Comparing Decomposition-based and Automatically Component-Wise Designed Multi-objective Evolutionary Algorithms

Leonardo C. T. Bezerra, Manuel López-Ibáñez, and Thomas Stützle

IRIDIA, Université libre de Bruxelles (ULB), Brussels, Belgium
{lteonaci,manuel.lopez-ibanez,stuetzle}@ulb.ac.be

Abstract. A main focus of current research on evolutionary multi-objective optimization (EMO) is the study of the effectiveness of EMO algorithms for problems with many objectives. Among the several techniques that have led to the development of more effective algorithms, decomposition and component-wise design have presented particularly good results. But how do they compare? In this work, we conduct a systematic analysis that compares algorithms produced using the MOEA/D decomposition-based framework and the AutoMOEA component-wise design framework. In particular, we identify a version of MOEA/D that outperforms the best known MOEA/D algorithm for several scenarios and confirms the effectiveness of decomposition on problems with three objectives. However, when we consider problems with five objectives, we show that MOEA/D is unable to outperform SMS-EMOA, being often outperformed by it. Conversely, automatically designed AutoMOEAs display competitive performance on three-objective problems, and the best and most robust performance among all algorithms considered for problems with five objectives.

Keywords: Multi-objective Optimization, Evolutionary Algorithms, Decomposition, Component-wise Design, Automatic Configuration

1 Introduction

Over the past years, research on evolutionary multi-objective optimization (EMO) has focused on the development of effective algorithms for many-objective optimization, as evidenced by the number of recent publications on this topic [20]. Many are the reasons that stirred this interest. First, Pareto dominance becomes a weak relation as the number of objectives increases. As a result, the number of feasible solutions that are incomparable becomes too large to give algorithms that rely on Pareto dominance enough convergence pressure [1, 13]. Second, the number of applications of many-objective optimization has demanded more effective algorithms for this scenario. In particular, many real-world engineering problems can be modeled as many-objective optimization problems, where constraints are considered objectives [10]. Finally, the number of solutions needed to accurately approximate Pareto fronts grows exponentially with the number of objectives [13].

Among the many different search techniques proposed for improving the effectiveness of many-objective algorithms, indicator- and decomposition-based approaches have shown very good results [16,23,28]. In particular, decomposition is an old search paradigm originally applied by EMO already two decades ago [11], which has recently regained prominence with the proposal of the MOEA/D framework [26]. In this search paradigm, the original multi-objective problem is decomposed into simpler, single-objective subproblems by means of scalarizations. Originally, this approach was not pursued by the EMO community in general, particularly because decomposition-based algorithms may waste function evaluations searching in directions that do not present Pareto-optimal solutions. However, the best-known MOEA/D algorithm [27], which won the IEEE CEC 2009 competition on multi-objective optimization [28], uses a dynamic resource allocation strategy to overcome this drawback. Unfortunately, no performance assessment concerning this version of MOEA/D has been reported so far using large and representative benchmark sets on which other EMO algorithms have typically been tested.

More recently, another promising paradigm for devising effective EMO algorithms was proposed, namely the *component-wise design* [5]. This paradigm proposes reusing algorithmic components from well-known EMO algorithms in novel ways, thus leading to new designs. Concretely, given a flexible template, algorithms can be created by plugging in a set of desired components. In the original proposal, authors have automatically designed several algorithms using the component-wise design framework for continuous and combinatorial optimization [5]. We call the algorithms resulting from this automatic configuration process AutoMOEAs in what follows. The AutoMOEAs devised for many-objective optimization problems have shown competitive performance when compared to several Pareto- and indicator-based algorithms on a large set of three- and five-objective benchmark test problems, being able to match (and often surpass) the performance of the original algorithms from which the AutoMOEAs components were gathered.

In its current stage, the component-wise AutoMOEA framework only contains components from Pareto-based and indicator-based algorithms. However, given the interesting results the AutoMOEAs were already able to achieve, in this work we conduct a systematic performance assessment to understand how they compare to the effective decomposition-based approach. In particular, we consider several MOEA/D algorithms, including the version that won the IEEE CEC competition, and the AutoMOEAs designed in the original component-wise design paper [5]. To make this analysis more representative, we also include two effective indicator-based algorithms, SMS-EMOA [3] and IBEA [29], as well as the two best known Pareto-based algorithms, NSGA-II [8] and SPEA2 [30]. Furthermore, we consider a wide benchmark test set comprising the DLTZ [9] and WFG [12] benchmarks with three and five objectives, as well as several different problem sizes. In all scenarios considered, algorithms are properly tuned to perform at their best.

The investigation we conduct in this paper produces many interesting insights. First, we show that the MOEA/D algorithm that won the IEEE CEC competition is unable to outperform some of the other algorithms considered. Particularly for the five-objective WFG set, this version is clearly outperformed by SMS-EMOA. Second, we show that a straightforward alternative version of MOEA/D is able to consistently outperform the version used in the IEEE CEC competition, and also outperforms all other algorithms for the WFG set with three-objective problems. Nevertheless, SMS-EMOA still presents better results than this improved MOEA/D version on the five-objective WFG set. Finally, we show that the AutoMOEAs match the best-performing algorithms on all WFG scenarios, and outperform them on the 5-objective DTLZ set.

The remainder of this paper is organized as follows. We review the EMO search paradigms we consider in Section 2. Next, we describe the decomposition-based and the component-wise design paradigms in Sections 3 and 4, respectively. In particular, we detail the designs of the algorithms that are used in the experimental evaluation. The experimental setup is given in Section 5 followed by the presentation and discussion of the results in Section 6. We conclude and discuss future work in Section 7.

2 Search paradigms in multi-objective optimization

In this section, we briefly review the search paradigms found in the EMO literature that we use in this performance assessment. While this review is not exhaustive, the algorithms we highlight in each of the paradigms are the most representative and most effective in the literature for their corresponding paradigm [5]. In particular, this represents a major improvement over other recent experimental analysis conducted on the effectiveness of EMO algorithms for many-objective optimization [16], which have used representative but not the most effective algorithms for each paradigm.

Pareto-based approaches. Early EMO algorithms tried to find approximation fronts as diverse and close to the optimal front as possible mostly thanks to the convergence pressure provided by Pareto dominance. Among these, we highlight **NSGA-II** [8] and **SPEA2** [30]. Although these algorithms use different mechanisms, both are based on pushing the population towards convergence by favoring nondominated solutions, while simultaneously trying to maintain a population as diverse as possible. In EMO, diversity is a measure of the different trade-offs among the objectives considered, rather than an attempt to prevent stagnation as in the single-objective optimization literature. For most of the test problems considered then, these Pareto-based approaches were able to perform quite effectively [9,31]. However, the majority of these test cases considered two or three objectives only.

Indicator-based approaches. As the performance assessment of EMO algorithms reached a mature stage, researchers observed that quality indicators could be used within algorithms to direct their search in a Pareto-compliant way. More importantly, the convergence pressure provided by these quality indicators does not weaken as the number of objective increases. Within this paradigm, we highlight **IBEA** [29] and **SMS-EMOA** [3]. IBEA uses a binary quality indicator to compare solutions. In particular, the most effective version of IBEA uses the binary ϵ-indicator [5, 29]. By contrast, SMS-EMOA uses the exclusive hypervolume contribution to direct its search. Although theoretical complexity analysis shows that this indicator can become exponentially costly as the number of objectives increases [2], empirical analysis has shown that recent efficient algorithms [2, 24, 25] give a runtime reasonable for practical purposes [19].

Decomposition-based approaches. Decomposition is one of the earliest search paradigms in EMO [11]. It is based on the principle that tackling single-objective subproblems is an easier task than facing the original multi-objective problem. However, in continuous optimization, decomposition was initially considered inefficient in comparison with other EMO algorithms, mostly due to the number of function evaluations that it may waste while searching along directions that do not present Pareto-optimal solutions. More recently, the **MOEA/D** [26] framework stirred the research on this paradigm, primarily when a variant of MOEA/D won the IEEE CEC 2009 competition on multi-objective optimization [27,28]. This variant improves over the original MOEA/D by using *dynamic resource allocation*, i.e., favoring search directions where the algorithm is progressing better. However, no performance assessment concerning this version of MOEA/D has been reported so far using a large and representative benchmark set where other EMO algorithms are typically tested.

Component-wise design. Proposed as a comprehensive design paradigm, the component-wise design aims at gathering the potential of the different existing EMO search paradigms. In its current version, the AutoMOEA framework [5] provides a flexible template and a collection of algorithmic components comprising both Pareto-based and indicator-based paradigms. Given an application, designers can then tailor algorithms to their target application. To demonstrate the potential of the component-wise design, the authors used an automatic configuration tool to automatically design various **AutoMOEAs** for the most-used continuous benchmarks [5], as well as for several combinatorial problems [6]. In particular, the AutoMOEAs designed for five-objective problems presented outstanding performance, matching the best-performing algorithms for the WFG benchmark, and outperforming all of them for the DTLZ benchmark [5].

In the following sections, we detail the specific variants of MOEA/D and AutoMOEA that are the focus of our performance assessment.

3 MOEA/D

Although it may be understood as an algorithmic framework, MOEA/D was originally proposed as a stand-alone algorithm [26]. Later, improved versions were also proposed as stand-alone algorithms [15, 27]. For this reason, from now on we always refer to the different MOEA/D algorithms rather than to instantiations of a more general framework.

The common underlying structure shared by all MOEA/D algorithms considered in this work is the structure of the original MOEA/D algorithm, to which we will refer simply as **MOEA/D**. MOEA/D simultaneously explores the different search directions defined by the weight vectors of scalarization methods such as weighted linear sums or Tchebychev utility functions. Another particular feature presented by MOEA/D is the selection mechanism, namely, variation is applied to randomly selected parents from local neighborhoods, built for each search direction. Although the algorithm maintains a single global population, these local neighborhoods are meant to help the algorithm progress along the search directions employed.

A couple of years later, a new version of MOEA/D was proposed [27]. **MOEA/D$_{\text{DRA-DE}}$**, as we will call it, uses *dynamic resource allocation* (DRA) and the differential evolution (DE) variation operator. The DRA strategy works as follows. Initially, each of the N weight vectors is given the same utility value. At each iteration, MOEA/D$_{\text{DRA-DE}}$ selects a subset $\frac{N}{\nu}$ to explore via tournament selection based on the utility values of the weights. Once the weights have been selected, DE variation is applied to each search direction. In this version, however, a parameter δ regulates whether the target vector will be randomly chosen from the local neighborhood or from the whole population. Finally, a subset of the selection set (local neighborhood or population) is used to update the search reference point for the current weight. The size of this subset is regulated by an additional parameter ϕ. Every 50 iterations, the utility values of the weights are recomputed.

The differences between MOEA/D and MOEA/D$_{\text{DRA-DE}}$ are substantial, particularly given the number of parameters used to define the DRA strategy. In addition, since MOEA/D$_{\text{DRA-DE}}$ uses a different variation operator from all the other EMO algorithms we consider here, it is not really possible to assess whether improvements over the original MOEA/D (and other algorithms) could be explained solely by DRA, by DE, or by the combination of both components. For this reason, we consider an alternative version of MOEA/D$_{\text{DRA-DE}}$ that we call **MOEA/D$_{\text{DRA-SBX}}$**. The only difference between MOEA/D$_{\text{DRA-DE}}$ and MOEA/D$_{\text{DRA-SBX}}$ is how a trial vector (or offspring) is generated, that is, MOEA/D$_{\text{DRA-SBX}}$ uses the SBX crossover operator, instead of DE variation, to produce a single solution at a time.

Below we summarize all the MOEA/D algorithms we consider in this work. All versions use Tchebychev utility functions to search the objective space.

MOEA/D: original MOEA/D algorithm [26], with SBX crossover and no dynamic resource allocation.

MOEA/D$_{\text{DRA-DE}}$: MOEA/D algorithm used in the 2009 IEEE CEC competition [27]. This algorithm uses DE variation and dynamic resource allocation.

MOEA/D$_{\text{DRA-SBX}}$: alternative version of the MOEA/D algorithm used in the 2009 IEEE CEC competition [27]. This algorithm uses the SBX crossover operator and dynamic resource allocation.

4 AutoMOEA

The AutoMOEA component-wise design framework explores the concept that existing algorithmic components can lead to more effective designs than existing stand-alone algorithms if components are combined in more effective ways. This idea has been used in other multi-objective metaheuristics and led to the development of effective algorithms that significantly outperformed existing approaches from which algorithmic components were gathered [4, 18]. Concerning EMO, the AutoMOEA framework is based on a template where components can be selected from existing Pareto- and indicator-based approaches.

The core structure of AutoMOEA algorithms are no different from traditional evolutionary algorithms. Starting from an initial population, select a mating pool of solutions from the population, apply variation operators to this pool, and replace solutions from the old population with these new offspring. AutoMOEA algorithms may also use an external bounded-size archive to store nondominated solutions, which is updated at the end of each iteration. The flexibility of the template relies heavily on the general preference relations used by the main components, namely mating, environmental, and external archive selection. For assembling a preference relation, AutoMOEA uses a tuple comprising a dominance-based set-partitioning, an indicator-based refinement, and a diversity metric. Concretely, solutions are partitioned in dominance-equivalent classes using a set-partitioning method, such as the ones originally proposed by Pareto-based approaches like NSGA-II or SPEA2. Since these partitions may contain incomparable solutions, indicator-based refinement relations are used, as in indicator-based approaches such as IBEA or SMS-EMOA. Finally, if solutions are still incomparable, diversity metrics are employed to ensure the population represents different trade-offs between the objectives.

Two other design concepts behind AutoMOEA provide additional flexibility to this framework. First, each of the main components may use a different preference relation, as proposed by more recent indicator-based algorithms like SMS-EMOA. Second, an AutoMOEA algorithm may use an internal bounded-size archive instead of a fixed-size population to increase the convergence pressure of the algorithms when required, as in algorithms such as PAES [14].

Below we summarize all the AutoMOEA algorithms we consider in this work. These algorithms are instantiations of the general AutoMOEA framework and have been automatically designed in [5] for the DTLZ and WFG benchmark sets with three and five objectives. The main components used by these algorithms are given in Table 1, where BuildMatingPool is the mating selection procedure, Replacement is the environmental selection procedure, and Replacement_{Ext} is the external archive truncation method.

$\text{AutoMOEA}_{\text{D3}}$ is an instantiation of AutoMOEA for 3-objective DTLZ problems. This algorithm uses a fixed-size population, random mating selection, and steady-state environmental selection based on dominance depth-rank and the binary ϵ-indicator (I_ϵ). In addition, $\text{AutoMOEA}_{\text{D3}}$ uses an external

Table 1. Algorithm components of the AutoMOEAs used in this work. From top to bottom, AutoMOEA$_{D3}$, AutoMOEA$_{D5}$, AutoMOEA$_{W3}$, and AutoMOEA$_{W5}$.

BuildMatingPool				Replacement				Replacement$_{Ext}$	
Selection	SetPart	Quality	Diversity	SetPart	Quality	Diversity	Removal	Quality	Diversity
random	—	—	—	depth-rank	I_ϵ	—	—	I_H^1	sharing
tourn.	count	I_H^1	crowding	depth	I_ϵ	crowding	sequential	I_H^1	crowding
random	—	—	—	strength	I_H^h	kNN	—	I_H^1	kNN
tourn.	—	I_H^1	crowding	—	I_H^1	sharing	sequential	I_ϵ	kNN

archive based on exclusive hypervolume contribution (I_H^1) and fitness sharing diversity.

AutoMOEA$_{D5}$ is an instantiation of AutoMOEA for 5-objective DTLZ problems. This algorithm uses a fixed-size population, mating selection based on deterministic tournament, and a mating preference relation that comprises dominance count set-partitioning, the exclusive hypervolume contribution as refinement, and crowding diversity. The environmental selection used by AutoMOEA$_{D5}$ is based on a preference relation that comprises dominance depth set-partitioning, the binary ϵ-indicator, crowding diversity, and sequential solution removal. In addition, AutoMOEA$_{D3}$ uses an external archive based on the exclusive hypervolume contribution and crowding diversity.

AutoMOEA$_{W3}$ is an instantiation of AutoMOEA for 3-objective WFG problems. This algorithm uses a fixed-size population, random mating selection, and steady-state environmental selection based on dominance strength, the shared hypervolume contribution (I_H^h), and nearest neighbor diversity. In addition, AutoMOEA$_{W3}$ uses an external archive based on the exclusive hypervolume contribution and nearest neighbor diversity.

AutoMOEA$_{W5}$ is an instantiation of AutoMOEA for 5-objective WFG problems. This algorithm uses a bounded internal archive, mating selection based on deterministic tournament, and a mating preference relation that comprises the exclusive hypervolume contribution and crowding diversity. The environmental selection used by AutoMOEA$_{W5}$ is based on a preference relation that comprises the exclusive hypervolume contribution, fitness sharing diversity, and sequential solution removal. In addition, AutoMOEA$_{D3}$ uses an external archive based on the binary ϵ-indicator and nearest neighbor diversity.

In the next section, we present the experimental setup we use in this work for the performance comparison of the different EMO paradigms.

5 Experimental setup

The experimental setup we use in this work is the same used in the original component-wise design [5]. Since we use the same experimental setup, we use

Table 2. Parameter space for tuning all MOEA/D algorithms.

Parameter	ρ	MOEAD$_{\text{DRA}}$ δ	ϕ	t_{size}	ν	η_m
Domain	$[0.1, 1]$	$[0, 1]$	$[0.01, 1]$	$\{1, 2, \ldots, 20\}$	$\{2, 3, \ldots, 10\}$	$\{1, \ldots, 50\}$

the same tuned settings for all algorithms except for the MOEA/D variants, which were not considered in the original paper and we tune them here. The benchmark sets we use are the DTLZ [9] and WFG [12] functions (DTLZ1–7 and WFG1–9), with three and five objectives. Concerning the number of variables n, we consider problems with $n \in \{20, 21, \ldots, 60\} \setminus n_{\text{testing}}$ for tuning, and $n_{\text{testing}} = \{30, 40, 50\}$ for testing. For both testing and tuning, algorithms are given 10 000 function evaluations per run, and all experiments are run on a single core of Intel Xeon E5410 CPUs, running at 2.33GHz with 6MB of cache size under Cluster Rocks Linux version 6.0/CentOS 6.3. For each problem instance, the approximation fronts produced by the algorithms are normalized to the range $[1, 2]$ to prevent issues due to dissimilar domains. Finally, we compute the hypervolume for each front using $r_i = 2.1$, $i = 1, 2, \ldots, M$ as reference point, where M is the number of objectives considered.

We tune the MOEA/D algorithms using irace [17] and the hypervolume as the quality measure, following the same procedure used for tuning all the other algorithms. In particular, for each tuning scenario, irace stops after 20 000 runs. For the original MOEA/D, the population size is given by the number of divisions in the objective space $N_{\text{divisions}}$ and the number of objectives. Since the population size can grow exponentially with the number of objectives, we use different ranges for each scenario: for the 3-objective problems, we use $N_{\text{divisions}} \in \{1, 2, \ldots, 30\}$, whereas for 5-objective problems we use $N_{\text{divisions}} \in \{1, 2, \ldots, 10\}$. For both MOEA/D$_{\text{DRA}}$ algorithms, the population size can be freely selected, and hence we use $\mu \in \{100, 200, \ldots, 500\}$. The remaining parameters tuned for the MOEA/D algorithms are given in Table 2. In particular, parameter ρ controls the size of the local neighborhoods $(\rho \cdot \mu)$, parameter t_{size} is the size of the tournament used by the DRA strategy, and parameter η_m is the distribution index used by the polynomial mutation operator. For more details about any of the remaining parameters, we refer to Section 3 and to the original MOEA/D papers [26,27]. Finally, for the algorithms that use the SBX crossover, the tuning range of parameters p_c (crossover probability) and η_c (the distribution index) is the same used by all other algorithms. By contrast, when DE variation is used, there are two other parameters: the crossover probability $CR \in [0, 1]$ and the scale factor $F \in [0.1, 2]$. For brevity, the tuned settings selected for the MOEA/D algorithms are provided as supplementary material [7].

To compare algorithms, we run each algorithm 25 times and evaluate them based on the relative hypervolume of the approximation fronts they produce w.r.t. the actual Pareto optimal fronts. More precisely, we use the same Pareto

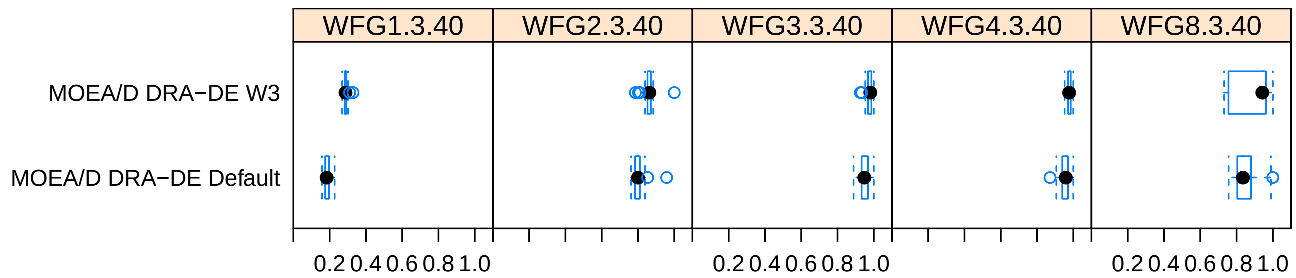

Fig. 1. Boxplots of the relative hypervolume achieved by MOEA/$D_{\text{DRA-DE}}$ using default or tuned parameter settings on selected 3-objective 40-variable WFG problems.

fronts used by [5]. Given an approximation front A and the Pareto front for a problem instance P, the relative hypervolume of A equals $I_H(A)/I_H(P)$. A relative hypervolume of 1.0 means the algorithm was able to perfectly approximate the Pareto front for the problem considered. Algorithms are then compared based on boxplots of these relative hypervolumes. To draw overall conclusions, we aggregate results through rank sums and test for significant differences using Friedman's test with 99% confidence level. Since we generate a large set of results, we only discuss the most representative ones here. The full set of results is provided as supplementary material [7].

6 Results and discussion

Before proceeding to the actual comparison between the different search paradigms, we start this section with boxplots on selected 3-objective WFG problems (Fig. 1) to demonstrate the effect of the tuning on the performance of MOEA/$D_{\text{DRA-DE}}$, which can also be observed for other MOEA/D algorithms. In particular, the label "W3" indicates that the MOEA/$D_{\text{DRA-DE}}$ algorithm has been tuned for 3-objective WFG problems. This notation is also used in all remaining boxplots to make it explicit that all algorithms have been properly tuned for the scenarios in which they are compared. Concerning the parameter settings used by MOEA/$D_{\text{DRA-DE}}$, the most interesting remark is the very low δ values for both 3-objective benchmarks ($\delta \leq 0.1$), which indicate that local neighborhoods are rarely used by this algorithm in these scenarios. By contrast, for all DTLZ scenarios MOEA/$D_{\text{DRA-SBX}}$ uses extremely high δ values ($\delta \geq 0.93$), and MOEA/D uses extremely large niche sizes ($\rho \geq 0.97$). Altogether, these settings indicate that the effectiveness of the local neighborhood component is tightly related to the benchmark, number of objectives, and variation operator considered. We then proceed to an analysis per benchmark.

6.1 Analysis on the DTLZ benchmark set

The performance comparison of all algorithms on the DTLZ benchmark set is given in Fig. 2. Results for 3-objective problems are shown on the top row, while the bottom row depicts the performance assessment on the 5-objective problems.

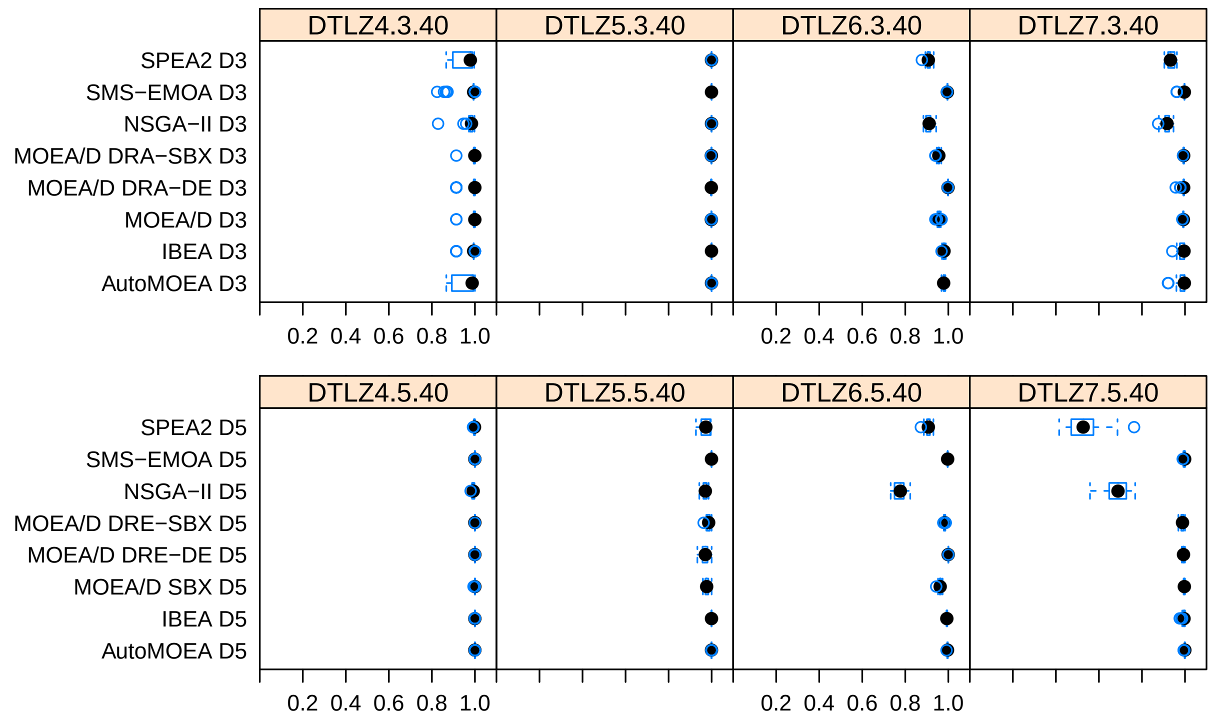

Fig. 2. Boxplots of the relative hypervolume achieved by all algorithms on selected DTLZ problems with 40 variables. Top: 3 objectives. Bottom: 5 objectives.

These results confirm insights previously identified in the literature [5]. First, the overall difficulty of this benchmark is low, as reflected by the very high relative hypervolumes achieved by most algorithms. In fact, we do not show the plots for DTLZ1–3 because they are identical to the plots shown for the 3-objective DTLZ5 problem, i.e., all algorithms are able to well approximate the Pareto optimal fronts used as reference. Second, the Pareto-based approaches are the ones that present worst-quality results among all algorithms. Although one can notice this already for 3-objective problems, it becomes far more evident when 5-objective problems are considered.

Regarding the performance of the remaining algorithms on the 3-objective problems, the only problem that actually poses difficulties for some algorithms is DTLZ6, where MOEA/$D_{\text{DRA-DE}}$ and SMS-EMOA presents results better than all other algorithms considered. The pattern observed in the boxplots is confirmed by the rank sum analysis given in Table 3. For the 3-objective DTLZ set, no difference can be observed between the four top-performing algorithms. Interestingly, MOEA/$D_{\text{DRA-DE}}$ ranks sixth, alongside IBEA. Concerning the 5-objective problems, $\text{AutoMOEA}_{\text{D5}}$, SMS-EMOA, and IBEA appear to always accurately approximate the Pareto fronts (Fig. 2, bottom), while the MOEA/D versions sometimes face difficulties, such as for problems DTLZ5 and DTLZ6. However, when we consider the rankings over the whole 5-objective DTLZ set (Table 3), $\text{AutoMOEA}_{\text{D5}}$ ranks first with much lower ranks than all other algorithms. No significant difference is observed between MOEA/$D_{\text{DRA-DE}}$, MOEA/D, and SMS-EMOA, nor between MOEA/$D_{\text{DRA-SBX}}$, and IBEA. As expected, the Pareto-based algorithms rank last.

Table 3. Rank sum analysis depicting overall performance on all scenarios. The best ranked algorithms are shown on top. Algorithms in boldface present rank sums not significantly worse than the best ranked algorithm. Algorithms within the same block are not significantly different, in terms of ranking, to the first algorithm of the same block.

3-obj DTLZ	5-obj DTLZ	3-obj WFG	5-obj WFG
SMS-EMOA$_{\text{D3}}$	**AutoMOEA$_{\text{D5}}$**	**MOEA/D$_{\text{DRA-SBXW3}}$**	**AutoMOEA$_{\text{W5}}$**
MOEA/D$_{\text{D3}}$	MOEA/D$_{\text{DRA-DED5}}$	MOEA/D$_{\text{DRA-DEW3}}$	**SMS-EMOA$_{\text{W5}}$**
AutoMOEA$_{\text{D3}}$	MOEA/D$_{\text{D5}}$	AutoMOEA$_{\text{W3}}$	MOEA/D$_{\text{DRA-SBXW5}}$
MOEA/D$_{\text{DRA-SBXD3}}$	SMS-EMOA$_{\text{D5}}$	SPEA2$_{\text{W3}}$	MOEA/D$_{\text{DRA-DEW5}}$
IBEA$_{\text{D3}}$	MOEA/D$_{\text{DRA-SBXD5}}$	SMS-EMOA$_{\text{W3}}$	MOEA/D$_{\text{W5}}$
MOEA/D$_{\text{DRA-DED3}}$	IBEA$_{\text{D5}}$	IBEA$_{\text{W3}}$	IBEA$_{\text{W5}}$
SPEA2$_{\text{D3}}$	NSGA-II$_{\text{D5}}$	NSGA-II$_{\text{W3}}$	SPEA2$_{\text{W5}}$
NSGA-II$_{\text{D3}}$	SPEA2$_{\text{D5}}$	MOEA/D$_{\text{W3}}$	NSGA-II$_{\text{W5}}$

6.2 Analysis on the WFG benchmark set

Results for the WFG benchmark set are much more heterogeneous, confirming that this benchmark set is far more difficult for EMO algorithms than the DTLZ one. The performance comparison for 3-objective problems is given in Fig. 3. In fact, it is difficult to even find patterns on the performance of the algorithms. Given any pair of algorithms, one cannot visually identify the best approach when all problems are considered, which confirms that most algorithms perform very similarly in all problems. For this reason, we proceed to the rank sum analysis, also given in Table 3. Surprisingly, the algorithm that achieves lowest rank sums is MOEA/D$_{\text{DRA-SBX}}$, outperforming MOEA/D$_{\text{DRA-DE}}$ again for a 3-objective benchmark set. Ranking second, MOEA/D$_{\text{DRA-DE}}$, AutoMOEA$_{\text{W3}}$, and SPEA2 show equivalent rank sums. This indicates that some components from SPEA2 are indeed particularly effective for this benchmark, since AutoMOEA$_{\text{W3}}$ heavily relies on SPEA2 components. The indicator-based approaches come right after, and MOEA/D ranks last this time. This reinforces the contribution of our experimental analysis, since previous results led to the conclusion that MOEA/D was particularly effective for 3-objective benchmarks in general [16].

The performance assessment for the 5-objective WFG benchmark is given in Fig. 4. Again it is difficult to make an overall analysis since the results vary per problem instance, nonetheless both AutoMOEA$_{\text{W5}}$ and SMS-EMOA perform consistently well. Concerning the MOEA/D algorithms, all present very similar performances. The rank sum analysis (Table 3) confirms these observations: AutoMOEA$_{\text{W5}}$ and SMS-EMOA present nearly identical rank sums, outperforming all MOEA/D algorithms, which present equivalent rank sums.

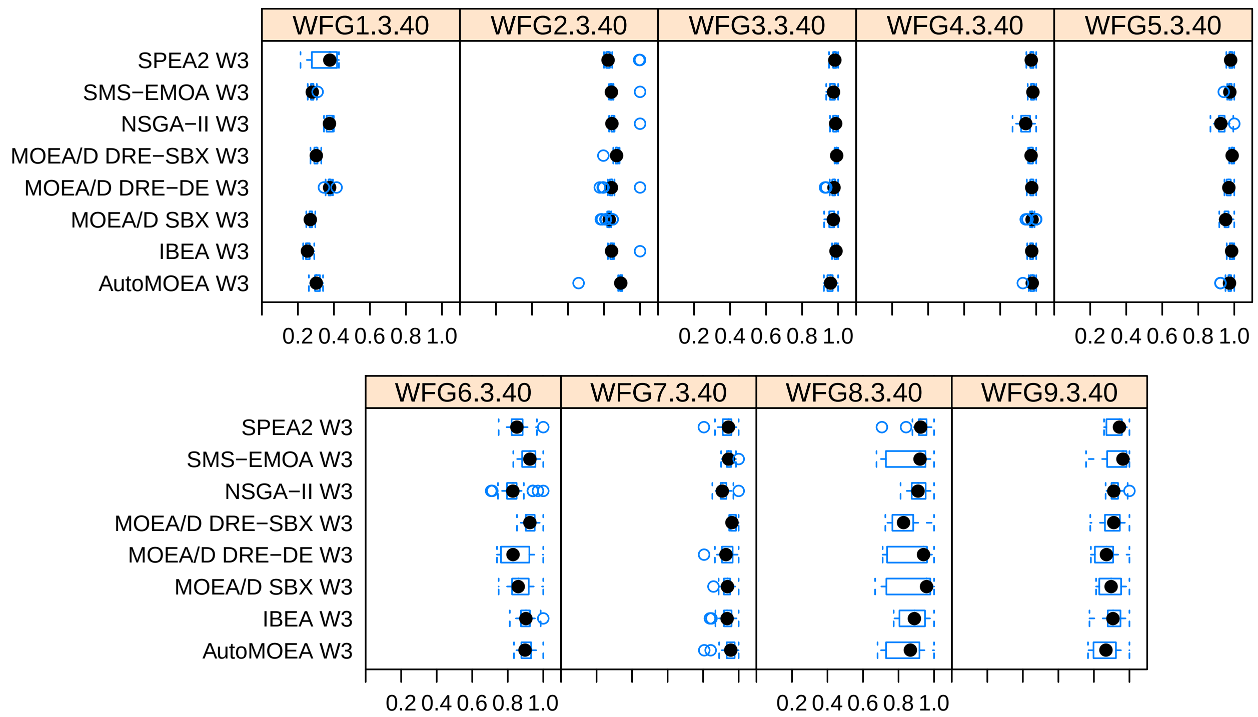

Fig. 3. Boxplots of the relative hypervolume achieved by all algorithms on WFG problems with 40 variables and 3 objectives.

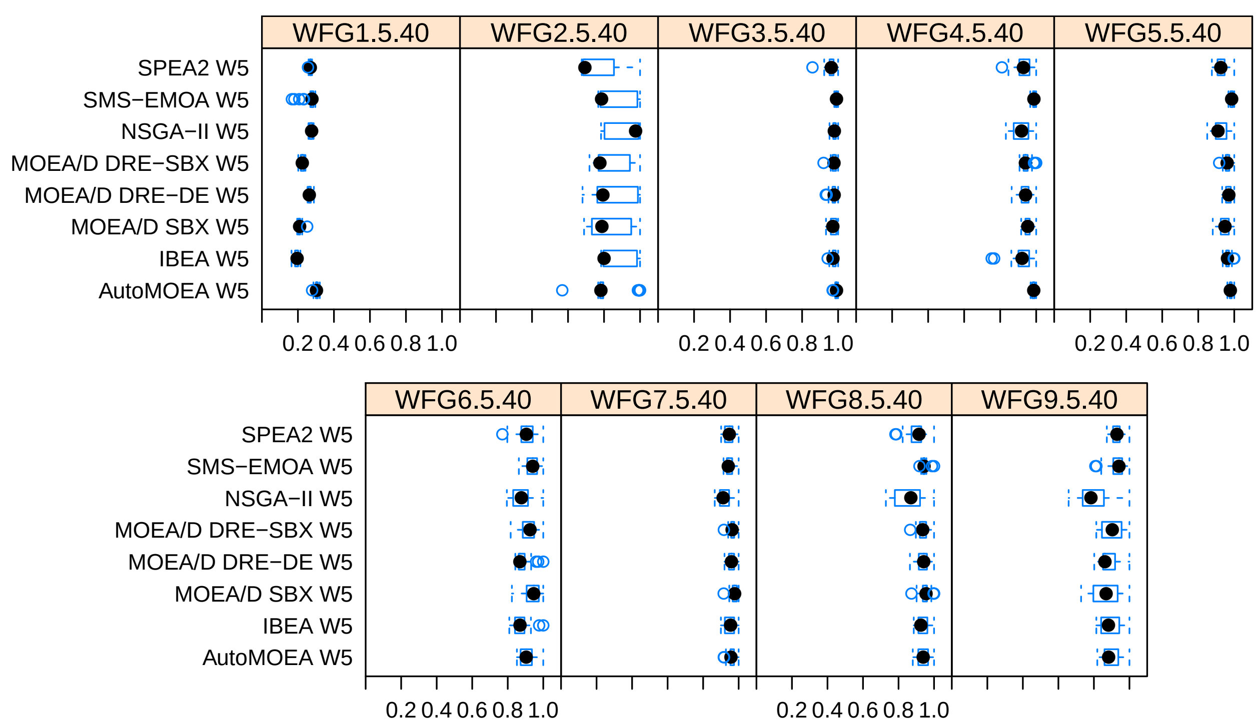

Fig. 4. Boxplots of the relative hypervolume achieved by all algorithms on WFG problems with 40 variables and 5 objectives.

7 Conclusions

The decomposition-based EMO paradigm has drawn a strong interest from the EMO community due to the possibility of devising more effective algorithms, particularly for many-objective optimization problems. In this paper, we have shown that, considering the most used benchmark sets from the EMO literature, MOEA/D is competitive or superior to other state-of-the-art EMO algorithms only on scenarios with three objectives. We have also shown that, neither the dynamic resource allocation (DRA) nor the differential evolution (DE) operator adopted by the IEEE CEC 2009 competition MOEA/D algorithm (MOEA/$D_{\text{DRA-DE}}$) actually led to improvements over the original MOEA/D version for most of the scenarios considered in this paper. The only scenario that proved an exception to these two conclusions is the WFG benchmark with 3-objective problems. In this particular scenario, MOEA/$D_{\text{DRA-DE}}$ performed very competitively, but since it was outperformed by MOEA/$D_{\text{DRA-SBX}}$ (the same algorithm using the SBX crossover operator instead of DE variation), we see that the component that actually leads to this significant performance improvement over the original MOEA/D is the DRA. Moreover, for most of the scenarios considered, the DE operator did not improve the performance of the original MOEA/D. Since related work has shown that DE variation can often improve the performance of other algorithms [22], we hypothesize that the interaction between the decomposition approach and DE is responsible for this.

Concerning the effectiveness of the recently proposed AutoMOEAs, we see that these algorithms are generally able to match the performance of indicator- and decomposition-based algorithms for scenarios with three objectives, and to outperform most of them when five objectives are considered. The high performance of the AutoMOEAs designed for 3-objective problems is in fact impressive, since extensive research has been conducted on this type of application scenario, leading to very effective human-designed algorithms. Achieving the same performance with automatically designed algorithms is remarkable. Even more exciting, the AutoMOEAs designed for 5-objective scenarios show a very robust and competitive performance. For the DTLZ benchmark, the difference in the rank sums between the $\text{AutoMOEA}_{\text{D5}}$ and the best performing indicator- and decomposition-based algorithms is such that it indicates that $\text{AutoMOEA}_{\text{D5}}$ consistently produces better approximation fronts than the others. For the WFG benchmark, $\text{AutoMOEA}_{\text{W5}}$ matches the performance of SMS-EMOA, outperforming all MOEA/D versions. These results indicate the potential of the component-wise design approach, since we have attained this performance level by combining only two among the different effective EMO search paradigms.

Although the results for the current stage of the component-wise design approach are already quite convincing, further research efforts in this direction could potentially improve even further the performance of newly designed AutoMOEAs. Besides including algorithmic components proposed for algorithms from other paradigms, more effective automatic configuration tools (or longer tuning budgets) could lead to even more effective designs. However, it is also imperative to develop this research field towards practical application requirements. For instance, since many real-world problems are computationally demanding, the number of function evaluations desired might not allow offline tuning. One possible way to work around this problem is to devise several automatic designs for different benchmarks and learn problem features that could help understand better the effectiveness of individual algorithmic components.

Acknowledgments. The research presented in this paper has received funding from the COMEX project within the Interuniversity Attraction Poles Programme of the Belgian Science Policy Office. L.C.T. Bezerra, M. López-Ibáñez and T. Stützle acknowledge support from the Belgian F.R.S.-FNRS, of which they are a FRIA doctoral fellow, a postdoctoral researcher and a senior research associate, respectively.

References

1. Aguirre, H.: Advances on many-objective evolutionary optimization. In: Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation. pp. 641–666. GECCO '13 Companion, ACM (2013)
2. Beume, N., Fonseca, C.M., López-Ibáñez, M., Paquete, L., Vahrenhold, J.: On the complexity of computing the hypervolume indicator. IEEE Trans. Evol. Comput. 13(5), 1075–1082 (2009)
3. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. Eur. J. Oper. Res. 181(3), 1653–1669 (2007)
4. Bezerra, L.C.T., López-Ibáñez, M., Stützle, T.: Automatic generation of multi-objective ACO algorithms for the biobjective knapsack problem. In: Dorigo, M., et al. (eds.) ANTS 2012, LNCS, vol. 7461, pp. 37–48. Springer (2012)
5. Bezerra, L.C.T., López-Ibáñez, M., Stützle, T.: Automatic component-wise design of multi-objective evolutionary algorithms. Tech. Rep. TR/IRIDIA/2014-012, IRIDIA, Université Libre de Bruxelles, Belgium, Brussels, Belgium (2014)
6. Bezerra, L.C.T., López-Ibáñez, M., Stützle, T.: Automatic design of evolutionary algorithms for multi-objective combinatorial optimization. In: Bartz-Beielstein, T., et al. (eds.) PPSN 2014, LNCS, vol. 8672, pp. 508–517. Springer (2014)
7. Bezerra, L.C.T., López-Ibáñez, M., Stützle, T.: Comparing decomposition-based and automatically component-wise designed multi-objective evolutionary algorithms. `http://iridia.ulb.ac.be/supp/IridiaSupp2015-002/` (2015)
8. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. 6(2), 182–197 (2002)
9. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. In: Abraham, A., et al. (eds.) Evolutionary Multiobjective Optimization, pp. 105–145. Advanced Information and Knowledge Processing, Springer, London, UK (2005)
10. Fleming, P.J., Purshouse, R.C., Lygoe, R.J.: Many-objective optimization: An engineering design perspective. In: Coello Coello, C.A., et al. (eds.) EMO, LNCS, vol. 3410, pp. 14–32. Springer (2005)
11. Hajela, P., Lin, C.Y.: Genetic search strategies in multicriterion optimal design. Structural Optimization 4(2), 99–107 (1992)
12. Huband, S., Hingston, P., Barone, L., While, L.: A review of multiobjective test problems and a scalable test problem toolkit. IEEE Trans. Evol. Comput. 10(5), 477–506 (2006)
13. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Evolutionary many-objective optimization: A short review. In: IEEE CEC. pp. 2419–2426. IEEE Press (2009)
14. Knowles, J.D., Corne, D.: Approximating the nondominated front using the Pareto archived evolution strategy. Evol. Comput. 8(2), 149–172 (2000)

15. Li, H., Zhang, Q.: Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. IEEE Trans. Evol. Comput. 13(2), 284–302 (2009)
16. Li, M., Yang, S., Liu, X., Shen, R.: A comparative study on evolutionary algorithms for many-objective optimization. In: Purshouse et al. [20], pp. 261–275
17. López-Ibáñez, M., Dubois-Lacoste, J., Stützle, T., Birattari, M.: The irace package, iterated race for automatic algorithm configuration. Tech. Rep. TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, Belgium (2011)
18. López-Ibáñez, M., Stützle, T.: The automatic design of multi-objective ant colony optimization algorithms. IEEE Trans. Evol. Comput. 16(6), 861–875 (2012)
19. Nowak, K., Märtens, M., Izzo, D.: Empirical performance of the approximation of the least hypervolume contributor. In: Bartz-Beielstein, T., et al. (eds.) PPSN 2014, LNCS, vol. 8672, pp. 662–671. Springer (2014)
20. Purshouse, R.C., et al. (eds.): Evolutionary Multi-Criterion Optimization - 7th International Conference, EMO 2013, Sheffield, UK, March 19-22, 2013. Proceedings, LNCS, vol. 7811. Springer (2013)
21. Reed, P.M.: Many-objective visual analytics: Rethinking the design of complex engineered systems. In: Purshouse et al. [20], pp. 1–1
22. Tušar, T., Filipič, B.: Differential evolution versus genetic algorithms in multiobjective optimization. In: Obayashi, S., et al. (eds.) EMO. LNCS, vol. 4403, pp. 257–271. Springer (2007)
23. Wagner, T., Beume, N., Naujoks, B.: Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In: Obayashi, S., et al. (eds.) EMO, LNCS, vol. 4403, pp. 742–756. Springer (2007)
24. While, L., Bradstreet, L.: Applying the WFG algorithm to calculate incremental hypervolumes. In: IEEE CEC. pp. 1–8. IEEE Press (2012)
25. While, L., Bradstreet, L., Barone, L.: A fast way of calculating exact hypervolumes. IEEE Trans. Evol. Comput. 16(1), 86–95 (2012)
26. Zhang, Q., Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. 11(6), 712–731 (2007)
27. Zhang, Q., Liu, W., Li, H.: The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In: IEEE CEC. pp. 203–208. IEEE Press (2009)
28. Zhang, Q., Suganthan, P.N.: Special session on performance assessment of multiobjective optimization algorithms/CEC'09 MOEA competition. `http://dces.essex.ac.uk/staff/qzhang/moeacompetition09.htm` (2009)
29. Zitzler, E., Künzli, S.: Indicator-based selection in multiobjective search. In: Yao, X., et al. (eds.) PPSN, LNCS, vol. 3242, pp. 832–842. Springer (2004)
30. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In: Giannakoglou, K.C., et al. (eds.) EUROGEN. pp. 95–100. CIMNE, Barcelona, Spain (2002)
31. Zitzler, E., Thiele, L., Deb, K.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evol. Comput. 8(2), 173–195 (2000)