Benchmarking MOEAs for Multi- and Many-objective Optimization Using an Unbounded External Archive

Ryoji Tanabe
Japan Aerospace Exploration Agency (JAXA)
Sagamihara City, Japan
rt.ryoji.tanabe@gmail.com

Akira Oyama
Japan Aerospace Exploration Agency (JAXA)
Sagamihara City, Japan
oyama@flab.isas.jaxa.jp

ABSTRACT

While a large number of multi-objective evolutionary algorithms (MOEAs) for many-objective optimization problems (MaOPs) have been proposed in the past few years, an exhaustive benchmarking study has never been performed. Moreover, most previous studies evaluated the performance of MOEAs based on nondominated solutions in the final population at the end of the search. In this paper, we exhaustively investigate the convergence performance of 21 MOEAs using an unbounded external archive that stores all nondominated solutions found during the search process. Surprisingly, the experimental results for the WFG functions with up to six objectives indicate that several recently proposed MOEAs perform significantly worse than classical MOEAs. Moreover, the performance rank among the 21 MOEAs significantly depends on the number of function evaluations. Thus, the previously reported performance of MOEAs on MaOPs as well as the widely used benchmarking methodology must be carefully reconsidered.

CCS CONCEPTS

•Mathematics of computing → Evolutionary algorithms;

KEYWORDS

Benchmarking, Evolutionary Multi-objective Optimization

ACM Reference format:
Ryoji Tanabe and Akira Oyama. 2016. Benchmarking MOEAs for Multi- and Many-objective Optimization Using an Unbounded External Archive. In *Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17),* 8 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

An unconstrained (bound-constrained) multi-objective continuous optimization problem (MOP) can be formulated as follows:

$$\begin{aligned} &\text{minimize} \quad \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), ..., f_M(\boldsymbol{x}))^{\mathrm{T}} \\ &\text{subject to} \quad \boldsymbol{x} \in \mathbb{S} \subseteq \mathbb{R}^D \end{aligned} \tag{1}$$

where, $\boldsymbol{f} : \mathbb{S} \rightarrow \mathbb{R}^M$ is an objective function vector that consists of M potentially conflicting objective functions, and $\mathbb{R}^M$ is the objective function space. Here, $\boldsymbol{x} = (x_1, ..., x_D)^{\mathrm{T}}$ is a D-dimensional solution vector, and $\mathbb{S} = \Pi_{j=1}^{D}[a_j, b_j]$ is the bound-constrained search space, where $a_j \leq x_j \leq b_j$ for each index $j \in \{1, ..., D\}$.

We say that $\boldsymbol{x}^1$ dominates $\boldsymbol{x}^2$ and denote $\boldsymbol{x}^1 \prec \boldsymbol{x}^2$ iff $f_i(\boldsymbol{x}^1) \leq f_i(\boldsymbol{x}^2)$ for all $i \in \{1, ..., M\}$ and $f_i(\boldsymbol{x}^1) < f_i(\boldsymbol{x}^2)$ for at least one index i. Here, $\boldsymbol{x}^*$ is a Pareto optimal solution if there exists no $\boldsymbol{x} \in \mathbb{S}$ such that $\boldsymbol{x} \prec \boldsymbol{x}^*$. Moreover, $\boldsymbol{f}(\boldsymbol{x}^*)$ is a Pareto optimal objective function vector. The set of all $\boldsymbol{x}^*$ in $\mathbb{S}$ is the Pareto optimal solution set (PS), and the set of all $\boldsymbol{f}(\boldsymbol{x}^*)$ is the Pareto frontier (PF). The goal of an MOP is to find a set of nondominated solutions that are well-distributed and close to the PF in the objective function space.

A multi-objective evolutionary algorithm (MOEA) is an efficient approach for solving MOPs [16]. However, while classical MOEAs (e.g., NSGA-II [5] and SPEA2 [40]) perform relatively well for MOPs with $M \leq 3$, their performance significantly degrades for MOPs with $M \geq 4$ [30]. An MOP with $M \geq 4$ is referred to as a many-objective continuous optimization problem (MaOP). In the past few years, researchers in the evolutionary computation community have attempted to design new MOEAs that can handle a large number of objectives [16]. Table 1 shows 21 MOEAs, including recently proposed methods for MaOPs (e.g., NSGA-III [6]) and classical methods which were originally designed only for MOPs with $M \leq 3$ (e.g., NSGA-II [5]).

Which MOEA is best? While a number of MOEAs for MaOPs have been proposed in 2014 – 2016 as shown in Table 1, as far as we know, an exhaustive benchmarking study has never been performed. In almost all previous studies, a proposed MOEA was compared with only a few algorithms/heuristics selected by the authors. Although a number of previous studies have examined benchmarking MOEAs for MaOPs (e.g., [23, 25, 30]), the number of methods for comparison was limited to $8 \sim 9$.

Moreover, most importantly, in the previous benchmarking studies [23, 25, 30] and in all of the papers listed in Table 1, MOEAs were compared based on nondominated solutions in the final population at the end of the search. Note that the performance rank decided by the above traditional benchmarking procedure strongly depends on the maximum number of function evaluations ($\text{FEvals}^{\max}$). In other words, for example, the conclusion "MOEA_1 performs better than MOEA_2" obtained by an experiment with $\text{FEvals}^{\max} = 7 \times 10^5$ cannot be applied when $\text{FEvals}^{\max}$ is set to 2×10^4. However, as far as we know, the standard setting of $\text{FEvals}^{\max}$ for the end-of-run results-based traditional comparison has never been defined in the evolutionary computation community, except for some competitions (e.g., [38]), and thus researchers set $\text{FEvals}^{\max}$ to an arbitrary number[1]. Table 1 shows the $\text{FEvals}^{\max}$ used in each of the previous studies. These values of $\text{FEvals}^{\max}$ differ significantly from each other. For example, $\text{FEvals}^{\max}$ was set to 1×10^4 in the HypE paper [2], whereas, in the recent studies, $\text{FEvals}^{\max}$ was set to 5.5×10^5

[1] We do not have an intention to claim that researchers set $\text{FEvals}^{\max}$ to an arbitrary number for *all possible problems*. Note that $\text{FEvals}^{\max}$ cannot be set to an arbitrary number in practice since some real-world problems require the execution of a simulation that takes a long time to evaluate the solution [15, 26].

Conference'17, Washington, DC, USA
2016. 978-x-xxxx-xxxx-x/YY/MM…$15.00
DOI: 10.1145/nnnnnnn.nnnnnnn

Table 1: Properties of 21 MOEAs benchmarked in this paper: C1 Pareto-dominance based MOEAs, C2 relaxed-dominance based MOEAs, C3 decomposition based MOEAs, C4 indicator based MOEAs, and C5 reference-vector based MOEAs. We also show $\text{FEvals}^{\max}$ used in each paper and its publication year. MOEA/D-07 and MOEA/D-09 correspond to the original names "MOEA/D" and "MOEA/D-DE", respectively.

MOEAs	C1	C2	C3	C4	C5	Year	$\text{FEvals}^{\max}$
NSGA-II [5]	✓					2002	2.5×10^4
NSGA-III [6]	✓				✓	2014	5.5×10^5
SPEA2 [40]	✓					2001	1.0×10^6
SPEA2+SDE [21]	✓					2014	1.0×10^5
GrEA [33]		✓				2013	1.0×10^5
VaEA [32]	✓				✓	2016	5.5×10^5
θ-DEA [34]	✓		✓		✓	2016	5.5×10^5
MOEA/D-07 [36]			✓			2007	7.5×10^4
MOEA/D-09 [18]			✓			2009	3.0×10^5
MOEA/D-DRA [37]			✓			2009	3.0×10^5
MOEA/D-STM [20]			✓			2014	3.0×10^5
MOEA/DD [19]	✓		✓			2015	5.5×10^5
MOEA/D-DU [35]			✓			2016	2.6×10^5
I-DBEA [1]	✓		✓			2015	1.4×10^6
EFR-RR [35]			✓			2016	2.6×10^5
RVEA [4]			✓		✓	2016	2.8×10^5
$\text{IBEA}_{\epsilon+}$ [39]				✓		2004	2.0×10^4
IBEA_{HD} [39]				✓		2004	2.0×10^4
HypE [2]	✓			✓		2011	1.0×10^4
BiGE [22]				✓		2015	1.0×10^5
MOMBI-II [8]				✓		2015	4.4×10^5

[6, 19, 32, 34]. A number of previous studies (e.g., [19–21, 33, 34]) have reported the poor performance of HypE. However, HypE was originally designed for optimization for $\text{FEvals}^{\max} = 1 \times 10^4$, and a performance comparison with a larger $\text{FEvals}^{\max}$ may seem unfair.

In addition, most previous studies used nondominated solutions in the population (at the end of the search) for calculating the hypervolume (HV) value. In general, MOEAs maintain (nondominated) solutions obtained during the search process in the population, the size of which is limited. When using nondominated solutions in the population for the HV calculation, a monotonic increase of the HV over time (which is equal to the number of function evaluations) cannot be ensured [7, 10, 24, 27]. Thus, we cannot exactly evaluate the performance of MOEAs using such a traditional evaluation methodology.

Taking into account the above considerations, in this paper, we exhaustively investigate the convergence performance of the 21 MOEAs listed in Table 1 on MaOPs with up to six objectives using an unbounded external archive. The unbounded external archive stores *all* nondominated solutions found during the search process and can be introduced into any MOEAs without any changes in their original algorithms [3, 7, 10, 24, 27]. When using the unbounded external archive, the issue described above can be addressed [24, 27]. Since, in practice, users of MOEAs want to know nondominated solutions whenever possible, we believe that the benchmark methodology using the unbounded external archive is more practical. The unbounded external archive was also adopted in the recently proposed BBOB-biobj benchmark suite [29] in the COCO framework[2]. When the number of obtained nondominated solutions in the unbounded external archive is too large, a method that selects only representative solutions (e.g., [13]) should be applied to them.

Table 2: Properties of the WFG functions.

Function	Shape of PF	Multimodality	Separability	Others
WFG1	Mixed		✓	Biased
WFG2	Discontinuous	✓		
WFG3	Degenerate			
WFG4	Nonconvex	✓	✓	
WFG5	Nonconvex		✓	Deceptive
WFG6	Nonconvex			
WFG7	Nonconvex		✓	Biased
WFG8	Nonconvex			Biased
WFG9	Nonconvex			Deceptive, Biased

This paper is inspired by two recent studies [3, 29]. In [3], Brockhoff et al. established the data profile-based methodology for visualizing the convergence performance of MOEAs, and their work was extended to the recently designed BBOB-biobj functions [29] in the COCO platform. We believe that the COCO with the BBOB-biobj suite [29] is one of most reliable benchmarking platforms. However, unfortunately, COCO *currently* provides only two-objective MOPs and the performance of MOEAs for MOPs with $M \geq 3$ cannot be evaluated using the *current* COCO platform. Although some previous studies evaluated the convergence performance of MOEAs [9, 26], these studies did not use the unbounded external archive.

This paper will be helpful to both users and algorithm designers of MOEAs. In practice, users want to apply a high-performance MOEA to real-world problems in order to obtain well-approximated nondominated solutions. Algorithm designers develop novel, state-of-the-art MOEAs by analyzing and improving well-performing MOEAs. Our benchmarking study will be useful in both cases. Moreover, all of the experimental data presented herein are available on the Website[3]. Thus, researchers can easily compare their newly designed MOEAs to the 21 MOEAs.

The remainder of this paper is organized as follows. Section 2 introduces the experimental settings used in this study. The experimental results are described and discussed in Section 3. We further discuss the experimental results and some previous work in Section 4. Finally, in Section 5, we conclude this paper and discuss our future work.

2 EXPERIMENTAL SETTINGS

In this benchmarking study, we used the nine WFG functions [11] with $M \in \{2, 3, 4, 5, 6\}$. Table 2 shows the properties of the WFG functions. As suggested in [11], the position parameter k was set

[2] http://coco.gforge.inria.fr/
[3] https://sites.google.com/site/benchmarkingmoeas/

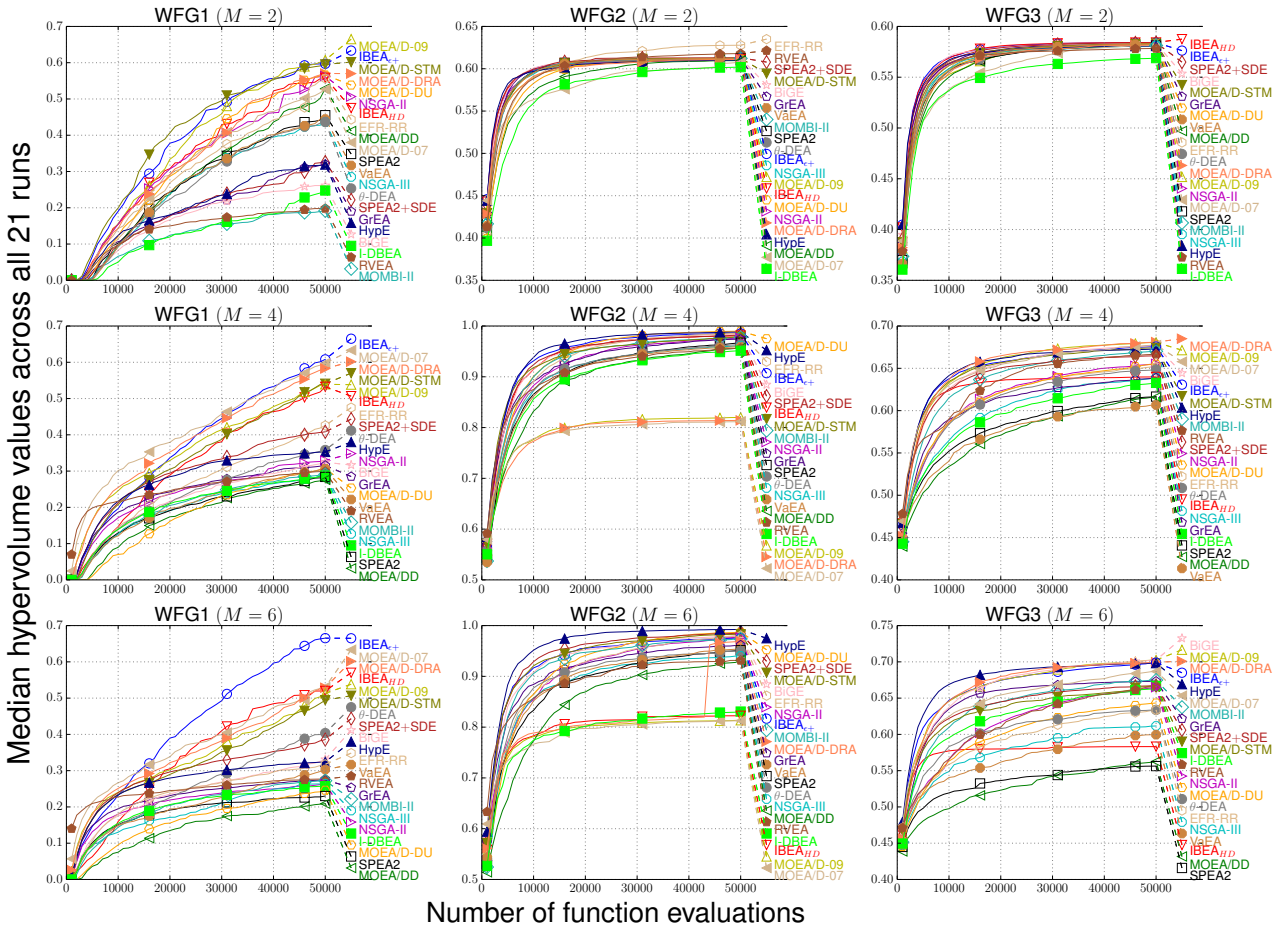


Figure 1: Convergence performance of the 21 MOEAs on the normalized WFG1, WFG2, and WFG3 functions with $M \in \{2, 4, 6\}$.

to $k = 2(M-1)$, and the distance parameter l was set to $l = 20$, where the number of variables D is $D = k + l$[4].

On all the WFG functions, the scale of each objective function is different. In general, an MOEA with an efficient normalization strategy for handling differently scaled objective function values performs well for such scaled MOPs [6]. Since objective function values of real-world problems are differently scaled, normalization strategies are mandatory in practice. However, in the 21 MOEAs, while some MOEAs adopt sophisticated normalization strategies (e.g., NSGA-III [6] and RVEA [4]), the others do not use any normalization strategies (e.g., NSGA-II [5] and MOEA/D-07 [36]). Since we want to evaluate the performance of MOEAs as fairly as possible, we used the normalized WFG functions to remove the effect of the normalization strategies as [14, 35]. For the normalized WFG functions, each objective function value $f_i(\boldsymbol{x}), i \in \{1, \ldots, M\}$ was normalized using the ideal point $(0, \ldots, 0)^{\mathrm{T}}$ and the nadir point $(2, \ldots, 2M)^{\mathrm{T}}$ as in [14, 35].

As suggested in [14, 35], the reference point for calculating the HV value was set to $(1.1, \ldots, 1.1)^{\mathrm{T}}$. Note that the nadir point of the normalized WFG functions is $(1, \ldots, 1)^{\mathrm{T}}$. In this setting, the HV range for all of the WFG functions is $[0, 1.1^M]$. We further normalized HV values $\in [0, 1.1^M]$ to the range $[0, 1]$ by divided by 1.1^M. The computational cost of the HV calculation exponentially increases both with M and the number of nondominated solutions in the unbounded external archive. In the COCO software with the two-objective BBOB-biobj functions [29], when a newly generated solution enters the unbounded external archive $\boldsymbol{A}^{\mathrm{unb}}$, the HV value of the nondominated solutions in $\boldsymbol{A}^{\mathrm{unb}}$ is recalculated immediately. However, for MaOPs with $M \geq 4$, since the computational cost of the HV calculation is very high, on-the-fly HV calculation like the COCO software with the two-objective BBOB-biobj functions is almost impossible. In order to address this issue, in this paper we calculated the HV value of the nondominated solutions in $\boldsymbol{A}^{\mathrm{unb}}$ only when FEvals $\in \{1 \times 10^3, 2 \times 10^3, \ldots, 4.9 \times 10^4, 5 \times 10^4\}$.

We used the SBX crossover and polynomial mutation for *all* 21 MOEAs. As suggested in [6], we set the control parameters of the variation operators as follows: $p_c = 1.0$, $\eta_c = 30$, $p_m = 1/D$, and $\eta_m = 20$. For most of the 21 MOEAs, we used the source codes

[4]In order to maintain consistency of properties of the WFG functions, k must be set to a number divisible by $(M-1)$. Thus, M and D are dependent on each other, and D could not be constant like the BBOB-biobj functions [29]. The scalability of MOEAs with respect to D will be investigated in our future work.

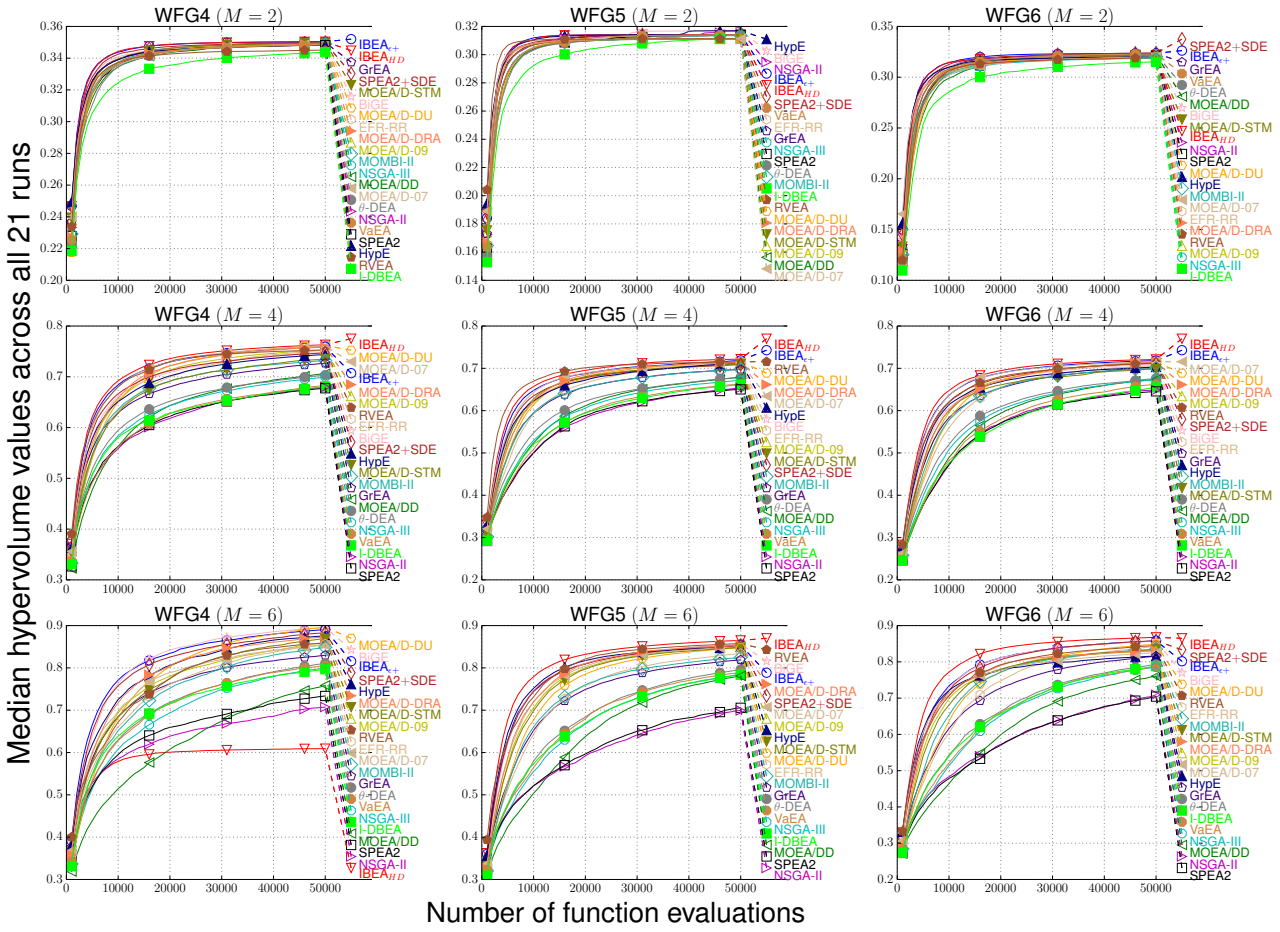


Figure 2: Convergence performance of the 21 MOEAs on the normalized WFG4, WFG5, and WFG6 functions with $M \in \{2, 4, 6\}$.

downloaded from websites as listed in Table S.1 in the supplementary file. We replaced the DE operator with the SBX crossover in MOEA/D-09, MOEA/D-DRA, and MOEA/D-STM. A slightly modified code of MOEA/D-09 was used as the source code of MOEA/D-07. The bug-fixed jMetal source code of IBEA$_{HD}$ was used in this study. We implemented IBEA$_{\epsilon+}$ by modifying the source code of IBEA$_{HD}$.

For each $M \in \{2, 3, 4, 5, 6\}$, we set the population size μ to 100, 92, 220, 212, and 184 respectively for GrEA, BiGE, SPEA2+SDE, and MOMBI-II because μ in their codes must be set as a multiple of four. For the remaining MOEAs, μ was set to $100, 91, 220, 210$, and 182 respectively. The μ values used in this paper have been widely used in previous studies such as [1, 6, 19, 35]. For the decomposition and reference vector based MOEAs, weight/reference vectors were generated using simplex-lattice design (for only $M = 6$, its two-layered version [6] was used). For GrEA, we set the *div* value as suggested in [4, 35]. The maximum number of function evaluations (FEvals$^{\max}$) was set to 5×10^4, and 21 independent runs were performed.

3 EXPERIMENTAL RESULTS

In this section, we investigate the performance of the 21 MOEAs listed in Table 1 on the WFG functions [11]. The experimental results for each function are described in Section 3.1. Then, in Section 3.2, we compare the MOEAs for all of the WFG functions. We also discuss the experimental results in Section 4.

3.1 Results for each WFG function

Figures 1, 2, and 3 show the results for the 21 MOEAs for each WFG function instance. Due to space constraints, we show only the results for the WFG functions with $M \in \{2, 4, 6\}$, but the results for $M \in \{3, 5\}$ are presented in Figures S.1 ~ S.3 in the supplemental file. Figures S.4 ~ S.6 in the supplemental file also show the detailed results for FEvals $\in \{3 \times 10^4, 3.1 \times 10^4, \ldots 5 \times 10^4\}$. In the following, we discuss the results for each WFG function:

• **WFG1**: IBEA$_{\epsilon+}$ and MOEA/D variants perform well in $M \in \{2, 4, 6\}$. For $M = 2$, MOEA/D-STM achieves the highest HV value until FEvals $= 3 \times 10^4$, but MOEA/D-09 and IBEA$_{\epsilon+}$ catch up with MOEA/D-STM for FEvals $> 3 \times 10^4$. For $M = 4$ and 6, the best performer is MOEA/D-07 for FEvals $< 3 \times 10^4$ and $< 1.3 \times 10^4$ respectively. After that, IBEA$_{\epsilon+}$ outperforms the other MOEAs.

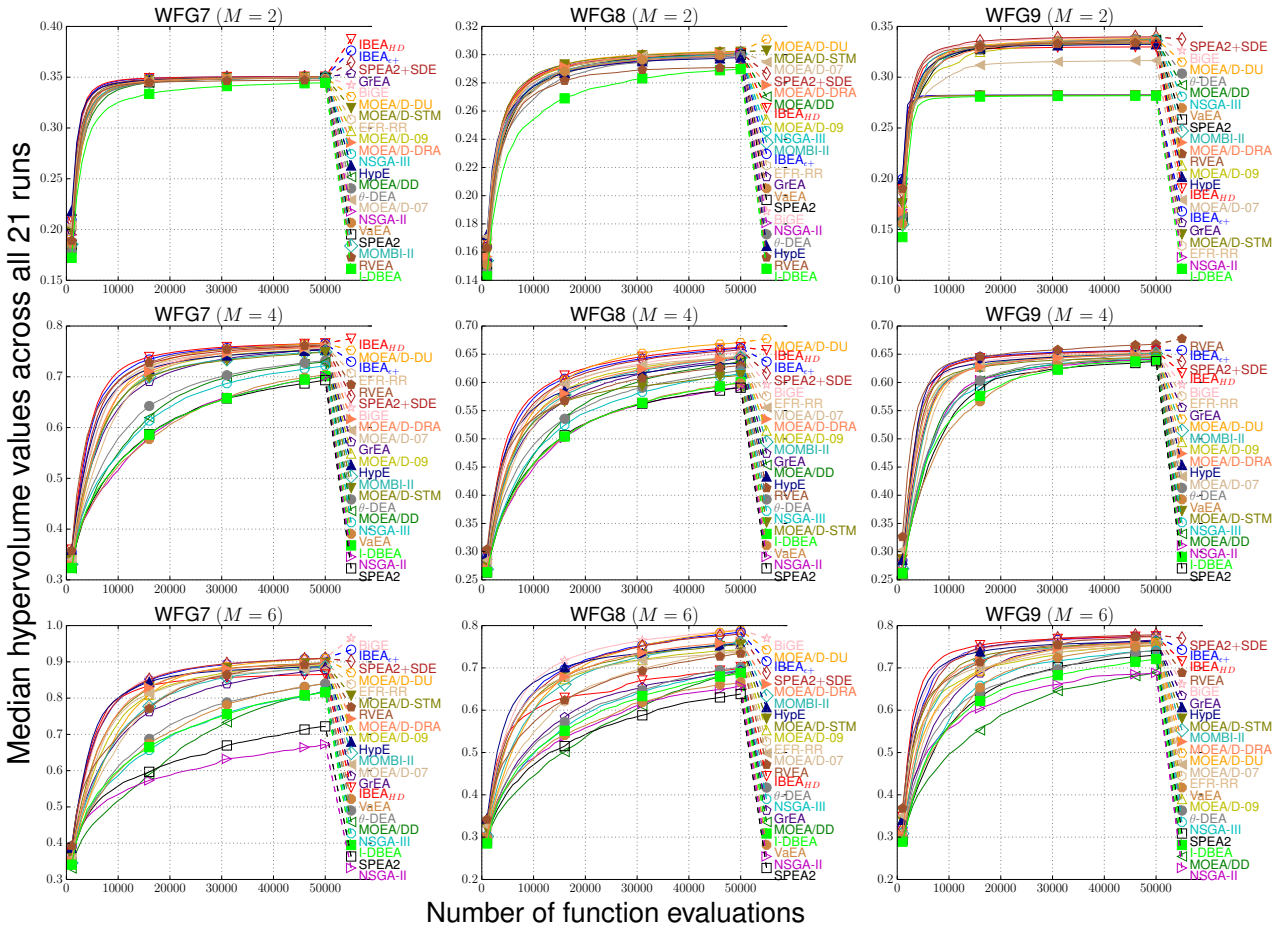


Figure 3: Convergence performance of the 21 MOEAs on the normalized WFG7, WFG8, and WFG9 functions with $M \in \{2, 4, 6\}$.

• **WFG2**: For $M = 2$, EFR-RR outperforms the other MOEAs after FEvals = 2×10^4. For $M = 4$ and $M = 6$, HypE and MOEA/D-DU achieve high HV values, whereas MOEA/D-07 and MOEA/D-09 perform poorly.

• **WFG3**: For $M = 2$, IBEA$_{HD}$ performs relatively well after FEvals = 1×10^4. However, with increasing M, the performance of IBEA$_{HD}$ deteriorates gradually. For $M = 4$ and $M = 6$, while MOEA/D-09, MOEA/D-DRA, IBEA$_{\epsilon+}$, and BiGE perform well for the larger FEvals, the best MOEA for the smaller FEvals is HypE.

• **WFG4**: For $M = 2$ and $M = 4$, IBEA$_{HD}$ exhibits good performance at all times, while for $M = 6$, its performance becomes inferior to the other MOEAs. On the other hand, for $M = 6$, BiGE and IBEA$_{\epsilon+}$ achieve high HV values, and MOEA/D-DU outperforms the remaining MOEAs at approximately FEvals = 5×10^4.

• **WFG5**: For $M = 2$ and $= 4$, the best MOEA is RVEA within FEvals = 8×10^3 and $= 2.1 \times 10^4$ respectively. For $M = 6$, IBEA$_{HD}$ exhibits the best performance on the six-objective WFG5 function.

• **WFG6**: For $M = 2$, IBEA$_{HD}$ and IBEA$_{\epsilon+}$ perform well for the smaller FEvals. The two IBEA variants also outperform the remaining MOEAs for $M = 4$. IBEA$_{HD}$ is also the best performer for $M = 6$.

• **WFG7**: The two IBEA variants show the best anytime performance for $M = 2$. For $M = 4$, IBEA$_{HD}$ and IBEA$_{\epsilon+}$ also perform well, and MOEA/D-DU and EFR-RR catch up them for larger FEvals. For $M = 6$, IBEA$_{\epsilon+}$ still performs well while the performance of IBEA$_{HD}$ is degraded. Moreover, BiGE and SPEA2+SDE are competitive with IBEA$_{\epsilon+}$.

• **WFG8**: MOEA/D-STM outperforms the other MOEAs for $M = 2$ at all times. For $M = 4$, the best MOEA is IBEA$_{HD}$ within FEvals = 2×10^4, and MOEA/D-DU outperforms the other MOEAs. Similarly, for $M = 6$, HypE achieves the highest HV value until FEvals = 1×10^4, and BiGE performs best after that.

• **WFG9**: For $M = 2$, SPEA2+SDE performs relatively well while the evolution of the five MOEAs (GrEA, MOEA/D-STM, EFR-RR, NSGA-II, and I-DBEA) clearly stagnates just after the beginning of the search. For $M = 4$, RVEA exhibits the good performance at all times and is clearly the best MOEA. For $M = 6$, IBEA$_{HD}$ performs well, but SPEA2+SDE and IBEA$_{\epsilon+}$ outperform IBEA$_{HD}$ for larger FEvals.

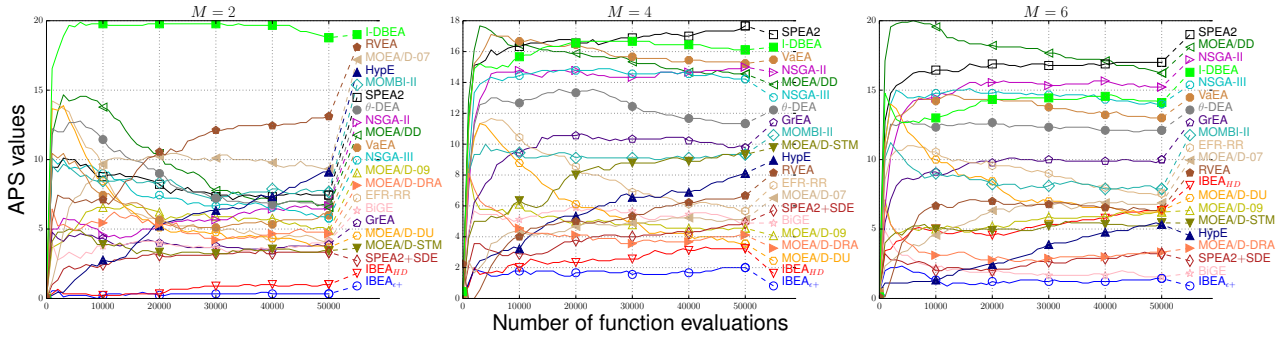


Figure 4: Results for the 21 MOEAs based on the APS for all WFG functions with $M \in \{2, 4, 6\}$ (lower is better).

3.2 Overall performance

While the previous section described the results for each WFG function, here, we describe the overall performance of the 21 MOEAs for all nine WFG functions with each M. For this comparison, we used the average performance score (APS) [2]. Suppose that n algorithms $\{A_1, ..., A_n\}$ are compared for a given problem instance using the HV values obtained in multiple runs. For each $i \in \{1, ..., n\}$ and $j \in \{1, ..., n\} \backslash \{i\}$, let $\delta_{i,j} = 1$, if A_j significantly outperforms A_i using the Wilcoxon rank-sum test with $p < 0.05$, otherwise $\delta_{i,j} = 0$. Then, the performance score $P(A_i)$ is defined as follows: $P(A_i) = \sum_{j \in \{1,...,n\} \backslash \{i\}}^{n} \delta_{i,j}$. $P(A_i)$ represents the number of algorithms outperforming A_i. The APS is the average of the $P(A_i)$ values for all problem instances. In other words, the APS value of A_i represents how good (relatively) the performance of A_i is among the n algorithms for all problem instances (lower is better).

Figure 4 shows the results for the 21 MOEAs for all the nine WFG functions with $M \in \{2, 4, 6\}$. We calculated the APS value for every FEvals $\in \{1, 1 \times 10^3, ..., 5 \times 10^4\}$. The results for $M \in \{3, 5\}$ can be found in Figure S.7 in the supplemental file. Below, we discuss the results for each algorithm category:

• **Seven Pareto dominance-based MOEAs**: Interestingly, in this category, the worst and best performers are SPEA2 and SPEA2+SDE respectively. SPEA2+SDE is an improved version of SPEA2 that incorporates the SDE strategy [21]. Based on these results, the SDE strategy appears to contribute significantly to the outstanding performance of SPEA2+SDE. For almost all M, the convergence speed of NSGA-III is lower than that of NSGA-II, and NSGA-II performs better than NSGA-III for the smaller FEvals, even when $M = 6$. NSGA-III is an improved version of NSGA-II for MaOPs by replacing the crowding distance-based selection with the reference vectors-based niching selection. The newly introduced niching selection appears to make the convergence speed of NSGA-III slow. The two relaxed-dominance based MOEAs (GrEA and θ-DEA) perform similarly to each other for the WFG functions with $M \geq 4$.

• **Nine decomposition-based MOEAs**: For $M = 2$, MOEA/D-STM clearly performs better than the remaining decomposition-based MOEAs. However, within FEvals $= 2 \times 10^3$, the classical MOEA/D-07 outperforms its improved variants. For $M \in \{4, 5, 6\}$, RVEA performs best for much smaller FEvals, followed by MOEA/D-07. For $M = 4$, after some FEvals, MOEA/D-DRA performs very well, but MOEA/D-DU outperforms MOEA/D-DRA after FEvals $= 4.3 \times 10^4$. For $M = 6$, the best performer is also MOEA/D-DRA, followed by MOEA/D-STM and MOEA/D-09. I-DBEA and MOEA/DD clearly perform worse than the other decomposition-based MOEAs including MOEA/D-07. In this category, only I-DBEA and MOEA/DD use the Pareto-dominance for the selection, and it might cause their poor performance. However, with increasing M, the APS value of I-DBEA decreases gradually, and so it is possible that I-DBEA performs well for MaOPs with a large number of objectives.

• **Five indicator-based MOEAs**: In this category, MOMBI-II (an R2-indicator based MOEA) is the worst performer, but it outperforms some MOEAs in the other categories. For $M \in \{2, 3, 4, 5\}$, the best performer is clearly IBEA$_{\epsilon+}$, followed by IBEA$_{HD}$. For $M = 6$, HypE performs very well within FEvals $= 1 \times 10^4$. However, as the search progresses, the APS value of HypE gradually deteriorates for all M. For FEvals $> 10^4$, IBEA$_{\epsilon+}$ outperforms the remaining indicator-based MOEAs, but the APS value of BiGE is gradually improved. Taking into account the results, when M is increasing, BiGE might perform well for a larger FEvals.

• **All 21 MOEAs**: For $M \in \{2, 3\}$, the best performer is IBEA$_{\epsilon+}$, followed by IBEA$_{HD}$. For $M \in \{4, 5, 6\}$, RVEA performs best for much smaller FEvals. For $M = 4$, IBEA$_{\epsilon+}$ still outperforms the remaining MOEAs. For $M \in \{5, 6\}$, among the 21 MOEAs, the best MOEA is HypE for the smaller FEvals, followed by IBEA$_{\epsilon+}$.

4 DISCUSSION

Here, we discuss the results described in Section 3. Unsurprisingly, the results show that the performance rank of the 21 MOEAs depends significantly for (i) the function type, (ii) M, and (iii) the number of function evaluations (FEvals). Thus, determining a best MOEA among the 21 MOEAs is difficult, even though for same problem instance. For example, for the six-objective WFG8 function, HypE performs best within FEvals $= 1 \times 10^4$, followed by BiGE. Keeping in mind this fact, in the following, we *roughly* discuss the performance of the 21 MOEAs based on the APS value (Figure 4).

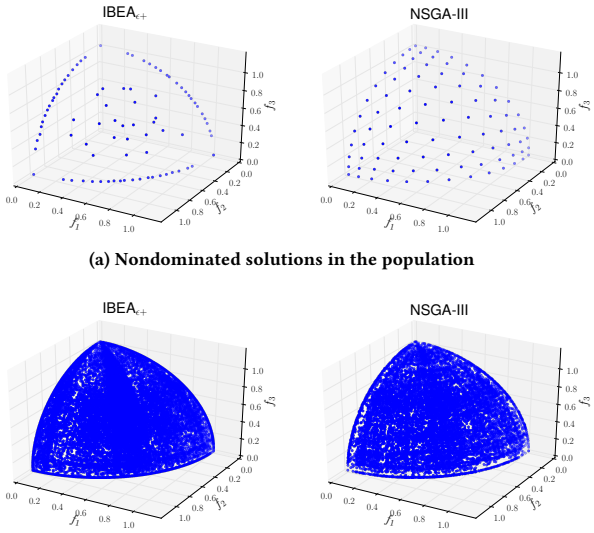
The performance degradation of NSGA-II and SPEA2 with increasing M is described in Section 3. This tendency is consistent with the previous study [30]. On the other hand, some recently proposed MOEAs (e.g., NSGA-III, MOEA/DD, and VaEA) perform poorly for the WFG functions with $M \leq 4$. The reason seems that these MOEAs were originally designed for MaOPs with a large number of objectives. The poor performance of MOEAs that were

specially designed for MaOPs for MOPs with a small number of objectives is consistent with the results of the previous study [23].

IBEA$_{\epsilon+}$, which was proposed in 2004, is the first indicator based MOEA and was not specially designed for MaOPs. The poor performance of IBEA$_{\epsilon+}$ for MaOPs as well as MOPs with $M < 4$ has been reported in numerous articles (e.g., [8, 12, 17, 30, 39]). However, surprisingly, in this paper, the classical IBEA$_{\epsilon+}$ shows the highest performance among the 21 MOEAs, including recently proposed state-of-the-art MOEAs, for the functions with two to six objectives. Moreover, for $M = 6$, RVEA performs best immediately after the start of the search, and HypE is the best MOEA up to FEvals = 1×10^4. HypE was proposed in 2011 and was specially designed for MaOPs with up to 50 objectives, but it is the most classical algorithm among the 21 MOEAs listed in Table 1. HypE is considered as a standard, benchmark algorithm and has frequently been compared to newly proposed MOEAs [19, 21, 33, 34]. In other words, it is widely believed that recently proposed MOEAs perform better than HypE. However, such conventional wisdom is clearly contradicted by the experimental results in this paper. In summary, the benchmarking results presented in Section 3 are not consistent with the results of the previous studies.

One reason for this may be the difference in the evaluation methodology. We evaluated the performance of the 21 MOEAs using the unbounded external archive, whereas previous studies evaluated MOEAs using the final population at the end of the search. In general, the population maintains the good (nondominated) solutions obtained during the search process of an MOEA, but the population size is limited. Thus, MOEAs require a carefully designed environmental selection method so that the solutions in the population are uniformly distributed. According to this policy, recently proposed MOEAs for MaOPs (e.g., NSGA-III, MOEA/DD, and VaEA) were designed. On the other hand, when using the unbounded external archive, MOEAs do not require such well-designed environmental selection method because all nondominated solutions obtained during the search process are automatically maintained in the external archive independent from the population. We believe that the unbounded external archive is beneficial for an MOEA that cannot maintain well-distributed solutions in the population.

For further investigation, we show the distribution of nondominated solutions in the population and the unbounded external archive obtained by IBEA$_{\epsilon+}$ and NSGA-III for the three-objective WFG4 function in Figure 5. As shown in Figure 5(a), while NSGA-III maintains the uniformly distributed solutions in the population, the solutions obtained by IBEA$_{\epsilon+}$ are biased to specific regions (the rim and the center of the PF). The biased distribution of IBEA$_{\epsilon+}$, as shown in Figure 5(b), is also reported in a previous study [8, 17, 30]. On the other hand, as shown in Figure 5(b), both MOEAs successfully maintain the densely distributed solutions covering the entire PF in the unbounded external archive. Moreover, the density of the distribution of the solutions obtained by NSGA-III is not so high as that of IBEA$_{\epsilon+}$. It is possible that IBEA$_{\epsilon+}$ can generate well-distributed solutions but cannot maintain them in the population. When using the unbounded external archive, this issue of IBEA$_{\epsilon+}$ is addressed. In fact, recent studies [17, 31] have reported the promising performance of IBEA$_{\epsilon+}$ variants with a (bounded) external archive for maintaining obtained nondominated solutions.



(a) Nondominated solutions in the population

(b) Nondominated solutions in the unbounded external archive

Figure 5: Distribution of nondominated solutions in (a) the population and (b) the unbounded external archive at FEvals = 5×10^4 for the three-objective WFG4 function. Data for a single run with a median HV value are shown (IBEA$_{\epsilon+}$ and NSGA-III).

Another reason for this might be the setting of the maximum number of function evaluations (FEvals$^{\max}$). As shown in Table 1, we do not know why, but FEvals$^{\max}$ used for comparative studies is increasing yearly. For example, while FEvals$^{\max}$ used in the HypE paper [2], published in 2011, was only 1×10^4, FEvals$^{\max}$ used in recent papers (e.g., NSGA-III [6], MOEA/DD [19], θ-DEA [34], and VaEA [32]) was 5.5×10^5. *The latter is 55 times larger than the former.* Note that some real-world problems require the execution of a very expensive simulation in order to evaluate the solution [15, 26], and thus, in practice, users of MOEAs cannot always set FEvals$^{\max}$ to a large number (e.g., 5.5×10^5). As mentioned above, many articles report the poor performance of HypE [19, 21, 33, 34], but we believe that this is because of the different settings of FEvals$^{\max}$. HypE was designed for optimization for FEvals$^{\max} \leq 1 \times 10^4$, but the settings of FEvals$^{\max}$ used in other previous studies (e.g., [19, 32, 34]) were clearly much larger. In fact, as shown in Figure 4, HypE performs very well within FEvals = 1×10^4 for $M = 6$. Although most previous studies determined the performance rank of MOEAs based only on the end-of-run results as described above, the end-of-run results do not provide sufficient information, as pointed out in [3], and might hide an important fact (i.e., the excellent performance of HypE for small FEvals).

5 CONCLUSION

We have investigated the convergence performance of the 21 MOEAs listed in Table 1 using the unbounded external archive that maintains all nondominated solutions obtained during the search process. Although the performance rank of the 21 MOEAs depend significantly on (i) the function type, (ii) M, and (iii) FEvals, it is significantly inconsistent with the results reported in previous studies. While the performance of some recently proposed MOEAs is

poor, some classical MOEAs perform very well (e.g., the two IBEA variants [39] and HypE [2]) *in this experimental study*. We also discussed why the performance rank obtained in this paper is significantly different from that in previous studies. Although a large number of MOEAs for MaOPs have been proposed, we conclude that their performance and the benchmarking methodology must be carefully reconsidered in light of the experimental results.

We used the (normalized) WFG functions [11] for benchmarking the 21 MOEAs, but it is pointed out that they have some unnatural, exploitable features [3, 14]. For example, MOEAs with systematically, uniformly generated reference and weight vectors (e.g., NSGA-III, RVEA, and MOEA/D variants) can successfully find well-approximated nondominated solutions for the WFG functions because the shape of the distribution of the reference and weight vectors is the same as the shape of the PF [14]. The comparison of MOEAs for other benchmark functions and the original WFG functions is an area for future research. Due to the expensive cost of the exact HV calculation, running an MOEA for MaOPs with $M \geq 7$ for FEvals $> 5 \times 10^4$ is very difficult. In order to address these issues, we will use a hypervolume approximation method [2]. Since the main objective of this paper was to benchmark MOEAs, we only used the HV indicator. Analyzing the performance of MOEAs using various performance indicators is an interesting future direction.

The performance of MOEAs depends on control parameter settings and can be improved by parameter tuning [27]. For example, [28] reveals that a simple parameter tuning significantly improves NSGA-III. Although the results show that NSGA-III with the default parameter setting [6] performs poorly, NSGA-III with tuned parameters may perform well. Its investigation is our future direction.

Due to space constraints, in this paper, the performance of MOEAs was only qualitatively discussed (e.g., MOEA_1 performs better than MOEA_2), not quantitatively discussed (e.g., MOEA_1 is two times faster than MOEA_2) as in [3]. Our future work will quantitatively analyze the performance of the 21 MOEAs by measuring the run-lengths to reach a predefined HV target value as in COCO.

ACKNOWLEDGMENT

This research is supported by the HPCI System Research Project "Research and development of multiobjective design exploration and high-performance computing technologies for design innovation" (Project ID:hp160203).

REFERENCES

[1] M. Asafuddoula, T. Ray, and R. A. Sarker. 2015. A Decomposition-Based Evolutionary Algorithm for Many Objective Optimization. *IEEE TEVC* 19, 3 (2015), 445–460.
[2] J. Bader and E. Zitzler. 2011. HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. *Evol. Comput.* 19, 1 (2011), 45–76.
[3] D. Brockhoff, T. Tran, and N. Hansen. 2015. Benchmarking Numerical Multiobjective Optimizers Revisited. In *GECCO*. 639–646.
[4] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff. 2016. A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization. *IEEE TEVC* 20, 5 (2016), 773–791.
[5] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE TEVC* 6, 2 (2002), 182–197.
[6] K. Deb and H. Jain. 2014. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE TEVC* 18, 4 (2014), 577–601.
[7] J. E. Fieldsend, R. M. Everson, and S. Singh. 2003. Using unconstrained elite archives for multiobjective optimization. *IEEE TEVC* 7, 3 (2003), 305–323.
[8] R. H. Gómez and C. A. Coello Coello. 2015. Improved Metaheuristic Based on the R2 Indicator for Many-Objective Optimization. In *GECCO*. 679–686.
[9] D. Hadka and P. M. Reed. 2012. Diagnostic Assessment of Search Controls and Failure Modes in Many-Objective Evolutionary Optimization. *Evol. Comput.* 20, 3 (2012), 423–452.
[10] T. Hanne. 1999. On the convergence of multiobjective evolutionary algorithms. *EJOR* 117, 3 (1999), 553–564.
[11] S. Huband, P. Hingston, L. Barone, and R. L. While. 2006. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE TEVC* 10, 5 (2006), 477–506.
[12] H. Ishibuchi, N. Akedo, and Y. Nojima. 2015. Behavior of Multiobjective Evolutionary Algorithms on Many-Objective Knapsack Problems. *IEEE TEVC* 19, 2 (2015), 264–283.
[13] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima. 2009. Selecting a small number of representative non-dominated solutions by a hypervolume-based solution selection approach. In *FUZZ-IEEE*. 1609–1614.
[14] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima. 2017. Performance of Decomposition-Based Many-Objective Algorithms Strongly Depends on Pareto Front Shapes. *IEEE TEVC* 21, 2 (2017), 169–190.
[15] Y. Jin. 2011. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evol. Comput.* 1, 2 (2011), 61–70.
[16] B. Li, J. Li, K. Tang, and X. Yao. 2015. Many-Objective Evolutionary Algorithms: A Survey. *ACM C. S.* 48, 1 (2015), 13:1–13:35.
[17] B. Li, K. Tang, J. Li, and X. Yao. 2016. Stochastic Ranking Algorithm for Many-Objective Optimization Based on Multiple Indicators. *IEEE TEVC* 20, 6 (2016), 924–938.
[18] H. Li and Q. Zhang. 2009. Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE TEVC* 13, 2 (2009), 284–302.
[19] K. Li, K. Deb, Q. Zhang, and S. Kwong. 2015. An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition. *IEEE TEVC* 19, 5 (2015), 694–716.
[20] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang. 2014. Stable Matching-Based Selection in Evolutionary Multiobjective Optimization. *IEEE TEVC* 18, 6 (2014), 909–923.
[21] M. Li, S. Yang, and X. Liu. 2014. Shift-Based Density Estimation for Pareto-Based Algorithms in Many-Objective Optimization. *IEEE TEVC* 18, 3 (2014), 348–365.
[22] M. Li, S. Yang, and X. Liu. 2015. Bi-goal evolution for many-objective optimization problems. *Artif. Intell.* 228 (2015), 45–65.
[23] M. Li, S. Yang, X. Liu, and R. Shen. 2013. A Comparative Study on Evolutionary Algorithms for Many-Objective Optimization. In *EMO*. 261–275.
[24] M. López-Ibáñez, J. D. Knowles, and M. Laumanns. 2011. On Sequential Online Archiving of Objective Vectors. In *EMO*. 46–60.
[25] J. Maltese, B. M. Ombuki-Berman, and A. P. Engelbrecht. 2017. A Scalability Study of Many-Objective Optimization Algorithms. *IEEE TEVC* (2017), (in press).
[26] A. J. Nebro, J. José Durillo, C. A. Coello Coello, F. Luna, and E. Alba. 2008. A Study of Convergence Speed in Multi-objective Metaheuristics. In *PPSN*. 763–772.
[27] A. Radulescu, M. López-Ibáñez, and T. Stützle. 2013. Automatically Improving the Anytime Behaviour of Multiobjective Evolutionary Algorithms. In *EMO*. 825–840.
[28] R. Tanabe and A. Oyama. 2017. The Impact of Population Size, Number of Children, and Number of Reference Points on the Performance of NSGA-III. In *EMO*. 606–621.
[29] T. Tusar, D. Brockhoff, N. Hansen, and A. Auger. 2016. COCO: The Bi-objective Black Box Optimization Benchmarking (bbob-biobj) Test Suite. *CoRR* (2016).
[30] T. Wagner, N. Beume, and B. Naujoks. 2007. Pareto-, Aggregation-, and Indicator-Based Methods in Many-Objective Optimization. In *EMO*. 742–756.
[31] H. Wang, L. Jiao, and X. Yao. 2015. Two_Arch2: An Improved Two-Archive Algorithm for Many-Objective Optimization. *IEEE TEVC* 19, 4 (2015), 524–541.
[32] Y. Xiang, Y. Zhou, M. Li, and Z. Chen. 2017. A Vector Angle-Based Evolutionary Algorithm for Unconstrained Many-Objective Optimization. *IEEE TEVC* 21, 1 (2017), 131–152.
[33] S. Yang, M. Li, X. Liu, and J. Zheng. 2013. A Grid-Based Evolutionary Algorithm for Many-Objective Optimization. *IEEE TEVC* 17, 5 (2013), 721–736.
[34] Y. Yuan, H. Xu, B. Wang, and X. Yao. 2016. A New Dominance Relation-Based Evolutionary Algorithm for Many-Objective Optimization. *IEEE TEVC* 20, 1 (2016), 16–37.
[35] Y. Yuan, H. Xu, B. Wang, B. Zhang, and X. Yao. 2016. Balancing Convergence and Diversity in Decomposition-Based Many-Objective Optimizers. *IEEE TEVC* 20, 2 (2016), 180–198.
[36] Q. Zhang and H. Li. 2007. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE TEVC* 11, 6 (2007), 712–731.
[37] Q. Zhang, W. Liu, and H. Li. 2009. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In *IEEE CEC*. 203–208.
[38] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari. 2008. *Multiobjective optimization test instances for the CEC 2009 special session and competition.* Technical Report. Univ. of Essex.
[39] E. Zitzler and S. Künzli. 2004. Indicator-Based Selection in Multiobjective Search. In *PPSN*. 832–842.
[40] E. Zitzler, M. Laumanns, and L. Thiele. 2001. *SPEA2: Improving the Strength Pareto Evolutionary Algorithm.* Technical Report. ETHZ.