

J Math Model Algor  
DOI 10.1007/s10852-014-9255-y

---

# Generalized Multiobjective Evolutionary Algorithm Guided by Descent Directions

Roman Denysiuk · Lino Costa · Isabel Espírito Santo

Received: 10 August 2013 / Accepted: 19 March 2014  
© Springer Science+Business Media Dordrecht 2014

**Abstract** This paper proposes a generalized descent directions-guided multiobjective algorithm (DDMOA2). DDMOA2 uses the scalarizing fitness assignment in its parent and environmental selection procedures. The population consists of leader and non-leader individuals. Each individual in the population is represented by a tuple containing its genotype as well as the set of strategy parameters. The main novelty and the primary strength of our algorithm is its reproduction operator, which combines the traditional local search and stochastic search techniques. To improve efficiency, when the number of objective is increased, descent directions are found only for two randomly chosen objectives. Furthermore, in order to increase the search pressure in high-dimensional objective space, we impose an additional condition for the acceptance of descent directions found for leaders during local search. The performance of the proposed approach is compared with those produced by representative state-of-the-art multiobjective evolutionary algorithms on a set of problems with up to 8 objectives. The experimental results reveal that our algorithm is able to produce highly competitive results with well-established multiobjective optimizers on all tested problems. Moreover, due to its hybrid reproduction operator, DDMOA2 demonstrates superior performance on multimodal problems.

**Keywords** Multiobjective optimization · Multiobjective evolutionary algorithms · Performance assessment

---

Proceedings of the 13th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE'13, held in Almeria, Spain 24-27 June, 2013.

R. Denysiuk (✉)  
Algoritmi R&D Center, University of Minho, Braga, Portugal  
e-mail: roman.denysiuk@algoritmi.uminho.pt

L. Costa · I. E. Santo  
Department of Production and Systems Engineering, University of Minho, Braga, Portugal  
e-mail: lac@dps.uminho.pt

I. E. Santo  
e-mail: iapinho@dps.uminho.pt

## 1 Introduction

Many real-world optimization problems involve the simultaneous optimization of several conflicting objectives. These problems are called multiobjective optimization problems. Assuming minimization, a multiobjective optimization problem with  $m$  objectives and  $n$  decision variables can be formulated mathematically as follows:

$$\begin{aligned} & \text{minimize: } \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))^T \\ & \text{subject to: } \mathbf{x} \in \Omega \end{aligned} \quad (1)$$

where  $\mathbf{x}$  is the decision vector defined in the feasible decision space  $\Omega \subseteq \mathbb{R}^n$  and  $\Omega = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}$ ,  $\mathbf{l}$  and  $\mathbf{u}$  are the lower and upper bounds of the decision variables, respectively, and  $\mathbf{f}(\mathbf{x})$  is the objective vector defined in the objective space  $\mathbb{R}^m$ . The goal of multiobjective optimization is to find a set of well-distributed compromise solutions representing the different trade-offs with respect to the given objective functions. This set is known as the set of Pareto optimal solutions.

Classical methods for solving multiobjective optimization problems mostly rely on scalarization that means converting the problem into a single-objective optimization problem with a real valued objective function, which depends on some parameters and is termed the scalarizing function. Repeated runs with different parameter settings are used to find multiple Pareto optimal solutions [20]. However, there are a few classical methods which attempt to find multiple Pareto optimal solutions in a single simulation run [22].

On the other hand, evolutionary algorithms (EAs) have emerged as a powerful alternative for solving multiobjective optimization problems. Evolutionary algorithms are particularly suitable for this task because they deal simultaneously with a set of possible solutions, called population, which allows to find several members of the Pareto optimal set in a single run of the algorithm, instead of having to perform series of separate runs. A number of multiobjective evolutionary algorithms have been proposed during the last two decades and successfully applied to solve a large number of multiobjective optimization problems [4, 6].

The current research in evolutionary multiobjective optimization (EMO) mainly focuses on the fitness assignment and selection of fittest individuals in multiobjective search. However, recombination operators are crucial to the performance of evolutionary algorithms. Moreover, combinations of EAs and local search methods often outperforms traditional evolutionary algorithms in single-objective optimization [5]. Although a few hybrid approaches for multiobjective optimization have been developed [17], hybrid multiobjective evolutionary algorithms remain to be an under-explored research area in multiobjective optimization.

A hybrid multiobjective evolutionary algorithm (DDMOA) was proposed in [9], and further studied in [11]. DDMOA showed highly competitive results when compared with representative state-of-the-art multiobjective algorithms. However, several issues were identified as potential weaknesses of DDMOA. In particular, in DDMOA the population consists only of nondominated solutions that can cause only a few or even one solution being presented in the population. In turn, that can severely deteriorate the performance of the algorithm on multi-modal problems. Another weakness is that the original DDMOA is not applicable to many-objective problems due to its selection operator. Moreover, with increasing number of objectives, finding descent directions for all objectives using local search may be computationally expensive.

In [10], an improved algorithm, called DDMOA2, was proposed in order to overcome the aforementioned drawbacks. DDMOA2 demonstrated the superior performance on a set of problems with different characteristics comparing with its predecessor [10]. In this paper, we extend DDMOA2 to deal with problems having more than three objectives. In order to

increase the search pressure in high-dimensional objective space we impose an additional condition for the acceptance of descent directions calculated for leaders in local search procedure. The performance of the algorithm is studied on a set of scalable test problems and results are compared with representative state-of-the-art EMO algorithms.

The remainder of this paper is organized as follows. In Section 2, we give a detailed description of DDMOA2. In Section 3, we present the methodology used to validate our approach and report on the obtained experimental results. In Section 4, we conclude and discuss some possible future work.

## 2 DDMOA2

The main loop of DDMOA2 is given by Algorithm 1. It is a hybrid evolutionary algorithm with  $(\mu + \lambda)$  selection scheme, where in each generation, the selection of leaders and the adaptation of the strategy parameters of all population members are followed by the successive application of parent selection, mutation, and environmental selection.

---

### Algorithm 1 DDMOA2

---

```

1:  $g \leftarrow 0$ ;
2: initialization:  $P^{(g)}, W = \{w_1, \dots, w_\mu\}$ ;
3: repeat
4:    $P^{(g)} \leftarrow \text{leaderSelection}(P^{(g)});$ 
5:    $P^{(g)} \leftarrow \text{updateSearchMatrix}(P^{(g)});$ 
6:    $P^{(g)} \leftarrow \text{updateStepSize}(P^{(g)});$ 
7:    $P^{(g)} \leftarrow \text{parentSelection}(P^{(g)});$ 
8:    $P^{(g)} \leftarrow \text{mutation}(P^{(g)});$ 
9:    $P^{(g+1)} \leftarrow \text{environmentalSelection}(P^{(g)});$ 
10:   $g \leftarrow g + 1$ ;
11: until the stopping criterion is met
12: output:  $P^{(g)}$ ;

```

---

In DDMOA2, an individual  $a_i$  ( $i \in \{1, \dots, \mu\}$ ) in the current population  $P^{(g)}$  in generation  $g$  is a tuple of the form  $[x_i, \delta_i, S_i, \sigma_i]$ , where  $x_i \in \mathbb{R}^n$  is the decision vector,  $\delta_i > 0$  is the step size used for local search,  $S_i \in \mathbb{R}^{n \times 2}$  is the search matrix, and  $\sigma_i > 0$  is the step size used for reproduction.

In the following, the components of DDMOA2 are discussed in more detail.

### 2.1 Initialization Procedure

The algorithm starts by generating a set of weight vectors  $W = \{w_1, \dots, w_\mu\}$  and initializing the population of size  $\mu$  using Latin hypercube sampling [19]. The strategy parameters of each population member are initialized taking default values. The search matrix  $S^{(0)}$  is initialized by simply generating a zero matrix of size  $n \times 2$ .

### 2.2 Leader Selection Procedure

Each generation of DDMOA2 is started by selecting leaders of the current population. A leader is a population member that performs the best on at least one weight vector. Thus,

leaders are selected as follows. First, the objective values of all individuals in the population are normalized:

$$\bar{f}_i = \frac{f_i - f_i^{\min}}{f_i^{\max} - f_i^{\min}}, \quad \forall i \in \{1, \dots, m\} \tag{2}$$

where  $f_i^{\min}$  and  $f_i^{\max}$  are the minimum and maximum values of the  $i$ -th objective in the current population, respectively, and  $\bar{f}_i \in [0, 1], \forall i \in \{1, \dots, m\}$  is the normalized objective value. For each weight vector, the fitness of each population member is calculated on a given weight vector using the weighted Chebyshev method, which after normalization of objectives can be defined as:

$$f_{\text{fitness}} = \max_{1 \leq i \leq m} \{w_i \bar{f}_i(\mathbf{x})\} \tag{3}$$

where  $f_{\text{fitness}}$  is the fitness of the population member  $\mathbf{x}$  on the weight vector  $\mathbf{w}$ . An individual having the best fitness on a given weight vector is a leader. It should be noted that one leader can have the best fitness value on several weight vectors.

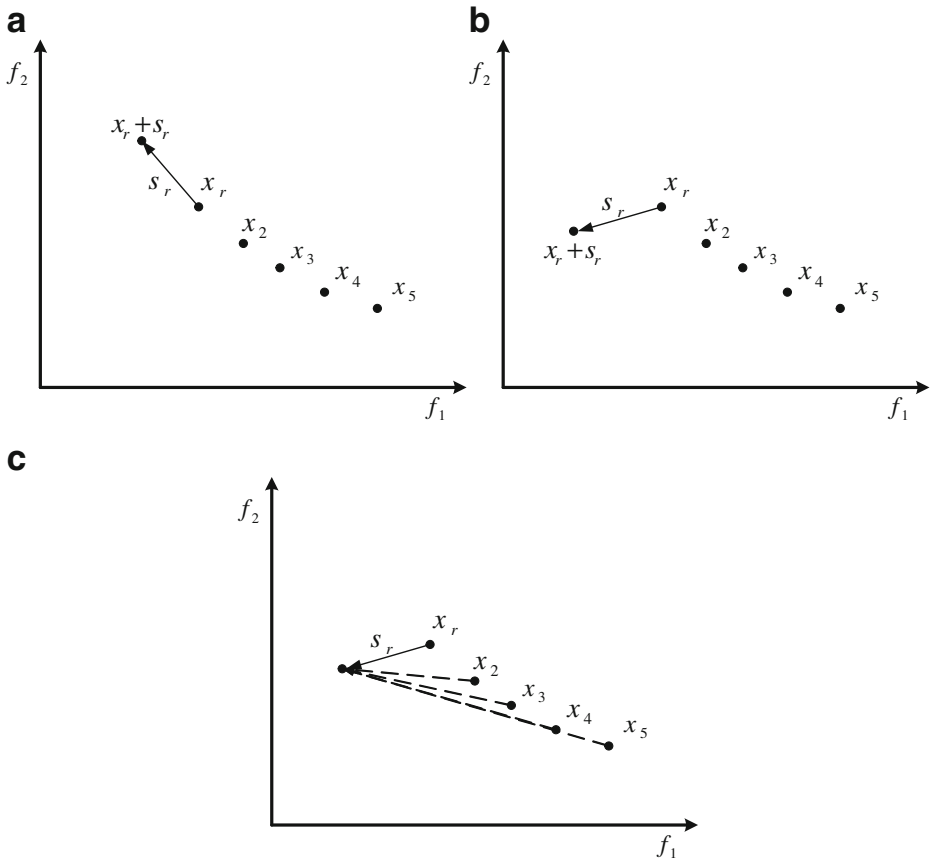
### 2.3 Update Search Matrix Procedure

Each generation of DDMOA2 is started by updating the search matrices of all individuals in the current population invoking `updateSearchMatrix` procedure. In DDMOA2, independently on the dimensionality of the objective space descent directions are found only for two randomly chosen objectives. Thus, the search matrix of each population member contains two columns that store descent directions for these randomly chosen objectives. The motivation behind the finding descend directions only for two instead of all objectives is that with an increasing number of objectives to find descend direction for all objectives using local search may become expensive.

In the beginning of `updateSearchMatrix` procedure two objectives are chosen at random, and only leaders of the current population are considered while all the other individuals are temporarily discarded.

Thereafter, for the first chosen objective, the resulting population is sorted in ascending order and partitioned into  $\alpha$  equal parts. Thus,  $\alpha$  subpopulations are defined in order to promote different reference points for the computation of descent directions. It follows that in each subpopulation, representative individual  $\mathbf{x}_r$  is selected. A representative of the subpopulation is a solution with the smallest value of the corresponding objective function among other solutions in the subpopulation and  $\delta > \delta_{\text{tol}}$ . Thus, if the solution with the smallest value of the corresponding objective has  $\delta \leq \delta_{\text{tol}}$  then solution with the second smallest value is selected as representative and so on. After that, descent direction for corresponding objective function is computed for representative using coordinate search [23]. During coordinate search, step size  $\delta$  of representative is reduced if no decrease in the objective function value is found. Each time, a trial solution is calculated and this solution is compared with the current population. If this trial solution has a smaller value for at least one objective compared with each member of the current population then it is added to the population, assuming the default values of strategy parameters.

In order to increase the search pressure, descent direction  $\mathbf{s}_r$  for representative  $\mathbf{x}_r$  is accepted during coordinate search if trial solution  $\mathbf{x}_r + \mathbf{s}_r$  is not worse in all objectives than  $\mathbf{x}_r$ , and direction  $\mathbf{s}_r$  leads to the decrease of the corresponding objective (objective for which descent direction is calculated). To better understand this procedure consider an example illustrated in Fig. 1. Where a subpopulation is presented and coordinate search is used to find



**Fig. 1** Computation of descent directions

descent directions with respect to  $f_1$ . Simply calculating descent direction  $s_r$  for subpopulation representative  $x_r$ , a descent direction which shown in Fig. 1a can be found. Although this direction leads to a decrease with respect to  $f_1$ , it does not lead to the promising region of the search. Imposing the condition that trial solution  $x_r + s_r$  cannot be worse in all objectives and it must be better than  $x_r$  in  $f_1$ , we exclude the possibility of obtaining a descent direction shown in Fig. 1a. An example of a descent direction that satisfies such condition is presented in Fig. 1b.

When descent direction  $s_r$  for subpopulation representative is found, descent directions for all other subpopulation members are computed as follows:

$$s_i = x_r - x_i + s_r, \tag{4}$$

where  $s_i$  is the descent direction for the  $i$ -th subpopulation member,  $x_i$  is the decision vector of the  $i$ -th subpopulation member,  $x_r$  is the subpopulation representative, and  $s_r$  is the descent direction for the representative. As an example, descent directions computed for other individuals in the subpopulation are shown by dashed arrows in Fig. 1c. The calculated descent directions are stored in the first column of the search matrices of the corresponding individuals. Thereafter, the same procedure for finding descent directions is performed for

the second chosen objective and the results are stored in the second column of the search matrices.

After the search matrices of leaders of the current population are updated, the search matrices of all other population members need to be updated. To do this, we use a simple stochastic procedure as follows. For each non-leader individual, a leader is randomly chosen and this leader shares its search matrix, i.e., non-leader individual's search matrix is equal to the selected leader's search matrix. This way, at the end of the `updateSearchMatrix` procedure, the search matrices of all population members are updated. Usually, after this procedure the population size is greater than  $\mu$  as some promising solutions satisfying the aforementioned conditions are added to the population during coordinate search.

## 2.4 Update Step Size Procedure

Before generating offspring, the step size of each population member needs to be updated. However, there is no common rule to update the step size  $\sigma$ , but it must be done carefully to ensure convergence to the Pareto set, starting from a larger value at the beginning and gradually reducing it during the generations. DDMOA2 uses the following rule for updating the step size of each population member:

$$\sigma = \max\{\exp(\tau N(0, 1)) \sigma_0 \left(1 - \frac{3 \text{funEval}}{\text{maxEval}}\right), \delta_{\text{tol}}\} \quad (5)$$

where  $\tau$  is the learning parameter ( $\tau = 1/\sqrt{2n}$ ),  $N(0, 1)$  is a random number sampled from the normal distribution with mean 0 and standard deviation 1,  $\sigma_0$  is the initial value of the step size,  $\text{funEval}$  is the current number of function evaluations, and  $\text{maxEval}$  is the maximum number of function evaluations. The idea here is to exponentially reduce the step size depending on the current number of function evaluations and multiply it by a scaling factor (adapted from evolution strategies [1]), thereby obtaining different step size values among the population members.

## 2.5 Parent Selection Procedure

In each generation,  $\lambda$  offspring individuals are generated by mutating the correspondent parent. The decision on how many offspring are produced from each population member is made in `parentSelection` procedure. At the beginning, it is assumed that none offspring is generated by each individual. Then, binary tournament selection based on scalarizing fitness is performed to identify which population members will be mutated in order to produce offspring. This selection process is based on the idea proposed in [15] and fosters promising individuals to produce more offspring.

The procedure starts by normalizing the objective values of all individuals in the population as defined in (2). Then, the selection process is performed in two stages. In the first stage, only leaders of the current population are considered. For each weight vector, two individuals are randomly selected and the fitness of the corresponding individual is calculated as defined in (3). Individual having smaller fitness value is a winner and the number of times it is mutated is augmented by one. Thereafter, all leaders are removed from the population and for each weight vector binary tournament selection is performed on the resulting population in the same way as it is done for leaders. It should be noted that some population members might be mutated several times as a result of this procedure, while others will not produce any offspring at all.

## 2.6 Mutation Procedure

After identifying which population members have to be mutated, offspring are generated in `mutation` procedure. For the corresponding individual, the mutation is performed as follows:

$$\mathbf{x}' = \mathbf{x} + \sigma \mathbf{S} \mathbf{v} \quad (6)$$

where  $\sigma$  is the step size,  $\mathbf{S}$  is the search matrix and  $\mathbf{v}$  is a column vector of random numbers sampled from the uniform distribution ( $\forall i \in \{1, 2\} : v_i \sim \mathbb{U}(0, 1)$ ). In order to guarantee that each new solution  $\mathbf{x}' = (x'_1, \dots, x'_n)^T$  belongs to  $\Omega$ , projection is applied to each component of the decision vector:

$$\mathbf{x}' = \min\{\max\{\mathbf{x}', \mathbf{l}\}, \mathbf{u}\}$$

After offspring  $\mathbf{x}'$  is repaired, it is evaluated and added to the population.

## 2.7 Environmental Selection Procedure

At the end of each generation,  $\mu$  fittest individuals are selected in `environmentalSelection` procedure from the enlarged population. DDMOA2 uses the selection mechanism proposed in [14].

First, the objective values of all individuals are normalized as defined in (2). Next, the following steps are performed:

1. matrix  $\mathbf{M}$  is calculated, which stores metrics for each population member on each weight vector (for each population member, a metric on a corresponding weight vector is computed as defined in (3));
2. for each column (weight vector), the minimum and second smallest metric value are found;
3. for each column, metric values are scaled by the minimum value found, except for the row which gave the minimum value, this result is scaled by the second lowest value;
4. for each row (population member), the minimum scaled value is found, this value represents individual's fitness;
5. the resulting column vector is sorted, and  $\mu$  individuals with the smallest fitness values are selected.

Normalizing objective values allows to cope with differently scaled objectives, while the weighted Chebyshev method can find optimal solutions in convex and nonconvex regions of the Pareto front. When the stopping criterion is met, the algorithm returns its final population.

## 3 Performance Assessment

In order to validate our algorithm, we compare its performance with that produced by DDMOA [11] and four state-of-the-art EMO algorithms NSGA-II [7], IBEA [24] and MOEA/D [18], and MSOPS2 [14] on the DTLZ test suite [8] with 30 decision variables having between 2 and 8 objectives.

### 3.1 Performance Indicators

To assess the outcomes produced by the algorithms we use the following quality indicators:

- The unary additive epsilon indicator [26]. This indicator is based on the concept of additive  $\epsilon$ -dominance and defined with respect to a reference set  $R$  as:

$$I_\epsilon = \inf_{\epsilon \in \mathbb{R}} \{ \forall \mathbf{a}' \in R \exists \mathbf{a} \in A : \mathbf{a} \preceq_\epsilon \mathbf{a}' \}. \tag{7}$$

The epsilon indicator gives the minimum factor  $\epsilon$  such that any objective vector in  $R$  is  $\epsilon$ -dominated by at least one objective vector in  $A$ . Smaller values of  $I_\epsilon$  are preferable.

- The hypervolume indicator [25]. It can be defined as the Lebesgue measure  $\Lambda$  of the union of hypercuboids in the objective space:

$$I_H = \Lambda \left( \bigcup_{\mathbf{a} \in A \wedge \mathbf{r} \in R} \{ f_1(\mathbf{a}'), \dots, f_m(\mathbf{a}') : \mathbf{a} < \mathbf{a}' < \mathbf{r} \} \right) \tag{8}$$

where  $A = \{\mathbf{a}_1, \dots, \mathbf{a}_{|A|}\}$  is an approximation set, and  $R = \{\mathbf{r}_1, \dots, \mathbf{r}_{|R|}\}$  is an appropriately chosen reference set. The higher value of  $I_H$ , the more preferable an approximation set is.

- Inverted generational distance (IGD) [3]. If the size of the discretized Pareto front  $\mathcal{PF} = \{\mathbf{a}'_1, \dots, \mathbf{a}'_{|\mathcal{PF}|}\}$  is large enough to represent the true Pareto optimal front, this indicator could measure both the diversity and convergence of an approximation set  $A = \{\mathbf{a}_1, \dots, \mathbf{a}_{|A|}\}$  and is defined as:

$$I_{IGD} = \frac{\sum_{\mathbf{a}' \in \mathcal{PF}} d(\mathbf{a}', A)}{|\mathcal{PF}|} \tag{9}$$

where  $d(\mathbf{a}', A)$  is the minimum Euclidean distance between  $\mathbf{a}'$  and the points in an approximation set  $A$ . Smaller values of  $I_{IGD}$  are preferable.

To calculate the epsilon and IGD indicators, for all problems we generate 1,000 uniformly distributed points along the Pareto front. To calculate the hypervolume indicator, we use the nadir point for the corresponding problem as a reference point. Thus, only those members of the approximation set, which dominate the nadir point, are used for calculating the hypervolume. If there is no point in the approximation set that dominates the nadir point, then the hypervolume is equal to zero. Further, solutions used to calculate the hypervolume are normalized using the utopian and nadir points for the corresponding problem. When the number of objectives is more than 6 the hypervolume is approximated using  $10^6$  uniformly sampled points, otherwise the exact computation of the hypervolume is used.

Since we are dealing with stochastic algorithms we want to provide the results with statistical confidence. Therefore, we perform pairwise comparison of the algorithms with respect to the quality indicators using a nonparametric Wilcoxon rank sum test [13]. All tests are performed at significance level of  $\alpha = 0.05$ .



### 3.2 Experimental Setup

The MATLAB<sup>®</sup> implementations of DDMOA2, DDMOA, and MSOPS2<sup>1</sup> are used, IBEA is used within the PISA [2] framework,<sup>2</sup> whereas NSGA-II and MOEA/D are used within the jMetal [12] framework.<sup>3</sup> For each algorithm, 30 independent runs are performed on each problem with a population size of  $\mu = 200$ , running for 60,000 function evaluations. All other parameters for the algorithms use the default settings. The default parameters for DDMOA2 are: the initial step size for local search  $\delta^{(0)} = 0.4$ , the initial step size for reproduction  $\sigma^{(0)} = 5$ , the number of subpopulations  $\alpha = 5$ , and the tolerance for step size  $\delta_{\text{tol}} = 10^{-3}$ .

### 3.3 Experimental Results

Tables 1, 2, and 3 present the median values of the quality indicators and the statistical comparison of the algorithms with respect to the obtained indicator values. In Table 1, small values of the epsilon indicator suggest that approximation sets produced by DDMOA2 are relatively close to the true Pareto fronts. From Table 2, it can be seen that the hypervolume indicator values calculated for the results produced by DDMOA2 are always greater than zero. This suggests that approximations returned by DDMOA2 are within the bounds of the Pareto fronts for the considered problems. Analyzing Table 3, conclusions consistent with those based on the previous two indicators can be drawn from the IGD values. DDMOA2 performs statistically better on DTLZ1,3 in all dimensions, providing a competitive performance on the other problems and being statistically better than DDMOA on the majority of problems having more than three objectives. Thus, the obtained indicator values emphasize that DDMOA2 can provide adequate approximations in terms of the proximity to the Pareto front as well as the diversity of the obtained solutions. From the obtained results, we can conclude that DDMOA2 performs generally better on multimodal problems in all dimensions. This directly results from the DDMOA2 operator for performing the search in the decision space. The strategy based on local search to find descend directions for two randomly chosen objectives and using their linear combination to generate offspring appears to be an effective to explore the search space. Furthermore, the introduced scalarizing fitness assignment allows to deal with problems having more than three objectives. On the other hand, the performance of DDMOA becomes increasingly poor when the number of objectives grows, due to the fact that it does not possess a proper selection operator. Since DDMOA uses local search for all objectives, an extensive exploration of the search space is carried out, allowing to find a number of solutions in the promising regions even without a proper selection operator for dealing with high-dimensional problems. This results in a competitive performance on some four and five-objective problems. However, this approach becomes inefficient in higher dimensions, spending the whole computational budget during a small number of generations. MOEA/D works better on the problems with degenerated Pareto fronts beyond three objectives, whereas DDMOA2 is the second best concerning all the indicators on DTLZ6. IBEA produces the best performance with respect to the epsilon and hypervolume indicators on the DTLZ2,4, and 7 test problems having more than three objectives. It is not surprising,

<sup>1</sup>available at <http://code.evanhughes.org>

<sup>2</sup>available at <http://www.tik.ee.ethz.ch/pisa>

<sup>3</sup>available at <http://jmetal.sourceforge.net>

**Table 1** Median values of the epsilon indicator after 30 runs (the lower the better)

	DDMOA2	DDMOA	NSGA-II	IBEA	MOEA/D	MSOPS2
2-objectives						
DTLZ1	0.002 <sup>2,3,4,5,6</sup>	1.81 <sup>3,4,5,6</sup>	5.50 <sup>6</sup>	3.28 <sup>3,6</sup>	5.56 <sup>6</sup>	149.20
DTLZ2	0.003 <sup>2,3,4,6</sup>	0.004 <sup>3,4,6</sup>	0.01	0.01	0.003 <sup>1,2,3,4,6</sup>	0.01
DTLZ3	0.003 <sup>2,3,4,5,6</sup>	10.30 <sup>3,4,5,6</sup>	13.95 <sup>5,6</sup>	14.85 <sup>5,6</sup>	38.43 <sup>6</sup>	401.21
DTLZ4	0.003 <sup>2,3,4,5,6</sup>	0.004 <sup>3,4,6</sup>	0.01 <sup>6</sup>	0.01 <sup>6</sup>	0.004 <sup>3,4,6</sup>	0.41
DTLZ5	0.003 <sup>2,3,4,6</sup>	0.004 <sup>3,4,6</sup>	0.01	0.01	0.003 <sup>1,2,3,4,6</sup>	0.01 <sup>3,4</sup>
DTLZ6	0.003 <sup>2,3,4,6</sup>	0.004 <sup>3,4,6</sup>	1.22 <sup>6</sup>	0.23 <sup>3,6</sup>	0.003 <sup>1,2,3,4,6</sup>	12.39
DTLZ7	0.01 <sup>4,5,6</sup>	0.004 <sup>1,3,4,5,6</sup>	0.01 <sup>4,5,6</sup>	0.02 <sup>6</sup>	0.02 <sup>6</sup>	3.77
3-objectives						
DTLZ1	0.03 <sup>2,3,4,5,6</sup>	0.79 <sup>3,4,5,6</sup>	14.87 <sup>6</sup>	2.83 <sup>3,6</sup>	8.92 <sup>3,6</sup>	187.81
DTLZ2	0.10 <sup>5</sup>	0.05 <sup>1,3,5</sup>	0.09 <sup>1,5</sup>	0.05 <sup>1,3,5</sup>	0.10	0.05 <sup>1,3,4,5</sup>
DTLZ3	0.10 <sup>2,3,4,5,6</sup>	3.06 <sup>3,4,5,6</sup>	36.93 <sup>6</sup>	12.98 <sup>3,6</sup>	18.69 <sup>6</sup>	592.26
DTLZ4	0.25 <sup>6</sup>	0.05 <sup>1,3,5,6</sup>	0.09 <sup>1,5,6</sup>	0.05 <sup>1,3,5,6</sup>	0.25 <sup>6</sup>	0.76
DTLZ5	0.02 <sup>6</sup>	0.003 <sup>1,3,4,5,6</sup>	0.01 <sup>1,5,6</sup>	0.01 <sup>1,5,6</sup>	0.02 <sup>1,6</sup>	0.02
DTLZ6	0.02 <sup>3,4,6</sup>	0.003 <sup>1,3,4,5,6</sup>	4.08 <sup>6</sup>	0.25 <sup>3,6</sup>	0.02 <sup>1,3,4,6</sup>	11.89
DTLZ7	0.10 <sup>5,6</sup>	0.06 <sup>1,3,5,6</sup>	0.09 <sup>1,5,6</sup>	0.06 <sup>1,3,5,6</sup>	0.24 <sup>6</sup>	5.41
4-objectives						
DTLZ1	0.07 <sup>2,3,4,5,6</sup>	0.67 <sup>3,4,5,6</sup>	95.80 <sup>6</sup>	2.61 <sup>3,6</sup>	4.25 <sup>3,6</sup>	157.63
DTLZ2	0.16 <sup>3,5</sup>	0.13 <sup>1,3,5</sup>	0.18	0.11 <sup>1,2,3,5</sup>	0.16 <sup>3</sup>	0.11 <sup>1,2,3,5</sup>
DTLZ3	0.17 <sup>2,3,4,5,6</sup>	0.75 <sup>3,4,5,6</sup>	173.66 <sup>6</sup>	11.19 <sup>3,6</sup>	7.70 <sup>3,6</sup>	624.71
DTLZ4	0.25 <sup>6</sup>	0.15 <sup>1,3,5,6</sup>	0.18 <sup>1,5,6</sup>	0.11 <sup>1,2,3,5,6</sup>	0.25 <sup>6</sup>	0.46
DTLZ5	0.05 <sup>2,3</sup>	0.21	0.10 <sup>2</sup>	0.03 <sup>1,2,3</sup>	0.02 <sup>1,2,3,4</sup>	0.02 <sup>1,2,3,4,5</sup>
DTLZ6	0.03 <sup>2,3,4,6</sup>	0.15 <sup>3,4,6</sup>	9.83 <sup>6</sup>	0.35 <sup>3,6</sup>	0.02 <sup>1,2,3,4,6</sup>	12.24
DTLZ7	0.25 <sup>2,5,6</sup>	0.61 <sup>6</sup>	0.40 <sup>2,5,6</sup>	0.11 <sup>1,2,3,5,6</sup>	0.66 <sup>6</sup>	7.62
5-objectives						
DTLZ1	0.13 <sup>2,3,4,5,6</sup>	0.53 <sup>3,4,5,6</sup>	285.73	2.41 <sup>3,6</sup>	3.70 <sup>3,6</sup>	155.77 <sup>3</sup>
DTLZ2	0.23 <sup>2,3,5</sup>	0.24 <sup>3,5</sup>	0.56	0.14 <sup>1,2,3,5,6</sup>	0.32 <sup>3</sup>	0.17 <sup>1,2,3,5</sup>
DTLZ3	0.28 <sup>2,3,4,5,6</sup>	0.56 <sup>3,4,5,6</sup>	540.15	12.69 <sup>3,6</sup>	12.57 <sup>3,6</sup>	529.19
DTLZ4	0.27 <sup>3,5,6</sup>	0.24 <sup>1,3,5,6</sup>	0.48	0.14 <sup>1,2,3,5,6</sup>	0.29 <sup>3,6</sup>	0.53
DTLZ5	0.12 <sup>2,3</sup>	0.36	0.33	0.05 <sup>1,2,3</sup>	0.03 <sup>1,2,3,4,6</sup>	0.03 <sup>1,2,3,4</sup>
DTLZ6	0.18 <sup>2,3,4,6</sup>	0.24 <sup>3,4,6</sup>	12.16	0.48 <sup>3,6</sup>	0.02 <sup>1,2,3,4,6</sup>	12.22
DTLZ7	1.29 <sup>2,6</sup>	2.52 <sup>6</sup>	0.62 <sup>1,2,5,6</sup>	0.42 <sup>1,2,3,5,6</sup>	0.77 <sup>1,2,6</sup>	9.43
6-objectives						
DTLZ1	0.14 <sup>2,3,4,5,6</sup>	0.26 <sup>3,4,5,6</sup>	367.58	2.25 <sup>3,6</sup>	1.67 <sup>3,6</sup>	143.80 <sup>3</sup>
DTLZ2	0.28 <sup>2,3,5</sup>	0.42 <sup>3,5</sup>	1.75	0.18 <sup>1,2,3,5,6</sup>	0.56 <sup>3</sup>	0.21 <sup>1,2,3,5</sup>
DTLZ3	0.35 <sup>2,3,4,5,6</sup>	0.71 <sup>3,4,5,6</sup>	985.27	9.33 <sup>3,6</sup>	6.95 <sup>3,6</sup>	546.73 <sup>3</sup>
DTLZ4	0.30 <sup>2,3,5</sup>	0.36 <sup>3</sup>	2.39	0.18 <sup>1,2,3,5,6</sup>	0.35 <sup>3</sup>	0.24 <sup>2,3</sup>
DTLZ5	0.09 <sup>2,3</sup>	0.52 <sup>3</sup>	1.07	0.06 <sup>1,2,3</sup>	0.02 <sup>1,2,3,4,6</sup>	0.04 <sup>1,2,3,4</sup>
DTLZ6	0.12 <sup>2,3,4,6</sup>	0.71 <sup>3,6</sup>	12.15	0.61 <sup>2,3,6</sup>	0.02 <sup>1,2,3,4,6</sup>	12.34
DTLZ7	1.31 <sup>2,6</sup>	2.90 <sup>6</sup>	0.79 <sup>1,2,6</sup>	0.55 <sup>1,2,3,5,6</sup>	0.81 <sup>1,2,6</sup>	11.40
7-objectives						
DTLZ1	0.16 <sup>2,3,4,5,6</sup>	0.37 <sup>3,4,5,6</sup>	380.68	1.64 <sup>3,5,6</sup>	7.74 <sup>3,6</sup>	129.96 <sup>3</sup>
DTLZ2	0.33 <sup>2,3,5</sup>	0.66 <sup>3</sup>	2.43	0.21 <sup>1,2,3,5,6</sup>	0.61 <sup>2,3</sup>	0.23 <sup>1,2,3,5</sup>
DTLZ3	0.61 <sup>2,3,4,5,6</sup>	0.66 <sup>3,4,5,6</sup>	1136.72	8.44 <sup>3,6</sup>	6.86 <sup>3,6</sup>	506.88 <sup>3</sup>
DTLZ4	0.33 <sup>2,3,5</sup>	0.54 <sup>3</sup>	2.74	0.21 <sup>1,2,3,5,6</sup>	0.37 <sup>2,3</sup>	0.28 <sup>2,3,5</sup>
DTLZ5	0.12 <sup>2,3</sup>	0.61 <sup>3</sup>	1.61	0.06 <sup>1,2,3</sup>	0.02 <sup>1,2,3,4,6</sup>	0.06 <sup>1,2,3</sup>
DTLZ6	0.13 <sup>2,3,4,6</sup>	0.83 <sup>3,6</sup>	11.77	0.73 <sup>3,6</sup>	0.01 <sup>1,2,3,4,6</sup>	11.88
DTLZ7	3.72 <sup>2,6</sup>	5.23 <sup>6</sup>	1.34 <sup>1,2,6</sup>	0.64 <sup>1,2,3,6</sup>	0.81 <sup>1,2,3,6</sup>	12.27
8-objectives						
DTLZ1	0.17 <sup>2,3,4,5,6</sup>	1.85 <sup>3,5,6</sup>	399.04	1.37 <sup>3,5,6</sup>	8.06 <sup>3,6</sup>	116.28 <sup>3</sup>
DTLZ2	0.46 <sup>2,3,5</sup>	1.01 <sup>3</sup>	2.62	0.24 <sup>1,2,3,5,6</sup>	0.60 <sup>2,3</sup>	0.25 <sup>1,2,3,5</sup>
DTLZ3	0.60 <sup>2,3,4,5,6</sup>	1.00 <sup>3,6</sup>	1209.79	8.14 <sup>3,6</sup>	13.54 <sup>3,6</sup>	442.80 <sup>3</sup>
DTLZ4	0.34 <sup>2,3,5</sup>	0.68 <sup>3</sup>	2.74	0.23 <sup>1,2,3,5,6</sup>	0.41 <sup>2,3</sup>	0.28 <sup>1,2,3,5</sup>
DTLZ5	0.29 <sup>2,3</sup>	0.76 <sup>3</sup>	1.62	0.07 <sup>1,2,3</sup>	0.02 <sup>1,2,3,4,6</sup>	0.04 <sup>1,2,3,4</sup>
DTLZ6	0.23 <sup>2,3,4,6</sup>	1.00 <sup>3,6</sup>	11.16	0.85 <sup>3,6</sup>	0.02 <sup>1,2,3,4,6</sup>	11.19
DTLZ7	3.42 <sup>2,6</sup>	4.89 <sup>6</sup>	2.14 <sup>1,2,6</sup>	0.76 <sup>1,2,3,5,6</sup>	0.80 <sup>1,2,3,6</sup>	13.55

The superscripts 1, 2, 3, 4, 5, and 6 indicate whether the respective algorithm performs significantly better than DDMOA2, DDMOA, NSGA-II, IBEA, MOEA/D, and MSOPS2 respectively

**Table 2** Median values of the hypervolume indicator after 30 runs (the higher the better)

	DDMOA2	DDMOA	NSGA-II	IBEA	MOEA/D	MSOPS2
2-objectives						
DTLZ1	0.50 <sup>2,3,4,5,6</sup>	0.00	0.00 <sup>2</sup>	0.00 <sup>2</sup>	0.00	0.00 <sup>2</sup>
DTLZ2	0.21 <sup>3,4,5</sup>	0.21 <sup>1,3,4,5,6</sup>	0.21	0.21	0.21 <sup>3,4</sup>	0.21 <sup>1,3,4,5</sup>
DTLZ3	0.21 <sup>2,3,4,5,6</sup>	0.00	0.00	0.00	0.00	0.00
DTLZ4	0.21 <sup>3,4,5,6</sup>	0.21 <sup>1,3,4,5,6</sup>	0.21 <sup>5,6</sup>	0.21 <sup>5,6</sup>	0.21 <sup>6</sup>	0.00
DTLZ5	0.21 <sup>3,4,5</sup>	0.21 <sup>1,3,4,5,6</sup>	0.21 <sup>4</sup>	0.21	0.21 <sup>4</sup>	0.21 <sup>1,3,4,5</sup>
DTLZ6	0.21 <sup>3,4,6</sup>	0.21 <sup>1,3,4,5,6</sup>	0.00	0.03 <sup>3,6</sup>	0.21 <sup>1,3,4,6</sup>	0.00
DTLZ7	0.34 <sup>3,4,5,6</sup>	0.34 <sup>1,3,4,5,6</sup>	0.34 <sup>4,5,6</sup>	0.33 <sup>5,6</sup>	0.33 <sup>6</sup>	0.00
3-objectives						
DTLZ1	0.78 <sup>2,3,4,5,6</sup>	0.00	0.00 <sup>2</sup>	0.00 <sup>2</sup>	0.00 <sup>3,4</sup>	0.00 <sup>2,5</sup>
DTLZ2	0.41 <sup>3,5</sup>	0.43 <sup>1,3,5</sup>	0.40 <sup>5</sup>	0.43 <sup>1,2,3,5,6</sup>	0.40	0.43 <sup>1,3,5</sup>
DTLZ3	0.41 <sup>2,3,4,5,6</sup>	0.00	0.00 <sup>2</sup>	0.00 <sup>2</sup>	0.00 <sup>3,4</sup>	0.00 <sup>2,5</sup>
DTLZ4	0.32 <sup>5,6</sup>	0.43 <sup>1,3,5,6</sup>	0.40 <sup>5,6</sup>	0.43 <sup>1,2,3,5,6</sup>	0.32 <sup>6</sup>	0.18
DTLZ5	0.22 <sup>5</sup>	0.22 <sup>1,3,4,5,6</sup>	0.22 <sup>1,5,6</sup>	0.22 <sup>1,3,5,6</sup>	0.22	0.22 <sup>5</sup>
DTLZ6	0.22 <sup>3,4,6</sup>	0.22 <sup>1,3,4,5,6</sup>	0.00	0.07 <sup>3,6</sup>	0.22 <sup>1,3,4,6</sup>	0.00
DTLZ7	0.34 <sup>5,6</sup>	0.36 <sup>1,3,4,5,6</sup>	0.34 <sup>5,6</sup>	0.36 <sup>1,3,5,6</sup>	0.26 <sup>6</sup>	0.00
4-objectives						
DTLZ1	0.88 <sup>2,3,4,5,6</sup>	0.00	0.00	0.00	0.00 <sup>3,4</sup>	0.00 <sup>5</sup>
DTLZ2	0.50 <sup>3,5</sup>	0.55 <sup>1,3,5</sup>	0.46	0.60 <sup>1,2,3,5,6</sup>	0.47 <sup>3</sup>	0.57 <sup>1,2,3,5</sup>
DTLZ3	0.51 <sup>2,3,4,5,6</sup>	0.01 <sup>3,4,5,6</sup>	0.00	0.00	0.00 <sup>3,4</sup>	0.00 <sup>5</sup>
DTLZ4	0.47 <sup>6</sup>	0.56 <sup>1,3,5,6</sup>	0.47	0.60 <sup>1,2,3,5,6</sup>	0.48	0.41
DTLZ5	0.18 <sup>2,3</sup>	0.09	0.15 <sup>2</sup>	0.18 <sup>1,2,3</sup>	0.19 <sup>1,2,3,4</sup>	0.19 <sup>1,2,3,4,5</sup>
DTLZ6	0.18 <sup>2,3,4,6</sup>	0.11 <sup>3,4,6</sup>	0.00	0.03 <sup>3,6</sup>	0.19 <sup>1,2,3,4,6</sup>	0.00
DTLZ7	0.34 <sup>3,5,6</sup>	0.33 <sup>3,5,6</sup>	0.29 <sup>5,6</sup>	0.37 <sup>1,2,3,5,6</sup>	0.14 <sup>6</sup>	0.00
5-objectives						
DTLZ1	0.91 <sup>2,3,4,5,6</sup>	0.00	0.00	0.00	0.00 <sup>2,3,4</sup>	0.00 <sup>5</sup>
DTLZ2	0.56 <sup>2,3,5</sup>	0.53 <sup>3,5</sup>	0.10	0.71 <sup>1,2,3,5,6</sup>	0.38 <sup>3</sup>	0.66 <sup>1,2,3,5</sup>
DTLZ3	0.55 <sup>2,3,4,5,6</sup>	0.49 <sup>3,4,5,6</sup>	0.00	0.00	0.00 <sup>3,4</sup>	0.00 <sup>5</sup>
DTLZ4	0.55 <sup>3,5,6</sup>	0.60 <sup>1,3,5,6</sup>	0.10	0.71 <sup>1,2,3,5,6</sup>	0.52 <sup>3,6</sup>	0.41 <sup>3</sup>
DTLZ5	0.13 <sup>2,3</sup>	0.05	0.04	0.15 <sup>1,2,3</sup>	0.16 <sup>1,2,3,4</sup>	0.17 <sup>1,2,3,4,5</sup>
DTLZ6	0.11 <sup>2,3,4,6</sup>	0.06 <sup>3,4,6</sup>	0.00	0.01 <sup>3,6</sup>	0.17 <sup>1,2,3,4,6</sup>	0.00
DTLZ7	0.27 <sup>2,3,5,6</sup>	0.26 <sup>3,5,6</sup>	0.17 <sup>5,6</sup>	0.32 <sup>1,2,3,5,6</sup>	0.03 <sup>6</sup>	0.00
6-objectives						
DTLZ1	0.93 <sup>2,3,4,5,6</sup>	0.28 <sup>3,4,5,6</sup>	0.00	0.00	0.00 <sup>3,4</sup>	0.00 <sup>5</sup>
DTLZ2	0.60 <sup>2,3,5</sup>	0.29 <sup>3</sup>	0.00	0.78 <sup>1,2,3,5,6</sup>	0.29 <sup>3</sup>	0.71 <sup>1,2,3,5</sup>
DTLZ3	0.59 <sup>2,3,4,5,6</sup>	0.21 <sup>3,4,5,6</sup>	0.00	0.00	0.00 <sup>3,4</sup>	0.00 <sup>5</sup>
DTLZ4	0.64 <sup>2,3,5</sup>	0.50 <sup>3</sup>	0.00	0.78 <sup>1,2,3,5,6</sup>	0.52 <sup>3</sup>	0.67 <sup>2,3,5</sup>
DTLZ5	0.13 <sup>2,3</sup>	0.01 <sup>3</sup>	0.00	0.14 <sup>1,2,3</sup>	0.15 <sup>1,2,3,4</sup>	0.15 <sup>1,2,3,4</sup>
DTLZ6	0.11 <sup>2,3,4,6</sup>	0.05 <sup>3,6</sup>	0.00	0.00 <sup>3,6</sup>	0.15 <sup>1,2,3,4,6</sup>	0.00
DTLZ7	0.22 <sup>2,3,5,6</sup>	0.12 <sup>3,5,6</sup>	0.05 <sup>5,6</sup>	0.28 <sup>1,2,3,5,6</sup>	0.01 <sup>6</sup>	0.00
7-objectives						
DTLZ1	0.91 <sup>2,3,4,5,6</sup>	0.18 <sup>3,4,5,6</sup>	0.00	0.00	0.00 <sup>3,4</sup>	0.00 <sup>5</sup>
DTLZ2	0.65 <sup>2,3,5</sup>	0.05 <sup>3</sup>	0.00	0.83 <sup>1,2,3,5,6</sup>	0.28 <sup>2,3</sup>	0.77 <sup>1,2,3,5</sup>
DTLZ3	0.50 <sup>2,3,4,5,6</sup>	0.44 <sup>3,4,5,6</sup>	0.00	0.00	0.00 <sup>3,4</sup>	0.00 <sup>5</sup>
DTLZ4	0.69 <sup>2,3,5</sup>	0.25 <sup>3</sup>	0.00	0.83 <sup>1,2,3,5,6</sup>	0.58 <sup>2,3</sup>	0.72 <sup>2,3,5</sup>
DTLZ5	0.11 <sup>2,3</sup>	0.02 <sup>3</sup>	0.00	0.12 <sup>1,2,3</sup>	0.14 <sup>1,2,3,4,6</sup>	0.14 <sup>1,2,3,4</sup>
DTLZ6	0.11 <sup>2,3,4,6</sup>	0.00	0.00 <sup>2</sup>	0.00 <sup>3,6</sup>	0.15 <sup>1,2,3,4,6</sup>	0.00 <sup>2</sup>
DTLZ7	0.20 <sup>3,5,6</sup>	0.18 <sup>3,5,6</sup>	0.01 <sup>5,6</sup>	0.25 <sup>1,2,3,5,6</sup>	0.00 <sup>6</sup>	0.00
8-objectives						
DTLZ1	0.88 <sup>2,3,4,5,6</sup>	0.00	0.00 <sup>2</sup>	0.00 <sup>2</sup>	0.00 <sup>3,4</sup>	0.00 <sup>2,5</sup>
DTLZ2	0.63 <sup>2,3,5</sup>	0.00	0.00 <sup>2</sup>	0.86 <sup>1,2,3,5,6</sup>	0.32 <sup>2,3</sup>	0.78 <sup>1,2,3,5</sup>
DTLZ3	0.56 <sup>2,3,4,5,6</sup>	0.00	0.00 <sup>2</sup>	0.00 <sup>2</sup>	0.00 <sup>3,4</sup>	0.00 <sup>2,5</sup>
DTLZ4	0.75 <sup>2,3,5</sup>	0.09 <sup>3</sup>	0.00	0.87 <sup>1,2,3,5,6</sup>	0.59 <sup>2,3</sup>	0.80 <sup>1,2,3,5</sup>
DTLZ5	0.06 <sup>2,3</sup>	0.00	0.00 <sup>2</sup>	0.11 <sup>1,2,3</sup>	0.14 <sup>1,2,3,4</sup>	0.14 <sup>1,2,3,4</sup>
DTLZ6	0.08 <sup>2,3,4,6</sup>	0.00	0.00 <sup>2</sup>	0.00 <sup>2</sup>	0.14 <sup>1,2,3,4,6</sup>	0.00 <sup>2</sup>
DTLZ7	0.17 <sup>2,3,5,6</sup>	0.10 <sup>3,5,6</sup>	0.00 <sup>6</sup>	0.23 <sup>1,2,3,5,6</sup>	0.00 <sup>3,6</sup>	0.00

The superscripts 1, 2, 3, 4, 5, and 6 indicate whether the respective algorithm performs significantly better than DDMOA2, DDMOA, NSGA-II, IBEA, MOEA/D, and MSOPS2 respectively

**Table 3** Median values of the IGD indicator after 30 runs (the lower the better)

	DDMOA2	DDMOA	NSGA-II	IBEA	MOEA/D	MSOPS2
2-objectives						
DTLZ1	0.001 <sup>2,3,4,5,6</sup>	2.49 <sup>3,4,5,6</sup>	7.39 <sup>6</sup>	3.64 <sup>3,5,6</sup>	7.14 <sup>6</sup>	187.18
DTLZ2	0.002 <sup>3,4,5,6</sup>	0.002 <sup>1,3,4,5,6</sup>	0.002 <sup>4</sup>	0.01	0.002 <sup>3,4,6</sup>	0.002 <sup>3,4</sup>
DTLZ3	0.002 <sup>2,3,4,5,6</sup>	14.23 <sup>3,5,6</sup>	18.22 <sup>5,6</sup>	14.65 <sup>3,5,6</sup>	43.19 <sup>6</sup>	484.43
DTLZ4	0.002 <sup>3,4,5,6</sup>	0.002 <sup>3,4,5,6</sup>	0.002 <sup>4,6</sup>	0.01 <sup>6</sup>	0.002 <sup>3,4,6</sup>	0.43
DTLZ5	0.002 <sup>3,4,5,6</sup>	0.002 <sup>1,3,4,5,6</sup>	0.002 <sup>4</sup>	0.01	0.002 <sup>3,4,6</sup>	0.002 <sup>3,4</sup>
DTLZ6	0.002 <sup>2,3,4,6</sup>	0.002 <sup>3,4,6</sup>	1.46 <sup>6</sup>	0.24 <sup>3,6</sup>	0.002 <sup>1,2,3,4,6</sup>	16.60
DTLZ7	0.003 <sup>4,5,6</sup>	0.002 <sup>1,3,4,5,6</sup>	0.002 <sup>1,4,5,6</sup>	0.01 <sup>6</sup>	0.01 <sup>4,6</sup>	2.95
3-objectives						
DTLZ1	0.02 <sup>2,3,4,5,6</sup>	1.18 <sup>3,4,5,6</sup>	21.17 <sup>6</sup>	3.46 <sup>3,6</sup>	12.68 <sup>3,6</sup>	264.76
DTLZ2	0.05 <sup>3,4,5</sup>	0.04 <sup>1,3,4,5</sup>	0.05 <sup>4</sup>	0.07	0.05 <sup>3,4</sup>	0.04 <sup>1,2,3,4,5</sup>
DTLZ3	0.05 <sup>2,3,4,5,6</sup>	4.54 <sup>3,4,5,6</sup>	50.93 <sup>6</sup>	12.53 <sup>3,6</sup>	20.56 <sup>6</sup>	755.22
DTLZ4	0.29 <sup>6</sup>	0.04 <sup>1,3,4,5,6</sup>	0.05 <sup>1,4,5,6</sup>	0.07 <sup>6</sup>	0.29 <sup>6</sup>	0.34
DTLZ5	0.01 <sup>5,6</sup>	0.002 <sup>1,3,4,5,6</sup>	0.002 <sup>1,4,5,6</sup>	0.01 <sup>1,5,6</sup>	0.01	0.01
DTLZ6	0.01 <sup>3,4,6</sup>	0.002 <sup>1,3,4,5,6</sup>	5.92 <sup>6</sup>	0.29 <sup>3,6</sup>	0.01 <sup>1,3,4,6</sup>	17.70
DTLZ7	0.09 <sup>4,5,6</sup>	0.04 <sup>1,3,4,5,6</sup>	0.06 <sup>1,4,5,6</sup>	0.10 <sup>6</sup>	0.12 <sup>6</sup>	3.64
4-objectives						
DTLZ1	0.05 <sup>2,3,4,5,6</sup>	1.03 <sup>3,4,5,6</sup>	142.99 <sup>6</sup>	3.23 <sup>3,6</sup>	6.07 <sup>3,6</sup>	238.22
DTLZ2	0.15 <sup>4,5</sup>	0.11 <sup>1,3,4,5</sup>	0.13 <sup>1,4,5</sup>	0.15	0.15 <sup>4</sup>	0.11 <sup>1,2,3,4,5</sup>
DTLZ3	0.17 <sup>2,3,4,5,6</sup>	0.76 <sup>3,4,5,6</sup>	256.79 <sup>6</sup>	10.94 <sup>3,6</sup>	11.56 <sup>3,6</sup>	811.13
DTLZ4	0.32	0.18 <sup>1,5</sup>	0.13 <sup>1,2,4,5,6</sup>	0.15 <sup>1,2,5</sup>	0.31	0.23
DTLZ5	0.02 <sup>2,3,4</sup>	0.17	0.06 <sup>2</sup>	0.03 <sup>2,3</sup>	0.01 <sup>1,2,3,4</sup>	0.01 <sup>1,2,3,4,5</sup>
DTLZ6	0.02 <sup>2,3,4,6</sup>	0.18 <sup>3,4,6</sup>	15.58 <sup>6</sup>	0.44 <sup>3,6</sup>	0.01 <sup>1,2,3,4,6</sup>	18.54
DTLZ7	0.24 <sup>2,5,6</sup>	0.29 <sup>6</sup>	0.18 <sup>1,2,4,5,6</sup>	0.23 <sup>1,2,5,6</sup>	0.30 <sup>6</sup>	4.78
5-objectives						
DTLZ1	0.10 <sup>2,3,4,5,6</sup>	0.52 <sup>3,4,5,6</sup>	420.76	3.03 <sup>3,6</sup>	6.04 <sup>3,6</sup>	233.29 <sup>3</sup>
DTLZ2	0.27 <sup>3,5</sup>	0.21 <sup>1,3,4,5</sup>	0.45	0.23 <sup>1,3,5</sup>	0.28 <sup>3</sup>	0.19 <sup>1,2,3,4,5</sup>
DTLZ3	0.32 <sup>2,3,4,5,6</sup>	0.43 <sup>3,4,5,6</sup>	745.81	12.30 <sup>3,6</sup>	22.60 <sup>3,6</sup>	749.27
DTLZ4	0.37 <sup>3,5</sup>	0.30 <sup>1,3,5</sup>	0.43	0.23 <sup>1,2,3,5,6</sup>	0.41	0.37
DTLZ5	0.05 <sup>2,3</sup>	0.29	0.28	0.05 <sup>2,3</sup>	0.01 <sup>1,2,3,4,6</sup>	0.02 <sup>1,2,3,4</sup>
DTLZ6	0.15 <sup>2,3,4,6</sup>	0.36 <sup>3,4,6</sup>	20.63	0.63 <sup>3,6</sup>	0.01 <sup>1,2,3,4,6</sup>	19.24 <sup>3</sup>
DTLZ7	0.49 <sup>2,5,6</sup>	0.60 <sup>6</sup>	0.33 <sup>1,2,4,5,6</sup>	0.41 <sup>1,2,5,6</sup>	0.54 <sup>2,6</sup>	5.44
6-objectives						
DTLZ1	0.12 <sup>2,3,4,5,6</sup>	0.30 <sup>3,4,5,6</sup>	546.48	2.69 <sup>3,6</sup>	2.04 <sup>3,6</sup>	226.61 <sup>3</sup>
DTLZ2	0.35 <sup>2,3,5</sup>	0.38 <sup>3,5</sup>	2.12	0.31 <sup>1,2,3,5</sup>	0.46 <sup>3</sup>	0.28 <sup>1,2,3,4,5</sup>
DTLZ3	0.43 <sup>2,3,4,5,6</sup>	0.71 <sup>3,4,5,6</sup>	1214.80	9.04 <sup>3,6</sup>	10.13 <sup>3,6</sup>	746.23 <sup>3</sup>
DTLZ4	0.45 <sup>2,3,5</sup>	0.49 <sup>3,5</sup>	3.28	0.30 <sup>1,2,3,5,6</sup>	0.54 <sup>3</sup>	0.38 <sup>1,2,3,5</sup>
DTLZ5	0.05 <sup>2,3</sup>	0.48 <sup>3</sup>	0.93	0.06 <sup>2,3</sup>	0.02 <sup>1,2,3,4,6</sup>	0.02 <sup>1,2,3,4</sup>
DTLZ6	0.08 <sup>2,3,4,6</sup>	0.68 <sup>3,4,6</sup>	21.57	0.85 <sup>3,6</sup>	0.01 <sup>1,2,3,4,6</sup>	19.05 <sup>3</sup>
DTLZ7	0.59 <sup>2,5,6</sup>	0.89 <sup>6</sup>	0.55 <sup>1,2,5,6</sup>	0.53 <sup>1,2,3,5,6</sup>	0.73 <sup>2,6</sup>	6.41
7-objectives						
DTLZ1	0.17 <sup>2,3,4,5,6</sup>	0.38 <sup>3,4,5,6</sup>	599.41	2.08 <sup>3,5,6</sup>	13.02 <sup>3,6</sup>	204.23 <sup>3</sup>
DTLZ2	0.45 <sup>2,3,5</sup>	0.66 <sup>3</sup>	3.19	0.38 <sup>1,2,3,5</sup>	0.53 <sup>2,3</sup>	0.34 <sup>1,2,3,4,5</sup>
DTLZ3	0.59 <sup>2,3,4,5,6</sup>	0.60 <sup>3,4,5,6</sup>	1480.19	8.14 <sup>3,6</sup>	9.09 <sup>3,6</sup>	684.02 <sup>3</sup>
DTLZ4	0.53 <sup>2,3,5</sup>	0.66 <sup>3</sup>	4.14	0.37 <sup>1,2,3,5,6</sup>	0.59 <sup>2,3</sup>	0.43 <sup>1,2,3,5</sup>
DTLZ5	0.07 <sup>2,3</sup>	0.51 <sup>3</sup>	1.53	0.07 <sup>2,3</sup>	0.01 <sup>1,2,3,4,6</sup>	0.03 <sup>1,2,3,4</sup>
DTLZ6	0.08 <sup>2,3,4,6</sup>	0.73 <sup>3,4,6</sup>	21.56	1.06 <sup>3,6</sup>	0.01 <sup>1,2,3,4,6</sup>	17.84 <sup>3</sup>
DTLZ7	0.94 <sup>2,6</sup>	1.21 <sup>6</sup>	0.79 <sup>1,2,6</sup>	0.66 <sup>1,2,3,5,6</sup>	0.82 <sup>1,2,6</sup>	6.48
8-objectives						
DTLZ1	0.20 <sup>2,3,4,5,6</sup>	2.29 <sup>3,5,6</sup>	624.34	1.60 <sup>3,5,6</sup>	11.41 <sup>3,6</sup>	183.06 <sup>3</sup>
DTLZ2	0.52 <sup>2,3,5</sup>	1.16 <sup>3</sup>	3.64	0.44 <sup>1,2,3,5</sup>	0.54 <sup>2,3</sup>	0.39 <sup>1,2,3,4,5</sup>
DTLZ3	0.62 <sup>2,3,4,5,6</sup>	1.14 <sup>3,6</sup>	1526.52	7.80 <sup>3,6</sup>	17.38 <sup>3,6</sup>	615.36 <sup>3</sup>
DTLZ4	0.58 <sup>2,3,5</sup>	0.80 <sup>3</sup>	4.35	0.43 <sup>1,2,3,5,6</sup>	0.65 <sup>2,3</sup>	0.49 <sup>1,2,3,5</sup>
DTLZ5	0.17 <sup>2,3</sup>	0.65 <sup>3</sup>	1.62	0.07 <sup>1,2,3</sup>	0.01 <sup>1,2,3,4,6</sup>	0.02 <sup>1,2,3,4</sup>
DTLZ6	0.18 <sup>2,3,4,6</sup>	0.75 <sup>3,4,6</sup>	21.28	1.22 <sup>3,6</sup>	0.01 <sup>1,2,3,4,6</sup>	16.88 <sup>3</sup>
DTLZ7	1.17 <sup>2,6</sup>	1.46 <sup>6</sup>	1.10 <sup>2,6</sup>	0.76 <sup>1,2,3,5,6</sup>	0.90 <sup>1,2,3,6</sup>	7.95

The superscripts 1, 2, 3, 4, 5, and 6 indicate whether the respective algorithm performs significantly better than DDMOA2, DDMOA, NSGA-II, IBEA, MOEA/D, and MSOPS2 respectively

**Table 4** Mean ranks achieved by the algorithms

Indicator	DDMOA2	DDMOA	NSGA-II	IBEA	MOEA/D	MSOPS2
Epsilon	2.5918	3.2041	4.8367	2.6327	3.1224	4.6122
Hypervolume	2.4082	3.2143	4.7755	2.9082	3.7143	3.9796
IGD	2.4898	2.9388	4.7143	3.1429	3.3878	4.3265

since these problems do not present much difficulties in terms of the convergence and the selection procedure in IBEA relies on the concept of  $\epsilon$ -dominance. Although MSOPS2 produces a relatively poor performance in the performed experiments, it behaves the best with respect to the IGD indicator on DTLZ2 with more than two objectives and provides highly competitive performance on DTLZ5 with more than three objectives concerning all the quality indicators. Although NSGA-II outperforms the other algorithms on four and five-objective DTLZ7 in terms of IGD, its performance becomes increasingly poor when the number of objectives is increased. Though an issue of the NSGA-II scalability was discussed in several previous studies [16, 21], it is an established state-of-the-art EMO algorithm and serves as an important reference.

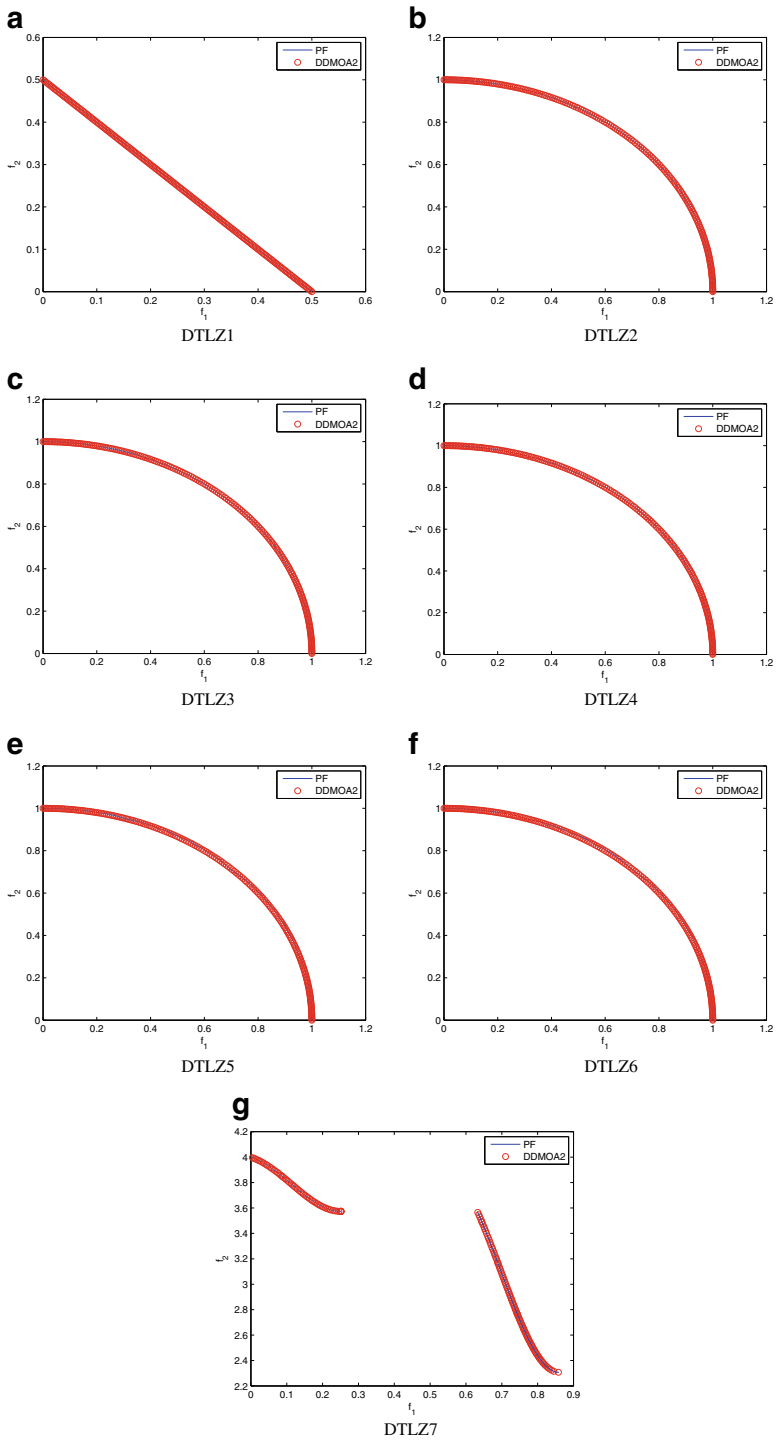
To compare the overall performance of the algorithms on all the considered problems, we rank the algorithms with respect to the quality indicators on each problem. It should be noted that testing DTLZ problems in 7 different dimensions we have a total of 49 distinct problems. Table 4 presents the mean rank calculated with respect to each quality indicator for each algorithm. From this table, we can see that DDMOA2 has the best mean ranks regarding all three indicators. Therefore, we can conclude that DDMOA2 produces a highly competitive overall performance, despite it does not outperform the other algorithms on the majority of problems.

Additionally, the approximation sets with the best values of IGD obtained by DDMOA2 on the two and three-objective problems are shown in Figs. 2 and 3. From the presented plots we can see that DDMOA2 is capable to converge and provide an adequate distribution of solutions along the Pareto front for all the two and three-objective DTLZ test problems.

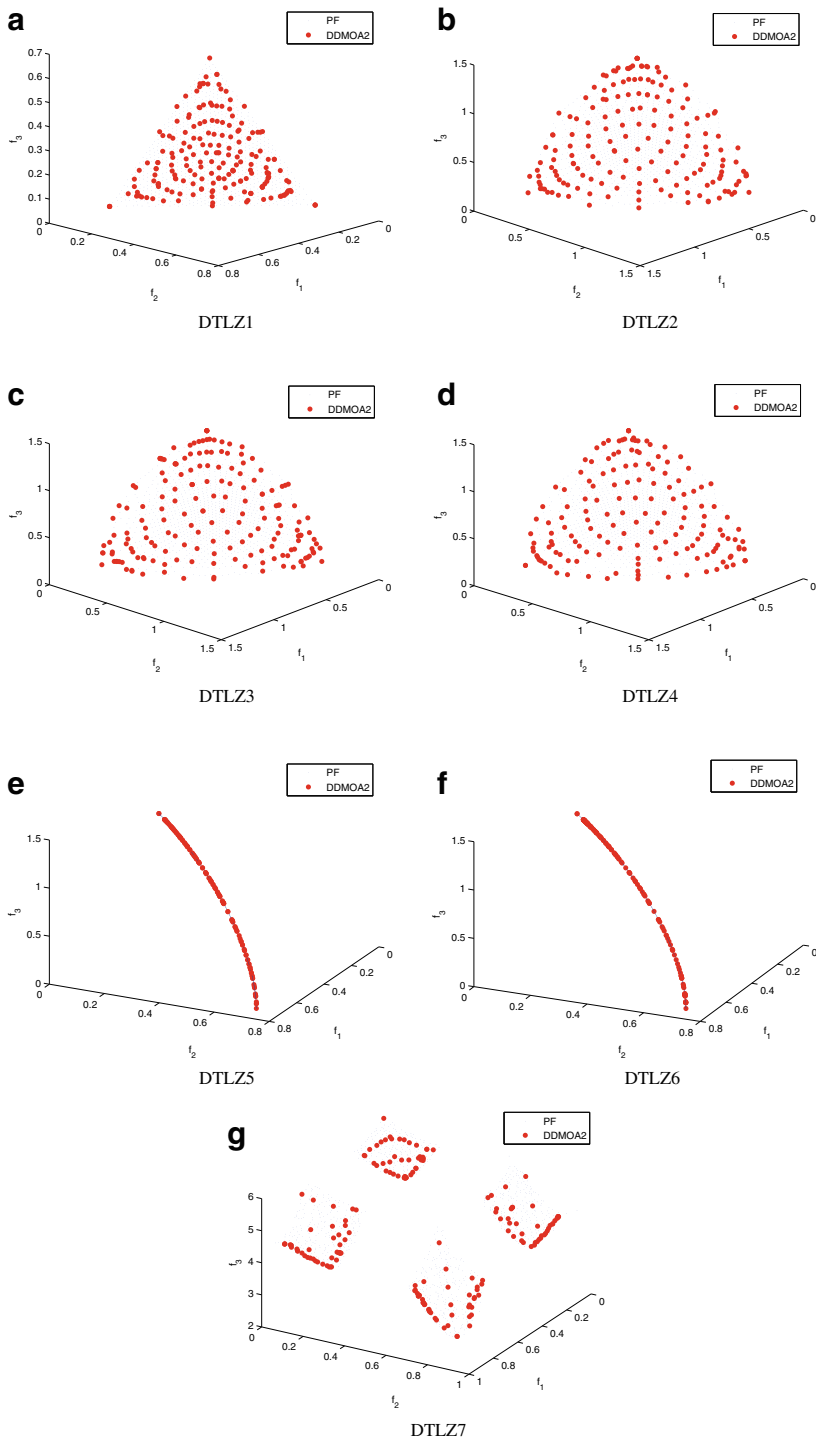
## 4 Conclusions

In this paper, we presented a generalized descent directions-guided multiobjective algorithm. We extended the previous version of algorithm in order to deal with many-objective problems. For this purpose, the scalarizing fitness assignment is incorporated into the parent and environmental selection procedures. In order to select promising individuals in the population for applying local search, we introduce the concept of population leaders. On the other hand, in order to improve the efficiency of the algorithm, especially on problems with a large number of objectives, local search procedure is used to find descent directions only for two randomly chosen objectives. Furthermore, the condition to accept a descent direction for a leader during the local search is defined to increase the search pressure in a high-dimensional objective space.

The empirical evaluation of the proposed approach is carried out on problems with up to 8 dimensions. The obtained results showed that DDMOA2 produces a highly competitive performance when compared with state-of-the-art multiobjective evolutionary algorithms.



**Fig. 2** Performance of DDMOA2 on the two-objective DTLZ test suite



**Fig. 3** Performance of DDMOA2 on the three-objective DTLZ test suite

In our experiments, DDMOA2 outperformed the other algorithms with respect to three quality indicators on multimodal problems in all dimensions. Thus, the proposed strategy to perform the search in the decision space combined with a proper selection scheme appears to be effective at solving many-objective optimization problems.

As future work, we intend to further study the adaptation of algorithm's parameters. In order to alleviate the computational burden endured computing descent directions using local search, the information about last successful mutations can be used to reduce the number of times local search is invoked. Furthermore, the recombination of individuals during the evolution can be introduced to improve global search ability of the algorithm.

**Acknowledgments** This work has been supported by FCT Fundação para a Ciência e Tecnologia in the scope of the project: PEst-OE/EEI/UI0319/2014.

## References

1. Beyer, H.-G., Schwefel, H.-P.: Evolution strategies: a comprehensive introduction. *Nat. Comput.* **1**(1), 3–52 (2002)
2. Bleuler, S., Laumanns, M., Thiele, L., Zitzler, E.: PISA: a platform and programming language independent interface for search algorithms. In: *Proceedings of the Conference on Evolutionary Multi-Criterion Optimization*, pp. 494–508. EMO'03 (2003)
3. Bosman, P.A.N., Thierens, D.: The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Trans. Evol. Comput.* **7**(2), 174–188 (2003)
4. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and Evolutionary Computation, 2nd edn. Springer (2007)
5. Costa, L., Espírito Santo, I., Denysiuk, R., Fernandes, E.M.G.P.: Hybridization of a genetic algorithm with a pattern search augmented Lagrangian method. In: *Proceedings of the Conference on Engineering Optimization*, p. 1195. EngOpt'10 (2010)
6. Deb, K.: *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley-Interscience Series in Systems and Optimization. Wiley (2001)
7. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective 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 multi-objective optimization. Technical Report 112, Swiss Federal Institute of Technology, Zurich, Switzerland (2001)
9. Denysiuk, R., Costa, L., Espírito Santo, I.: DDMOA: Descent directions based multiobjective algorithm. In: *Proceedings of the Conference on Computational and Mathematical Methods in Science and Engineering*, pp. 460–471. CMMSE'12 (2012)
10. Denysiuk, R., Costa, L., Espírito Santo, I.: DDMOA2: Improved descent directions-based multiobjective algorithm. In: *Proceedings of the Conference on Computational and Mathematical Methods in Science and Engineering*, pp. 513–524. CMMSE'13 (2013)
11. Denysiuk, R., Costa, L., Espírito Santo, I.: A new hybrid evolutionary multiobjective algorithm guided by descent directions. *J. Math. Model. Algoritm. Oper. Res.* **12**(3), 233–251 (2013)
12. Durillo, J.J., Nebro, A.J.: Metal: a Java framework for multi-objective optimization. *Adv. Eng. Softw.* **42**(10), 760–771 (2011)
13. García, S., Molina, D., Lozano, M., Herrera, F.: A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization. *J. Heuristics* **15**(6), 617–644 (2009)
14. Hughes, E.J.: MSOPS-II: A general-purpose many-objective optimiser. In: *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 3944–3951. CEC'07 (2007)
15. Ishibuchi, H., Doi, T., Nojima, Y.: Incorporation of scalarizing fitness functions into evolutionary multi-objective optimization algorithms. In: *Proceedings of the Conference on Parallel Problem Solving from Nature*, pp. 493–502. PPSN'06 (2006)
16. Khare, V.R., Yao, X., Deb, K.: Performance scaling of multi-objective evolutionary algorithms. In: *Proceedings of the Conference on Evolutionary Multi-Criterion Optimization*, pp. 376–390. EMO'03 (2003)



17. Knowles, J., Corne, D.: Memetic algorithms for multiobjective optimization: issues, methods and prospects. *Recent Adv. Memet. Algorith. Stud. Fuzziness Soft Comput.* **166**, 313–352 (2005)
18. 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)
19. Loh, W.L.: On Latin hypercube sampling. *Ann. Stat.* **33**(6), 2058–2080 (1996)
20. Miettinen, K.: *Nonlinear multiobjective optimization*. International Series in Operations Research and Management Science, vol. 12. Kluwer Academic Publishers (1999)
21. Purshouse, R.C., Fleming, P.J.: Evolutionary many-objective optimisation: an exploratory analysis. In: *Proceedings of the IEEE Congress on Evolutionary Computation, CEC'03*, pp. 2066–2073 (2003)
22. Shukla, P.K., Deb, K.: On finding multiple Pareto-optimal solutions using classical and evolutionary generating methods. *Eur. J. Oper. Res.* **181**(3), 1630–1652 (2007)
23. Torczon, V.: On the convergence of pattern search algorithms. *SIAM J. Optim.* **7**, 1–25 (1997)
24. Zitzler, E., Künzli, S.: Indicator-based selection in multiobjective search. In: *Proceedings of the Conference on Parallel Problem Solving from Nature, PPSN'04*, pp. 832–842 (2004)
25. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms - A case comparative case study. In: *Proceedings of the Conference on Parallel Problem Solving from Nature, PPSN'98*, pp. 292–304 (1998)
26. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Grunert da Fonseca, V.: Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Trans. Evol. Comput.* **7**(2), 117–132 (2003)