Université Libre de Bruxelles

*Institut de Recherches Interdisciplinaires
et de Développements en Intelligence Artificielle*

To DE or not to DE? Multi-Objective Differential Evolution Revisited from a Component-Wise Perspective

L. C. T. Bezerra, M. López-Ibáñez, and T. Stützle

IRIDIA – Technical Report Series

Technical Report No.
TR/IRIDIA/2015-001

February 2015

IRIDIA – Technical Report Series
ISSN 1781-3794

Published by:
IRIDIA, *Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle*
Université Libre de Bruxelles
Av F. D. Roosevelt 50, CP 194/6
1050 Bruxelles, Belgium

Technical report number TR/IRIDIA/2015-001

To DE or not to DE? Multi-Objective Differential Evolution Revisited from a Component-Wise Perspective

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. Differential evolution (DE) research for multi-objective optimization can be divided into proposals that either consider DE as a stand-alone algorithm, or see DE as an algorithmic component that can be coupled with other algorithm components from the general evolutionary multiobjective optimization (EMO) literature. Contributions of the latter type have shown that DE components can greatly improve the performance of existing algorithms such as NSGA-II, SPEA2, and IBEA. However, several experimental factors have been left aside from that type of algorithm design, compromising its generality. In this work, we revisit the research on the effectiveness of DE for multi-objective optimization, improving it in several ways. In particular, we conduct an iterative analysis on the algorithmic design space, considering DE and environmental selection components as factors. Results show a great level of interaction between algorithm components, indicating that their effectiveness depends on how they are combined. Some designs present state-of-the-art performance, confirming the effectiveness of DE for multi-objective optimization.

Keywords: Multi-objective Optimization, Evolutionary Algorithms, Differential Evolution, Component-wise Design

1 Introduction

Differential evolution (DE) [15] plays an important role in single-objective optimization and has led to the development of a number of effective optimization algorithms for both constrained and unconstrained continuous problems [6]. In particular, one of the most attractive features of DE is its simplicity and its ability to outperform classical genetic algorithms (GAs) [13]. As a result, a number of research proposals have extended DE algorithms to tackle multi-objective optimization problems (MOPs) in the Pareto sense [6, 10, 14]. In general, extensions follow different paths on how to adapt DE to deal with Pareto optimality, and these stand-alone algorithms have been compared to well-known GA-based algorithms such as NSGA-II [7] or SPEA2 [20] to test their effectiveness. Interestingly, two research groups independently proposed the same DE algorithm at about the same time: DEMO [14] and GDE3 [10]. To highlight the effectiveness of this algorithm, we remark that it ranked among the top five best-performing algorithms at the 2009 CEC competition on multi-objective optimization [18].

In the most comprehensive study conducted so far on DE for multi-objective optimization, Tušar and Filipič [17] have considered DEMO as a template for instantiating DE algorithms. Concretely, DEMO uses DE for exploring the decision space, but uses the environmental selection strategy of NSGA-II. The authors then considered the possibility of using other environmental selection approaches, and compared three top-performing GA-based algorithms, NSGA-II, SPEA2, and IBEA [19] with DE versions of these algorithms, aliased DEMO$^{\text{NS-II}}$, DEMO$^{\text{SP2}}$ and DEMO$^{\text{IB}}$. By performing pairwise comparisons between algorithms that differ only in the underlying search mechanism (GA or DE), the DE operators were shown to obtain more accurate approximations of the Pareto front and DEMO$^{\text{SP2}}$ was found to best balance convergence and diversity [16].

We extend here this excellent earlier work by carrying out a more profound component-wise analysis [3, 4] of the design of DE algorithms for MOPs. Our analysis shows that a more fine-grained view of DE components can lead to new insights. In the original analysis only the environmental selection strategy was a component to be set in the DEMO template. However, the DE-part of DEMO differs from traditional GAs in more than one component. In addition to the *DE variation operator*, there is an *online replacement strategy*, i.e., newly generated solutions are compared to existing solutions as soon as they are created, enforcing a higher convergence pressure. In fact, the latter component was found to be the key improvement of DEMO over earlier DE adaptations to MOPs [14]. However, when we consider the DEMO versions that use environmental selection strategies from IBEA and SPEA2 instead of the original DEMO algorithm that uses the environmental selection from NSGA-II, we show that the online replacement strategy is not always beneficial to the effectiveness of the DEMO versions. In other words, while DEMO was an improvement over existing NSGA-II based DE algorithms because of its online replacement strategy, the other DEMO versions present the same (or, sometimes, worst) performance than versions of IBEA and SPEA2 that simply use the DE variation operator.

Furthermore, we consider several factors that affect the conclusions in the original analysis. First, in the original paper, the quality indicator used by IBEA and DEMO$^{\text{IB}}$ was the binary hypervolume difference, whereas strong evidence points to a better performance of IBEA when using the binary epsilon indicator [2, 19]. Second, the analysis conducted in the original paper was done using the default parameter settings traditionally adopted by the EMO community for the benchmarks considered. However, we have recently shown that tuning the numerical parameters of EMO algorithms can significantly improve their performance [2], altering their relative performance. Finally, although the original paper considered a representative number of benchmark functions, they all used the same number of variables. In this work, we consider several different problem sizes to ensure scalability issues do not compromise the generality of our results.

Algorithm 1 componentWiseDE template

```
1: Initialize(pop)
2: repeat
3:     Variate(pop)
4:     Reduce(pop)
5: until termination criteria met
Output: pop
```

Algorithm 2 DE variation

```
Input: pop
1: repeat
2:     trial ← DE_operator(target)
3:     OnlineReplace(pop, target, trial)
4: until #offspring produced
```

Algorithm 3 GA variation

```
Input: pop
1: pool ← Select(pop)
2: pop_new ← GA_operators(pool)
3: pop ← pop ∪ pop_new
```

The remainder of this paper is organized as follows. Section 2 presents our component-wise approach to differential evolution, and how we instantiate both DE-based and GA-based algorithms using a flexible template. Section 3 presents the intermediate algorithmic designs we use in this work to understand the contribution of the individual DE components we consider. The experimental setup used for this assessment is given in Section 4. We split the discussion of the results in two parts. In Section 5, we compare algorithms grouped by environmental selection strategy. In Section 6, we compare all algorithms among themselves and to a well-known efficient EMO algorithm, SMS-EMOA [1]. We do so to put the results in perspective, since we have recently shown that SMS-EMOA performs consistently well for the experimental setup considered here [2]. Finally, we conclude and discuss future work in Section 7.

2 Differential evolution from a component-wise view

Several articles in the literature propose how to adapt DE algorithms to multi-objective optimization. However, the differences among most of these algorithms are quite small. From a very high-level perspective, multi-objective DE algorithms can be represented using the template defined by Algorithms 1 and 2. The general template displayed in Algorithm 1 could actually represent any of the most used evolutionary computation approaches (GA, DE or evolution strategies). Starting from an initial population (line 1), variation operators and environmental selection are applied to a population to promote evolution, until a given stopping criterion is reached.

In DE algorithms, the variation procedure is carried out as displayed in Algorithm 2. The DE operator produces a trial vector from an existing target vector of the population. Although the single-objective optimization literature presents many different strategies for this operation, the multi-objective DE algorithms proposed so far use the *DE/rand/1/bin* approach [15]. The most significant difference between the existing DE proposals is encapsulated in procedure

OnlineReplace (line 3). In earlier algorithms, the trial vector $\boldsymbol{x}_{\text{trial}}$ only replaced the target vector $\boldsymbol{x}_{\text{target}}$ if $\boldsymbol{x}_{\text{trial}}$ dominated $\boldsymbol{x}_{\text{target}}$. In this case, no environmental replacement is necessary, since the population size is always constant. Later, algorithms considered the option of adding the trial vector to the population in case both trial and target vectors were nondominated. In this case, the population size might double at each iteration, and hence environmental replacement strategies are employed after the variation is concluded, to reduce the population to its original size. While this prevents algorithms from early stagnation, it may as well slow down their convergence. We refer to these two replacement versions as *online replacement strategies*, since trial solutions may replace target solutions during the variation stage, before the actual population management represented by procedure Reduce happens. However, some multi-objective DE algorithms do not consider online replacement at all. In this case, solutions are created by the DE operator, but are only compared to the population altogether, when procedure Reduce is executed. These three different options for online solution replacement are listed in the bottom part of Table 1.

The three different DEMO versions considered by Tušar and Filipič [17] can be easily instantiated using this template as follows (all three versions use DE variation and (non)dominance online solution replacement):

DEMO$^{\textbf{NS-II}}$ uses environmental selection strategy proposed for NSGA-II, i.e., nondominated sorting with tie-breaking according to crowdedness.

DEMO$^{\textbf{SP2}}$ uses the environmental selection strategy proposed for SPEA2, i.e., sorting according to dominance strength and tie-breaking according to nearest neighbor density estimation.

DEMO$^{\textbf{IB}}$ uses the environmental selection strategy proposed for IBEA, i.e., sorting according to the binary ϵ-indicator (I_ϵ).

In an analogous fashion, the original GA-based algorithms NSGA-II, SPEA2 and IBEA can be instantiated using the same template. To do so, instead of a DE-based variation, we use a traditional GA variation approach, outlined by Algorithm 3. The mating selection (line 1) is done according to the fitness of the individuals, which is computed using the same strategies adopted for the environmental replacement in the respective GA-based algorithms. Besides the previously discussed algorithms, the component-wise template presented here could also be used to instantiate other algorithms. We will discuss this in more detail in the next section.

3 Investigating intermediate designs

As explained in the previous section, the three original DEMO versions [17] comprise more than a single atomic DE-related algorithmic component. Concretely, it is a combination of the DE variation operator and an online replacement strategy. Although the DEMO versions of NSGA-II, SPEA2, and IBEA have indeed shown performance improvements over the original algorithms, it remains unclear how each of these individual components contribute to these performance

Table 1. Algorithmic options of a component-wise multi-objective DE template.

Component	Domain	Description
Variate	{DE variation, GA variation	Underlying variation options
Reduce	{NSGA-II, SPEA2, IBEA	Environmental selection approaches
OnlineReplace	{dominance, (non)dominance none	Online solution replacement criterion (this component only takes effect when DE variation is used)

gains. To properly assess the effectiveness of these components, we propose a set of intermediate algorithmic designs: $\mathbf{DE^{NS\text{-}II}}$, $\mathbf{DE^{SP2}}$, and $\mathbf{DE^{IB}}$ which are identical to the DEMO variants except that they do not use online solution replacement. Moreover, the only difference between these DE versions and the original versions of NSGA-II, SPEA2 and IBEA is the use of the DE variation operator. For instance, considering the case of NSGA-II, $\mathrm{DE^{NS\text{-}II}}$, and $\mathrm{DEMO^{NS\text{-}II}}$, the first uses traditional GA selection and variation, while the latter two use DE variation. However, while $\mathrm{DEMO^{NS\text{-}II}}$ may replace solutions as soon as they are created, $\mathrm{DE^{NS\text{-}II}}$ replaces solutions only at the environmental selection stage (procedure Reduce of Algorithm 1).

In the next section, we present the experimental setup in which we use these intermediate designs to properly investigate the effectiveness of the DE operators used by the different DEMO versions.

4 Experimental setup

The benchmark sets we consider here include all unconstrained DTLZ [8] and WFG [9] functions (DTLZ1–7 and WFG1–9). Since both benchmark sets offer scalability as to the number of variables and objectives, we explore this feature to increase the representativeness of our investigation. We consider versions of these problems with three and five objectives. Concerning the number of variables n, we consider problems with $n \in \{20, 21, \ldots, 60\}$. Furthermore, to ensure that numerical parameters do not affect our performance assessment of the DE components, we initially tune all algorithms, but we use disjoint sets for tuning and testing to prevent overfitting. More precisely, we use problems with sizes $n_{testing} = \{30, 40, 50\}$ for testing, and problems with sizes $n \in \{20, 21, \ldots, 60\} \setminus n_{testing}$ for tuning. For both testing and tuning, 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. The remaining details about tuning and testing are given below.

Table 2. Parameter space for tuning all MOEAs for continuous optimization.

			GA variation			*DE variation*	
Parameter	$\mu = \lvert\mathtt{pop}\rvert$	$\lambda = \lvert\mathtt{pop_{new}}\rvert$	t_{size}	p_c, p_m	η_c, η_m	CR	F
Domain	$\{10, 20, \ldots, 100\}$	1 or $\lambda_r \cdot \mu$ $\lambda_r \in [0.1, 2]$	$\{2, 4, 8\}$	$[0, 1]$	$\{1, 2, \ldots, 50\}$	$[0, 1]$	$[0.1, 2]$

4.1 Tuning setup

The automatic parameter configuration tool we use in this work is irace [11]. Although it was originally proposed for configuring single-objective optimization algorithms, it can be adapted for multi-objective optimization by using the hypervolume indicator [12]. Concretely, for each problem considered by irace, candidate configurations are run for a maximum number of function evaluations (10 000, following [2]). The approximation fronts they produce are then normalized to the range $[1, 2]$ to prevent issues due to dissimilar domains. Finally, we compute the hypervolume for each front using $r_i = 2.1$, $i = 1, \ldots, M$ as reference point, where M is the number of objectives considered.

The parameter space we consider for tuning all algorithms is given in Table 2. Parameter μ applies to both DE-based and GA-based algorithms. The following six parameters (λ, t_{size}, p_c, p_m, η_c, η_m) only apply to GA-based algorithms. In particular, we highlight that all GA-based algorithms use SBX crossover and polynomial mutation, as commonly done in the literature [1, 8, 9]. Parameter t_{size} controls the size of the deterministic tournament used for mating selection. The probability of applying the crossover operator to a given pair of individuals is controlled by parameter p_c. Analogously, the probability of applying the mutation operator to a given individual is controlled by parameter p_m. In addition, we consider two different mutation schemes: (i) *bitwise*, which sets the mutation probability per variable $p_v = 1/n$; and (ii) *fixed*, where p_v becomes a parameter $\in [0.01, 1]$. Finally, η_c and η_m are the distribution indices for the SBX crossover and polynomial mutation, respectively. The remaining two parameters (CR and F) in Table 2 concern DE variation. They control the number of variables affected by the operator (parameter CR) and the strength of the changes (parameter F).

There are two additional parameters that concern only SPEA2 and IBEA. The original version of SPEA2 contains an additional parameter k for its k-th nearest neighborhood density estimation strategy in the mating selection. Here, besides the default value, which is computed according to the population size and we denote with $k_{\text{method}} = \textit{default}$, we also give irace the possibility of configuring k directly, with $k \in \{1, 2, \ldots, 9\}$. For IBEA, as previously discussed, several different binary quality indicators can be used. Here we allow irace to select between the two most commonly adopted [19], the binary hypervolume indicator (I_H^-) and the binary ϵ-indicator (I_ϵ). Additionally, irace is given the flexibility to set different quality indicators for mating and for environmental selection if that leads the algorithm to better performance. Algorithms are tuned for each benchmark set (DTLZ or WFG) and for each number of objectives (3 or 5); that is, for each algorithm X, we obtain four tuned variants: X_{D3}, X_{D5}, X_{W3} and X_{W5}. For brevity, the tuned settings for all algorithms considered in this work are provided as supplementary material [5].

4.2 Testing setup

For comparing the tuned 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 Pareto optimal fronts. Since the latter are typically infinite, we generate, for each problem instance, a Pareto front with 10 000 Pareto-optimal solutions following the methodology described in the papers where the benchmarks were proposed [8, 9]. Given an approximation front A generated by an algorithm when applied to a problem instance and the Pareto front P of the same problem instance, 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.

The comparison is done visually by means of boxplots, and analytically through rank sums. Since we generate a large set of results, we only discuss the most representative ones here. In particular, many of the DTLZ problems have been found to be easy for EMO algorithms, creating a ceiling effect in the results. For this reason, we focus the discussion on the WFG benchmark and provide the analysis on the DTLZ benchmark as supplementary material [5]. Additionaly, due to the large amount of results we produce, we present here the results for $n = 40$. Similar results were found for $n \in \{30, 50\}$, and are also provided as supplementary material.

5 Experimental analysis grouped by environmental selection strategy

To investigate how each algorithm component individually affects the performance of the different DEMO versions, we first conduct an analysis where algorithms are grouped by the environmental selection strategy they employ.

5.1 NSGA-II, DE$^{\text{NS-II}}$, and DEMO$^{\text{NS-II}}$

The boxplots of the relative hypervolume achieved by the algorithms that use the environmental selection strategy proposed for NSGA-II are given in Figures 1 and 2. For the 3-objective problems (Figure 1), we observe very heterogeneous results. For some problems such as WFG7 and WFG8 there is almost no difference between the algorithms, indicating that the DE components are unable to improve the performance of the original NSGA-II. However, for problems such as WFG1, WFG2, WFG4, and WFG6, the performance of NSGA-II can be improved, sometimes by a large margin, such as for WFG1 and WFG2. When we consider the effectiveness of the DE components, we see that sometimes

Table 3. Sum of ranks depicting the overall performance of algorithms grouped by environmental selection strategy. Algorithms in boldface present rank sums not significantly higher than the lowest ranked for a significance level of 95%.

3 objectives			5 objectives		
DEMO$^{\text{NS-II}}_{\text{W3}}$ (1259.5)	**DE$^{\text{NS-II}}_{\text{W3}}$ (1321)**	NSGA-II$_{\text{W3}}$ (1469.5)	**DE$^{\text{NS-II}}_{\text{W5}}$ (1257)**	DEMO$^{\text{NS-II}}_{\text{W5}}$ (1393)	NSGA-II$_{\text{W5}}$ (1400)
DEMO$^{\text{SP2}}_{\text{W3}}$ (1281)	**SPEA2$_{\text{W3}}$ (1299.5)**	DE$^{\text{SP2}}_{\text{W3}}$ (1469.5)	**DEMO$^{\text{SP2}}_{\text{W5}}$ (1259)**	**DE$^{\text{SP2}}_{\text{W5}}$ (1346.5)**	SPEA2$_{\text{W5}}$ (1444.5)
DE$^{\text{IB}}_{\text{W3}}$ (1212)	**DEMO$^{\text{IB}}_{\text{W3}}$ (1246.5)**	IBEA$_{\text{W3}}$ (1591.5)	**DEMO$^{\text{IB}}_{\text{W5}}$ (1215.5)**	**DE$^{\text{IB}}_{\text{W5}}$ (1225.5)**	IBEA$_{\text{W5}}$ (1609)

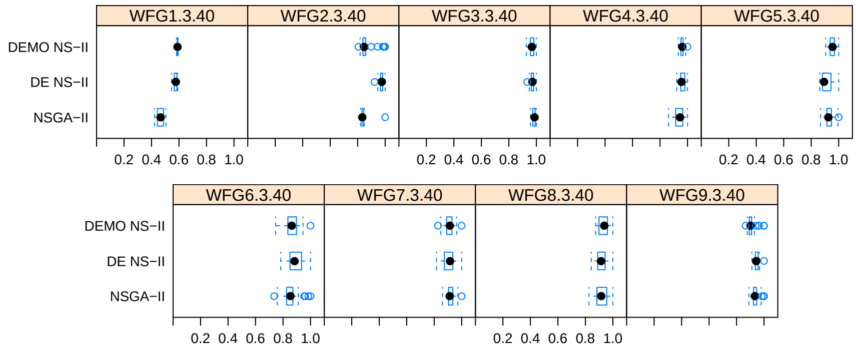

Fig. 1. Boxplots of the relative hypervolume achieved by algorithms that use the environmental selection strategy of NSGA-II (WFG problems, 40 variables, 3 objectives).

using both components (as in DEMO$^{\text{NS-II}}$) is beneficial (e.g., WFG1, WFG5, and WF8), but for other problems it is better to use the DE variation without the online replacement strategy as in DE$^{\text{NS-II}}$ (e.g., WFG2, WFG6, and WFG9). Particularly for WFG9, using both components simultaneously worsens the performance of NSGA-II. When we aggregate results for all runs and sizes of 3-objective WFG problems in a rank sum analysis (Table 3), we see that both DE-based algorithms improve over NSGA-II, but no significant difference can be found among DEMO$^{\text{NS-II}}$ and DE$^{\text{NS-II}}$ using Friedman's test at 95% confidence level.

The performance shown by NSGA-II, DE$^{\text{NS-II}}$, and DEMO$^{\text{NS-II}}$ on the 5-objective WFG problems (see Fig. 2) is quite different. This time, using both DE components (DEMO$^{\text{NS-II}}$) is only beneficial for problems WFG1, WFG4, WFG5, and WFG8. In the other problems, the online replacement leads to results worse even than the ones achieved by the original NSGA-II. However, when we consider only the DE variation (DE$^{\text{NS-II}}$), we see that the performance of NSGA-II is improved for most functions, except for WFG2 and WFG5. When we aggregate results for all 5-objective problems, we see that DE$^{\text{NS-II}}$ indeed ranks first, with significantly lower rank sums than the remaining algorithms (Table 3).

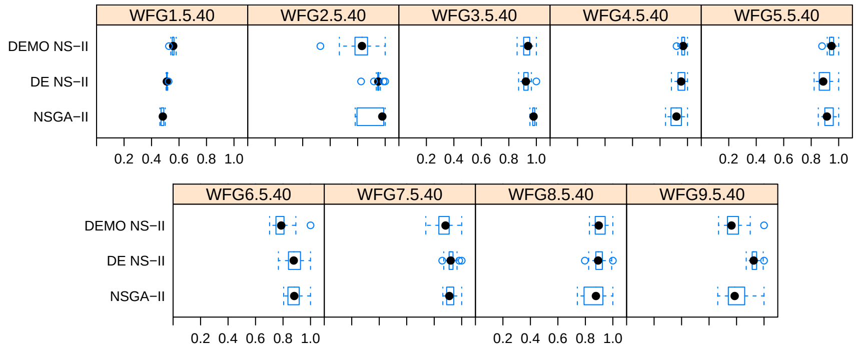

Fig. 2. Boxplots of the relative hypervolume achieved by algorithms that use the environmental selection strategy of NSGA-II (WFG problems, 40 variables, 5 objectives).

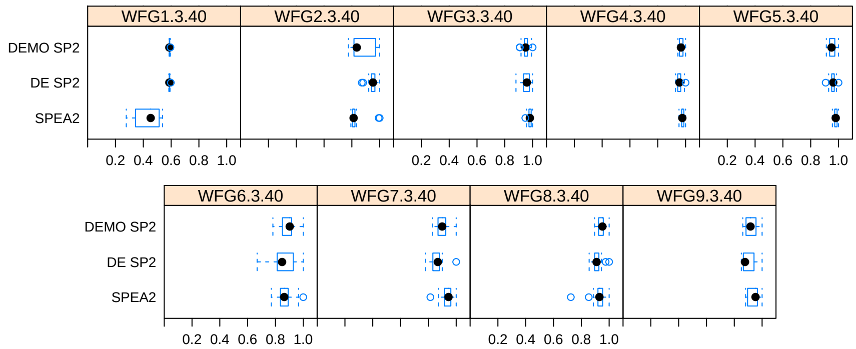

Fig. 3. Boxplots of the relative hypervolume achieved by algorithms that use the environmental selection strategy of SPEA2 (WFG problems, 40 variables, 3 objectives).

5.2 SPEA2, DE$^{\mathbf{SP2}}$, and DEMO$^{\mathbf{SP2}}$

The boxplots of the relative hypervolume achieved by the algorithms that use the environmental selection strategy proposed for SPEA2 are given in Figures 3 and 4. This time the 3-objective problems (Figure 3) show a more clear separation between problems for which DE components lead to improvements and problems for which they worsen the performance of the original SPEA2. For the first group (WFG1, WFG2, and WFG6), we see that there is no pattern as to whether the online replacement is a suitable component for improving SPEA2. However, for the problems where DE components do not lead to performance improvements, typically the version that uses online replacement (that is, DEMO$^{\text{SP2}}$) shows better results than the version that does not use it (that is, DE$^{\text{SP2}}$). When we aggregate results for all 3-objective problems, we see that SPEA2 and DEMO$^{\text{SP2}}$ show equivalent results, while DE$^{\text{SP2}}$ shows significantly higher rank sums than both.

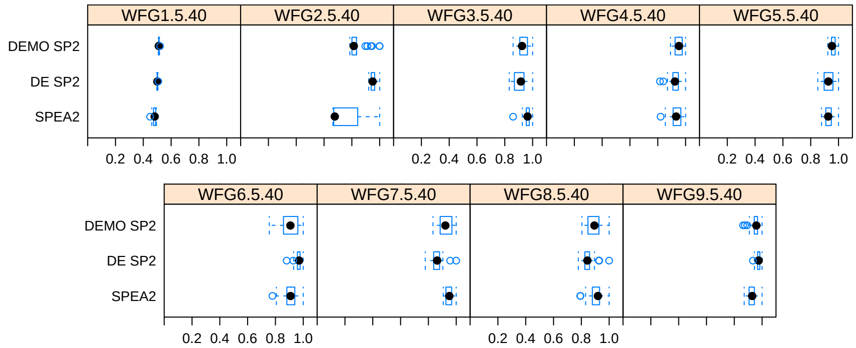

Fig. 4. Boxplots of the relative hypervolume achieved by algorithms that use the environmental selection strategy of SPEA2 (WFG problems, 40 variables, 5 objectives).

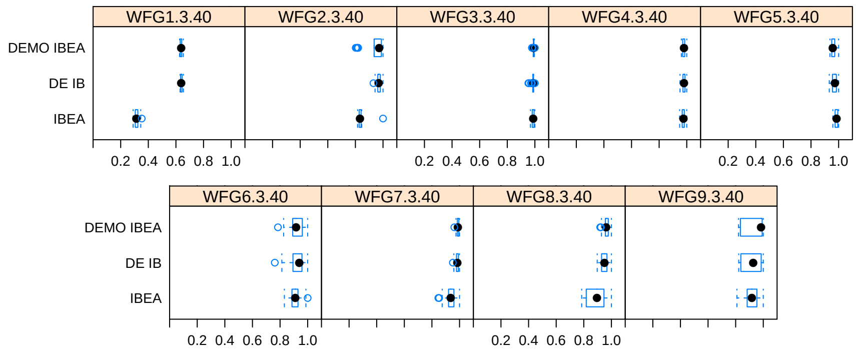

Fig. 5. Boxplots of the relative hypervolume achieved by algorithms that use the environmental selection strategy of IBEA (WFG problems, 40 variables, 3 objectives).

For the 5-objective WFG problems (see Figure 4), the online replacement component plays a more important role than in the 3-objective problems. For most problems, the performance of DE^{SP2} and DEMO^{SP2} is quite different: while DEMO^{SP2} outperforms SPEA2 for most problems, DE^{SP2} worsens the performance of SPEA2 for nearly half of the problems considered. The main exception is WFG2, where DE^{SP2} has the best performance among all algorithms. When all 5-objective problems are considered (Table 3), DEMO^{SP2} ranks first with rank sums significantly lower than DE^{SP2} and SPEA2, which respectively rank second and third. Despite its erratic behavior, DE^{SP2} also presents significantly lower rank sums than SPEA2.

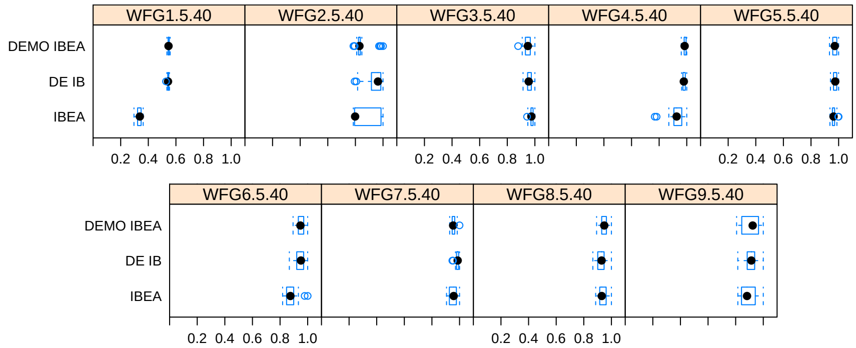

Fig. 6. Boxplots of the relative hypervolume achieved by algorithms that use the environmental selection strategy of IBEA (WFG problems, 40 variables, 5 objectives).

5.3 IBEA, DE$^{\text{IB}}$, and DEMO$^{\text{IB}}$

The boxplots of the relative hypervolume achieved by the algorithms that use the environmental selection strategy proposed for IBEA are given in Figures 5 and 6. The results for the 3-objective problems achieved by these indicator-based versions are far more homogeneous than the results shown before for NSGA-II and SPEA2 environmental selection strategies. In almost all situations, DE$^{\text{IB}}$ and DEMO$^{\text{IB}}$ perform nearly identically. Moreover, the DE-based variants always outperform the GA-based version, except for problems WFG3–WFG5, where the original IBEA was already very effective. These results indicate that, for 3-objective problems, the online replacement component is not an effective component when combined with the indicator-based environmental selection strategy proposed by IBEA.

The results for the 5-objective problems (see Figure 6) are somehow consistent with the results on the 3-objective problems. However, on the 5-objective problems, online replacement leads to performance changes. For some problems, such as WFG2 and WFG7, DE$^{\text{IB}}$ finds better results than DEMO$^{\text{IB}}$. The opposite happens for problems WFG8 and WFG9. When we aggregate across all problems (Table 3), we see that these two algorithms get nearly the same rank sum, and that IBEA gets significantly worse rank sums.

5.4 Overall remarks

Overall, the DE operator leads algorithms to better results on problems WFG1, WFG2, WFG6, and WFG9. As common characteristics, WFG1 and WFG2 present convex geometry, WFG1 and WFG9 present some form of bias, and WFG6 and WFG9 present a complex non-separable reduction [9]. As for the online replacement component, the only problem for which we can say that it is beneficial is the WFG8 problem. However, since the DE operator typically worsens the performance of the original algorithms for this problem, we see that the online replacement is only weakening the effects of the DE operator. Although these results might seem to contradict the results presented by the authors of DEMO, we see that the environmental selection strategy from NSGA-II represents a special case here. $\text{DEMO}^{\text{NS-II}}$ in fact improves over $\text{DE}^{\text{NS-II}}$ and NSGA-II, particularly for functions where NSGA-II faces difficulties [9]. However, this is most likely explained by the poor performance of NSGA-II rather than by the effectiveness of the online replacement strategy.

6 Comparison to SMS-EMOA

In this section we compare all algorithms with SMS-EMOA. In a recent comparison using the same experimental setup, SMS-EMOA was found to be very effective for the benchmarks considered in this work [2].

For the 3-objective problems (Figure 7) we see that, in general, the DE-based algorithms are never clearly worse than SMS-EMOA, except for the WFG6 problem. Particularly for WFG1 and WFG2, the differential evolution operator leads to a significant performance improvement. However, the online replacement is not effective for these two problems regardless of the environmental selection strategy employed, and often worsens the performance of the algorithms. When we aggregate across all 3-objective problems considered (Table 4), we see that DE^{IB} and DEMO^{IB} achieve significantly lower rank sums than all other algorithms. DEMO^{SP2} and SPEA2 rank second, along with SMS-EMOA. These results confirm that DE algorithmic components can indeed lead to significant performance improvements, but that the interactions between them and the environmental selection are also significant.

The comparison between all algorithms for 5-objective problems is given in Figure 8. This time the environmental selection strategy becomes very important for the effectiveness of the algorithms. As expected, dominance-based approaches (NSGA-II and SPEA2) are not as effective for many-objective scenarios, and hence even the DE versions of these algorithms are not able to perform as well as the indicator-based algorithms. However, the performance improvements provided by the DE variation to IBEA is such that both DE^{IB} and DEMO^{IB} become the top-performing algorithms, even though IBEA itself did not perform as competitively as SMS-EMOA. These results indicate that, if coupled with proper many-objective search mechanisms, DE algorithmic components can possibly improve state-of-the-art algorithms, such as SMS-EMOA.

7 Conclusions

This paper has examined how the individual components of DE interact with the components of various EMO algorithms. In particular, we studied the underlying variation operator (GA or DE), the environmental selection strategy (NSGA-II, SPEA2, or IBEA), and the use of an online replacement strategy. For the DTLZ benchmark, results presented a ceiling effect, and hence we focused our analysis on the WFG benchmark. For both three or five objectives, results showed that

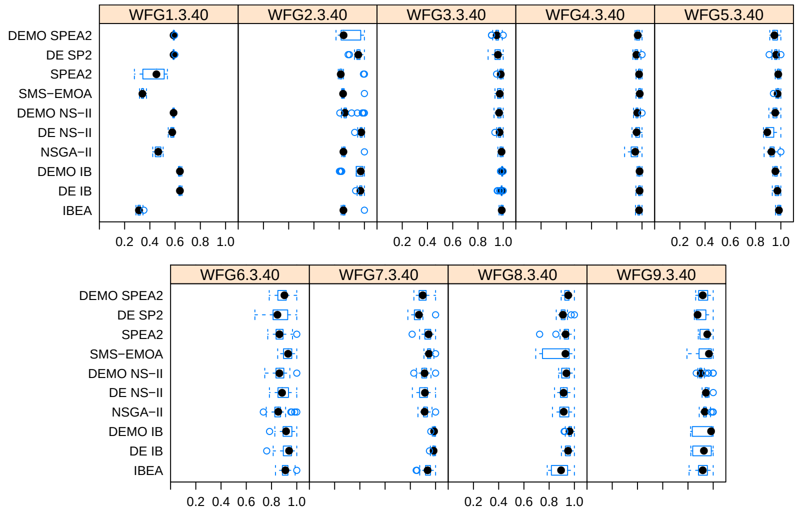

Fig. 7. Relative hypervolume boxplots: 3-objective WFG problems with 40 variables.

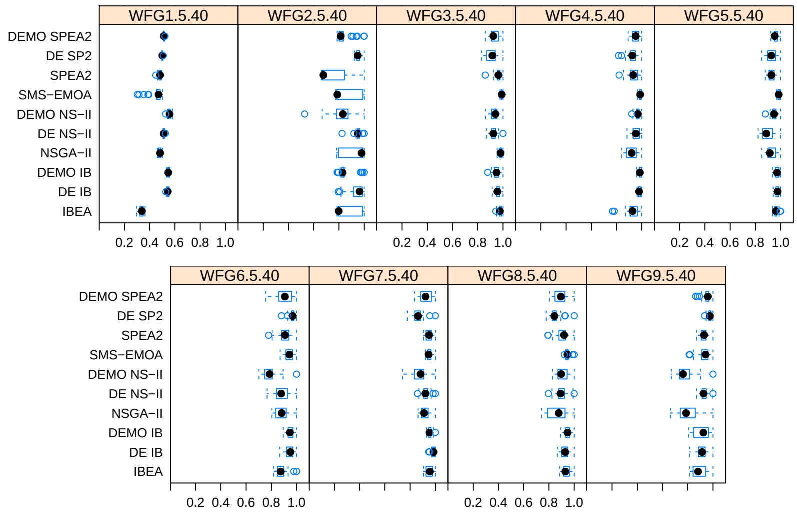

Fig. 8. Relative hypervolume boxplots: 5-objective WFG problems with 40 variables.

the DE-operator improves the algorithms in most problems and that there is a strong interaction between this component and environmental selection. However, for the online replacement component, results almost always indicated that this component is not effective, except when combined with NSGA-II environmental selection.

Table 4. Sum of ranks depicting the overall performance of all algorithms. ΔR is the critical rank sum difference for Friedman's test with 95% confidence. Algorithms in boldface present rank sums not significantly higher than the lowest ranked.

3 objectives ($\Delta R = 271$)	5 objectives ($\Delta R = 265$)
$\mathbf{DE^{IB}}_{\mathbf{W3}}$ (2532)	**$\mathbf{DE^{IB}}_{\mathbf{W5}}$ (2493)**
$\mathbf{DEMO^{IB}}_{\mathbf{W3}}$ (2535)	**$\mathbf{DEMO^{IB}}_{\mathbf{W5}}$ (2506)**
$\text{DEMO}^{\text{SP2}}{}_{\text{W3}}$ (3738.5)	$\text{SMS-EMOA}_{\text{W5}}$ (2891.5)
SPEA2_{W3} (3764.5)	IBEA_{W5} (3930)
$\text{SMS-EMOA}_{\text{W3}}$ (3798)	$\text{DEMO}^{\text{SP2}}{}_{\text{W3}}$ (3932.5)
IBEA_{W3} (3924)	$\text{DE}^{\text{NS-II}}{}_{\text{W5}}$ (4089)
$\text{DEMO}^{\text{NS-II}}{}_{\text{W3}}$ (3972.5)	$\text{DE}^{\text{SP2}}{}_{\text{W5}}$ (4123.5)
$\text{DE}^{\text{NS-II}}{}_{\text{W3}}$ (4094.5)	SPEA2_{W5} (4271.5)
$\text{DE}^{\text{SP2}}{}_{\text{W3}}$ (4325.5)	$\text{DEMO}^{\text{NS-II}}{}_{\text{W5}}$ (4426.5)
$\text{NSGA-II}_{\text{W3}}$ (4440.5)	$\text{NSGA-II}_{\text{W5}}$ (4461)

These results represent a significant contribution of our investigation. Before our work, it was believed that the online replacement component was critical to the effectiveness of multi-objective DE algorithms [14]. Furthermore, this result reinforces the value of the component-wise design approach [2], which advocates that components should be jointly investigated to account for interactions. In fact, the component-wise design of effective DE-based algorithms is an important next step for this research. Here, we have shown that, when coupled with the environmental selection strategy from IBEA and used with numerical parameters properly tuned, a very effective algorithm can be devised. Concretely, this DEMO^{IB} algorithm has consistently outperformed SMS-EMOA, an algorithm that was recently shown to be very effective on the benchmarks considered here. It is then natural to envision the possibility of designing even more effective algorithms if a large set of components is considered, as in the automatic component-wise design methodology [2, 12].

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. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. Eur. J. Oper. Res. 181(3), 1653–1669 (2007)
2. 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)
3. 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)

4. Bezerra, L.C.T., López-Ibáñez, M., Stützle, T.: Deconstructing multi-objective evolutionary algorithms: An iterative analysis on the permutation flowshop. In: Pardalos, P.M., et al. (eds.) LION, LNCS, vol. 8426, pp. 57–172. Springer (2014)
5. Bezerra, L.C.T., López-Ibáñez, M., Stützle, T.: To DE or not to DE? Multi-objective differential evolution revisited from a component-wise perspective. `http://iridia.ulb.ac.be/supp/IridiaSupp2015-001/` (2015)
6. Das, S., Suganthan, P.N.: Differential evolution: a survey of the state-of-the-art. IEEE Trans. Evol. Comput. 15(1) (2011)
7. 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)
8. 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)
9. 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)
10. Kukkonen, S., Lampinen, J.: GDE3: the third evolution step of generalized differential evolution. In: IEEE CEC. pp. 443–450. IEEE Press (2005)
11. 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)
12. López-Ibáñez, M., Stützle, T.: The automatic design of multi-objective ant colony optimization algorithms. IEEE Trans. Evol. Comput. 16(6), 861–875 (2012)
13. Price, K., Storn, R.M., Lampinen, J.A.: Differential Evolution: A Practical Approach to Global Optimization. Springer, New York, NY (2005)
14. Robič, T., Filipič, B.: DEMO: Differential evolution for multiobjective optimization. In: Coello Coello, C.A., et al. (eds.) EMO. LNCS, vol. 3410, pp. 520–533. Springer (2005)
15. Storn, R., Price, K.: Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. 11(4), 341–359 (1997)
16. Tušar, T.: Design of an Algorithm for Multiobjective Optimization with Differential Evolution. M.sc. thesis, Faculty of Computer and Information Science, University of Ljubljana (2007)
17. 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)
18. 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)
19. 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)
20. 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)