

# Differential Evolution Versus Genetic Algorithms in Multiobjective Optimization

Tea Tušar and Bogdan Filipič

Department of Intelligent Systems, Jožef Stefan Institute  
Jamova 39, SI-1000 Ljubljana, Slovenia  
`tea.tusar@ijs.si`, `bogdan.filipic@ijs.si`

**Abstract.** This paper presents a comprehensive comparison between the performance of state-of-the-art genetic algorithms NSGA-II, SPEA2 and IBEA and their differential evolution based variants DEMO<sup>NS-II</sup>, DEMO<sup>SP2</sup> and DEMO<sup>IB</sup>. Experimental results on 16 numerical multiobjective test problems show that on the majority of problems, the algorithms based on differential evolution perform significantly better than the corresponding genetic algorithms with regard to applied quality indicators. This suggests that in numerical multiobjective optimization, differential evolution explores the decision space more efficiently than genetic algorithms.

## 1 Introduction

Differential Evolution (DE) [1] is a simple yet powerful algorithm that outperforms Genetic Algorithms (GAs) on many numerical singleobjective optimization problems [2]. In this paper we show that DE can achieve better results than GAs also on numerical multiobjective optimization problems (MOPs). To this end, we compare three state-of-the-art Multiobjective Evolutionary Algorithms (MOEAs), namely NSGA-II [3], SPEA2 [4] and IBEA [5], to their counterparts – algorithms that use the same environmental selection, but DE instead of GAs for exploring the decision space. While DE-based algorithms for multiobjective optimization have already been proposed in the past (see Related Work in Section 3), comparisons between these approaches and GA-based algorithms lack: (a) a wide choice of difficult test problems with more than two objectives, (b) performance assessment with Pareto compliant indicators, and (c) inferences about algorithm performance based on statistical tests. The comparison in this paper includes all these usually omitted features.

The paper is further organized as follows. Section 2 introduces the basic GA as the underlying algorithm for NSGA-II, SPEA2 and IBEA, while the proposed algorithm DEMO is explained in detail in Section 3. Section 4 outlines the experiments, whose results are presented and discussed in Section 5. Section 6 concludes the paper with a summary of the results.

## 2 Multiobjective Optimization with the Basic GA

Most of the efforts spent on adapting GAs to multiobjective optimization have been focusing on finding new approaches for environmental selection. These

approaches try to produce good approximations of the Pareto optimal front by incorporating different preferences. For example, the environmental selection in NSGA-II [3] first ranks the individuals using nondominated sorting. To distinguish between individuals with the same rank, the crowding distance metric is used, which prefers individuals from less crowded regions of the objective space. SPEA2 [4] works similarly, calculating the raw fitness of the individuals according to Pareto dominance relations between them and using a density measure to break the ties. The individuals that reside close together in the objective space are discouraged from entering the archive of best solutions. IBEA [5], on the other hand, uses a different approach. The fitness of individuals is determined only according to the value of a predefined indicator. This indicator has to be dominance preserving and no other explicit diversity preserving mechanism (such as crowding in NSGA-II or density in SPEA2) is applied.

While directing all attention to environmental selection, the popular algorithms NSGA-II, SPEA2 and IBEA use practically the same algorithm for exploring the decision space. It is therefore possible to describe all three algorithms using a unifying framework, which will be called *Basic Genetic Algorithm* in the remainder of this paper. This algorithm is presented in Fig. 1. After initialization of the populations  $\mathcal{P}$  and  $\mathcal{Q}$ , which is slightly different in NSGA-II, SPEA2 and IBEA<sup>1</sup>, the evolutionary steps of selection, crossover and mutation are repeated until a stopping criterion is met. In environmental selection, one of the previously described approaches is used to calculate the fitness of the individuals. This fitness is used again when comparing individuals in tournament selection. Figure 2 shows the variation operators on individuals encoded as real vectors. In case of combinatorial MOPs, different operators need to be used.

#### Basic Genetic Algorithm for Multiobjective Optimization

1. Initialize populations  $\mathcal{P}_0$  and  $\mathcal{Q}_0$ .
2. Set  $t = 0$ .
3. Repeat:
  - 3.1. Set  $t = t + 1$ .
  - 3.2. Calculate the objectives for new individuals from  $\mathcal{P}_{t-1}$  and  $\mathcal{Q}_{t-1}$ .
  - 3.3. Get  $\mathcal{P}_t$  from  $\mathcal{P}_{t-1}$  and  $\mathcal{Q}_{t-1}$  with environmental selection.
  - 3.4. If stopping criterion met, return nondominated individuals from  $\mathcal{P}_t$ .
  - 3.5. Fill the mating pool  $\mathcal{M}_t$  using tournament selection on  $\mathcal{P}_t$ .
  - 3.6. Apply variation to individuals from  $\mathcal{M}_t$  to get  $\mathcal{Q}_t$  (see Fig. 2).

**Fig. 1.** Outline of the basic genetic algorithm

<sup>1</sup> While NSGA-II initializes the population  $\mathcal{P}_0$  with randomly created individuals and sets  $\mathcal{Q}_0$  to be empty, in SPEA2,  $\mathcal{P}_0$  represents the archive of best solutions and is therefore initially empty, while  $\mathcal{Q}_0$  is filled with randomly created individuals. IBEA originally uses a single population of variable size instead of two separate populations. Without altering its performance, we can assume that IBEA uses two populations, which are initialized in the same way as in NSGA-II.

**Variation***Input:* Mating pool  $\mathcal{M}_t$ 

1. Create empty population  $\mathcal{Q}_t$ .
2. For each pair of individuals  $I_i, I_{i+1}$  ( $i = 1, 3, \dots$ ) from  $\mathcal{M}_t$  do:
  - 2.1. Modify the individuals  $I_i, I_{i+1}$  with uniform crossover.
  - 2.2. Modify the individuals  $I_i, I_{i+1}$  with simulated binary crossover.
  - 2.3. Modify the individual  $I_i$  with polynomial mutation.
  - 2.4. Modify the individual  $I_{i+1}$  with polynomial mutation.
  - 2.5. Add individuals  $I_i$  and  $I_{i+1}$  to  $\mathcal{Q}_t$ .

*Output:* Population  $\mathcal{Q}_t$ **Fig. 2.** Variation of real-coded individuals**3 Multiobjective Optimization with DE**

DE [1] is a simple evolutionary algorithm that encodes solutions as vectors and uses operations such as vector addition, scalar multiplication and exchange of components (crossover) to construct new solutions from the existing ones. When a new solution, also called *candidate*, is constructed, it is compared to its parent. If the candidate is better than its parent, it replaces the parent in the population. Otherwise, the candidate is discarded. As a steady-state algorithm, DE implicitly incorporates elitism, i.e. no solution can be deleted from the population unless a better solution is found. While being a very successful optimization method, DE's greatest limitation originates in its encoding. As no vector representation of solution exists for combinatorial problems, DE can only be applied in numerical optimization.

**3.1 Related Work**

DE has been adapted to solve MOPs in several ways. In the early approaches (PDE [6] and GDE [7]), only the concept of Pareto dominance was used to compare the individuals. The candidate replaced its parent only if it (weakly) dominated it. Otherwise, it was discarded. This is a rather strict demand, especially when the number of objectives is high. Many subsequent approaches (PDEA [8], MODE [9], NSDE [10], GDE2 [11], DEMO [12], GDE3 [13] and NSDE-DCS [14]) used nondominated sorting and/or the crowding distance metric to calculate the fitness of individuals. Only recently, new algorithms that do not follow the environmental selection of NSGA-II were proposed ( $\varepsilon$ -MyDE [15] and DEMORS [16]). To our best knowledge, no algorithms that combine DE with the environmental selection of SPEA2 or IBEA have been presented so far.

**3.2 DEMO<sup>NS-II</sup>, DEMO<sup>SP2</sup> and DEMO<sup>IB</sup>**

The idea presented here is to use DE for exploring the decision space and environmental selection from either NSGA-II, SPEA2 or IBEA to select the best

individuals for the next population. This idea is implemented in the algorithm called DEMO (Differential Evolution for Multiobjective Optimization)<sup>2</sup>.

The outline of DEMO is presented in Figs. 3 and 4. In the main loop, the candidate replaces the parent if it dominates it. If the parent dominates the candidate, the candidate is discarded. Otherwise (when the candidate and parent are nondominated with regard to each other), the candidate is added to the population. This step is repeated until *popSize* number of candidates are created. After that, we get a population of size between *popSize* and  $2 \times \text{popSize}$ . If the population has enlarged, it is truncated to *popSize* using environmental selection.

### Differential Evolution for Multiobjective Optimization

1. Evaluate the initial population  $\mathcal{P}$  of random individuals.
2. While stopping criterion not met, do:
  - 2.1. For each individual  $P_i$  ( $i = 1, \dots, \text{popSize}$ ) from  $\mathcal{P}$  repeat:
    - (a) Create candidate  $C$  from parent  $P_i$  (see Fig. 4).
    - (b) Calculate the objectives of the candidate.
    - (c) If the candidate dominates the parent, the candidate replaces the parent. If the parent dominates the candidate, the candidate is discarded. Otherwise, the candidate is added in the population.
  - 2.2. If the population has more than *popSize* individuals, apply environmental selection to get the best *popSize* individuals.
  - 2.3. Randomly enumerate the individuals in  $\mathcal{P}$ .
3. Return nondominated individuals from  $\mathcal{P}$ .

**Fig. 3.** Outline of DEMO

### Candidate creation

*Input:* Parent  $P_i$

1. Randomly select three individuals  $P_{i_1}, P_{i_2}, P_{i_3}$  from  $\mathcal{P}$ , where  $i, i_1, i_2$  and  $i_3$  are pairwise different.
2. Calculate candidate  $C$  as  $C = P_{i_1} + F(P_{i_2} - P_{i_3})$ , where  $F$  is a scaling factor.
3. Modify the candidate with binary crossover with the parent  $P_i$ .
4. Repair the candidate if it falls out of bounds of the decision space.

*Output:* Candidate  $C$

**Fig. 4.** Candidate creation using scheme *DE/rand/1/bin*

Note that the newly created candidates that enter the population (either by replacement or by addition) instantly take part in the creation of subsequent candidates. This helps achieving fast convergence to the Pareto optimal front. Moreover, it resembles very closely the steady-state mechanism of DE. This is why we prefer the described approach to a somewhat more straightforward way to use DE

<sup>2</sup> DEMO is a generalization of the DEMO/parent variant presented in [12], which used the *DE/rand/1/bin* scheme [2] and environmental selection from NSGA-II.

in the basic GA, which consists of replacing the variation phase (see Fig. 2) with candidate creation (as in Fig. 4) for each individual from the mating pool.

In candidate creation, the use of vector addition can result in candidates that fall out of bounds of the decision space. In such cases, many repair methods are possible. We address this problem by replacing the candidate value violating the boundary constraints with the closest boundary value. In this way, the candidate becomes feasible with as few alterations to it as possible and there is no need for making a new candidate. It is important to note, however, that this repair method may yield more boundary individuals and is biased for problems where the Pareto optimal set lies on one of the bounds of the decision space.

DEMO, as described in Fig. 3, can incorporate arbitrary environmental selection. In the remainder of the paper we will use  $\text{DEMO}^{\text{NS-II}}$ ,  $\text{DEMO}^{\text{SP}^2}$  and  $\text{DEMO}^{\text{IB}}$  to denote the variants of DEMO that use environmental selection from NSGA-II, SPEA2 and IBEA, respectively.

## 4 Experimental Setup

To compare the presented algorithms, extensive experiments on 16 test problems were performed. The focus of the experiments was on comparing  $\text{DEMO}^{\text{NS-II}}$  to NSGA-II,  $\text{DEMO}^{\text{SP}^2}$  to SPEA2, and  $\text{DEMO}^{\text{IB}}$  to IBEA, and not on comparing algorithms with different environmental selection among themselves. Such a comparison can be found, for example, in [5].

### 4.1 Test Problems

Two test problem suits were used in the experiments. The first consists of the first seven DTLZ test problems from [17] and the second of the nine WFG test problems presented in [18]. Both suits comprise difficult problems that present many challenges for multiobjective optimizers, such as the existence of many local Pareto optimal fronts, uneven distribution of points on the Pareto optimal front, nonseparable variables etc.

Let  $n$  and  $m$  denote the dimensionality of the decision space and variable space, respectively. Each of the 16 problems was used three times – each time with a different number of objectives ( $m = 2, 3$  and  $4$ ). The other parameters were set as follows:

- The parameters of DTLZ problems were set as recommended in [17], i.e.  $n = m + k - 1$ , where  $k = 5$  for DTLZ1,  $k = 10$  for problems DTLZ2 to DTLZ6 and  $k = 20$  for DTLZ7.
- Parameters of the WFG test suite are: the number of position related parameters  $k$ , number of distance related parameters  $l$  and number of objectives  $m$ . The number of decision variables is calculated as  $n = k + l$ . Because of some additional requirements ( $l$  must be an even number for WFG2 and WFG3, and  $k$  must be divisible by  $m - 1$ ), we used the following setting:  $k = 6$  and  $l = 4$  (consecutively  $n = 10$ ), which satisfies all the requirements for  $m = 2, 3$  and  $4$ .

All test problems suppose minimization of all objectives.

## 4.2 Parameters of the Algorithms

The experiments with NSGA-II, SPEA2 and IBEA were performed using the PISA environment [19]. The parameter settings for the basic GA, used by all three algorithms, are the same as the ones used in the comparison between NSGA-II and SPEA2 on the DTLZ1 problem in [17]:

- population size = 100,
- number of generations = 300,
- tournament size = 2,
- size of the mating pool = 100,
- individual crossover probability = 1,
- variable probability for SBX crossover = 1,
- distribution index for crossover  $\eta_c = 15$ ,
- variable uniform crossover probability = 0.5,
- individual mutation probability = 1,
- variable polynomial mutation probability =  $1/n$ ,
- distribution index for mutation  $\eta_m = 20$ .

The parameters of all three variants of DEMO were set as in [12] (except for the number of generations, which equals the number of generations used by the basic GA):

- population size = 100,
- number of generations = 300,
- DE selection scheme = *DE/rand/1/bin*,
- scaling factor  $F = 0.5$ ,
- probability used in binary crossover = 0.3.

DEMO<sup>IB</sup> and IBEA used additional parameters: indicator =  $I_{HD}$ <sup>3</sup>, scaling factor  $\kappa = 0.05$  and reference point for the hypervolume calculation  $\rho = (2, \dots, 2) \in \mathbb{R}^m$ . Each algorithm was run on each problem 30 times.

## 4.3 Performance Assessment

The performance assessment was carried out using PISA and the guidelines from [20] and [21]. Consider for example the comparison between DEMO<sup>NS-II</sup> and NSGA-II on one problem. Firstly, the bounds of approximation sets of both algorithms were calculated so that the approximation sets could be normalized to the interval [1, 2]. After that, a dominance rank was calculated for each of the 60 approximation sets by simply counting the number of approximation sets that are better than the observed one. The Mann-Whitney rank sum test was used to discover if there are significant differences between the dominance ranks of the two algorithms.

<sup>3</sup> The same set of experiments was performed also with the  $I_{\varepsilon+}$  indicator. Because of space limitations, we report only the results obtained using  $I_{HD}$  as they are less favorable for DEMO<sup>IB</sup>. The interested reader can access all results from <http://dis.ijs.si/tea/EM02007/demo.htm>.

Additional assessment was carried out using unary quality indicators. From the approximation sets of both algorithms, the set containing only nondominated solutions was computed and used as the reference set for the unary indicators  $I_{\varepsilon+}^1$  and  $I_{R2}^1$ . Other parameters of the  $I_{R2}^1$  indicator were:  $\rho = 0.01$ ,  $(0.9, \dots, 0.9)$  and  $(2.1, \dots, 2.1) \in \mathbb{R}^m$  served as the ideal and Nadir points, and 501, 496 and 455 uniformly spread parameter vectors were used for the problems with two, three and four objectives, respectively. The hypervolume indicator  $I_H$  used the point  $(2.1, \dots, 2.1) \in \mathbb{R}^m$  as the reference point. All three indicators were calculated for each approximation set of both algorithms. The significance of these outcomes was tested independently with the Fisher's independent permutation test. Because we used dominance ranking and three indicators on the same data, the significance level  $\alpha$  for all significance tests was reduced from 0.05 to 0.0125 using the Bonferroni correction.

The same procedure was repeated in comparing DEMO<sup>SP2</sup> to SPEA2 and DEMO<sup>IB</sup> to IBEA. The outcomes of these comparisons are presented in the next section.

## 5 Results and Discussion

Looking at the outcomes of dominance ranking (Tables 1, 3 and 5) we can observe that on many problems, approximation sets of DEMO achieve significantly better domination ranks than the approximation sets of the basic GA. Only rarely (see Subsection 5.3), the basic GA outperforms DEMO. On the majority of problems, however, there are no significant differences between the two algorithms with regard to dominance ranking.

As expected, when dominance ranking shows a significant difference between two algorithms, so do the three applied indicators (an exception is again explained in Subsection 5.3). On the majority of problems, DEMO achieves significantly better results with regard to the chosen indicator (see Tables 2, 4 and 6). Note that on a few problems (see for example DTLZ5 for  $m = 4$  in Table 2), DEMO is significantly better than the basic GA with regard to one indicator ( $I_{R2}^1$ ) and significantly worse with regard to another indicator ( $I_H$ ). This suggests that the outcomes of DEMO and the basic GA are incomparable on such problems.

Besides these results, we also investigated the plots of approximation sets (for  $m = 2$  and 3) and plots of attainment surfaces (for  $m = 2$ ) [22] to gain further insight into the comparison between DEMO and the basic GA. Despite statistical tests show that there is almost always a significant difference in indicator values of the two algorithms, in general no noticeable distinction was *visible* between the approximation sets (and attainment surfaces) of DEMO and the basic GA on problems DTLZ2, DTLZ4, DTLZ5, DTLZ7, WFG3, WFG4, WFG5, WFG8 and WFG9. On problems DTLZ1, DTLZ3 and DTLZ6, where it is very difficult to converge to the Pareto optimal front, and on the non-separable WFG6 problem, DEMO generally attained the Pareto optimal front more efficiently than the basic GA. On problems DTLZ3, WFG1, WFG2 and WFG7, DEMO achieved better spread of solutions along the Pareto optimal front than the basic GA.

**Table 1.** Outcomes of the Mann-Whitney rank sum test ( $\alpha = 0.0125$ ) on dominance ranking for DEMO<sup>NS-II</sup> and NSGA-II. The ‘▲’ ( $p$ -value) (‘▽’  $p$ -value’) denotes the problems, on which DEMO<sup>NS-II</sup> is significantly better (worse) than NSGA-II, while ‘-’ indicates there are no significant differences between the two algorithms.

	$m = 2$	$m = 3$	$m = 4$
DTLZ1	-	▲ $3.9 \times 10^{-13}$	▲ $7.9 \times 10^{-15}$
DTLZ2	-	-	-
DTLZ3	▲ $2.0 \times 10^{-11}$	▲ $3.9 \times 10^{-13}$	▲ $3.5 \times 10^{-12}$
DTLZ4	▲ 0.0052	-	-
DTLZ5	-	-	-
DTLZ6	▲ $4.1 \times 10^{-14}$	▲ $7.9 \times 10^{-15}$	▲ $3.9 \times 10^{-13}$
DTLZ7	-	-	▲ $1.5 \times 10^{-4}$
WFG1	-	▲ $1.6 \times 10^{-7}$	-
WFG2	-	-	-
WFG3	-	-	-
WFG4	-	-	-
WFG5	-	-	-
WFG6	▲ $1.5 \times 10^{-4}$	-	-
WFG7	-	-	-
WFG8	-	-	-
WFG9	-	-	-

**Table 2.** Outcomes of the Fisher-independent test ( $\alpha = 0.0125$ ) on indicator values for DEMO<sup>NS-II</sup> and NSGA-II. A ‘▲’ (‘▽’) under the indicator  $I$  means that DEMO<sup>NS-II</sup> is significantly better (worse) than NSGA-II regarding indicator  $I$ , while ‘-’ indicates there are no significant differences between the two algorithms regarding indicator  $I$

	$m = 2$			$m = 3$			$m = 4$		
	$I_{\varepsilon+}^1$	$I_H$	$I_{R2}^1$	$I_{\varepsilon+}^1$	$I_H$	$I_{R2}^1$	$I_{\varepsilon+}^1$	$I_H$	$I_{R2}^1$
DTLZ1	▲	▲	▲	▲	▲	▲	▲	▲	▲
DTLZ2	▲	▲	▲	▲	▲	▲	▲	▲	▲
DTLZ3	▲	▲	▲	▲	▲	▲	▲	▲	▲
DTLZ4	▲	▲	▲	▲	▲	▲	▲	▲	▲
DTLZ5	▲	▲	▲	▲	▲	▲	-	▽	▲
DTLZ6	▲	▲	▲	▲	▲	▲	▲	▲	▲
DTLZ7	▲	▲	▲	▲	▲	▲	▲	▲	▲
WFG1	▲	▲	▲	▲	▲	▲	▲	▲	▲
WFG2	▲	▲	▲	▲	▲	▲	▲	▲	▲
WFG3	-	▲	▲	▲	▲	▲	-	▽	▽
WFG4	-	▲	-	-	▲	▲	▲	▲	▲
WFG5	▽	-	▽	-	-	-	-	-	▲
WFG6	▲	▲	▲	▲	▲	▲	-	▲	▲
WFG7	▲	▲	▲	▲	▲	▲	-	▲	-
WFG8	▲	-	▲	-	-	-	-	-	▽
WFG9	-	▲	-	-	▲	▲	-	-	-



**Table 3.** Outcomes of the Mann-Whitney rank sum test ( $\alpha = 0.0125$ ) on dominance ranking for DEMO<sup>SP2</sup> and SPEA2. The ‘▲’ (‘▽’) denotes the problems, on which DEMO<sup>SP2</sup> is significantly better (worse) than SPEA2, while ‘-’ indicates there are no significant differences between the two algorithms.

	$m = 2$	$m = 3$	$m = 4$
DTLZ1	-	▲ $9.7 \times 10^{-13}$	▲ $3.2 \times 10^{-13}$
DTLZ2	-	-	-
DTLZ3	▲ $2.2 \times 10^{-11}$	▲ $2.0 \times 10^{-14}$	▲ $3.2 \times 10^{-13}$
DTLZ4	-	-	-
DTLZ5	-	-	-
DTLZ6	▲ $2.0 \times 10^{-14}$	▲ $7.9 \times 10^{-15}$	▲ $1.7 \times 10^{-12}$
DTLZ7	-	-	-
WFG1	-	▲ $2.7 \times 10^{-9}$	-
WFG2	-	-	-
WFG3	-	-	-
WFG4	-	-	-
WFG5	-	-	-
WFG6	▲ $2.6 \times 10^{-6}$	-	-
WFG7	-	-	-
WFG8	-	-	-
WFG9	-	-	-

**Table 4.** Outcomes of the Fisher-independent test ( $\alpha = 0.0125$ ) on indicator values for DEMO<sup>SP2</sup> and SPEA2. A ‘▲’ (‘▽’) under the indicator  $I$  means that DEMO<sup>SP2</sup> is significantly better (worse) than SPEA2 regarding indicator  $I$ , while ‘-’ indicates there are no significant differences between the two algorithms regarding indicator  $I$ .

	$m = 2$			$m = 3$			$m = 4$		
	$I_{\epsilon+}^1$	$I_H$	$I_{R2}^1$	$I_{\epsilon+}^1$	$I_H$	$I_{R2}^1$	$I_{\epsilon+}^1$	$I_H$	$I_{R2}^1$
DTLZ1	▲	▲	▲	▲	▲	▲	▲	▲	▲
DTLZ2	▲	▲	▲	▲	▲	▲	▲	▲	▲
DTLZ3	▲	▲	▲	▲	▲	▲	▲	▲	▲
DTLZ4	▲	▲	▲	▲	▲	▲	▲	▲	▲
DTLZ5	▲	▲	▲	▲	▲	▲	▽	-	▲
DTLZ6	▲	▲	▲	▲	▲	▲	▲	▲	▲
DTLZ7	▲	▲	▲	▲	▲	▲	▲	▲	▲
WFG1	▲	▲	▲	▽	-	-	▲	▲	▲
WFG2	▲	▲	▲	▲	▲	▲	▲	▲	▲
WFG3	▲	▲	▲	-	-	-	▽	-	-
WFG4	-	▲	-	-	▲	▲	-	▲	▲
WFG5	▽	▽	▽	▲	▲	▲	▲	▲	▲
WFG6	▲	▲	▲	▲	▲	▲	-	▲	▲
WFG7	▲	▲	▲	▲	▲	▲	▲	▲	▲
WFG8	▲	-	▲	▲	▲	▲	-	▲	▲
WFG9	-	-	-	▲	▲	▲	-	▲	▲

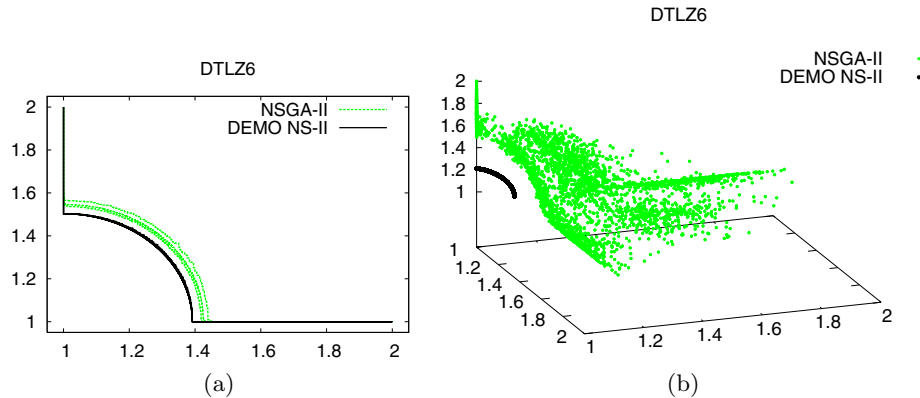
In the following subsections, we review the performance of DEMO and basic GA on selected problems in more detail.

**Table 5.** Outcomes of the Mann-Whitney rank sum test ( $\alpha = 0.0125$ ) on dominance ranking for DEMO<sup>IB</sup> and IBEA. The ‘▲ *p*-value’ (‘▽ *p*-value’) denotes the problems, on which DEMO<sup>IB</sup> is significantly better (worse) than IBEA, while ‘-’ indicates there are no significant differences between the two algorithms.

	<i>m</i> = 2	<i>m</i> = 3	<i>m</i> = 4
DTLZ1	-	-	-
DTLZ2	-	-	-
DTLZ3	-	▲ $3.0 \times 10^{-12}$	▲ $9.7 \times 10^{-12}$
DTLZ4	▲ 0.0013	▲ 0.0104	-
DTLZ5	-	-	-
DTLZ6	▲ $1.2 \times 10^{-13}$	▲ $2.4 \times 10^{-11}$	-
DTLZ7	▽ 0.0023	▽ $1.3 \times 10^{-7}$	▽ $1.8 \times 10^{-7}$
WFG1	-	▽ 0.0058	-
WFG2	-	-	-
WFG3	-	-	-
WFG4	-	-	-
WFG5	-	-	-
WFG6	▲ $6.1 \times 10^{-6}$	-	-
WFG7	-	-	-
WFG8	-	-	-
WFG9	-	-	-

**Table 6.** Outcomes of the Fisher-independent test ( $\alpha = 0.0125$ ) on indicator values for DEMO<sup>IB</sup> and IBEA. A ‘▲’ (‘▽’) under the indicator *I* means that DEMO<sup>IB</sup> is significantly better (worse) than IBEA regarding indicator *I*, while ‘-’ indicates there are no significant differences between the two algorithms regarding indicator *I*.

	<i>m</i> = 2			<i>m</i> = 3			<i>m</i> = 4		
	$I_{\epsilon+}^1$	$I_H$	$I_{R2}^1$	$I_{\epsilon+}^1$	$I_H$	$I_{R2}^1$	$I_{\epsilon+}^1$	$I_H$	$I_{R2}^1$
DTLZ1	▲	▲	▲	▲	▲	▲	▲	▲	-
DTLZ2	-	-	▲	-	▲	▲	-	-	▲
DTLZ3	-	▽	▽	▲	▲	▲	▲	▲	▲
DTLZ4	▲	▲	▲	▲	▲	▲	▲	▲	▲
DTLZ5	-	-	▲	▽	▽	▲	▲	▲	▲
DTLZ6	▲	▲	▲	▲	▲	▲	▲	▲	▲
DTLZ7	▽	▽	▽	▽	▽	▽	▽	▽	-
WFG1	▲	▲	▲	▲	▲	▲	▲	▲	▲
WFG2	▲	▲	▲	▲	▲	▲	-	▲	▲
WFG3	-	▲	▲	-	▲	▲	▽	▽	▽
WFG4	-	▽	▽	-	▲	▲	-	▲	▲
WFG5	▽	▽	▽	-	-	▲	-	▽	▲
WFG6	▲	▲	▲	▲	▲	▲	▲	▲	▲
WFG7	▲	▲	▲	-	▲	▲	-	▲	▲
WFG8	-	-	-	-	-	▽	-	-	▽
WFG9	▲	▲	-	-	▲	▲	-	-	▲



**Fig. 5.** Plots of normalized attainment surfaces and approximation sets of  $\text{DEMO}^{\text{NS-II}}$  and NSGA-II on the DTLZ6 problem: (a) the best, worst and 50%-attainment surfaces for each algorithm on the problem with two objectives; (b) 30 approximation sets for each algorithm on the problem with three objectives

### 5.1 $\text{DEMO}^{\text{NS-II}}$ vs. NSGA-II

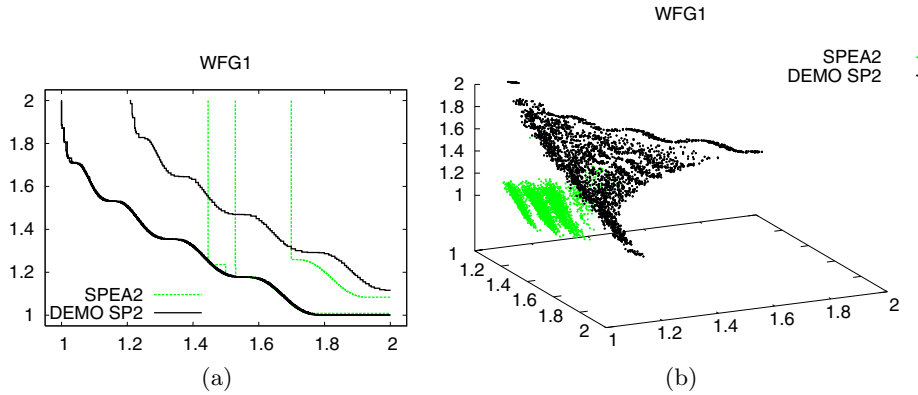
The comparison between DEMO and the basic GA is very favorable to DEMO, when nondominated sorting is used for environmental selection. Let us explore in more detail the outcomes of both algorithms on the DTLZ6 problem. The difficulty of this problem reflects in poor convergence of certain algorithms to the Pareto optimal front. Figure 5 shows that  $\text{DEMO}^{\text{NS-II}}$  reaches the Pareto optimal front for  $m = 2$  and  $m = 3$ , while NSGA-II does not. The most probable cause for such behavior is the repair method used by DEMO, since in this problem, the Pareto optimal set lies at the bounds of the decision space and boundary points are likely to be found after applying DEMO's repair method.

It is interesting to note, however, that on the only other problem (DTLZ7), where the Pareto optimal set lies on the bounds of the decision space, no big differences between approximation sets could be noticed. This is probably because on this problem, none of the algorithms has difficulties in reaching the Pareto optimal front.

### 5.2 $\text{DEMO}^{\text{SP}2}$ vs. SPEA2

Using the strength Pareto approach for environmental selection yields very similar results in the comparison between DEMO and the basic GA as the use of nondominated sorting. The findings from the previous subsection (on problems DTLZ6 and DTLZ7) hold also for  $\text{DEMO}^{\text{SP}2}$  and SPEA2. Similarly, some of the characteristics of the comparison between  $\text{DEMO}^{\text{SP}2}$  and SPEA2 on the WFG1 problem, which will be discussed shortly, are true also when comparing  $\text{DEMO}^{\text{NS-II}}$  and NSGA-II.

Consider now the WFG1 problem for  $m = 2$ . From the plot of attainment surfaces (see Fig. 6) we can see that  $\text{DEMO}^{\text{SP}2}$  reaches a wider portion of the



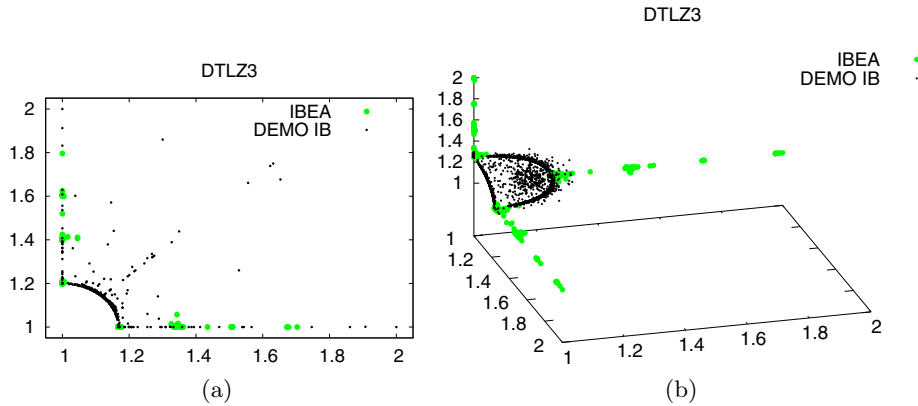
**Fig. 6.** Plots of normalized attainment surfaces and approximation sets of  $\text{DEMO}^{\text{SP}2}$  and SPEA2 on the WFG1 problem: (a) the best, worst and 50%-attainment surfaces for each algorithm on the problem with two objectives; (b) 30 approximation sets for each algorithm on the problem with three objectives

Pareto optimal front than SPEA2, while having comparable convergence properties in the best and average case (50%-attainment surface) and a little worse in the worst case. When this problem is tackled in three objectives,  $\text{DEMO}^{\text{SP}2}$  loses some of its convergence power while keeping the good coverage. SPEA2, on the other hand, still covers only a small part of the whole front, while achieving much better convergence than  $\text{DEMO}^{\text{SP}2}$ . Although this is not visible from the plots, we wish to point out that neither of the algorithms reached the Pareto optimal front for this problem.

There is an additional interesting aspect of the results on this problem, which is related to the performance assessment using dominance ranking and quality indicators. Note that Tables 3 and 4 show that  $\text{DEMO}^{\text{SP}2}$  is significantly better than SPEA2 with regard to dominance ranking, and significantly worse than SPEA2 with regard to the  $I_{\varepsilon+}^1$  indicator. This happens because the approximation sets of  $\text{DEMO}^{\text{SP}2}$  are never dominated, while the approximation sets of SPEA2 sometimes dominate each other. As a result, dominance ranking prefers  $\text{DEMO}^{\text{SP}2}$  to SPEA2 although approximation sets of SPEA2 are closer to the Pareto optimal front than approximation sets of  $\text{DEMO}^{\text{SP}2}$ .

### 5.3 $\text{DEMO}^{\text{IB}}$ vs. IBEA

From Tables 5 and 6 it is obvious that using indicator based environmental selection brought DEMO less improvement over the basic GA than using the other two approaches. For the first time, DEMO was outperformed with regard to dominance ranking. The DTLZ7 problem with  $2^{m-1}$  disconnected Pareto optimal regions proved to be very hard for  $\text{DEMO}^{\text{IB}}$ . While the convergence to the Pareto optimal front was not difficult, maintaining diverse solutions was hard for  $\text{DEMO}^{\text{IB}}$ . Out of 30 runs for each objective space dimensionality, DEMO converged to a single point 29 times for  $m = 2$ , 26 times for  $m = 3$  and 25 times



**Fig. 7.** Plots of normalized approximation sets of DEMO<sup>IB</sup> and IBEA on the DTLZ3 problem: (a) 30 approximation sets for each algorithm on the problem with two objectives; (b) 30 approximation sets for each algorithm on the problem with three objectives

for  $m = 4$ . Note, however, that in combination with all other approaches to environmental selection (including using  $I_{\varepsilon+}^1$  instead of  $I_{HD}$  in indicator based selection), DEMO could always maintain diverse solutions.

Let us analyze in more detail also the DTLZ3 problem, where the main difficulty rises from its  $3^{10} - 1$  local Pareto optimal fronts. As shown in the plots in Fig. 7, IBEA has more difficulties in reaching the Pareto optimal front than DEMO<sup>IB</sup>. In the case of two objectives, DEMO<sup>IB</sup> performs worse than IBEA in the worst case while achieving a much better spread in the best case. On the three-objective problem, DEMO<sup>IB</sup> achieves good results in all 30 runs, while IBEA still gets stuck in local optima and has a poor spread of solutions.

## 6 Conclusion

This paper compared the well-known multiobjective evolutionary algorithms NSGA-II, SPEA2 and IBEA to their DE-based variants DEMO<sup>NS-II</sup>, DEMO<sup>SP2</sup> and DEMO<sup>IB</sup> on 16 state-of-the-art benchmark problems (each with 2, 3 and 4 objectives). The results have shown that on 20% of the problems, DEMO achieved significantly better dominance ranks than the basic GA, while significantly worse dominance ranks were obtained on only 3% of the problems. Furthermore, DEMO outperformed the basic GA with regard to the used quality indicator on the majority (almost 83%) of the problems.

On the basis of these results we conclude that DE explores the decision space more efficiently than GAs also when multiple objectives need to be optimized. It is important to note, however, that DE and, consequently, DEMO are limited to vector representation of solutions and can therefore only be used in numerical optimization.

## Acknowledgment

The work presented in the paper was supported by the Slovenian Research Agency under Research Program P2-0209 *Artificial Intelligence and Intelligent Systems*. The authors wish to thank Eckart Zitzler for useful comments on algorithm performance assessment.

## References

1. Price, K.V., Storn, R.: Differential evolution – A simple evolution strategy for fast optimization. *Dr. Dobb's Journal* **22**(4) (1997) 18–24
2. Price, K., Storn, R.M., Lampinen, J.A.: *Differential Evolution: A Practical Approach to Global Optimization*. Springer-Verlag New York, Inc. (2005)
3. Deb, K., Pratap, A., Agrawal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **6**(2) (2002) 182–197
4. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength Pareto evolutionary algorithm. In: *Proceedings of Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems – EUROGEN 2001*. (2001) 95–100
5. Zitzler, E., Künzli, S.: Indicator-based selection in multiobjective search. In: *Proceedings of the Eighth International Conference on Parallel Problem Solving from Nature – PPSN VIII*. (2004) 832–842
6. Abbass, H.A., Sarker, R., Newton, C.: PDE: A Pareto-frontier differential evolution approach for multi-objective optimization problems. In: *Proceedings of the 2001 Congress on Evolutionary Computation – CEC 2001*. Volume 2. (2001) 971–978
7. Lampinen, J.: DE's selection rule for multiobjective optimization. Technical report, Lappeenranta University of Technology (2001)
8. Madavan, N.K.: Multiobjective optimization using a Pareto differential evolution approach. In: *Proceedings of the 2002 Congress on Evolutionary Computation – CEC 2002*. Volume 2. (2002) 1145–1150
9. Xue, F., Sanderson, A.C., Graves, R.J.: Pareto-based multi-objective differential evolution. In: *Proceedings of the 2003 Congress on Evolutionary Computation – CEC 2003*. Volume 2. (2003) 862–869
10. Iorio, A.W., Li, X.: Solving rotated multi-objective optimization problems using differential evolution. In: *Proceedings of the 17th Australian Joint Conference on Artificial Intelligence – AI 2004*. (2004) 861–872
11. Kukkonen, S., Lampinen, J.: An extension of generalized differential evolution for multi-objective optimization with constraints. In: *Proceedings of Parallel Problem Solving from Nature – PPSN VIII*. (2004) 752–761
12. Robič, T., Filipič, B.: DEMO: Differential evolution for multiobjective optimization. In: *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization – EMO 2005*. (2005) 520–533
13. Kukkonen, S., Lampinen, J.: GDE3: The third evolution step of generalized differential evolution. In: *Proceedings of the 2005 Congress on Evolutionary Computation – CEC 2005*. Volume 1. (2005) 443–450
14. Iorio, A.W., Li, X.: Incorporating directional information within a differential evolution algorithm for multi-objective optimization. In: *Proceedings of the 2006 Genetic and Evolutionary Computation Conference – GECCO 2006*. Volume 1. (2006) 675–682

15. Santana-Quintero, L.V., Coello Coello, C.A.: An algorithm based on differential evolution for multiobjective problems. In: Smart Engineering System Design: Neural Networks, Evolutionary Programming and Artificial Life. (2005) 211–220
16. Hernández-Díaz, A.G., Santana-Quintero, L.V., Coello Coello, C., Caballero, R., Molina, J.: A new proposal for multi-objective optimization using differential evolution and rough sets theory. In: Proceedings of the 2006 Genetic and Evolutionary Computation Conference – GECCO 2006. Volume 1. (2006) 675–682
17. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multi-objective optimization. In Abraham, A., Jain, R., Goldberg, R., eds.: Evolutionary Multiobjective Optimization: Theoretical Advances and Applications. Springer (2005) 105–145
18. Huband, S., Barone, L., White, L., Hingston, P.: A scalable multi-objective test problem toolkit. In: Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization – EMO 2005. (2005) 280–295
19. Bleuler, S., Laumanns, M., Thiele, L., Zitzler, E.: PISA – A platform and programming language independent interface for search algorithms. In: Proceedings of the Second International Conference on Evolutionary Multi-Criterion Optimization – EMO 2003. (2003) 494–508
20. Knowles, J.D., Thiele, L., Zitzler, E.: A tutorial on the performance assessment of stochastic multiobjective optimizers. TIK-Report No. 214, Computer Engineering and Networks Laboratory, ETH Zürich, Switzerland (2006)
21. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.D.: Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation* **7**(2) (2003) 117–132
22. Fonseca, C.M., Fleming, P.J.: On the performance assessment and comparison of multiobjective optimizers. In: Proceedings of Parallel Problem Solving from Nature – PPSN IV. (1996) 584–593