Fig. 15 (a) and Fig. 15 (b), uniformly distributed solution sets are obtained only when the number of selected solutions is 4 and 10. Those solution sets are highlighted in Fig. 15. Moreover, independent of the number of selected solutions, uniform solution sets are not obtained in Fig. 15 (c). This is because the reference point $\boldsymbol{r}$ = (1.2, 1.2, 1.2) is not appropriate for greedy selection. These observations in Fig. 15 suggest a future research topic: proposal of subset selection methods which can efficiently select an arbitrarily number of uniformly distributed solutions.

Whereas subset selection has mainly been studied for the hypervolume indicator in the literature, other indicators can be also used for subset selection such as IGD [45] and IGD$^+$ [46]. The choice of an indicator for subset selection is closely related to that for performance comparison, which is discussed in the next section.



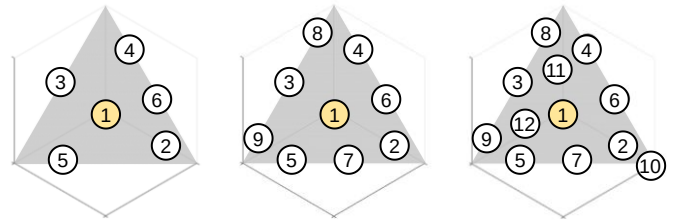
$|S| = 6.$ $|S| = 9.$ $|S| = 12.$

(a) Results of the distance-based greedy method.



$|S| = 6.$ $|S| = 9.$ $|S| = 12.$

(b) Results of the hypervolume-based greedy method ($r = 2$).



$|S| = 6.$ $|S| = 9.$ $|S| = 12.$

(c) Results of the hypervolume-based greedy method ($r = 1.2$).

Fig. 15. Illustrations of greedy subset selection. Subset selection is performed from 5050 uniformly distributed Pareto optimal solutions.

Performance comparison after selecting solution subsets of the pre-specified size from all the evaluated solutions suggests a new framework for EMO algorithm design where all the evaluated solutions are stored in an unbounded external archive separately from the current population [47]-[49]. Most EMO algorithms have been designed to find a good final population. However, if a subset of all the evaluated solutions is selected as the final result, it is not needed to find a good final population since the final population is not the final result in the new framework. Moreover, storing a good solution set as the current population is not always needed. This increases the flexibility in EMO algorithm design. Actually, different algorithm configurations of MOEA/D were obtained using a hyper-heuristic algorithm [49] depending on the performance evaluation scenario: the final population-based evaluation and the selected solution subset-based evaluation.

It is usually assumed in the EMO community that non-dominated solutions are presented as the final result to the decision maker. However, if hundreds of thousands of non-dominated solutions are obtained (see Fig. 14), it is not likely that the decision maker wants to examine all of those solutions. Thus, their subset is selected and shown to the decision maker. In this case, one unexplored research topic is the explanation of why the presented solution subset is selected [50]. Whereas this issue has not been discussed in the literature, clear explanations about solution subset selection are important to improve the practical applicability of EMO algorithms to real-world tasks. When hundreds of thousands of non-dominated solutions are obtained, it is likely that the decision maker wants to examine only its subset (e.g., 10, 100 or 1000 solutions). At the same time, he/she may want to know the reason why the presented subset is selected.

To summarize this section, it is important to compare EMO algorithms under multiple population size specifications since different algorithms work well under different specifications. When different population size specifications are used for different EMO algorithms, fair comparison can be performed by selecting a pre-specified number of solutions from the final population or from all the evaluated solutions in each run of each algorithm.

IV. Choice of Performance Indicators

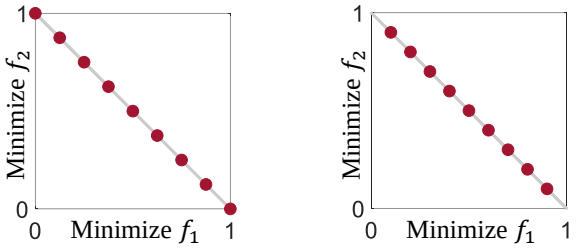
For performance comparison, the hypervolume [24] and IGD [45] indicators are often used. However, as shown in this section, different results are obtained depending on the reference point specification for hypervolume calculation and the reference point set specification for IGD calculation. The best solution set for each indicator is not always a uniform solution set which fully covers the entire Pareto front.

A. Difficulties in the Use of Hypervolume

One reason for the frequent use of the hypervolume indicator is that no other indicator is known as Pareto compliant [51]. Roughly speaking, comparison results between two solution sets by a Pareto compliant indicator are always consistent with Pareto dominance-based comparison results. Let S_1 and S_2 be two solution sets where S_1 is better than S_2 based on the Pareto dominance relation (see [16] for the "better" relation between solution sets). In this case, S_1 is always evaluated as better than S_2 by a Pareto compliant indicator. The hypervolume indicator is Pareto compliant whereas IGD is not. That is, S_1 always has a better hypervolume value than S_2 in the above-mentioned case while S_1 can have a worse IGD value than S_2. This issue will be further discussed later for IGD.

For a two-objective linear Pareto front, it has been theoretically shown that a uniform solution set is optimal for hypervolume maximization [52]-[54]. Let us consider the optimal distribution of μ solutions on a linear Pareto front for hypervolume maximization with the reference point $\boldsymbol{r} = (r, r)$. In Fig. 16, it is assumed that the Pareto front is a line between (0, 1) and (1, 0). When the inequality relation $r \geq 1 + 1/(\mu - 1)$ holds, a uniform solution set including (1, 0) and (0, 1) maximizes the hypervolume indicator as shown in Fig. 16 (a). Intuitively, Fig. 16 (a) shows the solution set with the best diversity and the best uniformity.

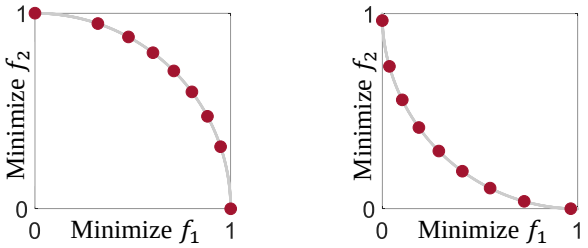
On the other hand, when $r < 1 + 1/(\mu - 1)$, the optimal distribution of solutions does not include the two extreme points (0, 1) and (1, 0), e.g., see Fig. 16 (b) with $\boldsymbol{r} = (1, 1)$. Note in Fig. 16 (b) that a uniformly distributed inside solution set is optimal. Figure 16 shows that different solution sets are optimal depending on the reference point specification. However, its effect is not large in the case of a two-objective linear Pareto front since the same solution set is always optimal when $r \geq 1 + 1/(\mu - 1)$. For the case of three or more objectives, its effect is totally different depending on the Pareto front shape as shown later in this section.



(a) $r \geq 1 + 1/(\mu - 1) = 9/8 = 1.125$. (b) $\boldsymbol{r} = (1, 1)$.

Fig. 16. Optimal solution distributions for hypervolume maximization with nine solutions ($\mu = 9$) for different reference points $\boldsymbol{r} = (r, r)$.

For nonlinear Pareto fronts, uniformly distributed solutions are not optimal [54]. Figure 17 shows two examples where SMS-EMOA [4] is used to find near-optimal distributions of nine solutions for DTLZ2 and Minus-DTLZ2 with $\boldsymbol{r} = (2, 2)$. In Fig. 17 (a), the two-objective DTLZ2 test problem [17] with no distance variable is used. Since the number of distance variables is zero, all feasible solutions are Pareto optimal (i.e., they are always on the Pareto front). Thus, the role of SMS-EMOA is to find the best distribution of nine solutions. The same setting as in the previous sections is used in Fig. 17 except for the population size ($\mu = 9$) and the termination condition (1,000,000 solution evaluations). To find a near-optimal distribution, the best solution set is selected from ten runs of SMS-EMOA. Figure 17 (a) shows the obtained solution set, which can be viewed as being very close to the optimal distribution since it is obtained using large computation load (i.e., 10 independent runs with 1,000,000 solution evaluations).

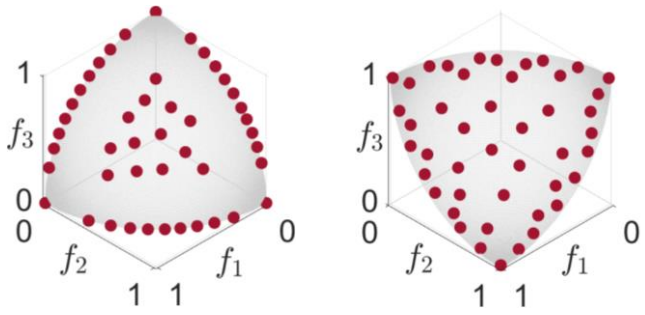


(a) Concave front (DTLZ2). (b) Convex front (Minus-DTLZ2).

Fig. 17. Optimal distributions of nine solutions for hypervolume maximization for the reference point $\boldsymbol{r} = (2, 2)$.

In the same manner, Fig. 17 (b) is obtained on the two-objective Minus-DTLZ2 test problem [23] with no distance variable. Figure 17 shows that the optimal distribution of solutions is not uniform on nonlinear Pareto fronts. More solutions are located around the center than the two extreme points (1, 0) and (0, 1) in Fig. 17. For the convex Pareto front in Fig. 17 (b), the two extreme points are not obtained. For example, the right-bottom point in Fig. 17 (b) is not (1, 0). This is because the two extreme points of the convex Pareto front do not have large hypervolume contributions [54], [55].

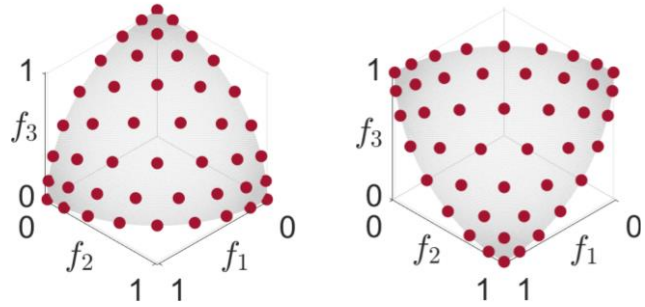
The non-uniformity of optimal distributions is more clearly demonstrated in Fig. 18 for three-objective problems. SMS-EMOA is used in the same manner as in Fig. 17 except for the population size (i.e., $\mu = 9$ in Fig. 17 and $\mu = 45$ in Fig. 18). Figure 18 (a) shows the obtained result for three-objective DTLZ2 with a concave Pareto front, and Fig. 18 (b) shows the obtained result for three-objective Minus-DTLZ2 with a convex Pareto front.



(a) DTLZ2. (b) Minus-DTLZ2.

Fig. 18. Hypervolume optimal distributions of 45 solutions for $\boldsymbol{r} = (2, 2, 2)$.

For comparison, two solution sets are generated using the 45 uniform weight vectors of Das & Dennis [44]. Figure 19 (a) is generated by calculating the intersection between the Pareto front of DTLZ2 and each weight vector. Figure 19 (b) is an inverted version of Fig. 19 (a). Intuitively, the solution sets in Fig. 19 are better than those in Fig. 18. However, each solution set in Fig. 18 has a better hypervolume value than the corresponding solution set in Fig. 19. These examples suggest that hypervolume-based comparison can be inconsistent with our intuition.



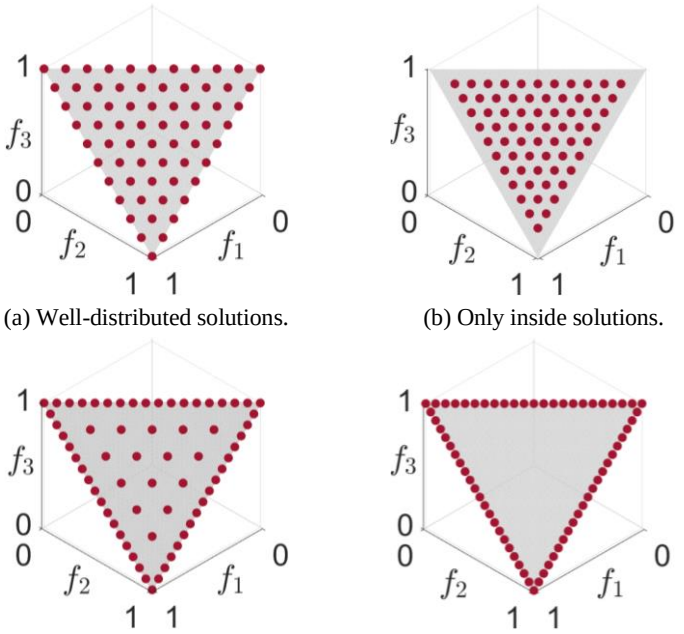
(a) DTLZ2. (b) Minus-DTLZ2.

Fig. 19. Solution sets generated by the 45 uniform weight vectors.

Next, dominant effects of the reference point are demonstrated in the case of inverted triangular Pareto fronts. For this purpose, four solution sets in Fig. 20 are generated on the Pareto front of the normalized three-objective Minus-DTLZ1 test problem. Its Pareto front satisfies the following relations: $f_1 + f_2 + f_3 = 2$ and $0 \leq f_i \leq 1$ ($i = 1, 2, 3$). In Fig. 20 (a), 66 solutions are uniformly generated. In Fig. 20 (b), 105 solutions are first uniformly generated. Next, 39 boundary solutions are removed. As a result, Fig. 20 (b) includes 66 inside solutions. In Fig. 20 (c), 36 solutions are first uniformly generated. Next, 21 boundary solutions are removed from the generated solutions to use the 15 inside solutions. Then, 51 uniformly distributed boundary solutions are added to the 15 inside solutions. In Fig. 20 (d), 66 boundary solutions are uniformly generated.

Intuitively, it is clear that Fig. 20 (a) is the best in Fig. 20. However, when the reference point is the same as or very close to the nadir point (1, 1, 1), Fig. 20 (b) is evaluated as the best.

By increasing the distance from the nadir point to the reference point (e.g., $\boldsymbol{r}$ = (10, 10, 10)), the solution set in Fig. 20 (d) can have the best hypervolume value. For further discussions, the four solution sets in Fig. 20 are compared using 1001 reference point specifications: $\boldsymbol{r} = (r, r, r)$ = (1, 1, 1), (1.01, 1.01, 1.01), (1.02, 1.02, 1.02), ..., (10, 10, 10). Table 1 summarizes comparison results. In Table 1, different solution sets are evaluated as the best for different reference points. The solution set in Fig. 20 (a) has the best hypervolume value only when $1.06 \leq r \leq 1.29$.



(a) Well-distributed solutions. (b) Only inside solutions.

(c) Mainly boundary solutions. (d) Only boundary solutions.

Fig. 20. Four solution sets with 66 solutions on an inverted triangular linear Pareto front.

Table 1. The best solution set among Fig. 20 (a)-(d) for each reference point specification: $\boldsymbol{r} = (r, r, r)$.

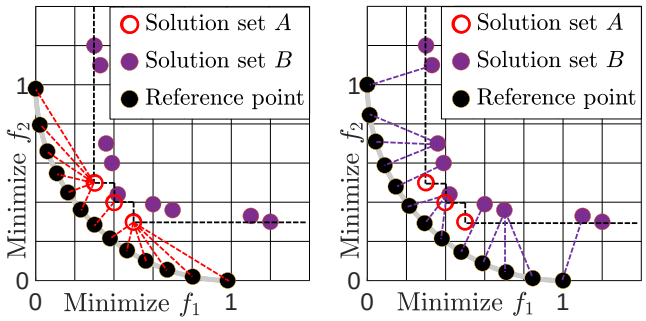
Reference point: r	[1, 1.05]	[1.06, 1.29]	[1.30, 6.08]	[6.09, 10]
Best solution set	(b)	(a)	(c)	(d)

Figure 20 and Table 1 clearly demonstrate dominant effects of the reference point on hypervolume-based comparison results. However, its importance has not always been well recognized [55] since its effect is small for test problems with triangular Pareto fronts (e.g., almost all DTLZ and WFG test problems). If the Pareto front in Fig. 20 is rotated (i.e., if the triangular linear Pareto front is used), Fig. 20 (a) is the best for all specifications of the reference point in Table 1.

In general, different comparison results are obtained from different reference point specifications. Thus, our suggestion is to use multiple specifications (e.g., r = 1.2, 2.0). If the same comparison results are obtained from different specifications, the results seem to be reliable. If different comparison results are obtained, it is advisable to use an additional indicator.

B. Difficulties in the Use of IGD

The IGD indicator [45] is also frequently used for performance comparison. One difficulty is that IGD is not Pareto compliant. Another difficulty is that calculated IGD values are strongly influenced by the reference point set specification. These two difficulties are explained in Fig. 21 where two solution sets A and B are compared. In Fig. 21 (a), 13 reference points, which maximize the hypervolume value, are generated in the same manner as in Fig. 16 and Fig. 17. That is, their distribution is hypervolume optimal. In Fig. 21 (b), they are generated uniformly. Under these two specifications of the reference point set, different solution sets have better IGD values in Fig. 21: A is better in Fig. 21 (a) whereas B is better in Fig. 21 (b). Figure 21 shows that different reference point sets for IGD calculation lead to different IGD-based comparison results. This issue is further discussed later.



(a) HV-optimal reference points. IGD(A): 0.2841, IGD(B): 0.2886.

(b) Uniform reference points. IGD(A): 0.3032, IGD(B): 0.2958.

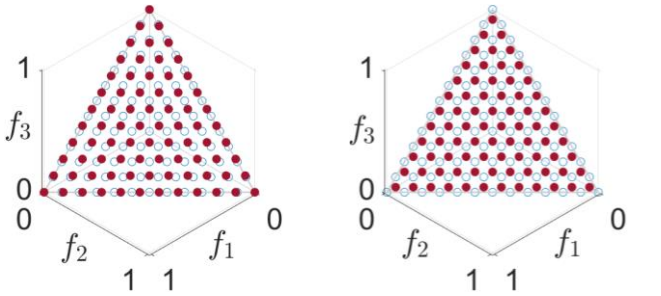
Fig. 21. Illustration of IGD-based performance comparison. In (a), solution set A has a better (i.e., smaller) IGD value. However, in (b), B has a better IGD value. Pareto dominance-based comparison shows that A is better than B.

Between the two solution sets A and B in Fig. 21, every solution in B is dominated by at least one solution in A. Thus, the "better" relation [16] holds: A is better than B. However, in Fig. 21 (b), B has a smaller IGD value than A. This observation shows that IGD is not a Pareto compliant indicator. It should be noted that solution set A always has a better hypervolume value than B independent of the reference point specification (as long as it is not too close to the Pareto front). A modified indicator IGD$^+$ [46] was proposed, which is weakly Pareto compliant (i.e., which never gives a worse evaluation to a better solution set). When A is better than B as in Fig. 21, a weakly Pareto compliant indicator can give the same evaluation to the two solution sets (whereas it never gives a worse evaluation to the better solution set A). This is the difference between "weakly Pareto compliant" and "Pareto compliant". A Pareto compliant indicator such as hypervolume always gives a strictly higher evaluation to a better solution set.

When two solution sets are well converged to the Pareto front, the Pareto dominance-based "better" relation is not likely to hold between them (i.e., either solution set is not likely to be better than the other). For two solution sets of a many-objective problem, the better relation is not likely to hold between them. Thus, the Pareto incompliance property of IGD is not always a serious problem.

The main difficulty in the IGD indicator is how to specify the reference point set since it has large effects on performance comparison results. Figure 22 shows three solution sets with 78

solutions in (a)-(c) and a reference point set with 91 points in (d) on a triangular linear Pareto front of a three-objective problem (e.g., the normalized DTLZ1 problem). Figures 22 (a) and 22 (d) are generated using Das & Dennis method [44] where the integer parameter H is specified as H = 11 and H = 12, respectively. In Das & Dennis method, all sides of the triangle are evenly divided into H intervals. Solution set C in Fig. 22 (c) is generated from Fig. 22 (d) by removing the bottom 13 solutions. Solution set B in Fig. 22 (b) is generated by moving down each solution in Fig. 22 (c) to the center of the neighboring three reference points. The IGD value of each solution set is shown in parentheses in Fig. 22 (a)-(c).



(a) Solution set A (IGD: 0.0421). (b) Solution set B (IGD: 0.0680).

(c) Solution set C (IGD: 0.0168). (d) Reference point set (H = 12).

Fig. 22. Three solution sets with 78 solutions and 91 reference points. The IGD value of each solution set is shown in parentheses.

Intuitively, Fig. 22 (a) looks the best since solutions are uniformly distributed over the entire Pareto front. Solution set C in Fig. 22 (c) looks the worst since solutions are biased. However, in Fig. 22, solution set C has the best (i.e., smallest) IGD value. This is because each solution in C overlaps with the corresponding reference point in Fig. 22 (d). This example demonstrates that comparison results based on IGD values can be counter-intuitive.

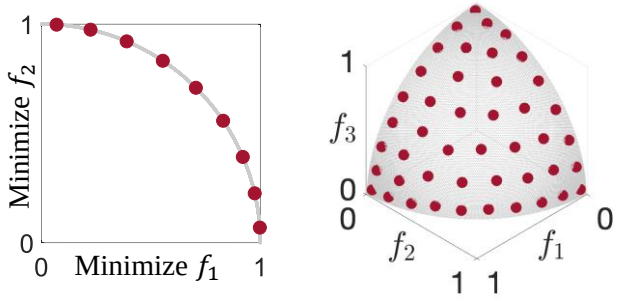
For further discussion on this issue, the three solution sets are compared by generating various reference point sets by Das & Dennis method using different values of H (e.g., H = 12 in Fig. 22 (d)). Table 2 summarizes experimental results.

When H = 11 in the third row of Table 2, the generated reference point set and solution set A are exactly the same. Thus the IGD value of A is zero. In this case, A is clearly the best and C is clearly the worst (based on the calculated IGD values), which is consistent with our intuition. However, performance comparison results in Table 2 are different depending on the reference point set specification. In particular, when the number of reference points is small (e.g., 66, 78, 91 in Table 2), their specification has a large effect on IGD-based comparison results. When a large number of reference points are used in Table 2 (e.g., 20100 reference points), solution set B has the best (i.e., smallest) IGD value. Moreover, in this case, the IGD values of A and C are almost the same (e.g., 0.0449 and 0.0451 when 20100 reference points are used). These results are not consistent with our intuitive evaluation of the three solution sets in Fig. 22 (a)-(c).

Table 2. IGD values of the three solution sets in Fig. 22 calculated for different reference point sets. The worst and best results for each reference point set are highlighted in blue and red fonts, respectively.

Value of H	Number of reference points	Solution set A	Solution set B	Solution set C
10	66	0.0415	0.0401	0.0519
11	78	0.0000	0.0354	0.0509
12	91	0.0421	0.0680	0.0168
13	105	0.0423	0.0507	0.0496
14	120	0.0425	0.0467	0.0491
15	136	0.0427	0.0453	0.0488
16	153	0.0428	0.0450	0.0481
17	171	0.0429	0.0436	0.0478
18	190	0.0431	0.0436	0.0474
19	210	0.0432	0.0428	0.0472
20	231	0.0433	0.0428	0.0472
50	1275	0.0444	0.0412	0.0457
100	5050	0.0447	0.0408	0.0453
200	20100	0.0449	0.0406	0.0451

One characteristic feature of the IGD indicator is that uniformly distributed solution sets are always highly evaluated, which is independent of the Pareto front shape. Using the IGD indicator in the SMS-EMOA algorithm, near-optimal solution sets of DTLZ2 with two and three objectives can be found as in Fig. 17 and Fig. 18. In computational experiments on each test problem, 10,000 solutions on the entire Pareto front are uniformly generated as reference points. The obtained IGD optimal distributions are shown in Fig. 23 where solutions are uniformly distributed. This means that uniform solution sets are highly evaluated by IGD. However, the IGD optimal distributions include no boundary solutions (i.e., no extreme solutions in Fig. 23 (a), no boundary solutions in Fig. 23 (b)).



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

Fig. 23. IGD optimal distributions of (a) nine solutions and (b) 45 solutions on nonlinear Pareto fronts. Uniformly distributed 10,000 reference points on the entire Pareto front are used for IGD calculation.

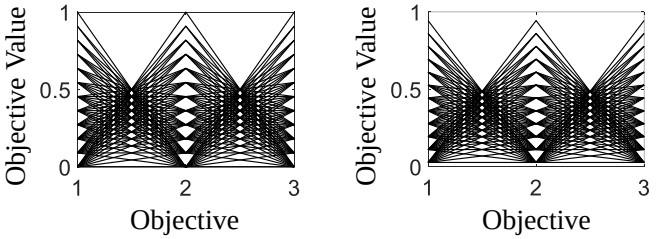
As in computational experiments in Fig. 23, a large number of uniformly distributed reference points are usually used for IGD calculation. Let us consider the normalized two-objective

linear Pareto front again, which is a straight line between (1, 0) and (0, 1). For this Pareto front, the optimal distribution of μ solutions for IGD minimization is the uniform distribution on the line segment between $(1/2\mu, 1 - 1/2\mu)$ and $(1 - 1/2\mu, 1/2\mu)$ when an infinitely large number of uniform reference points are used for IGD calculation [56]. That is, the IGD optimal distribution does not include the two extreme points (1, 0) and (0, 1). Independent of the curvature property (i.e., linear, convex, concave) of the Pareto front, the IGD optimal distribution does not include the two extreme points. In the case of three or more objectives, the IGD optimal distribution does not include boundary solutions when an infinitely large number of uniform reference points are used. This is the reason why solution set *B* with no boundary solutions is the best in Table 2 for a large number of reference points.

As shown by the experimental results in Fig. 22 and Table 2, the reference point specification has large effects on IGD-based comparison results especially when a small number of reference points are used. Our suggestion is to use many reference points to avoid unexpected biases caused by the overlapping between reference points and solutions (e.g., lucky matching of obtained solutions with the given reference points in Fig. 22 (c), unlucky discordance between them in Fig. 22 (b)).

Since the hypervolume and IGD indicators have clearly different optimal distributions of solutions, it is likely that different comparison results will be obtained from them. Thus, it is advisable to use both of them in performance comparison. When the use of the hypervolume indicator is unrealistic for many-objective problems, the IGD^+ indicator can be used since the hypervolume and IGD^+ indicators have similar optimal distributions [57], [58].

Since uniformly distributed and fully spread solution sets over the entire Pareto front (e.g., Fig. 22 (a)) often have worse IGD values than inside solution sets (e.g., Fig. 22 (b)), it is a good idea to visually examine each solution set using a parallel coordinate graph [59]. The spread of solutions can be visually examined for each objective even for many-objective problems. The two solution sets in Fig. 22 (a) and Fig. 22 (b) are shown using parallel coordinate graphs in Fig. 24 (a) and Fig. 24 (b), respectively. It is clear in Fig. 24 that Fig. 24 (b) has a smaller spread along each axis than Fig. 24 (a).

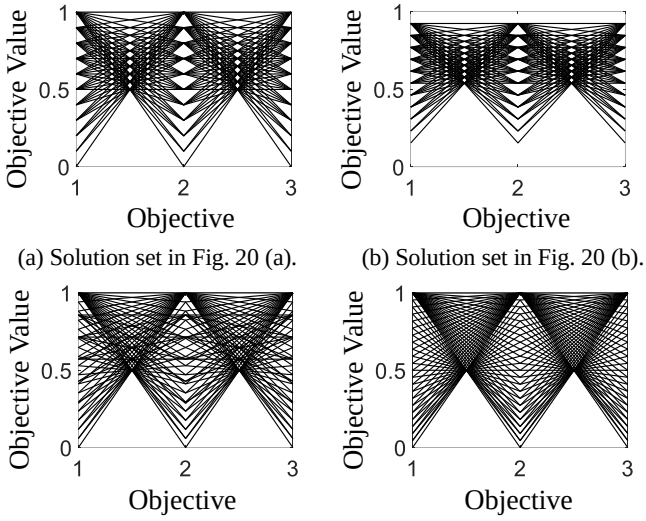


(a) Solution set in Fig. 22 (a). (b) Solution set in Fig. 22 (b).

Fig. 24. Parallel coordinate graphs of the two solution sets in Fig. 22.

One difficulty of parallel coordinate graphs is that they are often misleading about the uniformity of solutions. To clearly demonstrate this issue, the four solution sets in Fig. 20 are compared using parallel coordinate graphs in Fig. 25.

In Fig. 20, solutions in Fig. 20 (a) are well distributed while all solutions in Fig. 20 (d) are boundary solutions. However, Fig. 25 gives the following impression: Fig. 25 (d) has better uniformity than Fig. 25 (a)-(c). As explained in Table 1 in Section IV, the boundary solution set in Fig. 25 (d), i.e., in Fig. 20 (d), has the best hypervolume value if the reference point is far away from the Pareto front. Thus, if the parallel coordinate graph and the hypervolume indicator are used for comparing the four solution sets in Fig. 25, it is possible that Fig. 25 (d) is evaluated as the best solution set. However, the IGD value of Fig. 25 (d) is the worst among the four solution sets when a large number of uniform reference points are used for IGD calculation (since all solutions in Fig. 25 (d) are on the boundary of the Pareto front as shown in Fig. 20 (d)). These discussions support the necessity of using multiple indicators.



(a) Solution set in Fig. 20 (a). (b) Solution set in Fig. 20 (b).

(c) Solution set in Fig. 20 (c). (d) Solution set in Fig. 20 (d).

Fig. 25. Parallel coordinate graphs of the four solution sets in Fig. 20.

In this section, the evaluation of solution sets has been discussed using performance indicators by implicitly assuming that no preference information is available from the decision maker. However, real-world application problems usually have a decision maker. It is likely that the decision maker has some explicit or implicit preference about solution sets. In this situation, the choice of a performance indicator should be consistent with the preference of the decision maker. Recently, the choice of a performance indicator was discussed based on the decision maker's preference for multi-objective search-based software engineering (SBSE) problems in [60]. Practical guidelines on the choice of an indicator were shown for various types of preference information together with a review on the use of performance indicators for multi-objective SBSE problems.

V. Choice of Test Problems

As shown in Figs. 2-5 in Section II, the choice of test problems has large effects on performance comparison results. Recently, DTLZ [17] and WFG [18] have been frequently used as scalable problems for many-objective optimization. In [23], many-objective algorithms (NSGA-III, MOEA/DD, θ-DEA) were compared with classic algorithms (NSGA-II and four implementations of MOEA/D) using various test problems (i.e., DTLZ1-4, WFG1-9, Minus-DTLZ1-4 and Minus-WFG1-9)

with 3, 5, 8 and 10 objectives. In Table 3 and Table 4, the experimental results in [23] are summarized for the four groups of test problems. For each group of test problems, the percentage of test problems on which each algorithm shows the best results is calculated. For example, DTLZ1-4 includes 16 problems (4 problems with 3, 5, 8 and 10 objectives). When the hypervolume indicator was used to evaluate each algorithm, MOEA/DD showed the best performance on 12 out of the 16 DTLZ1-4 test problems in [23]. That is, MOEA/DD showed the best performance on 75% test problems. Thus, "75.00" is given in the corresponding cell in Table 3. In this manner, Table 3 and Table 4 are created from the reported results in [23] with respect to the hypervolume and IGD indicators, respectively.

Table 3. Percentage of test problems on which each algorithm has the best average hypervolume value among the eight algorithms for the reference point (1.1, ..., 1.1). The percentage is calculated for each of the four test problem suites (reproduced from Table III and Table IV of [23]). The highest percentage is highlighted in red font.

Algorithm	DTLZ1-4	WFG1-9	Minus-DTLZ1-4	Minus-WFG1-9
NSGA-III	6.25	8.33	**31.25**	**38.89**
θ-DEA	18.75	**66.67**	12.50	11.11
MOEA/DD	**75.00**	0.00	0.00	0.00
MOEA/D-PBI	0.00	0.00	18.75	0.00
MOEA/D-Tch	0.00	13.89	0.00	8.33
MOEA/D-WS	0.00	0.00	18.75	0.00
MOEA/D-IPBI	0.00	0.00	0.00	11.11
NSGA-II	0.00	11.11	18.75	30.56

Table 4. Percentage of test problems on which each algorithm has the best average IGD value among the eight algorithms in each of the four test problem suites (reproduced from Table VI and Table VII of [23]). The highest percentage is highlighted in red font.

Algorithm	DTLZ1-4	WFG1-9	Minus-DTLZ1-4	Minus-WFG1-9
NSGA-III	6.25	16.67	12.50	5.56
θ-DEA	18.75	**44.44**	0.00	0.00
MOEA/DD	**62.50**	11.11	0.00	0.00
MOEA/D-PBI	12.50	5.56	0.00	0.00
MOEA/D-Tch	0.00	5.56	0.00	8.33
MOEA/D-WS	0.00	0.00	6.25	0.00
MOEA/D-IPBI	0.00	0.00	12.50	16.67
NSGA-II	0.00	16.67	**68.75**	**69.44**

In Table 3 and Table 4, MOEA/DD is the best for DTLZ1-4, and θ-DEA is the best for WFG1-9. However, NSGA-II and NSGA-III show the best performance for their minus versions. These tables show that totally different comparison results can be obtained from different test problems. These tables also show that different results can be obtained from different indicators.

When the 10-objective HTNY19 test problem is used, none of the above-mentioned four algorithms (MOEA/DD, θ-DEA, NSGA-II and NSGA-III) works well. Their convergence to the Pareto front is very slow, which is similar to MOEA/D-PBI and MOEA/D-Tch in Fig. 4 in Section II. In [21], much better results were obtained for 10-objective HTNY19 by a modified NSGA-II algorithm, MOEA/D-WS, and MOEA/D-STM [61] than the above-mentioned four algorithms. These results show large effects of test problems on comparison results.

For fair comparison, it is advisable to use various test problems. Ideally, they have various characteristic features with respect to the following aspects: the number of objectives, the number of decision variables, the Pareto front curvature (linear, convex, concave), the Pareto front shape (triangular, inverted triangular, degenerate, disconnected), the difficulty of convergence, the difficulty of diversification, and the necessity of normalization. Since most test problems in DTLZ and WFG have similar and somewhat unrealistic properties [62], there is a need to create new test suites with various characteristic features (e.g., [63]). It is especially important to create a new test suite including various realistic test problems (e.g., [64]).

It is not likely that a single EMO algorithm works well on all test problems with different characteristics. Thus, it is an interesting research topic to create a recommendation system which suggests an appropriate EMO algorithm for a given problem based on its characteristics. Automatic design is also an interesting research direction to create a new algorithm using the characteristics of the given problem. There exist many other interesting and important research directions such as the design of a general-purpose robust EMO algorithm which works well on various test problems, and the design of a specialized EMO algorithm which works well on a special application task.

VI. Conclusions

This paper has explained various difficulties in fair performance comparison of EMO algorithms, which are related to the termination condition, the population size, performance indicators, and test problems. Those difficulties are not clear for the frequently-used DTLZ and WFG test suites. However, as pointed out in this paper, they become clear when test problems with different characteristics (e.g., different shape of Pareto fronts, different difficulty of convergence) are used for performance comparison. From the discussions in this paper, our suggestions are as follows:

(1) Evaluation of each algorithm as an anytime algorithm under multiple termination conditions.
(2) Evaluation of each algorithm under different population size specifications.
(3) Evaluation of each algorithm using selected subsets from all the evaluated solutions.
(4) Use of multiple reference point specifications in the hypervolume indicator.
(5) Use of a large number of uniformly distributed reference points over the entire Pareto front in the IGD indicator.
(6) Use of the IGD^+ indicator in the same manner as in (5) when the use of the hypervolume indicator is not realistic for many-objective problems.
(7) Visual examination of each solution set in addition to its evaluation using multiple performance indicators.
(8) Use a number of test problems with various characteristics including realistic test problems.

As demonstrated in this paper, different comparison results will be obtained from these suggestions (e.g., from the use of multiple performance indicators and different test problems). Such comparison results will suggest interesting research

directions for future studies on new algorithm development, indicator analysis, and new test problem design. All of them will support the future growth of the EMO research field. Of course, there exist a number of other important research topics related to performance comparison of EMO algorithms, which have not been discussed in this paper. The following are some examples: the handling of highly multi-modal problems where totally different solution sets can be obtained from different runs due to stochastic nature of evolutionary algorithms, the handling of multi-modal multi-objective optimization where the distributions of solutions in both the objective and decision spaces are important, parameter specifications in each EMO algorithm for each test problem, the design of robust performance indicators which are not sensitive to related specifications, the evaluation of the robustness of solution sets, and the utilization of preference information in solution set evaluation.

Acknowledgment

This work was supported by National Natural Science Foundation of China (Grant No. 61876075, 62002152), Guangdong Provincial Key Laboratory (Grant No. 2020B121201001), the Program for Guangdong Introducing Innovative and Enterpreneurial Teams (Grant No. 2017ZT07X386), The Stable Support Plan Program of Shenzhen Natural Science Fund (Grant No. 20200925174447003), Shenzhen Science and Technology Program (Grant No. KQTD2016112514355531).

References

[1] 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.

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

[3] 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.

[4] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653-1669, September 2007.

[5] 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.

[6] R. Wang, R. Purshouse, and P. Fleming, "Preference-inspired co-evolutionary algorithms for many-objective optimisation," *IEEE Trans. on Evolutionary Computation*, vol. 17, no. 4, pp. 474-491, August 2013.

[7] 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.

[8] 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.

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

[10] H. Wang, L. Jiao, and X. Yao, "Two_Arch2: An improved two-archive algorithm for many-objective optimization," *IEEE Trans. on Evolutionary Computation*, vol. 19, no. 4, pp. 524-541, August 2015.

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

[12] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. on Evolutionary Computation*, vol. 20, no. 5, pp. 773-791, October 2016.

[13] S. Jiang and S. Yang, "A strength Pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization," *IEEE Trans. on Evolutionary Computation*, vol. 21, no. 3, pp. 329-346, June 2017.

[14] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization," *IEEE Trans. on Evolutionary Computation*, vol. 22, no. 1, pp. 97-112, February 2018.

[15] Y. Tian, R. Cheng, X. Zhang, F. Cheng, Y. Jin, "An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility," *IEEE Trans. on Evolutionary Computation*, vol. 22, no. 4, pp. 609-622, August 2018.

[16] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. on Evolutionary Computation*, vol. 7, no. 2, pp. 117-132, April 2003.

[17] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in A. Abraham, L. Jain, and R. Goldberg (Eds.), *Evolutionary Multiobjective Optimization*, pp. 105-145, Springer-Verlag, London, 2005.

[18] 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, October 2006.

[19] R. Wang, Z. Zhou, H. Ishibuchi, T. Liao, and T. Zhang, "Localized weighted sum method for many-objective optimization," *IEEE Trans. on Evolutionary Computation*, vol. 22, no. 1, pp. 3-18, February 2018.

[20] H. Ishibuchi, T. Matsumoto, N. Masuyama, and Y. Nojima, "Effects of dominance resistant solutions on the performance of evolutionary multi-objective and many-objective algorithms," *Proc. of 2020 Genetic and Evolutionary Computation Conference*, pp. 507-515, July 08-12, 2020.

[21] L. M. Pang, H. Ishibuchi, and K. Shang, "NSGA-II with simple modification works well on a wide variety of many-objective problems," *IEEE Access*, vol. 8, pp. 190240-190250, October 2020.

[22] H. Ishibuchi, H. Masuda, and Y. Nojima, "Pareto fronts of many-objective degenerate test problems," *IEEE Trans. on Evolutionary Computation*, vol. 20, no. 5, pp. 807-813, October 2016.

[23] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes," *IEEE Trans. on Evolutionary Computation*, vol. 21, no.2, pp. 169-190, April 2017.

[24] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms – A comparative case study," *Proc. of 5th International Conference on Parallel Problem Solving from Nature*, pp. 292-301, September 27-30, 1998.

[25] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "A PlatEMO: A MATLAB platform for evolutionary multi-objective optimization," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73-87, November 2017.

[26] D. A. Van Veldhuizen D. A., Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations. Ph.D. Thesis, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, USA (1999).

[27] S. Zilberstein, "Using anytime algorithms in intelligent systems," *AI Magazine*, vol. 17, no. 3, pp. 73-83, Fall 1996.

[28] R. Tanabe, H. Ishibuchi, and A. Oyama, "Benchmarking multi- and many-objective evolutionary algorithms under two optimization scenarios," *IEEE Access*, vol. 5, pp. 19597-19619, December 2017.

[29] M. Li and X. Yao, "An empirical investigation of the optimality and monotonicity properties of multiobjective archiving methods," *Proc. of 10th International Conference on Evolutionary Multi-Criterion Optimization*, pp. 15-26, March 10-13, 2019.

[30] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "How to compare many-objective algorithms under different settings of population and archive sizes," *Proc. of 2016 IEEE Congress on Evolutionary Computation*, pp. 1149-1156, July 24-29, 2016.

[31] A. P. Guerreiro, C. M. Fonseca, and L. Paquete, "Greedy hypervolume subset selection in low dimensions," *Evolutionary Computation*, vol. 24, no. 3, pp. 521-544, September 2016.

[32] W. Chen, H. Ishibuchi, and K. Shang, "Lazy greedy hypervolume-based subset selection from large candidate solution sets," *Proc. of 2020 IEEE Congress on Evolutionary Computation*, pp. 1-8, July 19-24, 2020.

[33] J. E. Fieldsend, R. M. Everson, and S. Singh, "Using unconstrained elite archive for multiobjective optimization," *IEEE Trans. on Evolutionary Computation*, vol. 7, no. 3, pp. 305-323, June 2003.

[34] K. Bringmann, T. Friedrich, and P. Klitzke, "Generic postprocessing via subset selection for hypervolume and epsilon-indicator," *Proc. of 13th International Conference on Parallel Problem Solving from Nature*, pp. 518-527, September 13-17, 2014.

[35] L. C. T. Bezerra, M. López-Ibáñez, and T. Stützle, "Archiver effects on the performance of state-of-the-art multi- and many-objective evolutionary algorithms," *Proc. of 2019 Genetic and Evolutionary Computation Conference*, pp. 620-628, July 13-17, 2019.

[36] H. Ishibuchi, L. M. Pang, and K. Shang, "Population size specification for fair comparison of multi-objective evolutionary algorithms," *Proc. of 2020 IEEE International Conference on Systems, Man and Cybernetics*, pp. 1095-1102, October 11-14, 2020.

[37] Y. Nan, K. Shang, H. Ishibuchi, and L. He, "Reverse strategy for non-dominated archiving," *IEEE Access*, vol. 8, pp. 119458-119469, July 2020.

[38] T. Kuhn, C. M. Fonseca, L. Paquete, S. Ruzika, M. M. Duarte, J. R. Figueira, "Hypervolume subset selection in two dimensions: Formulations and algorithms," *Evolutionary Computation*, vol. 24, no. 3, pp. 411-425, September 2016.

[39] K. Bringmann and T. Friedrich, "An efficient algorithm for computing hypervolume contributions," *Evolutionary Computation*, vol. 18, no. 3, pp. 383-402, September 2010.

[40] K. Bringmann, T. Friedrich, and P. Klitzke, "Two-dimensional subset selection for hypervolume and epsilon-indicator," *Prof. of 2014 Genetic and Evolutionary Computation Conference*, pp. 589-596, July 12-16, 2014.

[41] R. J. Gomes, A. P. Guerreiro, T. Kuhn, and L. Paquete, "Implicit enumeration strategies for the hypervolume subset selection problem," *Computers and Operations Research*, vol. 100, pp. 244-253, December 2018.

[42] B. Groz and S. Maniu, "Hypervolume subset selection with small subsets," *Evolutionary Computation*, vol. 27, no. 4, pp. 611-637, December 2019.

[43] H. K. Singh, K. S. Bhattacharjee, and T. Ray, "Distance-based subset selection for benchmarking in evolutionary multi/many-objective optimization," *IEEE Trans. on Evolutionary Computation*, vol. 23, no. 5, pp. 904-912, October 2019.

[44] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 631–657, August 1998.

[45] M. R. Sierra and C. A. C. Coello, "A new multi-objective particle swarm optimizer with improved selection and diversity mechanisms," *Technical Report. CINVESTAV-IPN*, 2004.

[46] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Modified distance calculation in generational distance and inverted generational distance," *Proc. of 8th International Conference on Evolutionary Multi-Criterion Optimization*, pp. 110-125, March 29-April 1, 2015.

[47] H. Ishibuchi, L. M. Pang, and K. Shang, "A new framework of evolutionary multi-objective algorithms with an unbounded external archive," *Proc. of 24th European Conference on Artificial Intelligence*, pp. 283-290, August 29 - September 02, 2020.

[48] H. Ishibuchi, L. M. Pang, and K. Shang, "Evolutionary multi-objective optimization algorithm framework with three solution sets," *arXiv*: https://arxiv.org/abs/2012.07319, December 2020.

[49] L. M. Pang, H. Ishibuchi, and K. Shang, "Decomposition-based multi-objective evolutionary algorithm design under two algorithm frameworks," *IEEE Access*, vol. 8, pp. 163197-163208, September 2020.

[50] H. Ishibuchi, L. M. Pang, and K. Shang, "Solution subset selection for final decision making in evolutionary multi-objective optimization," *arXiv*: https://arxiv.org/abs/2006.08156, June 2020.

[51] E. Zitzler, D. Brockhoff, and L. Thiele, "The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration," *Proc. of 4th International Conference on Evolutionary Multi-Criterion Optimization*, pp. 862-876, March 5-8, 2007.

[52] M. Emmerich, A. Deutz, and N. Beume, "Gradient-based/evolutionary relay hybrid for computing Pareto front approximations maximizing the S-metric," *Proc. of 4th International Workshop on Hybrid Metaheuristics*, pp. 140-156, October 8-9, 2007.

[53] D. Brockhoff, "Optimal μ-distributions for the hypervolume indicator for problems with linear bi-objective fronts: Exact and exhaustive results," *Proc. of 8th International Conference on Simulated Evolution and Learning*, pp. 24-34, December 1-4, 2010.

[54] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, "Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications," *Theoretical Computer Science*, vol. 425, pp. 75-103, March 2012.

[55] H. Ishibuchi, R. Imada, Y. Setoguchi, and Y. Nojima, "How to specify a reference point in hypervolume calculation for fair performance comparison," *Evolutionary Computation*, vol. 26, no. 3, pp. 411-440, September 2018.

[56] H. Ishibuchi, R. Imada, Y. Setoguchi, and Y. Nojima, "Reference point specification in inverted generational distance for triangular linear Pareto front," *IEEE Trans. on Evolutionary Computation*, vol. 22, no. 6, pp. 961-975, December 2018.

[57] H. Ishibuchi, R. Imada, N. Masuyama, and Y. Nojima, "Comparison of hypervolume, IGD and IGD$^+$ from the viewpoint of optimal distributions of solutions," *Proc. of 10th International Conference on Evolutionary Multi-Criterion Optimization*, pp. 332-345, March 10-13, 2019.

[58] R. Tanabe and H. Ishibuchi, "An analysis of quality indicators using approximated optimal distributions in a 3-D objective space," *IEEE Trans. on Evolutionary Computation*, vol. 24, no. 5, pp. 853-867, October 2020.

[59] M. Li, L. Zhen, and X. Yao, "How to read many-objective solution sets in parallel coordinates," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, 88-97, November 2017.

[60] M. Li, T. Chen, and X. Yao, "How to evaluate solutions in Pareto-based search-based software engineering? A critical review and methodological guidance," *IEEE Trans. on Software Engineering*, 2020 (Early Access Paper).

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

[62] H. Ishibuchi, L. He, and K. Shang, "Regular Pareto front shape is not realistic," *Proc. of 2019 IEEE Congress on Evolutionary Computation*, pp. 2035-2042, June 10-13, 2019.

[63] R. Cheng, M. Li, Y. Tian, X. Zhang, S. Yang, Y. Jin, and X. Yao, "A benchmark test suite for evolutionary many-objective optimization," *Complex & Intelligent Systems*, vol. 3, no. 1, pp. 67-81, March 2017.

[64] R. Tanabe and H. Ishibuchi, "An easy-to-use real-world multi-objective optimization problem suite," *Applied Soft Computing*, vol. 89, Article 106078, April 2020.