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 (Accepted)*.

How to Compare Many-Objective Algorithms under Different Settings of Population and Archive Sizes

Hisao Ishibuchi, Yu Setoguchi, 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@, yu.setoguchi@ci., hiroyuki.masuda@ci., nojima@}cs.osakafu-u.ac.jp

***Abstract*—In the evolutionary multi-objective optimization community, algorithm comparison is usually performed under the same population size. However, this is not always fair because its best specification is usually different in each algorithm. In many-objective optimization, the number of solutions to be found may depend on the situation. If the decision maker wants to analyze the entire Pareto front, thousands of solutions may be needed. If the decision maker wants to choose a single final solution from some candidates after their quick checks, only a small number of representative solutions may be needed. In this paper, we discuss how to evaluate the ability of evolutionary many-objective optimization algorithms to find an arbitrarily specified number of non-dominated solutions. Our idea is the use of solution selection after the termination of each algorithm. We examine two scenarios: One is solution selection from the final population, and the other is from all of the examined solutions. Through computational experiments, first we demonstrate that performance comparison heavily depends on the population size. Then we examine the effects of solution selection from the final population and the examined solutions on comparison results.**

Keywords*—*Evolutionary many-objective optimization, solution selection, solution subset selection, performance comparison, evolutionary multi-objective optimization.

I. Introduction

In the last decade, many-objective optimization has been a very active research area in evolutionary computation [1]-[3]. A number of many-objective algorithms have been proposed in the literature [4]-[10]. New algorithms are usually compared with existing ones through computational experiments on well-known many-objective test problems such as DTLZ [11] and WFG [12]. However, fair performance comparison of many-objective algorithms is very difficult for the following reasons:

(i) Choice of Test Problems:

In general, there is no absolutely best algorithm for all problems [13]. Thus it is always the case that performance comparison depends on the choice of test problems. If we use very similar test problems, comparison results are likely to be biased by their common features. In this situation, not only performance comparison but also algorithm development may be heavily biased by the common features of the test problems. Actually, the design of test problems and the development of evolutionary multi-objective optimization (EMO) algorithms in the last two decades can be viewed as a kind of co-evolutionary progress [14]. Since DTLZ [11] and WFG [12] test problems were designed using a similar mechanism [15], it is very likely that some recently-proposed high-performance many-objective algorithms are over-fitting to the test problems.

(ii) Choice of Performance Indicators:

The hypervolume (HV) indicator [16] has been frequently used for comparing EMO algorithms. However, its exact calculation for many-objective problems is not easy due to its heavy computation load especially for a large solution set with more than ten objectives. While some fast calculation methods were proposed [17]-[19], approximate calculation [5] is usually used for test problems with more than ten objectives. The inverted generational distance (IGD) indicator [20], [21] has also been frequently used for many-objective problems (e.g., [6]-[10]) thanks to its simple calculation. The IGD indicator can be easily calculated even for many-objective problems. However, the IGD indicator is not Pareto compliant. That is, IGD-based comparison results are not always consistent with Pareto dominance [22]. Unfortunately, there is no other Pareto compliant unary performance indicator than the HV [23].

(iii) Setting of Reference Points in Performance Indicators:

The HV and IGD calculations need a reference point and a reference point set, respectively. That is, HV-based and IGD-based comparison results depend on their specifications. It was reported that totally different comparison results were obtained from their different settings [24], [25]. In the case of a multi-objective problem with two or three objectives, we can visually examine their locations in the low-dimensional objective space. Such a visual examination helps us to appropriately specify them. However, it is not easy to visually examine their locations in the high-dimensional objective space of a many-objective problem with four or more objectives.

(iv) Specification of the Population Size:

In the EMO community, the same population size is usually used for all algorithms in their performance comparison (even when a different population size is used for a different test problem). In many studies, it is specified in [100, 300]. This may be because 100-300 solutions are enough to approximate the Pareto front of a multi-objective problem with two or three objectives. It is also possible to show those solutions in its low-dimensional objective space. Many-objective algorithms are also often compared in a similar manner whereas thousands of solutions may be needed to approximate the Pareto front of a

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

many-objective problem. Moreover, it is difficult to show even 100 solutions in its high-dimensional objective space. These discussions imply that there is no reason for specifying the population size in [100, 300] for many-objective algorithms in their performance comparison. It was reported that different comparison results were obtained from different settings of the population size for many-objective knapsack problems [26].

(v) Handling of Algorithms with an External Population:

Some algorithms have an external (i.e., archive) population to store a number of non-dominated solutions. For example, all non-dominated solutions among the examined ones were to be stored as an external population in the original MOEA/D [27]. For fair comparison, only the internal population is usually used in performance evaluation of EMO algorithms. However, this is not necessarily fair because the external population can be much better than the internal population. If one algorithm with an external population is highly diversification-oriented and another with no external population is highly convergence-oriented, it may be unfair to use only the internal population for their performance comparison.

In this paper, we focus on the last two difficulties in the performance comparison of many-objective algorithms (i.e., population size specification and external population handling). This is because they have not been discussed in many studies. With respect to the other difficulties, we follow the frequently-used settings in recent studies on evolutionary many-objective optimization (while those settings still need further discussions as we have just explained). That is, many-objective algorithms are compared on the DTLZ and WFG test problems by the HV indicator, which is calculated in the normalized objective space.

Our idea for the handling of the last two difficulties is to select a pre-specified number of non-dominated solutions as a post processing procedure. When different algorithms with different settings of the population size are compared, solution selection is used to select the same number of non-dominated solutions from the final population of each algorithm. When different algorithms with/without an external population are compared, solution selection is used to select the same number of non-dominated solutions from all solutions examined during the execution of each algorithm.

This paper is organized as follows. In Section II, we briefly explain a heuristic solution selection method used in this paper as a post-processing procedure of many-objective algorithms. In Section III, we demonstrate that performance comparison results heavily depend on the specification of the population size through computational experiments on the DTLZ1-4 and WFG1-9 problems with three and five objectives. In Section IV and Section V, we report experimental results of solution selection from the final population and the examined solutions, respectively. In Section VI, we conclude this paper.

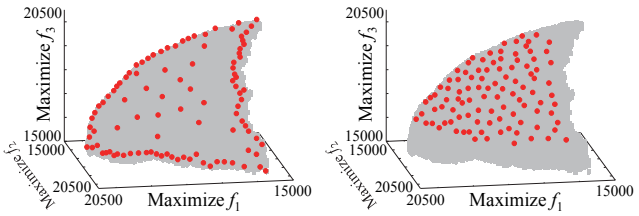
II. Solution Selection

Solution selection, which is often referred to as "subset selection" [28]-[30], is to select a pre-specified number of solutions from a given set of non-dominated solutions using a performance indicator. The HV indicator is usually used for solution selection [28]-[31]. In [28], [32], the ε-indicator was also used. Solution selection in these studies was discussed for multi-objective problems with two or three objectives.

Let us denote a given set of non-dominated solutions by S_{All}. Our solution selection problem is to select k solutions that maximize the HV indicator where k is the number of solutions to be finally presented to the decision maker:

$$\text{Maximize } HV(S) \text{ subject to } |S| = k \text{ and } S \subset S_{\mathrm{All}}. \qquad (1)$$

The optimal subset S^* of (1) depends on the setting of a reference point for the HV calculation. For example, when a reference point is far from the two-dimensional Pareto front of a three-objective problem, most solutions in S^* are along the boundary of the Pareto front as shown in Fig. 1 (a). However, when a reference point is close to the Pareto front, most solutions in S^* are inside the Pareto front as shown in Fig. 1 (b). In other words, we can control the distribution of solutions in S^* by the location of a reference point. In [33], [34], the HV maximization problem in (1) was generalized to multi-objective solution selection by using multiple reference points. In [35], [36], two-objective solution selection was formulated by using the number of solutions (i.e., k) as the second objective to be minimized together with the first objective $HV(S)$ in (1).



(a) A selected solution set. (b) Another selected solution set.

Fig. 1. Different results of solution selection for a three-objective 500-item knapsack problem. One hundred solutions are selected from a given set of non-dominated solutions using the HV indicator with a reference point (0, 0, 0) in (a) and a different reference point (16000, 16000, 16000) in (b).

In our former study [26], solution selection was used for performance comparison of different EMO algorithms with different settings of the population size. The same number of solutions were selected from the final population of each EMO algorithm. That is, solution selection was applied to the final population. In this paper, we examine two scenarios: One is solution selection from the final population (i.e., S_{All} is the set of all non-dominated solutions in the final population), and the other is from all of the examined solutions (i.e., S_{All} is the set of all non-dominated solutions among the examined solutions). The use of the second scenario is motivated by reported good results of solution selection from the examined solutions for two-objective problems in [28].

In our former study [26], we examined the following four solution selection methods from the final population for knapsack problems with four and six objectives: (i) Random selection, (ii) Greedy forward selection, (iii) Greedy backward selection, and (iv) Genetic optimization. When a large number of non-dominated solutions of a many-objective problem are included in S_{All}, it may be impractical to try to search for its true optimal subset S^* with k solutions in (1). So, we use the greedy forward selection method in this paper. This method starts with an empty subset S of S_{All}. The first solution $\boldsymbol{x}_{(1)}$ with

the largest HV value in S_{All} is found and moved from S_{All} to S. Then the next solution $\boldsymbol{x}_{(2)}$ in S_{All} is found to maximize the HV value of the subset $\{\boldsymbol{x}_{(1)}, \boldsymbol{x}_{(2)}\}$, i.e., to maximize $HV(S \cup \{\boldsymbol{x}_{(2)}\})$. The second solution $\boldsymbol{x}_{(2)}$ is moved from S_{All} to S. The third solution $\boldsymbol{x}_{(3)}$ is found to maximize $HV(S \cup \{\boldsymbol{x}_{(3)}\})$ and moved from S_{All} to S. In this manner, solution selection is performed one by one to find a subset S with k solutions.

III. Effects of the Population Size

In this section, we examine the effect of the population size on the performance of evolutionary many-objective algorithms through computational experiments on well-known frequently-used many-objective test problems: DTLZ1-4 and WFG1-9. We use their three-objective and five-objective instances.

We compare the following four algorithms: MOEA/D with the weighted Tchebycheff [27], NSGA-III [6], an evolutionary many-objective optimization algorithm based on dominance and decomposition (MOEA/DD [8]) and a simple but effective θ-dominance based evolutionary algorithm (θ-DEA [10]). The first two algorithms are representatives of well-known and frequently-used algorithms. The others are recently-proposed high-performance algorithms. All algorithms are based on a set of uniformly distributed weight vectors. NSGA-III and θ-DEA use a $(\mu+\mu)$ES-style generation update model while MOEA/D and MOEA/DD have a $(\mu+1)$ES-style model (i.e., steady-state generation update model).

In computational experiments, we use our implementation of MOEA/D. For the other algorithms, we use online available codes: θ-DEA and NSGA-III from [37] by the authors of the θ-DEA paper [10], and MOEA/DD from [38] by the authors of the MOEA/DD paper [8]. Our computational experiments are performed in a similar manner to those in recent studies on evolutionary many-objective optimization algorithms [6]-[10]. In the four algorithms, we use the same genetic operators with the same parameter specifications. As mutation, the polynomial mutation is used with the distribution index 20 and the mutation probability $1/n$ where n is the string length. As crossover, we use the SBX crossover with the distribution index 30 and the crossover probability 1.0.

We examine three settings of the population size in Table I: small, large and very large. The small setting of the population size is from [6]-[10]. The large setting (about 1000) and the very large setting (about 5000) are to examine the sensitivity of performance comparison results among the four algorithms to the population size. Since each algorithm has its own constraint on the population size specification, there are minor differences in the population size among the four algorithms in each of its three settings in Table I.

TABLE I. Three Settings of the Population Size: Small, Large and Very Large.

Population Size	Three-Objective (m = 3)		Five Objective (m = 5)	
	MOEA/D MOEA/DD	NSGA-III θ-DEA	MOEA/D MOEA/DD	NSGA-III θ-DEA
Small	91	92	210	212
Large	990	992	1001	1004
Very Large	5050	5052	5985	5988

The number of weight vectors in each algorithm for each test problem is the same as the population size of MOEA/D (and MOEA/DD) in Table I. In MOEA/D, a weight value 0 is replaced with 10^{-6}. The neighborhood size in MOEA/D is specified as 20, 200, 1000 for the small, large and very large settings of the population size, respectively. The total number of examined solutions is used as the termination condition of each algorithm: 50,000 for all three-objective problems and 200,000 for all five-objective problems. Similar termination conditions have been used in the literature [6]-[10]. Since we use the total number of examined solutions as the termination condition, the increase in the population size leads to the decrease in the total number of generations. For example, when the population size is 5050 in MOEA/D for a three-objective test problem, its execution is terminated in the middle of the 10th generation (whereas a small population of size 91 can be evolved by MOEA/D for 549 generations).

The number of decision variables is specified as $m + k - 1$ in DTLZ1-4 where m is the number of objectives and k is a parameter. The value of k is specified as $k = 5$ in DTLZ1 and $k = 10$ in DTLZ2-4. WFG1-9 have k position-related variables and l distance-related variables. They are specified as $k = m - 1$ and $l = 24 - (m - 1)$ for m-objective WFG1-9 problems.

We use the HV indicator for evaluating the performance of each algorithm. When the HV indicator is calculated, the m-dimensional objective space is normalized so that the ideal and nadir points are (0, 0, ..., 0) and (1, 1, ..., 1), respectively (i.e., so that the Pareto front is included in the m-dimensional unit cube $[0, 1]^m$). The reference point for the HV indicator is specified as (1.1, 1.1, ..., 1.1) in the normalized objective space. After the termination of each algorithm, this normalization is also used in our HV-based solution selection.

Each algorithm is applied to each test problem 20 times to calculate the average HV value. Experimental results under the small and very large settings of the population size are shown in Table II and Table III, respectively. For each test problem in each table, the best and worst results among the four algorithms are highlighted by yellow and gray, respectively. In Table II, good results are obtained by MOEA/DD and θ-DEA. However, MOEA/D looks the best in Table III (whereas it looks the worst in Table II). These observations clearly show that our performance comparison results heavily depend on the setting of the population size.

For almost all test problems, the best results are obtained by different algorithms between Table II and Table III. The best results are obtained by MOEA/D only for two problems (e.g., three-objective WFG1) in Table II with the small population size. In Table III with the very large population size, the best results are obtained by MOEA/DD only for the five-objective DTLZ2 problem and by θ-DEA only for two problems (e.g., five-objective DTLZ4). We show the relation between the average HV value and the population size for these problems in Figs. 2-4. Whereas the best results are obtained by a single algorithm for the three settings of the population size for each test problem in these figures, performance comparison results strongly depend on the population size for almost all test problems in our computational experiments as shown in Fig. 5 on the five-objective WFG3 problem.

TABLE II. EXPERIMENTAL RESULTS BY SMALL POPULATION.

Problem	m	MOEA/D	NSGA-III	MOEA/DD	θ-DEA
DTLZ1	3	1.069	1.118	1.120	1.119
	5	1.512	1.575	1.578	1.578
DTLZ2	3	0.700	0.744	0.745	0.745
	5	1.146	1.308	1.309	1.309
DTLZ3	3	0.682	0.701	0.729	0.715
	5	1.144	1.297	1.306	1.303
DTLZ4	3	0.412	0.713	0.745	0.745
	5	1.029	1.309	1.309	1.309
WFG1	3	0.994	0.788	0.861	0.813
	5	1.527	0.956	1.346	1.235
WFG2	3	1.150	1.231	1.228	1.236
	5	1.587	1.600	1.563	1.601
WFG3	3	0.806	0.824	0.779	0.820
	5	0.886	1.014	0.967	1.029
WFG4	3	0.668	0.735	0.726	0.735
	5	1.014	1.292	1.265	1.294
WFG5	3	0.620	0.689	0.679	0.689
	5	0.933	1.224	1.192	1.225
WFG6	3	0.625	0.695	0.685	0.694
	5	0.940	1.226	1.197	1.225
WFG7	3	0.669	0.735	0.726	0.736
	5	1.015	1.297	1.263	1.300
WFG8	3	0.622	0.672	0.663	0.672
	5	0.608	1.190	1.160	1.190
WFG9	3	0.620	0.688	0.677	0.682
	5	0.825	1.222	1.159	1.231

Yellow shows the best while gray shows the worst.

TABLE III. EXPERIMENTAL RESULTS BY VERY LARGE POPULATION.

Problem	m	MOEA/D	NSGA-III	MOEA/DD	θ-DEA
DTLZ1	3	1.147	0.000	0.000	0.000
	5	1.571	0.007	0.148	0.047
DTLZ2	3	0.794	0.650	0.658	0.654
	5	1.319	1.357	1.368	1.366
DTLZ3	3	0.762	0.000	0.000	0.000
	5	1.323	0.000	0.000	0.000
DTLZ4	3	0.795	0.484	0.226	0.490
	5	1.310	1.354	1.110	1.363
WFG1	3	0.894	0.068	0.035	0.074
	5	1.548	0.373	0.296	0.378
WFG2	3	1.211	0.970	0.948	0.966
	5	1.593	1.472	1.404	1.471
WFG3	3	0.841	0.692	0.686	0.698
	5	1.088	0.968	0.876	0.966
WFG4	3	0.774	0.618	0.612	0.613
	5	1.242	1.189	1.152	1.208
WFG5	3	0.728	0.505	0.502	0.499
	5	1.160	1.101	1.089	1.125
WFG6	3	0.735	0.442	0.435	0.437
	5	1.175	1.051	1.031	1.077
WFG7	3	0.781	0.553	0.558	0.548
	5	1.240	1.191	1.160	1.214
WFG8	3	0.697	0.488	0.480	0.483
	5	1.103	1.052	0.990	1.070
WFG9	3	0.739	0.505	0.519	0.504
	5	1.164	1.150	1.130	1.177

Yellow shows the best while gray shows the worst.

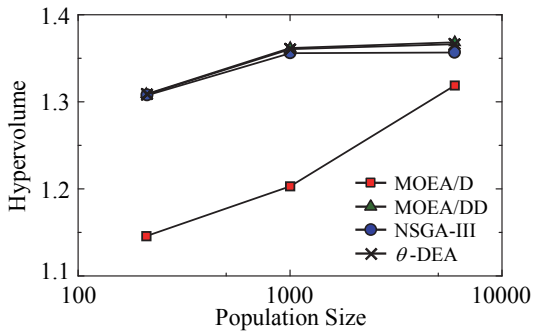


Fig. 2. Results on the five-objective DTLZ2 problem.

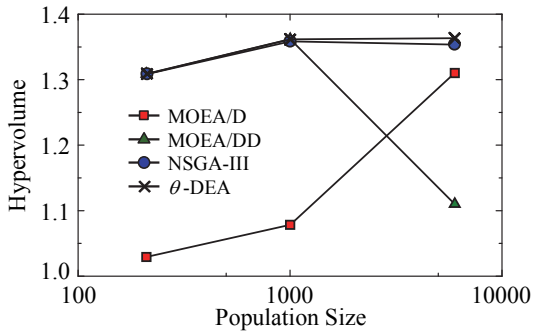


Fig. 3. Results on the five-objective DTLZ4 problem.

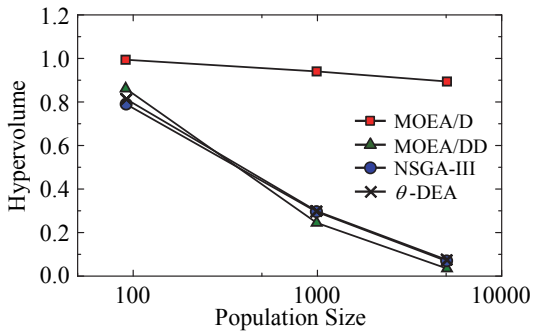


Fig. 4. Results on the three-objective WFG1 problem.

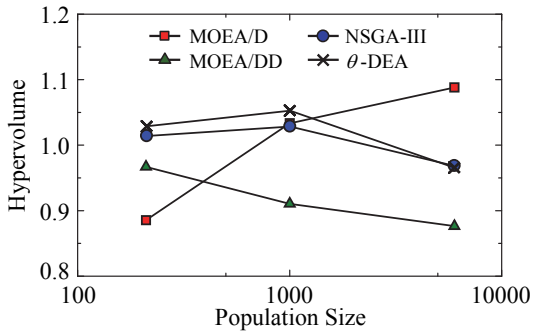


Fig. 5. Results on the five-objective WFG3 problem.

TABLE IV. BEST AVERAGE RESULTS FOR EACH POPULATION SIZE.

Problem	m	Small	Large	Very Large
DTLZ1	3	1.120	1.139	1.147
	5	1.578	1.589	1.571
DTLZ2	3	0.745	0.782	0.794
	5	1.309	1.362	1.368
DTLZ3	3	0.729	0.758	0.762
	5	1.306	1.324	1.323
DTLZ4	3	0.745	0.780	0.795
	5	1.309	1.362	1.363
WFG1	3	0.994	0.940	0.894
	5	1.527	1.546	1.548
WFG2	3	1.236	1.205	1.211
	5	1.601	1.593	1.593
WFG3	3	0.824	0.842	0.841
	5	1.029	1.053	1.088
WFG4	3	0.735	0.761	0.774
	5	1.294	1.276	1.242
WFG5	3	0.689	0.716	0.728
	5	1.225	1.232	1.160
WFG6	3	0.695	0.720	0.735
	5	1.226	1.219	1.175
WFG7	3	0.736	0.767	0.781
	5	1.300	1.307	1.240
WFG8	3	0.672	0.690	0.697
	5	1.190	1.189	1.103
WFG9	3	0.688	0.716	0.739
	5	1.231	1.241	1.177

TABLE V. BEST ALGORITHMS FOR EACH POPULATION SIZE.

Problem	m	Small	Large	Very Large
DTLZ1	3	MOEA/DD	MOEA/D	MOEA/D
	5	MOEA/DD	MOEA/DD	MOEA/D
DTLZ2	3	MOEA/DD	MOEA/DD	MOEA/D
	5	MOEA/DD	MOEA/DD	MOEA/DD
DTLZ3	3	MOEA/DD	MOEA/D	MOEA/D
	5	MOEA/DD	MOEA/DD	MOEA/D
DTLZ4	3	θ-DEA	θ-DEA	MOEA/D
	5	θ-DEA	MOEA/DD	θ-DEA
WFG1	3	MOEA/D	MOEA/D	MOEA/D
	5	MOEA/D	MOEA/D	MOEA/D
WFG2	3	θ-DEA	MOEA/D	MOEA/D
	5	θ-DEA	MOEA/D	MOEA/D
WFG3	3	NSGA-III	MOEA/D	MOEA/D
	5	θ-DEA	θ-DEA	MOEA/D
WFG4	3	θ-DEA	MOEA/D	MOEA/D
	5	θ-DEA	θ-DEA	MOEA/D
WFG5	3	θ-DEA	MOEA/D	MOEA/D
	5	θ-DEA	θ-DEA	MOEA/D
WFG6	3	NSGA-III	MOEA/D	MOEA/D
	5	NSGA-III	θ-DEA	MOEA/D
WFG7	3	θ-DEA	MOEA/D	MOEA/D
	5	θ-DEA	θ-DEA	MOEA/D
WFG8	3	θ-DEA	MOEA/D	MOEA/D
	5	NSGA-III	θ-DEA	MOEA/D
WFG9	3	NSGA-III	MOEA/D	MOEA/D
	5	θ-DEA	θ-DEA	θ-DEA

In Table IV, we show the best average HV value for each test problem under each setting of the population size. For example, the first value 1.120 in the "Small" column is the best result in Table II for the three-objective DTLZ1 problem. This best result 1.120 is obtained by MOEA/DD in Table II. The best algorithm corresponding to the best result in each cell of Table IV (e.g., MOEA/DD for 1.120) is shown in Table V. Among the three results in each row of Table IV, the largest and smallest average HV values are shown by yellow and gray, respectively. The corresponding algorithms in Table V are also shown by the same colors.

Each yellow cell in Table V, which shows the algorithm corresponding to the largest average HV value, does not necessarily means the best algorithm for each test problem. This is because each yellow cell in Table V is selected without taking into account the population size. For example, in the second row (i.e., "DTLZ1 with m = 3" row) of Table V, the obtained solution sets by MOEA/DD with the population size 91 are compared with those by MOEA/D with the population size 5050 for the three-objective DTLZ1 problem. For fair comparison, it is needed to compare different solution sets under the same size.

IV. SOLUTION SELECTION FROM THE FINAL POPULATION

Using the greedy forward selection method based on the HV indictor, we select 50 solutions from the final population in each run of each algorithm for each test problem. Independent of the population size, we always select 50 solutions. Of course, we can use other specifications as the number of solutions to be selected for performance comparison. Fig. 6 shows the results of solution selection for the five-objective WFG3 problem. The average HV values in Fig. 6 are decreased by solution selection from Fig. 5 especially when the population size is very large.

In the same manner as in Table IV and Table V, solution selection results from the final population are summarized in Table VI and Table VII where the best result after solution selection is highlighted by yellow for each test problem. It should be noted that the size of solution sets is always 50 in these tables. Thus the highlighted algorithm by yellow in Table VII can be viewed as the best algorithm for each test problem among the examined four algorithms.

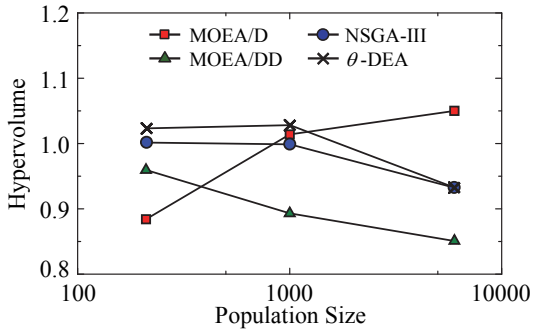


Fig. 6. Solution selection from the final population (5-objective WFG3).

TABLE VI. BEST AVERAGE RESULTS FOR EACH POPULATION SIZE AFTER THE SOLUTION SELECTION FROM THE FINAL POPULATION.

Problem	m	Small	Large	Very Large
DTLZ1	3	1.096	1.094	1.091
	5	1.544	1.544	1.534
DTLZ2	3	0.731	0.730	0.732
	5	1.234	1.230	1.210
DTLZ3	3	0.715	0.710	0.707
	5	1.230	1.179	1.215
DTLZ4	3	0.731	0.726	0.733
	5	1.234	1.233	1.201
WFG1	3	0.993	0.933	0.886
	5	1.527	1.544	1.546
WFG2	3	1.233	1.198	1.202
	5	1.598	1.592	1.590
WFG3	3	0.823	0.829	0.825
	5	1.023	1.028	1.050
WFG4	3	0.722	0.717	0.719
	5	1.219	1.152	1.155
WFG5	3	0.675	0.670	0.670
	5	1.149	1.104	1.071
WFG6	3	0.681	0.675	0.679
	5	1.151	1.097	1.088
WFG7	3	0.723	0.723	0.724
	5	1.226	1.184	1.154
WFG8	3	0.659	0.649	0.645
	5	1.111	1.051	1.013
WFG9	3	0.674	0.673	0.685
	5	1.152	1.106	1.077

TABLE VII. BEST ALGORITHMS FOR EACH POPULATION SIZE AFTER THE SOLUTION SELECTION FROM THE FINAL POPULATION.

Problem	m	Small	Large	Very Large
DTLZ1	3	MOEA/DD	MOEA/D	MOEA/D
	5	MOEA/DD	MOEA/DD	MOEA/D
DTLZ2	3	MOEA/DD	MOEA/D	MOEA/D
	5	MOEA/DD	MOEA/DD	MOEA/D
DTLZ3	3	MOEA/DD	MOEA/D	MOEA/D
	5	MOEA/DD	MOEA/D	MOEA/D
DTLZ4	3	MOEA/DD	θ-DEA	MOEA/D
	5	MOEA/DD	θ-DEA	MOEA/D
WFG1	3	MOEA/D	MOEA/D	MOEA/D
	5	MOEA/D	MOEA/D	MOEA/D
WFG2	3	θ-DEA	MOEA/D	MOEA/D
	5	θ-DEA	MOEA/D	MOEA/D
WFG3	3	NSGA-III	MOEA/D	MOEA/D
	5	θ-DEA	θ-DEA	MOEA/D
WFG4	3	θ-DEA	MOEA/D	MOEA/D
	5	θ-DEA	MOEA/DD	MOEA/D
WFG5	3	θ-DEA	MOEA/D	MOEA/D
	5	θ-DEA	θ-DEA	MOEA/D
WFG6	3	NSGA-III	MOEA/D	MOEA/D
	5	NSGA-III	MOEA/DD	MOEA/D
WFG7	3	θ-DEA	MOEA/D	MOEA/D
	5	θ-DEA	MOEA/DD	MOEA/D
WFG8	3	θ-DEA	MOEA/D	MOEA/D
	5	NSGA-III	θ-DEA	MOEA/D
WFG9	3	NSGA-III	MOEA/D	MOEA/D
	5	θ-DEA	θ-DEA	MOEA/D

Whereas the apparent best results are obtained by MOEA/D with the very large population size for many test problems in Table V before solution selection, the small population size seems to be the best setting in Table VII. However, the best results are still obtained from the very large population size for the six test problems in Table VII. This may be a surprising observation since less than 1% of solutions are selected from the very large final population.

V. SOLUTION SELECTION FROM THE EXAMINED SOLUTIONS

In general, the final population of an EMO algorithm is not always the best solution set. Thus it may be a good idea to search for a good solution set from the examined solutions. Fig. 7 shows the average HV values of 50 solutions selected from the examined solutions for the five-objective WFG3 problem. The comparison between Fig. 6 and Fig. 7 shows that the use of the examined solutions in Fig. 7 improves the results of the solution selection from the final population in Fig. 6. The improvement is clear especially when the population size is small (i.e., when the size of the final population is small).

The main difficulty in the use of the examined solutions for solution selection is the huge number of candidate solutions. Table VIII shows the average percentage of different non-dominated solutions among the examined ones. For example, about 50% (i.e., 100,000 solutions) of the examined solutions for the five-objective WFG3 problem are non-dominated.

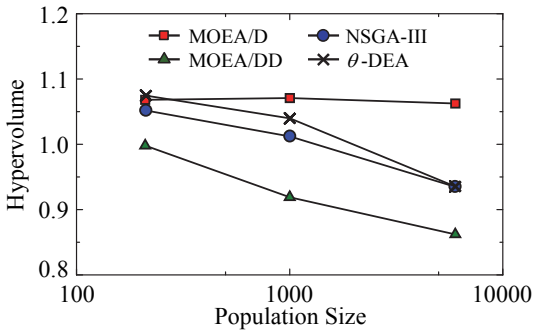


Fig. 7. Solution selection from the examined solutions (5-objective WFG3).

TABLE VIII. PERCENTAGE OF NON-DOMINATED SOLUTIONS.

Problem	m	Small	Large	Very Large
DTLZ1-4	3	21.9%	11.1%	6.1%
	5	41.7%	35.1%	19.2%
WFG1	3	1.8%	1.6%	1.6%
	5	7.8%	8.5%	7.8%
WFG2	3	7.9%	3.2%	1.8%
	5	28.3%	22.1%	8.3%
WFG3	3	24.4%	28.3%	13.4%
	5	55.5%	53.0%	46.8%
WFG4-9	3	16.4%	10.9%	6.4%
	5	33.4%	33.2%	24.4%

TABLE IX. BEST AVERAGE RESULTS FOR EACH POPULATION SIZE AFTER THE SOLUTION SELECTION FROM THE EXAMINED SOLUTIONS.

Problem	m	Small	Large	Very Large
DTLZ1	3	1.099	1.097	1.093
	5	1.547	1.545	1.547
DTLZ2	3	0.734	0.733	0.733
	5	1.241	1.236	1.241
DTLZ3	3	0.718	0.713	0.708
	5	1.239	1.189	1.240
DTLZ4	3	0.734	0.728	0.733
	5	1.241	1.238	1.227
WFG1	3	1.017	0.935	0.887
	5	1.536	1.549	1.549
WFG2	3	1.237	1.210	1.204
	5	1.602	1.601	1.598
WFG3	3	0.847	0.830	0.825
	5	1.075	1.071	1.062
WFG4	3	0.732	0.723	0.721
	5	1.240	1.240	1.240
WFG5	3	0.679	0.677	0.673
	5	1.159	1.159	1.159
WFG6	3	0.685	0.680	0.680
	5	1.160	1.161	1.164
WFG7	3	0.732	0.728	0.725
	5	1.236	1.235	1.237
WFG8	3	0.662	0.651	0.645
	5	1.117	1.057	1.101
WFG9	3	0.682	0.682	0.692
	5	1.165	1.158	1.182

TABLE X. BEST ALGORITHMS FOR EACH POPULATION SIZE AFTER THE SOLUTION SELECTION FROM THE EXAMINED SOLUTIONS.

Problem	m	Small	Large	Very Large
DTLZ1	3	MOEA/DD	MOEA/D	MOEA/D
	5	MOEA/DD	MOEA/DD	MOEA/D
DTLZ2	3	MOEA/DD	MOEA/D	MOEA/D
	5	θ-DEA	MOEA/DD	MOEA/D
DTLZ3	3	MOEA/DD	MOEA/D	MOEA/D
	5	MOEA/D	MOEA/DD	MOEA/D
DTLZ4	3	θ-DEA	θ-DEA	MOEA/D
	5	MOEA/DD	θ-DEA	MOEA/D
WFG1	3	MOEA/D	MOEA/D	MOEA/D
	5	MOEA/D	MOEA/D	MOEA/D
WFG2	3	NSGA-III	MOEA/D	MOEA/D
	5	θ-DEA	MOEA/D	MOEA/D
WFG3	3	θ-DEA	MOEA/D	MOEA/D
	5	θ-DEA	MOEA/D	MOEA/D
WFG4	3	MOEA/D	MOEA/D	MOEA/D
	5	MOEA/D	MOEA/D	MOEA/D
WFG5	3	MOEA/D	MOEA/D	MOEA/D
	5	MOEA/D	MOEA/D	MOEA/D
WFG6	3	MOEA/D	MOEA/D	MOEA/D
	5	MOEA/DD	MOEA/D	MOEA/D
WFG7	3	MOEA/D	MOEA/D	MOEA/D
	5	MOEA/D	MOEA/D	MOEA/D
WFG8	3	MOEA/DD	MOEA/D	MOEA/D
	5	NSGA-III	θ-DEA	MOEA/D
WFG9	3	NSGA-III	MOEA/D	MOEA/D
	5	θ-DEA	MOEA/D	MOEA/D

We apply the HV-based greedy forward selection method to all non-dominated solutions among the examined ones to select only 50 solutions. Experimental results are summarized in Table IX and Table X. As in Table VII, the best results in the yellow cells are obtained from the small population size for many test problems. However, the best algorithm for each test problem under the small population size is not the same in the three tables: Table V before solution selection, Table VII from the final population, and Table X from the examined solutions. Actually, in the third column of Table V under the small population size, the best results are obtained before solution selection by MOEA/D, NSGA-III, MOEA/DD and θ-DEA for 2, 5, 6 and 13 problems, respectively. However, after solution selection, the best results under the small population size are obtained by these algorithms for 2, 5, 8 and 11 test problems in the third column of Table VII, and for 10, 3, 7 and 6 test problems in the third column of Table X, respectively.

VI. CONCLUDING REMARKS

In this paper, we discussed difficulties of fair performance comparisons of evolutionary many-objective algorithms. First we demonstrated that totally different comparison results were obtained from different settings of the population size through computational experiments on the DTLZ1-4 and WFG1-9 problems with three and five objectives. When the population size was small in Table II, MOEA/D was the worst for almost all test problems. However, when the populations size was very large in Table III, MOEA/D was the best for almost all test problems. Next we reported performance comparison results after solution selection from the final population where only 50 solutions were selected. For many test problems, the best results were obtained from the small setting of the population size. Finally we reported performance comparison results after solution selection from the examined solutions. In this case, the best results were almost obtained from the small population size setting for many test problems. This is the most general performance comparison scheme since different algorithms with different settings of internal and external populations can be compared. An important observation in this paper is that the worst algorithm with respect to the small final population (i.e., MOEA/D in Table II for 22 out of the 26 test problems) was evaluated as being the best after solution selection from the examined solutions under the small population size setting (i.e., MOEA/D in the third column of Table X for 10 test problems).

REFERENCES

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

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

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

13, pp. 1-35, September 2015.

[4] 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, 2011.

[6] 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.

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

[8] 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.

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

[10] 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* (Available from the IEEE Xplore as an Early Access Paper).

[11] 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.

[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, October 2006.

[13] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. on Evolutionary Computation*, vol. 1, no. 1, pp. 67-82, April 1997.

[14] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Review of coevolutionary developments of evolutionary multi-objective and many-objective algorithms and test problems," *Proc. of 2014 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*, pp. 178-184, Orlando, Florida, USA, December 9-12, 2014

[15] H. Ishibuchi, H. Masuda, and Y. Nojima, "Pareto fronts of many-objective degenerate test problems," *IEEE Trans. on Evolutionary Computation* (Accepted: Available from the IEEE Xplore as an Early Access Paper).

[16] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms – A comparative case study," *Proc. of 5th Parallel Problem Solving from Nature (Lecture Notes in Computer Science 1498: PPSN V),* pp. 292-301, Amsterdam, Netherlands, September 27-30, 1998.

[17] L. While, L. Bradstreet, and L. Barone, "A fast way of calculating exact hypervolumes," *IEEE Trans. on Evolutionary Computation*, vol. 16, no. 1, pp. 86-95, February 2012.

[18] L. M. S. Russo and A. P. Francisco, "Quick hypervolume," *IEEE Trans. on Evolutionary Computation*, vol. 18, no. 4, pp. 481-502, August 2014.

[19] T. Watanabe, T. Tatsukawa, and A. Oyama, "On the fast hypervolume calculation method," *Proc. of 2015 IEEE Congress on Evolutionary Computation*, pp. 965-969, Sendai, Japan, May 25-28, 2015.

[20] C. A. C. Coello and M. R. Sierra, "A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm," *Lecture Notes in Computer Science 2972: MICAI 2004*, pp. 688-697, April 2004.

[21] M. R. Sierra and C. A. C. Coello, "A new multi-objective particle swarm optimizer with improved selection and diversity mechanisms," *Technical Report of CINVESTAV-IPN*, Nov. 23, 2004 (Available from http://delta.cs.cinvestav.mx/~ccoello/techreports/evocinv-05-2004.pdf).

[22] H. Ishibuchi, H. Masuda, and Y. Nojima, "A study on performance evaluation ability of a modified inverted generational distance indicator," *Proc. of Genetic and Evolutionary Computation Conference*, pp. 695-702, Madrid, Spain, July 11-15, 2015.

[23] 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 (Lecture Notes in Computer Science 4403: EMO 2007)*, pp. 862-876, Matsushima, Japan, March 5-8, 2007.

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

[25] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Difficulties in specifying reference points to calculate the inverted generational distance for many-objective optimization problems," *Proc. of 2014 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*, pp. 170-177, Orlando, Florida, USA, December 9-12, 2014.

[26] H. Ishibuchi, H. Masuda, and Y. Nojima, "Comparing solution sets of different size in evolutionary many-objective optimization," *Proc. of 2015 IEEE Congress on Evolutionary Computation*, pp. 2859-2866, Sendai, Japan, May 25-28, 2015.

[27] 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.

[28] 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, Ljubljana, Slovenia, September 13-17, 2014.

[29] T. Kuhn, C. M. Fonseca, L. Paquete, S. Ruzikam, M. M. Duarte, and J. R. Figueira, "Hypervolume subset selection in two dimensions: Formulations and algorithms," *Evolutionary Computation* (Online available as an Early Access Paper).

[30] A. P. Guerreiro, C. M. Fonseca, and L. Paquete, "Greedy hypervolume subset selection in the three-objective case," *Proc. of 2015 Genetic and Evolutionary Computation Conference*, pp. 671-678, Madrid, Spain, July 11-15, 2015.

[31] T. Glasmachers, "Optimized approximation sets for low-dimensional benchmark Pareto fronts," *Proc. of 13th International Conference on Parallel Problem Solving from Nature*, pp. 569-578, Ljubljana, Slovenia, September 13-17, 2014.

[32] K. Bringmann, T. Friedrich, and P. Klitzke, "Efficient computation of two-dimensional solution sets maximizing the epsilon-indicator," *Proc. of 2015 IEEE Congress on Evolutionary Computation*, pp. 970-977, Sendai, Japan, May 25-28, 2015.

[33] 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, Vancouver, Canada, July 12-16, 2014.

[34] 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, San Diego, CA, USA, October 5-8, 2014.

[35] 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, Montreal, Canada, July 8-12, 2009.

[36] 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, Jeju Island, Korea, August 20-24, 2009.

[37] http://learn.tsinghua.edu.cn:8080/2012310563/ManyEAs.rar (Codes were downloaded on October 1, 2015).

[38] http://www.cs.cityu.edu.hk/~51888309/ (Code was downloaded on November 13, 2015).