

Using an outward selective pressure for improving the search quality of the MOEA/D algorithm

Krzysztof Michalak

Received: 12 January 2014 / Published online: 13 February 2015

© The Author(s) 2015. This article is published with open access at Springerlink.com

Abstract In optimization problems it is often necessary to perform an optimization based on more than one objective. The goal of the multiobjective optimization is usually to find an approximation of the Pareto front which contains solutions that represent the best possible trade-offs between the objectives. In a multiobjective scenario it is important to both improve the solutions in terms of the objectives and to find a wide variety of available options. Evolutionary algorithms are often used for multiobjective optimization because they maintain a population of individuals which directly represent a set of solutions of the optimization problem. multiobjective evolutionary algorithm based on decomposition (MOEA/D) is one of the most effective multiobjective algorithms currently used in the literature. This paper investigates several methods which increase the selective pressure to the outside of the Pareto front in the case of the MOEA/D algorithm. Experiments show that by applying greater selective pressure in the outwards direction the quality of results obtained by the algorithm can be increased. The proposed methods were tested on nine test instances with complicated Pareto sets. In the tests the new methods outperformed the original MOEA/D algorithm.

Keywords Multiobjective optimization · Evolutionary algorithms · MOEA/D algorithm · Selective pressure

K. Michalak (✉)

Department of Information Technologies, Institute of Business Informatics,
Wrocław University of Economics, Wrocław, Poland
e-mail: krzysztof.michalak@ue.wroc.pl

1 Introduction

One of the steps of a decision making process is finding optimal solutions of various optimization problems. In cases when more than one objective is concerned it is often not possible to find a single optimal solution. Instead, the decision maker is presented with an entire set of possible solutions that represent various trade-offs between the objectives. A multiobjective optimization problem (MOP) can be formalized as follows:

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

where, Ω —the decision space, m —the number of objectives.

Because solutions are evaluated using multiple criteria it is usually not possible to order them linearly from the best one to the worst. Instead, solutions are compared using the Pareto domination relation defined as follows [4, 19]. Given two points x_1 and x_2 in the decision space Ω for which:

$$\begin{aligned} F(x_1) &= (f_1(x_1), \dots, f_m(x_1)) \\ F(x_2) &= (f_1(x_2), \dots, f_m(x_2)) \end{aligned} \quad (2)$$

we say that x_1 *dominates* x_2 ($x_1 \succ x_2$) iff:

$$\begin{aligned} \forall i \in \{1, \dots, m\} : f_i(x_1) &\leq f_i(x_2) \\ \exists i \in \{1, \dots, m\} : f_i(x_1) &< f_i(x_2) \end{aligned} \quad (3)$$

A solution x is said to be *nondominated* (*Pareto optimal*) iff:

$$\neg \exists x' \in \Omega : x' \succ x. \quad (4)$$

In the area of multiobjective problems solving evolutionary algorithms are often used [2, 30]. The advantage of evolutionary and other population-based approaches is that they maintain an entire population of candidate solutions which represent an approximation of the true Pareto front. Among multiobjective evolutionary algorithms the two well-known groups of approaches are methods based on Pareto domination relation and decomposition-based methods.

Algorithms based on Pareto domination relation use it for numerical evaluation of specimens or for selection. Pareto domination relation is used for evaluation of specimens in algorithms such as SPEA [32] and SPEA-2 [33]. The approach in which Pareto domination is used for selection is used for example in NSGA [23] and NSGA-II [7].

The multiobjective evolutionary algorithm based on decomposition (MOEA/D) was proposed by Zhang and Li [27]. Contrary to the algorithms that base their selection process on the Pareto domination relation the MOEA/D algorithm is a decomposition-based algorithm. In this algorithm the multiobjective optimization problem is decom-

posed to a number of single-objective problems. To date the MOEA/D algorithm has been applied numerous times and was found to outperform other algorithms not only in the case of benchmark problems [17,27], but also in the case of combinatorial problems [21] and various applications to real-life problems. The applications in which the MOEA/D algorithm was used include multi-band antenna design [9], deployment in wireless sensor networks [15], air traffic optimization [3], and route planning [24].

Since the MOEA/D algorithm was introduced many modified versions were proposed. Some approaches aim at achieving better performance by exploiting the advantages of various scalarization functions used in the MOEA/D framework. Ishibuchi et al. [12] proposed adaptive selection of either the weighted sum scalarization or Tchebycheff scalarization depending of the region of the Pareto front. An algorithm in which both scalarizations are used simultaneously was also proposed [13]. Other authors attempt using autoadaptive mechanisms in the context of the MOEA/D framework. For example, in the paper [28] a MOEA/D-DRA variant is proposed in which a utility value is calculated for each subproblem and computational resources are allocated to subproblems according to their utility values.

There are also hybrid algorithms in which the MOEA/D is combined with other optimization algorithms. Martinez and Coello Coello proposed hybridization with the nonlinear simplex search algorithm [26]. Li and Landa-Silva combined the MOEA/D algorithm with simulated annealing [16] and applied their approach to combinatorial problems. Another approach based on PSO and decomposition was proposed in [1]. In the case of combinatorial optimization a hybridization with the ant colony optimization (ACO) was attempted [14] in order to implement the “learning while optimizing” principle.

The main motivation for this paper is to research the possibility of increasing the diversity of the search, namely extending the Pareto front, by directing the selective pressure towards outer regions of the Pareto front. While it is a different approach than those discussed above it can be easily combined with many of the methods by other authors. The presented algorithm can be hybridized with other optimization algorithms and local search procedures in the same way as the original MOEA/D algorithm. For example, the approach proposed in [26] employs a simplex algorithm as a local search procedure to improve the solutions found by the MOEA/D algorithm. It works at a different level than the approach proposed in this paper which modifies the working of the MOEA/D algorithm itself. Therefore, hybrid approaches, such as described in papers [1,14,16,26] could use the weight generation scheme presented in this paper instead of the original one. Adaptive allocation of the resources discussed in [28] could also be attempted with the algorithm presented in this paper.

The MOEA/D algorithm employs the idea of decomposition of the original multi-objective problem to several single-objective problems. To obtain a single, scalar objective from several ones that are present in the original problem various aggregation approaches are used. Many authors use the weighted sum approach [19] and the Tchebycheff approach [27]. In both approaches a weight vector $\lambda = (\lambda_1, \dots, \lambda_m)$, $\lambda_i \geq 0$ for all $i \in \{1, \dots, m\}$, $\sum_{i=1}^m \lambda_i = 1$ is assigned to each specimen in the population. The formulation of a scalar problem in the Tchebycheff aggregation approach is as follows:

$$\begin{aligned} &\text{minimize } g^{te}(x|\lambda, z^*) = \max_{1 \leq i \leq m} \{ \lambda_i |f_i(x) - z_i^*| \} \\ &\text{subject to } x \in \Omega, \end{aligned} \tag{5}$$

where, λ —the weight vector assigned to the specimen, z^* —the reference point.

The reference point $z^* = (z_1^*, \dots, z_m^*)$ used in the Tchebycheff aggregation approach is composed of the best attainable values along all the objectives:

$$z^* = \min\{f_i(x) : x \in \Omega\}. \tag{6}$$

If the best possible values for all the objectives are not known in advance the reference point z^* can be calculated based on the values of objectives that were attained so far by the population.

The authors of the MOEA/D algorithm proposed the following way of generating weight vectors [27]. First, a value of a parameter H is chosen which determines the size of a step used for generating weight vectors. A set of weight vectors is generated by selecting all the possible m -element combinations of numbers from the set:

$$\left\{ \frac{0}{H}, \frac{1}{H}, \dots, \frac{H}{H} \right\} \tag{7}$$

such that:

$$\sum_{j=1}^m \lambda_j = 1. \tag{8}$$

This weight vector generation approach produces:

$$N = C_{H+m-1}^{m-1} \tag{9}$$

weight vectors. Because there is exactly one weight vector assigned to each specimen in the population the size of the population equals N . For a bi-objective optimization problem ($m = 2$) this weight vector generation procedure builds a set of vectors $\{\lambda^{(i)}\}$, $i \in \{0, \dots, H\}$, where:

$$\lambda^{(i)} = \left[\frac{i}{H}, \frac{H-i}{H} \right]. \tag{10}$$

The MOEA/D algorithm uses a concept of subproblem neighborhood for transferring information between specimens that represent solutions of scalar subproblems. The neighborhood size is controlled by the algorithm parameter T . For each weight vector $\lambda^{(i)}$ its neighborhood $B(i)$ is defined as a set of T weight vectors that are closest to $\lambda^{(i)}$ with respect to the Euclidean distance calculated in the m -dimensional space of vector weights. In the bi-objective case for $T = 2k + 1$, $k \in \mathbb{N}$ the neighborhood can be located symmetrically around $\lambda^{(i)}$. For $T = 2k$, $k \in \mathbb{N}$ it is not possible and one “excess” weight vector must remain on one side of $\lambda^{(i)}$ (for a study of the effects that this asymmetry has on the working of the MOEA/D algorithm please refer to [18]). In further considerations we assume, that this excess vector is included on the side with

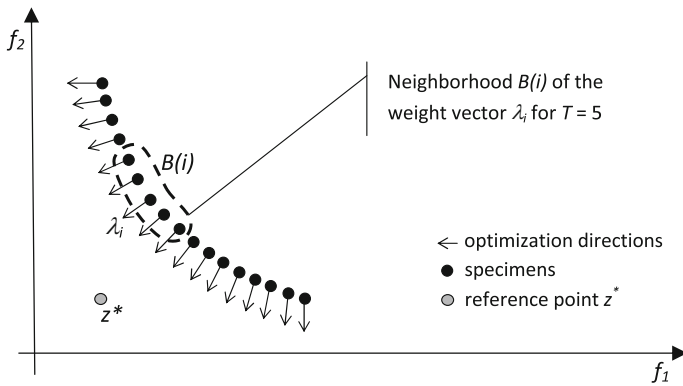


Fig. 1 Overview of concepts involved in the working of the MOEA/D algorithm in the case of minimization in a bi-objective problem

indices larger than i . Therefore, for weight vectors located near edges of the Pareto front the neighborhood is shifted and is equal to:

$$B(i) = \begin{cases} \{0, \dots, T - 1\} & \text{for } i < \lfloor (T - 1)/2 \rfloor \\ \{N - T, \dots, N - 1\} & \text{for } i \geq N - \lceil (T - 1)/2 \rceil. \end{cases} \tag{11}$$

For $\lambda^{(i)}$ located farther from the edges the neighborhood is:

$$B(i) = \begin{cases} \{i - k, \dots, i, \dots, i + k\} & \text{for } T = 2k + 1, k \in \mathbb{N} \\ \{i - k + 1, \dots, i, \dots, i + k\} & \text{for } T = 2k, k \in \mathbb{N} . \end{cases} \tag{12}$$

When a new specimen for the i th subproblem is to be generated, parent selection is performed among the specimens from the neighborhood $B(i)$. The newly generated offspring may then replace not only the current specimen for the i th subproblem, but also specimens assigned to the subproblems from the neighborhood $B(i)$. By definition, each vector $\lambda^{(i)}$ belongs to its own neighborhood $B(i)$. Concepts involved in the working of the MOEA/D algorithm are presented in Fig. 1 which shows the population (black dots) and optimization directions determined by the weight vectors (arrows) for a bi-objective case.

2 Increasing the outward selective pressure

The MOEA/D algorithm uses weight vectors to aggregate objectives of a multiobjective optimization problem into a single, scalar objective. Clearly, the coordinates of a weight vector $\lambda^{(i)}$ determine the importance of each objective $f_j, j \in \{1, \dots, m\}$ in i -th scalar subproblem.

Solutions in the neighborhood $B(i)$ participate in information exchange with the solution of a subproblem parameterized by weight vector $\lambda^{(i)}$. Thus, one can expect that the distribution of weight vectors in the neighborhood $B(i)$ may influence the

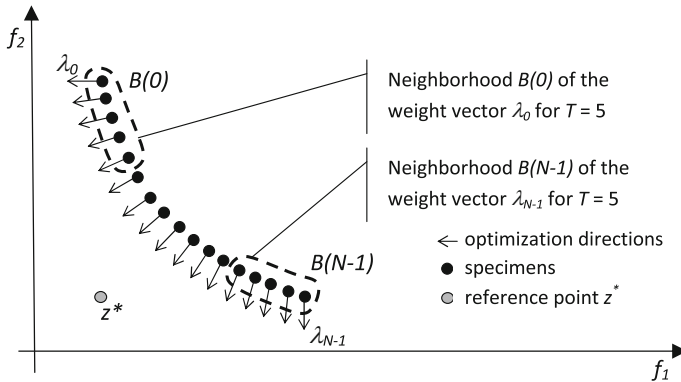


Fig. 2 The neighborhoods $B(0)$ and $B(N - 1)$ in a bi-objective minimization problem

selective pressure exerted on specimens that represent solutions of the i -th subproblem.

Weight vectors $\lambda^{(i)}$ located near the center of the weight vector set have all the coordinates close to $1/m$. In such a case the neighborhood $B(i)$ is located more or less symmetrically around the weight vector $\lambda^{(i)}$. For weight vectors located near or on the edge of the weight vector set the neighborhood $B(i)$ is placed asymmetrically. This effect is easy to illustrate in a bi-objective case. If $\lambda^{(i)} = [1/2, 1/2]$ the neighborhood $B(i)$ is symmetric as shown in Fig. 1. For weight vectors $\lambda^{(0)}$ and $\lambda^{(N-1)}$ the neighborhoods $B(0)$ and $B(N - 1)$ are placed entirely on one side of the weight vector as shown in Fig. 2.

Such placement of weight vectors in the neighborhood of the $\lambda^{(0)}$ vector causes a situation in which specimens assigned to the 0th subproblem can only exchange information with specimens that are evaluated using weight vectors pointing slightly downwards. Correspondingly, specimens that solve the subproblem $N - 1$ can only exchange information with specimens that are evaluated using weight vectors pointing slightly to the left. In both situations it can be expected that an inward selective pressure will be exerted on specimens on the edges of the Pareto front preventing them from extending towards higher values of objectives f_1 and f_2 . Other specimens located at positions closer than $T/2$ from the edge can also be influenced by such pressure, which can be expected to be smaller if a specimen is located farther from the edge.

Obviously, the selective pressure influences the behavior of the population and therefore it can be expected to influence also the shape of the Pareto front attained by the algorithm. The methods presented in this paper were developed based on an assumption that directing the selective pressure outwards will make the Pareto front spread wider. This assumption was positively verified by a preliminary test performed on a simple benchmark problem. The construction of this benchmark function and the results of the preliminary test are presented in Sect. 3.

In this paper three methods of directing the selective pressure outwards are proposed: “fold”, “reduce” and “diverge”. The first two of these methods differ from the standard MOEA/D algorithm in the way in which the neighborhoods are defined.

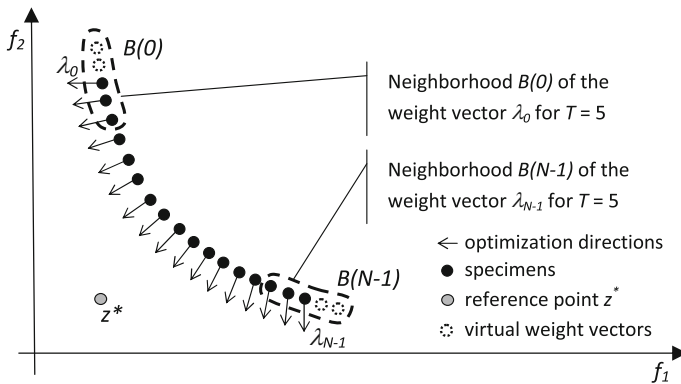


Fig. 3 Symmetric neighborhoods $B(0)$ and $B(N-1)$ including “virtual” weight vectors used in the “fold” method

In the “diverge” method the neighborhoods are defined in the same way as in the standard MOEA/D algorithm but weight vectors are calculated in a different way.

2.1 The “fold” method

In the “fold” method weight vectors are generated using the same procedure as proposed by the authors of the MOEA/D algorithm in [27]. This weight vector generation procedure is described in Sect. 1. It generates all the possible m -element combinations of numbers from the set (7) that sum to 1 as described by Eq. (8).

In the “fold” method the neighborhoods of weight vectors are defined differently than in the standard MOEA/D implementation. Each neighborhood $B(i)$ is placed symmetrically around the weight vector $\lambda^{(i)}$. In the case of weight vectors placed near the edges of the Pareto front the unavailable weight vectors are represented by special values -1 and N which correspond to “virtual” weight vectors as shown in Fig. 3.

In the MOEA/D algorithm the neighborhood $B(i)$ is used for selecting parents for genetic operations (crossover and mutation). Parent selection is done by randomly selecting a weight vector from the neighborhood $B(i)$. The specimen that corresponds to the selected weight vector is used in genetic operations. This selection process is slightly modified in the “fold” method in the case of neighborhoods which include “virtual” weight vectors. The probability of selecting each of the weight vectors (a “real” or a “virtual” one) is the same and equals $\frac{1}{|B(i)|}$. Therefore, with some probability, the selection procedure can choose one of the “virtual” weight vectors represented by a special value -1 or N . “Virtual” weight vectors do not have specimens assigned to them, so if a “virtual” weight vector is selected a “real” weight vector has to be selected instead. In a biobjective case the specimen corresponding to the outermost weight vector $\lambda^{(0)}$ or $\lambda^{(N-1)}$ is returned instead of any “virtual” weight vector. The probability $P_O(i)$ that the specimen corresponding to the outermost vector will be used is in such case equal to:

$$P_O(i) = \begin{cases} \frac{\lfloor \frac{T-1}{2} \rfloor - i + 1}{T} & \text{for } i = 0, \dots, \lfloor \frac{T-1}{2} \rfloor \\ \frac{\lceil \frac{T-1}{2} \rceil - N + i + 2}{T} & \text{for } i = N - \lceil \frac{T-1}{2} \rceil - 1, \dots, N - 1 \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

This probability satisfies inequalities:

$$\begin{aligned} P_O(i) &> \frac{1}{T} \text{ for } i < \lfloor \frac{T-1}{2} \rfloor, \\ P_O(i) &> \frac{1}{T} \text{ for } i > N - \lceil \frac{T-1}{2} \rceil - 1. \end{aligned} \tag{14}$$

Clearly, when the “fold” method is used for a biobjective problem the probability of selecting the outermost weight vector $\lambda^{(0)}$ or $\lambda^{(N-1)}$ increases in the neighborhoods surrounding weight vectors placed near the edge of the Pareto front.

The approach described above can easily be generalized to problems with more than two objectives. First, the “virtual” vectors are generated. These are vectors that satisfy the condition (8), but contain elements that are negative (e.g. $-1/H, -2/H$) or larger than 1 (e.g. $(H + 1)/H, (H + 2)/H$). For each “real” weight vector λ_i a preliminary neighborhood $B'(i)$ is formed by selecting T vectors closest to λ_i from both the “real” and “virtual” weight vectors. If a “virtual” weight vector λ_v is selected for the neighborhood $B'(i)$ it is replaced by a “real” weight vector λ_r closest to λ_v (cf. Figs. 4 and 5). If there are several “real” weight vectors at the same distance from λ_v

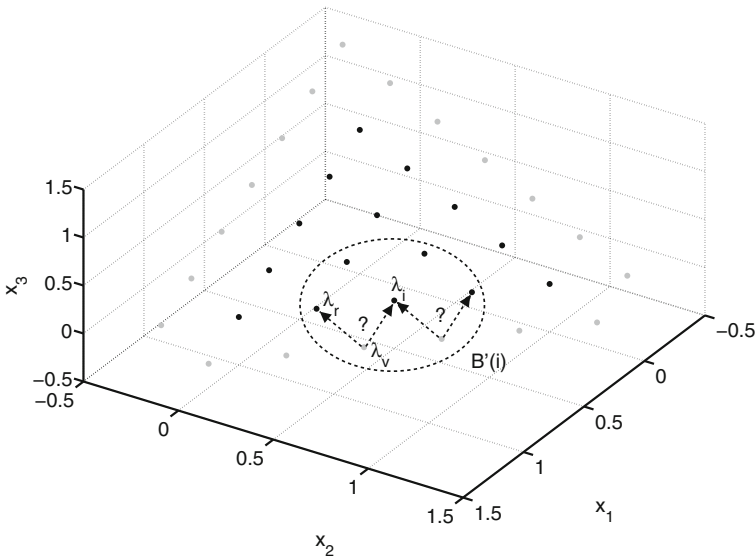


Fig. 4 Selection of “real” weight vectors for the neighborhood $B(i)$ of a weight vector λ_i used in the “fold” method

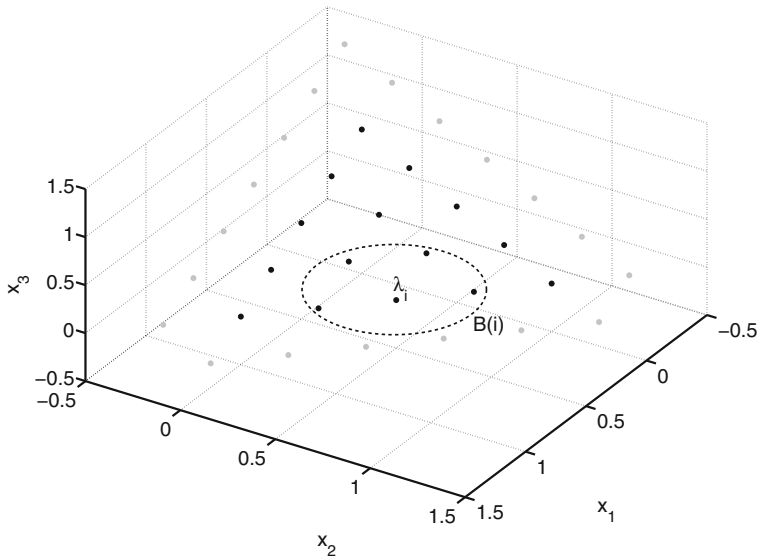


Fig. 5 Neighborhood $B(i)$ of a weight vector λ_i used in the “fold” method

one of them is selected at random. Obviously, such weight vector λ_r is placed on the edge of the weight vector set.

After the neighborhood construction procedure described above, the neighborhood $B(i)$ contains only the “real” weight vectors and the vectors placed on the edge of the set of “real” weight vectors have multiple copies in $B(i)$. Parent selection is performed by selecting elements from $B(i)$ with uniform probability. In neighborhoods $B(i)$ corresponding to weight vectors placed on the edge of the weight vector set some “virtual” weight vectors are replaced by “real” weight vectors. The selective pressure towards the center of the Pareto front is thus reduced.

2.2 The “reduce” method

In the “reduce” method the weight vectors are generated using a similar procedure as in the “fold” method. From the set of “real” and “virtual” weight vectors a preliminary neighborhood $B'(i)$ of a weight vector λ_i is generated by considering T closest neighbors of λ_i (both “real” and “virtual” weight vectors). From these T vectors only the “real vectors” are retained in $B(i)$. Contrary to the “fold” method no multiple copies of the weight vectors are placed in $B(i)$. The neighborhood $B(i)$ of a vector λ_i placed on the edge of the set of the “real” weight vectors looks as shown in Fig. 5. Therefore, the size of the neighborhood is decreased for such λ_i compared to the original MOEA/D algorithm.

In a biobjective case the size of the neighborhood $B(i)$ equals:

$$|B(i)| = \begin{cases} \lfloor \frac{T}{2} \rfloor + i + 1 < T & \text{for } i = 0, \dots, \lfloor \frac{T-1}{2} \rfloor - 1 \\ \lceil \frac{T}{2} \rceil + N - i - 1 < T & \text{for } i = N - \lceil \frac{T-1}{2} \rceil, \dots, N - 1 \\ T & \text{otherwise.} \end{cases} \quad (15)$$

When parents are selected, each neighborhood member can be selected with the probability $\frac{1}{|B(i)|}$, so there is no preference for the outermost weight vectors in a given neighborhood $B(i)$. However, the overall probability of selecting the outermost weight vector is increased compared to the standard MOEA/D algorithm because the neighborhoods near the edge of the Pareto front are smaller.

2.3 The “diverge” method

In the “diverge” method the weight vector generation procedure is modified. Instead of choosing values from the set (7) in this method the coordinates are taken from the set:

$$\left\{ \frac{0}{H} - \alpha, \dots, \frac{H}{H} + \alpha \right\}, \quad (16)$$

where:

α - a parameter that controls the divergence of weight vectors.

The condition (8) must be satisfied by all vectors $\lambda^{(0)}, \dots, \lambda^{(N-1)}$, i.e. all coordinates in each weight vector must sum to 1. Therefore, in a bi-objective case all the weight vectors have the form:

$$\lambda^{(i)} = \left[\frac{i}{H} + \frac{(2i - H)\alpha}{H}, \frac{H - i}{H} - \frac{(2i - H)\alpha}{H} \right], \quad (17)$$

where: $i = 0, \dots, H$.

Obviously,

$$\lambda^{(0)} = [-\alpha, 1 + \alpha], \quad \lambda^{(H)} = [1 + \alpha, -\alpha]. \quad (18)$$

The layout of the optimization directions determined by the weight vectors in the “diverge” method in a bi-objective minimization problem is presented in Fig. 6.

Note, that in this method some of the weight vectors have negative coordinates. The aggregation function used for decomposing the multiobjective problem into scalar ones must be defined in such a way that for the negative weights the definition remains correct. For example, negative weights can easily be used with the weighted sum decomposition. Other approaches, such as the Tchebycheff decomposition which minimizes $\max\{\lambda_i | f_i(x) - z_i^* | \}$ (cf. Eq. (5)) have to be modified in order to accommodate negative weights. Obviously, in the Tchebycheff decomposition only the objectives $f_i(x)$ corresponding to weight vectors coordinates $\lambda_i > 0$ have any influence on the value of the \max function (terms corresponding to negative weight vector coordinates are always not larger than 0). Thus, this particular decomposition method has to be modified in order to accommodate negative weight vector coordinates. A solution employed

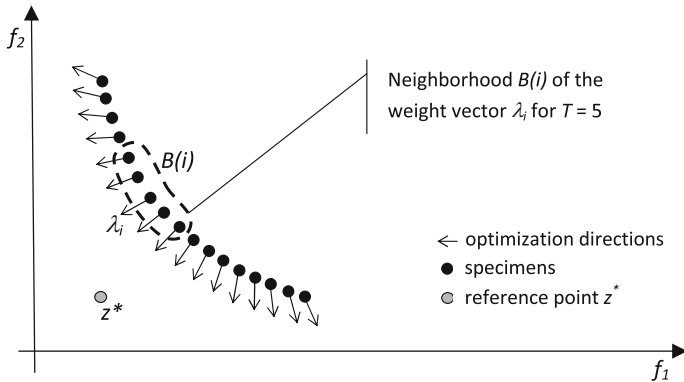


Fig. 6 The layout of the optimization directions determined by the weight vectors in the “diverge” method in a bi-objective minimization problem

in this paper is to calculate the distance from the nadir point $z^\#$ (a point with all the worst, instead of the best, coordinates) for these objectives that correspond to negative coordinates λ_i in the weight vector. Therefore in the “diverge” method the Tchebycheff decomposition is performed using Eq. (19).

$$\begin{aligned} \text{minimize } g^{te}(x|\lambda, z^*) &= \max_{1 \leq i \leq m} \{\phi(\lambda_i, f_i(x))\} \\ \text{subject to } x &\in \Omega, \end{aligned} \tag{19}$$

where: λ —the weight vector assigned to the specimen, ϕ —a function calculated as follows:

$$\phi(\lambda, y) = \begin{cases} \lambda|y - z_i^*| & \text{for } \lambda \geq 0 \\ \lambda|y - z_i^\#| & \text{for } \lambda < 0 \end{cases} \tag{20}$$

where: z^* —the reference point, $z^\#$ —the nadir point.

Similarly as the reference point z^* the nadir point $z^\#$ is updated during the runtime of the algorithm. It contains the worst objective values found so far by the algorithm. These are usually the objective values attained during the first generations of the evolution.

In the “diverge” method neighborhoods of weight vectors are defined in the same way as in the standard MOEA/D algorithm. The parent selection procedure is also the same as in the standard MOEA/D algorithm.

3 Preliminary tests

In the previous section three methods of directing the selective pressure outwards are proposed: “fold”, “reduce” and “diverge”. The design of these methods was based on a hypothesis that directing the selective pressure outwards will make the Pareto front spread wider. To verify this assumption a preliminary test was performed on a simple benchmark problem with a symmetric Pareto front designed in such a way that the

difference in the Pareto front extent should easily be visible. Also, the results produced by the MOEA/D algorithm for this benchmark problem are rather poor considering the simplicity of the problem. The solutions found by the MOEA/D algorithm tend to concentrate in the middle of the actual Pareto front. The solutions at the edges of the actual Pareto front are not found at all by the unmodified MOEA/D algorithm.

The geometric representation of this problem is as follows. First, a parameter d is set which determines the dimensionality of the search space which is a d -dimensional cube $\Omega = [0, 1]^d$. The two conflicting objectives are to approach two different points in this search space as close as possible. One of these points denoted A has all coordinates equal to $\frac{1}{4}$ and the other denoted B all coordinates equal to $\frac{3}{4}$:

$$A = \left[\frac{1}{4}, \dots, \frac{1}{4} \right] \in \mathbb{R}^d \tag{21}$$

$$B = \left[\frac{3}{4}, \dots, \frac{3}{4} \right] \in \mathbb{R}^d \tag{22}$$

The values of the two objective functions $f_1, f_2 : [0, 1]^d \rightarrow R$ are defined as distances to points A and B respectively. Formally, the optimization problem can be defined as:

$$\begin{aligned} &\text{minimize } F(x) = (\|x - A\|, \|x - B\|) \\ &\text{subject to } x \in [0, 1]^d, \end{aligned} \tag{23}$$

which is equal to:

$$\begin{aligned} &\text{minimize } F(x) = (f_1(x), f_2(x)) \\ &\text{subject to } x \in [0, 1]^d, \end{aligned} \tag{24}$$

where:

$$f_1(x) = \sqrt{\sum_{i=1}^d (x_i - A_i)^2} \tag{25}$$

$$f_2(x) = \sqrt{\sum_{i=1}^d (x_i - B_i)^2} \tag{26}$$

Clearly, the objectives are conflicting, and the Pareto set of this optimization problem is the line segment connecting the points A and B in the decision space $[0, 1]^d$. When the solution is located at one of the ends of this line segment one of the objectives is 0 while the other has the value of $\frac{\sqrt{d}}{2}$ which is the length of the line segment AB . The Pareto front is therefore a line segment in the objective space \mathbb{R}^2 with the equation:

$$y = -x + \frac{\sqrt{d}}{2}, \quad x \in \left[0, \frac{\sqrt{d}}{2}\right]. \quad (27)$$

Positioning the solution anywhere on the AB line segment is allowed, so all the points on this Pareto front can be attained.

Sets of all nondominated solutions obtained in 30 runs of the preliminary tests are presented in Fig. 7. Table 1 presents the minimum and the maximum values attained along each of the objectives by each method. Obviously, if the minimum values are lower and the maximum values are higher, the Pareto front found by the algorithm spreads wider which indicates a greater diversity of the search.

The Pareto fronts presented in Fig. 7 and the values presented in Table 1 clearly show that the unmodified MOEA/D algorithm performed worst in the preliminary test. The “fold” method seems to be only slightly better. On the other hand the “reduce” method improved the results significantly and the “diverge” method produced even more widely spread Pareto fronts. Thus, the results of the preliminary tests support the hypothesis that the diversity of the search can be improved by directing the selective pressure outwards to the edges of the Pareto front.

4 Experimental study

The experiments were performed in order to verify if increasing the outward selective pressure improves the results obtained by the algorithm. The experiments were performed on test problems with complicated Pareto sets F1–F9 described in [17]. Some of these test problems were also used during the CEC 2009 MOEA competition [29], namely F2 (as UF1), F5 (as UF2), F6 (as UF8), and F8 (as UF3).

The performance of the standard MOEA/D algorithm was compared to the performance of the three methods of directing the selective pressure outwards: “fold”, “reduce” and “diverge” described in Sect. 2. The “diverge” method requires setting the value of the parameter α which determines by how much the weight vectors diverge to the outside. To test the influence of this parameter on the algorithm performance the value of this parameter was set to $\alpha = 0.1, 0.2, 0.3, 0.4$ and 0.5 .

For each data set and each method of increasing the outward selective pressure 30 iterations of the test were performed. For comparison, tests using the standard version of the MOEA/D algorithm were also performed.

4.1 Performance assessment

In the case of multiobjective optimization problems evaluating the results obtained by an algorithm is more complicated than in the case of single-objective problems. Solutions produced by each run of an algorithm are characterized by two or more conflicting objectives and thus they may not be directly comparable. A common practice is to evaluate the entire set of solutions using a certain indicator which represents the quality of the whole solution set.

In this paper two indicators were used: the hypervolume (HV) [34] also known as the size of the objective space covered [31] and the Inverted Generational Distance

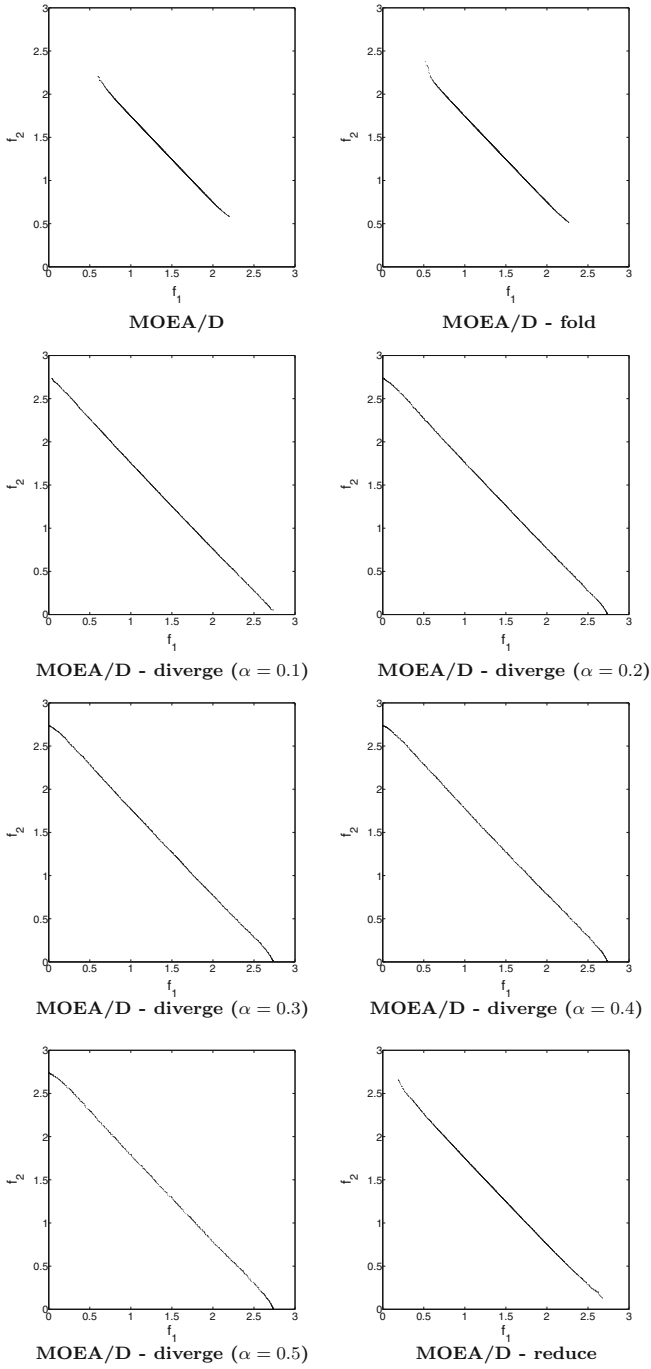


Fig. 7 Sets of all nondominated solutions obtained in 30 runs of the preliminary tests

Table 1 The minimum and the maximum values attained along each of the objectives by each method in preliminary tests on a problem with a symmetric Pareto front

Algorithm	$\min f_1$	$\max f_1$	$\min f_2$	$\max f_2$
MOEA/D	0.6015	2.2048	0.5840	2.2063
MOEA/D—fold	0.5188	2.2659	0.5131	2.3757
MOEA/D—diverge ($\alpha = 0.1$)	0.0332	2.7339	0.0519	2.7330
MOEA/D—diverge ($\alpha = 0.2$)	0.0108	2.7394	0.0109	2.7375
MOEA/D—diverge ($\alpha = 0.3$)	0.0061	2.7394	0.0065	2.7370
MOEA/D—diverge ($\alpha = 0.4$)	0.0045	2.7377	0.0046	2.7390
MOEA/D—diverge ($\alpha = 0.5$)	0.0023	2.7395	0.0040	2.7394
MOEA/D—reduce	0.1893	2.6765	0.1286	2.6611

(IGD) [17] also known as the distance from the representatives (or D -metric) [27]. For a given set of solutions P the hypervolume is the Lebesgue measure of the portion of objective space that is dominated by solutions in P collectively:

$$HV(P) = L\left(\bigcup_{x \in P} [f_1(x), r_1] \times \cdots \times [f_m(x), r_m]\right), \quad (28)$$

where: m —the dimensionality of the objective space, $f_i(\cdot)$, $i = 1, \dots, m$ —the objective functions, $R = (r_1, \dots, r_m)$ —a reference point, $L(\cdot)$ —the Lebesgue measure on \mathbb{R}^m .

In two and three dimensions the hypervolume corresponds to the area and volume respectively. Better Pareto fronts are those that have higher values of the HV indicator. The hypervolume indicator is commonly used in the literature to evaluate Pareto fronts and it has good theoretical properties. It has been proven [10] that maximizing the hypervolume is equivalent to achieving Pareto optimality. To the best of the knowledge of the author, this is the only currently known measure with this property.

The second indicator, the Inverted Generational Distance (IGD) measures how close the solutions in an approximation of a Pareto front P approach points from a predefined set P^* . The P^* set contains points that are uniformly distributed in the objective space along the real Pareto front of a given multiobjective optimization problem. Formally, for a given reference set P^* and an approximation of the Pareto front P the IGD indicator is calculated as:

$$IGD(P, P^*) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|}, \quad (29)$$

where: $d(v, P)$ —a minimum Euclidean distance between v and the points in P .

Better Pareto fronts are those that have lower values of the IGD indicator. The IGD indicator measures both the diversity and the convergence of P . Calculating the IGD indicator requires generating a set of points that are uniformly distributed

Table 2 Algorithm parameters (note, that population size increases with the increasing number of objectives)

Parameter name	Value
Number of generations (N_{gen})	500
Population size (N)	
(2 objectives)	300
(3 objectives)	595
Weight vector step size (H)	
(2 objectives)	299
(3 objectives)	33
Neighborhood size (T)	20
The probability that parent solutions are selected from the neighborhood (δ)	0.9
The maximum number of solutions that can be replaced by a child solution (η_T)	2
Crossover probability for the DE operator (CR)	1.0
Differential weight for the DE operator (F)	0.5
Mutation probability (P_{mut})	$1/n^{(*)}$
Mutation distribution index (η_{mut})	20
Decomposition method	Tchebycheff

(*) n —the number of decision variables

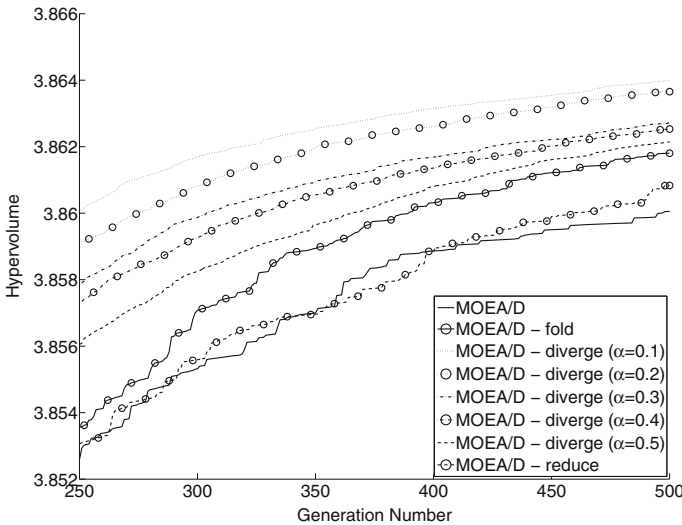


Fig. 8 Hypervolume values obtained for the F1 test problem. Median values from 30 runs

along the true Pareto front of a given multiobjective optimization problem. This can be a significant obstacle in the case of real-life optimization problems for which the true Pareto front is not known. For benchmark problems the true Pareto front is usually known so the points in the reference set P^* can be set exactly on this front.

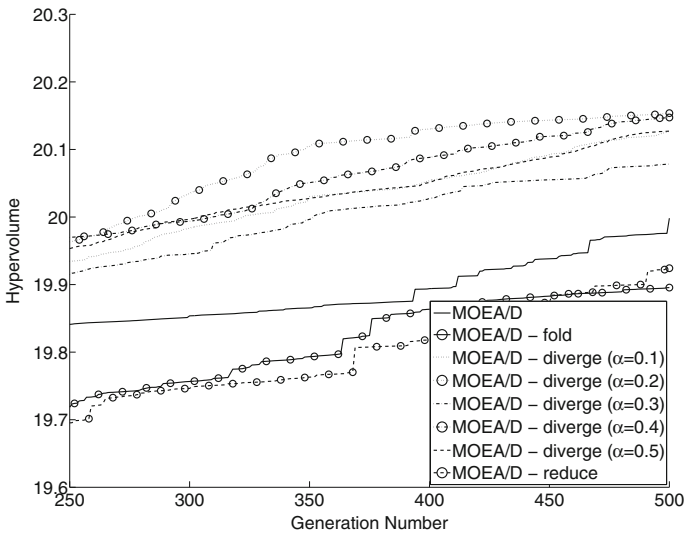


Fig. 9 Hypervolume values obtained for the F2 test problem. Median values from 30 runs

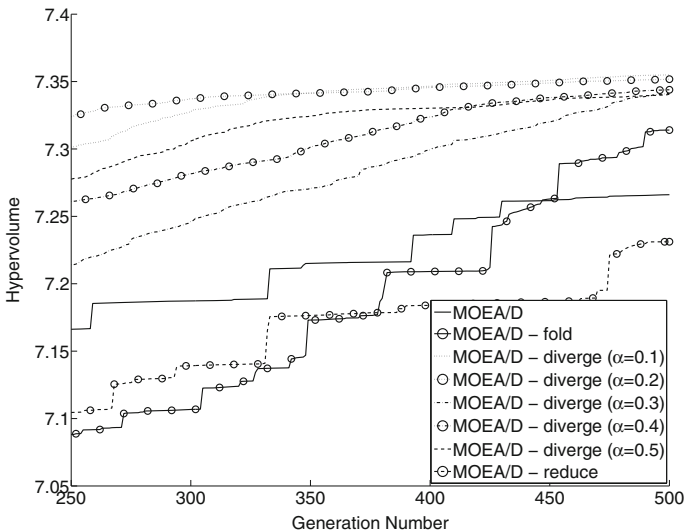


Fig. 10 Hypervolume values obtained for the F3 test problem. Median values from 30 runs

4.2 Parameter settings

The algorithm used in the experiments is based on the version of the MOEA/D algorithm described in [17] which is the same paper in which the test problems were proposed. For the “fold”, “reduce” and “diverge” methods neighborhood construction and specimen selection procedures were changed as described in Sect. 2. Other aspects

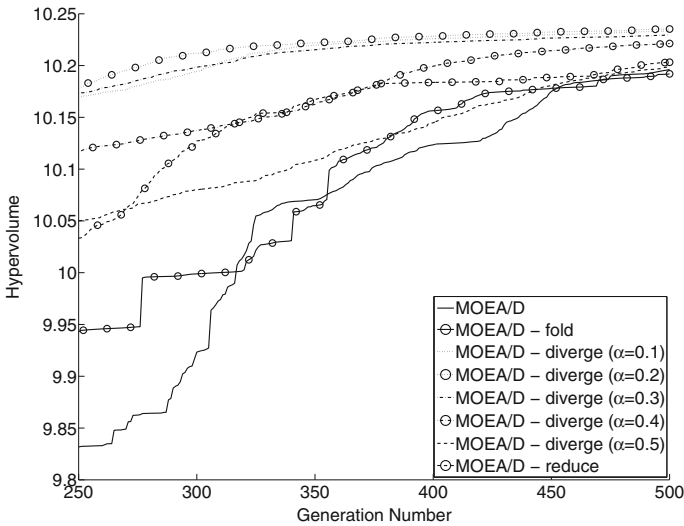


Fig. 11 Hypervolume values obtained for the F4 test problem. Median values from 30 runs

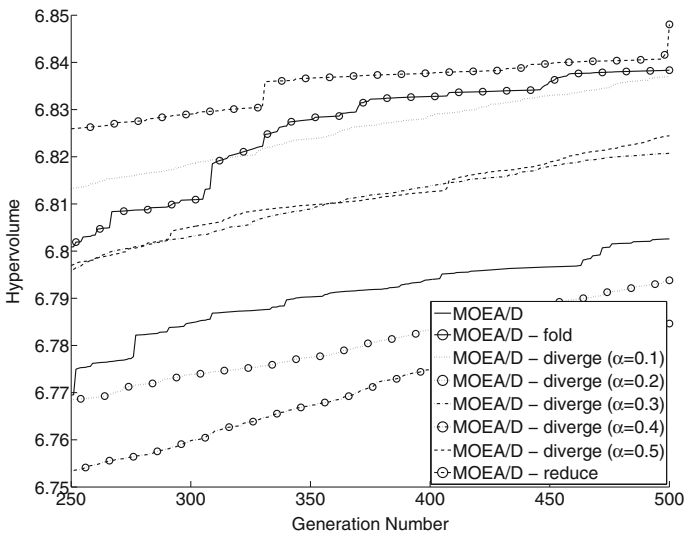


Fig. 12 Hypervolume values obtained for the F5 test problem. Median values from 30 runs

of how the algorithm worked were not changed with respect to the standard MOEA/D algorithm.

For each test problem and for each tested method proposed in this paper the algorithm was run for $N_{gen} = 500$ generations. Since each of the algorithm versions performs the same number of objective function evaluations the total number of objective function evaluations was the same in each test. In the case of problems F1–F9 the size of the population was set to $N = 300$ specimens for 2-objectives and to

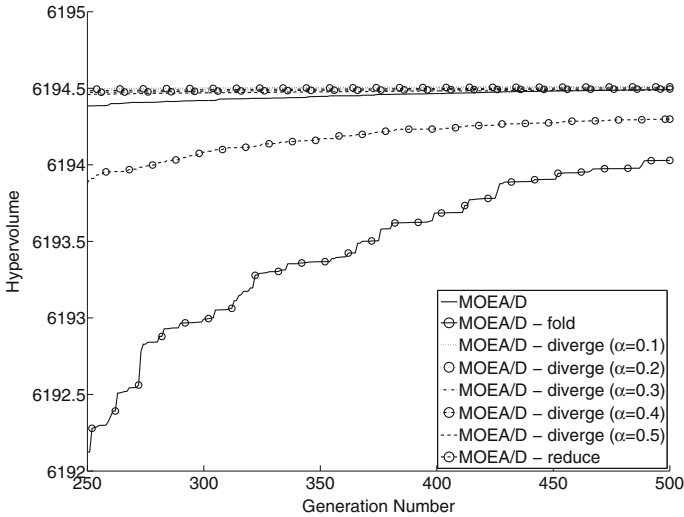


Fig. 13 Hypervolume values obtained for the F6 test problem. Median values from 30 runs

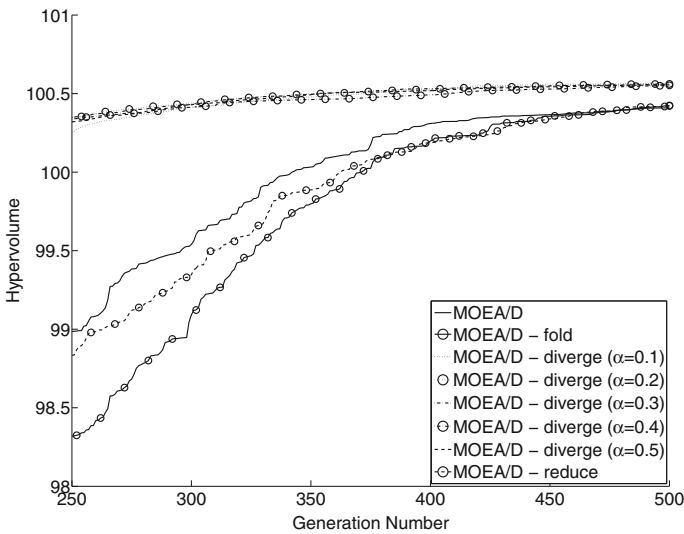


Fig. 14 Hypervolume values obtained for the F7 test problem. Median values from 30 runs

595 for 3-objectives. This corresponds to the weight vector step size $H = 299$ and $H = 33$ respectively. Both the number of generations and the population size were the same as used in the original paper on the problems with complicated Pareto sets [17].

The neighborhood size was set to $T = 20$. The version of the MOEA/D algorithm proposed in [17] uses two parameters that are intended to prevent premature convergence of the population. These parameters are δ and η_T . During parent selection, solutions are selected from the neighborhood $B(i)$ with probability δ and from

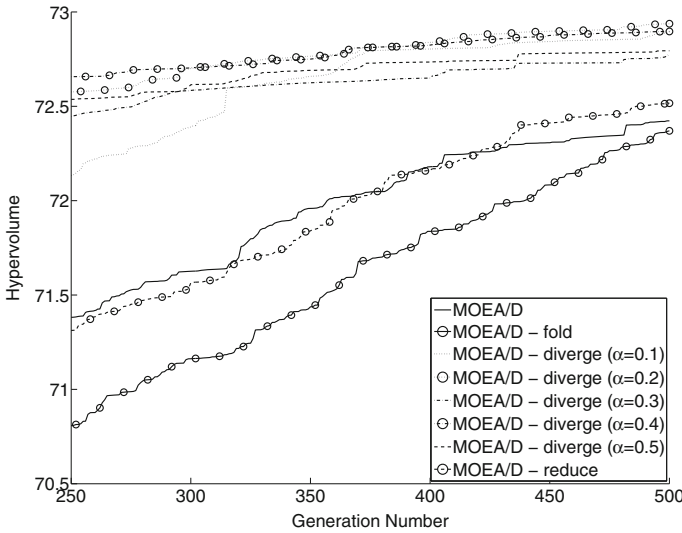


Fig. 15 Hypervolume values obtained for the F8 test problem. Median values from 30 runs

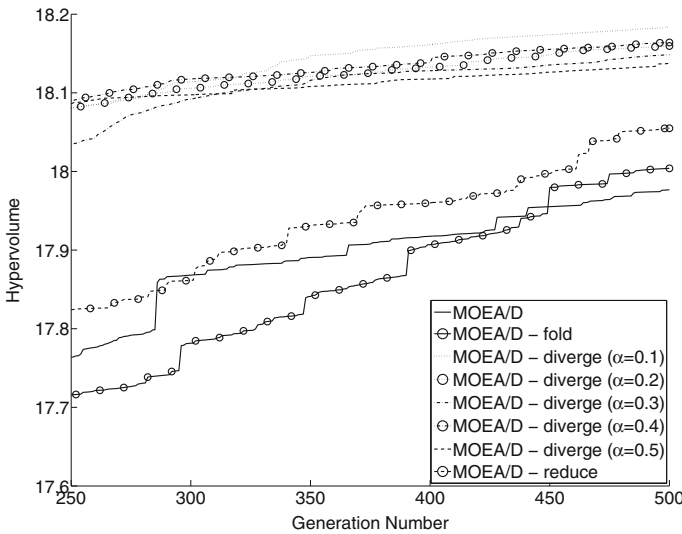


Fig. 16 Hypervolume values obtained for the F9 test problem. Median values from 30 runs

the entire population with probability $1 - \delta$. The δ parameter was set to 0.9. The η_T parameter determines the maximum number of solutions that can be replaced by a new child solution. This parameter is used to prevent a situation in which too many solutions in a certain neighborhood are replaced by a single child solution. The η_T parameter was set to 2.

The genetic operators were the same as used by the authors of the MOEA/D algorithm in [17] who suggested using the Differential Evolution (DE) operator [22] instead

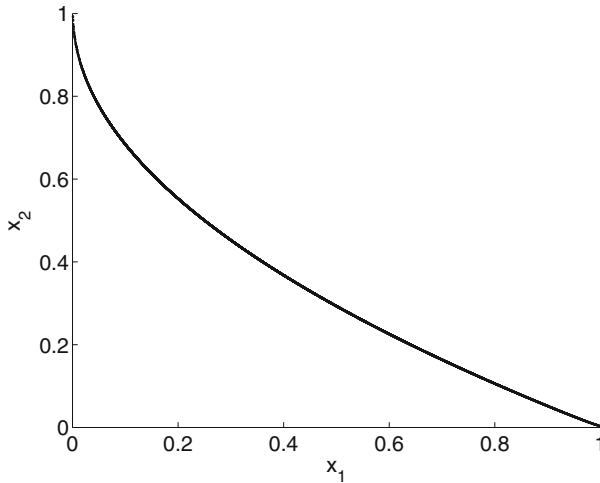


Fig. 17 Pareto front of all nondominated solutions obtained by the MOEA/D algorithm for the F1 problem in 30 runs

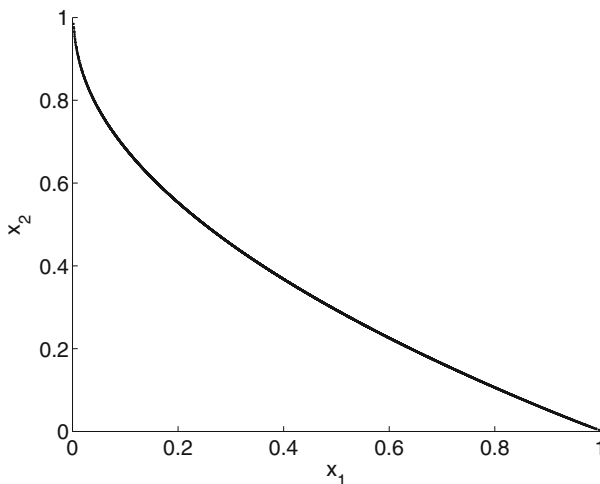


Fig. 18 Pareto front of all nondominated solutions obtained by the “diverge” method with $\alpha = 0.1$ for the F1 problem in 30 runs

of the Simulated Binary Crossover (SBX) operator [5] used in their previous paper on MOEA/D [27]. The Differential Evolution operator is parameterized by two parameters CR and F . The values of these parameters were set to $CR = 1.0$ and $F = 0.5$. Mutation was performed using the polynomial mutation operator [6, 11]. Following the parameterization proposed by the authors of the MOEA/D algorithm the distribution index for the polynomial mutation was set to 20. The probability of mutation was set to $P_{mut} = 1/n$ where n —the number of decision variables in a given problem. Tchebycheff decomposition was used for aggregating the objectives to a scalar

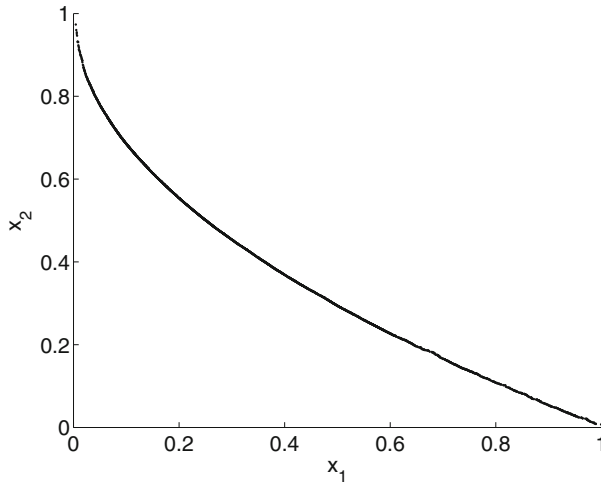


Fig. 19 Pareto front of all nondominated solutions obtained by the MOEA/D algorithm for the F2 problem in 30 runs

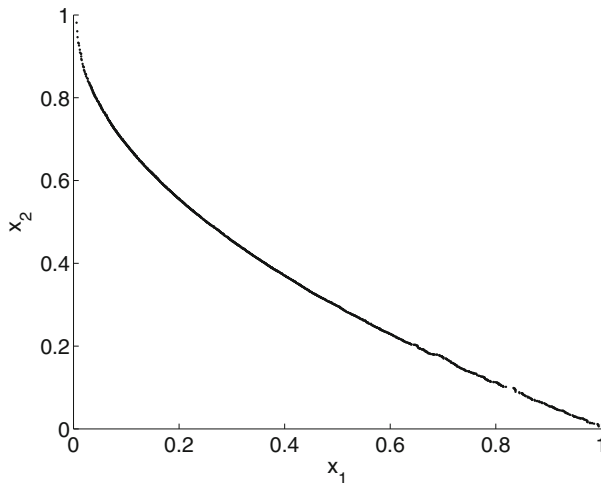


Fig. 20 Pareto front of all nondominated solutions obtained by the “diverge” method with $\alpha = 0.2$ for the F2 problem in 30 runs

objective function. All the parameters used in the experiments are summarized in Table 2.

5 Experiments

In the experiments solutions of the test problems F1–F9 were generated using the standard MOEA/D algorithm and using the three methods of directing the selective pressure outwards: “fold”, “reduce” and “diverge” described in Sect. 2. The “diverge”

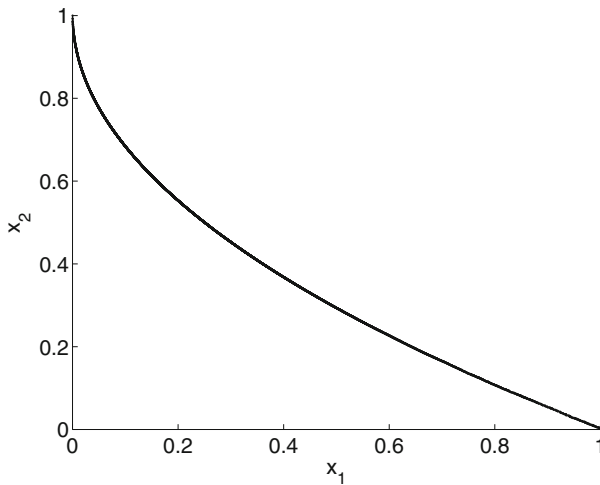


Fig. 21 Pareto front of all nondominated solutions obtained by the MOEA/D algorithm for the F3 problem in 30 runs

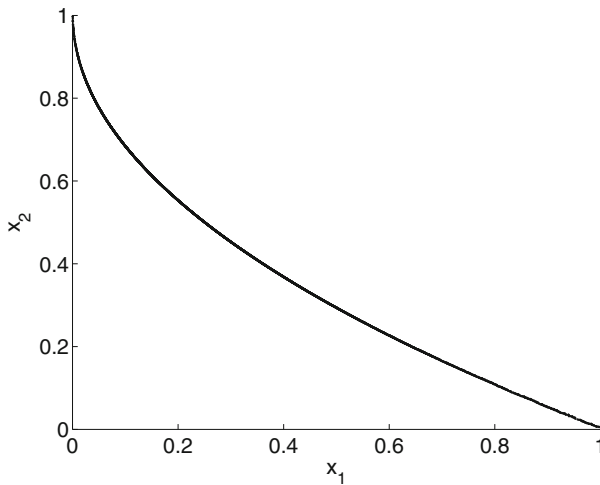


Fig. 22 Pareto front of all nondominated solutions obtained by the “diverge” method with $\alpha = 0.1$ for the F3 problem in 30 runs

method is parameterized by the parameter α which controls by how much the weight vectors diverge to the outside. In the experiments the value of this parameter was set to $\alpha = 0.1, 0.2, 0.3, 0.4$ and 0.5 . Each algorithm was run 30 times for each test problem. The number of generations in each run was $N_{gen} = 500$. Solutions generated by the algorithms were compared using the hypervolume (HV) and Inverted Generational Distance (IGD) indicators described in Sect. 4.1.

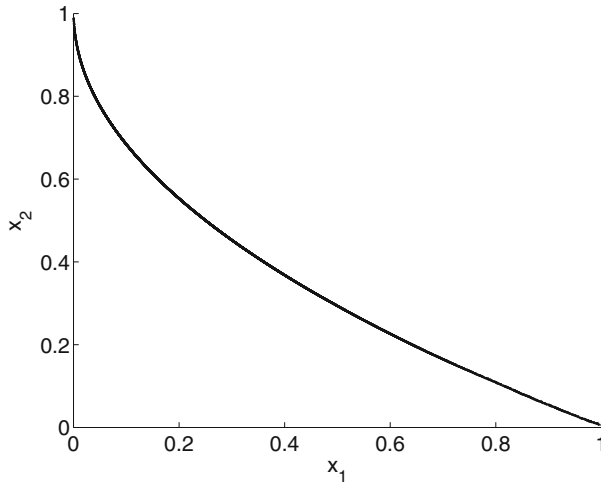


Fig. 23 Pareto front of all nondominated solutions obtained by the MOEA/D algorithm for the F4 problem in 30 runs

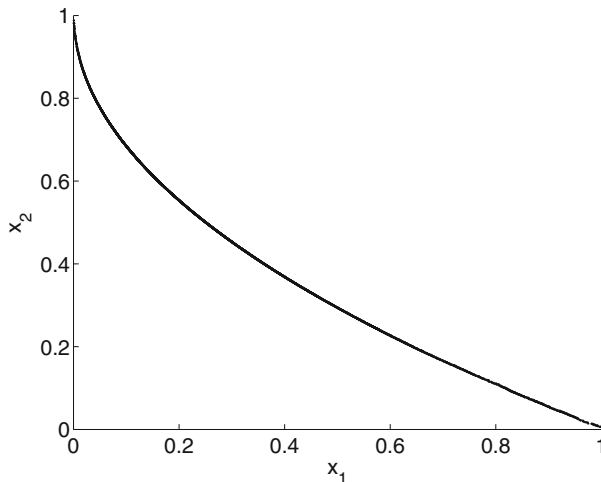


Fig. 24 Pareto front of all nondominated solutions obtained by the “diverge” method with $\alpha = 0.2$ for the F4 problem in 30 runs

Figures 8, 9, 10, 11, 12, 13, 14, 15, and 16 contain plots which present, for each of the test problems, the dynamic behavior of the hypervolume indicator. Presented values are medians from 30 runs performed in the experiments.

Figures 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, and 32 present the Pareto fronts obtained in the experiments for the MOEA/D algorithm and the best performing (in terms of the hypervolume indicator) of the modified versions of the algorithm proposed in this paper. The Pareto fronts in the figures were obtained by taking all the nondominated solutions from 30 runs of each of the algorithms.

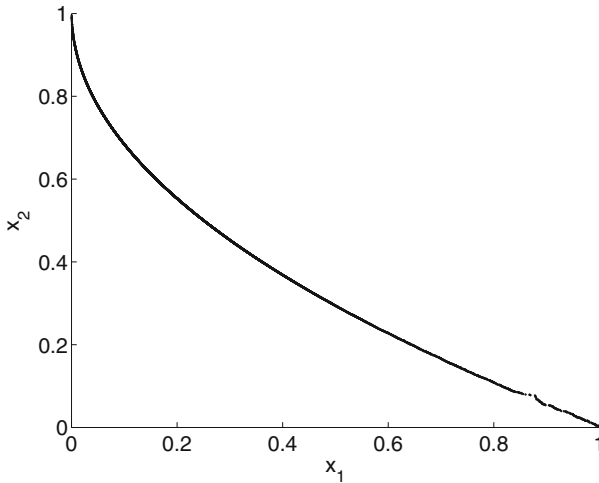


Fig. 25 Pareto front of all nondominated solutions obtained by the MOEA/D algorithm for the F5 problem in 30 runs

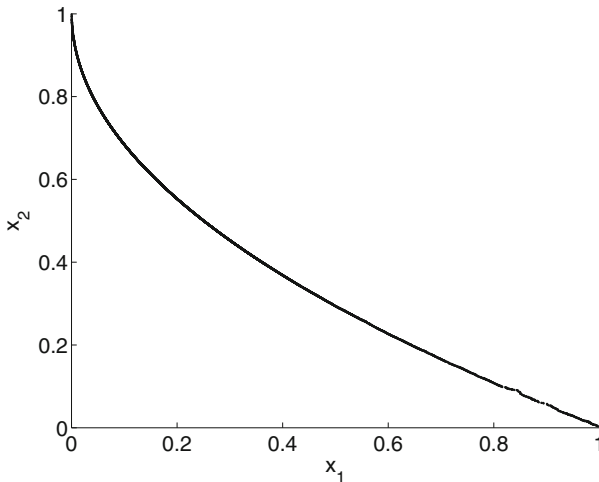


Fig. 26 Pareto front of all nondominated solutions obtained by the "reduce" method for the F5 problem in 30 runs

Figures 33, 34, and 35 contain boxplots which present, for each of the test problems, the distributions of the hypervolume indicator obtained at the last (500th) generation by each of the tested algorithms.

From the figures that present the dynamic behavior of the hypervolume indicator and from the boxplots that present the distribution of the hypervolume values it can be observed that the standard MOEA/D algorithm was outperformed by other algorithms - most often the ones using the "diverge" method. To validate this observation statistical testing was performed in which the results produced by each of the algo-

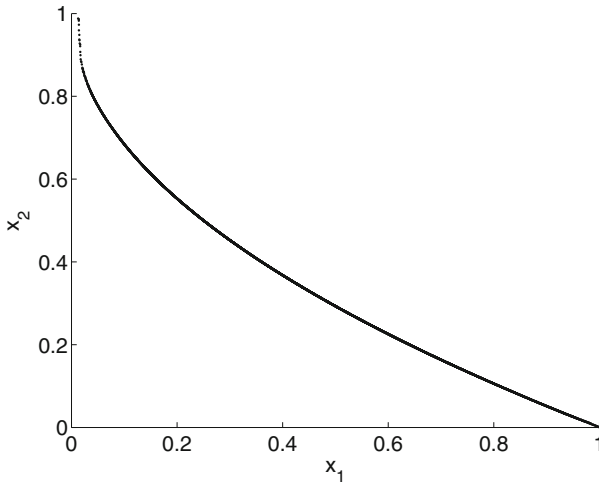


Fig. 27 Pareto front of all nondominated solutions obtained by the MOEA/D algorithm for the F7 problem in 30 runs

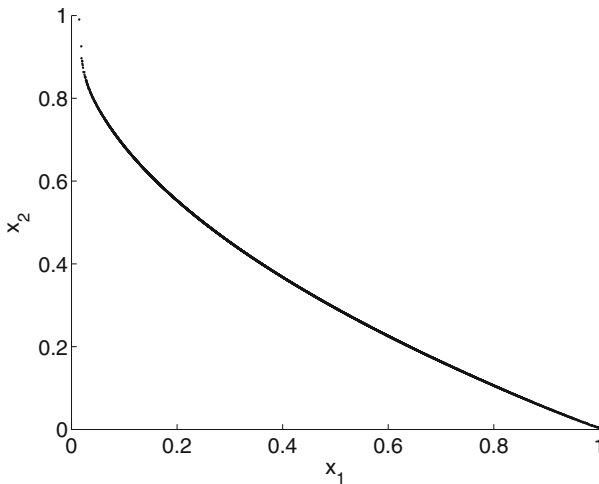


Fig. 28 Pareto front of all nondominated solutions obtained by the “diverge” method with $\alpha = 0.1$ for the F7 problem in 30 runs

rithms were compared to those produced by the standard MOEA/D algorithm. In most cases the normality of the distribution of hypervolume values cannot be guaranteed, therefore a test that does not require this assumption had to be chosen. In this paper the Wilcoxon signed rank test [20] introduced in [25] was used. This test was recommended in a recent survey [8] as one of the methods suitable for statistical comparison of results produced by metaheuristic optimization algorithms. The Wilcoxon test does not assume the normality of data distributions and has a null hypothesis which states the equality of medians.

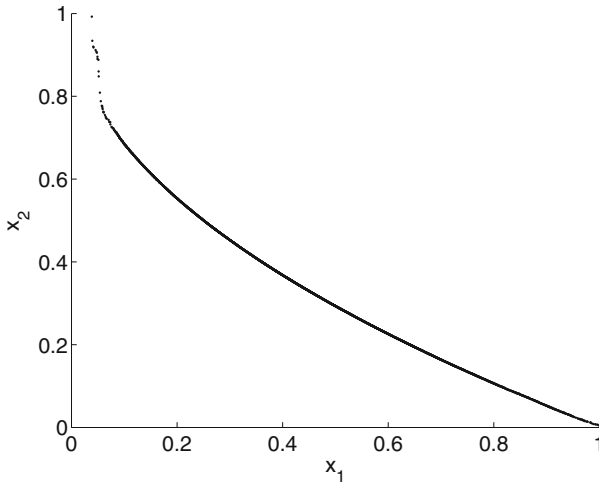


Fig. 29 Pareto front of all nondominated solutions obtained by the MOEA/D algorithm for the F8 problem in 30 runs

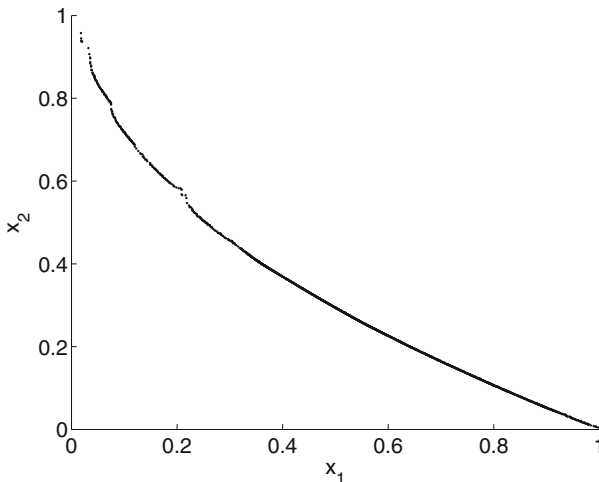


Fig. 30 Pareto front of all nondominated solutions obtained by the “diverge” method with $\alpha = 0.2$ for the F8 problem in 30 runs

Tables 3, 4, 5, 6, 7, 8, 9, 10, and 11 present results obtained by all the algorithms after 500 generations. The median hypervolume value from all the 30 runs is given and the results of the statistical test are presented. The p-value produced by the Wilcoxon test is the upper bound of the probability of obtaining the presented results if the null hypothesis (equality of medians) holds. Therefore, low p-values support a conclusion that the medians obtained for a given algorithm and for the standard MOEA/D algorithm are not equal. A value in the “interpretation” column has the following meaning. If the median for a given algorithm is lower than that for the standard MOEA/D

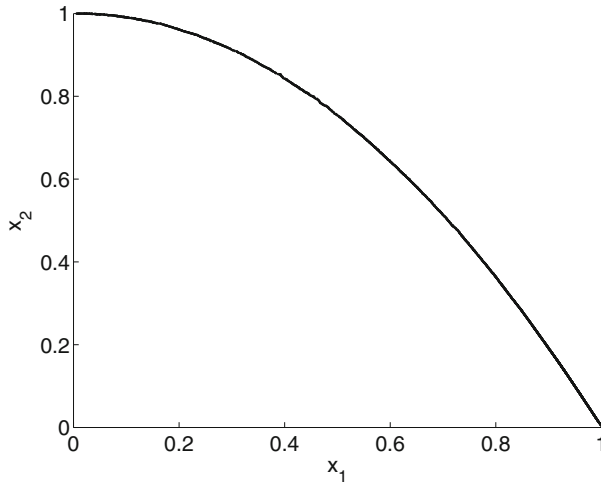


Fig. 31 Pareto front of all nondominated solutions obtained by the MOEA/D algorithm for the F9 problem in 30 runs

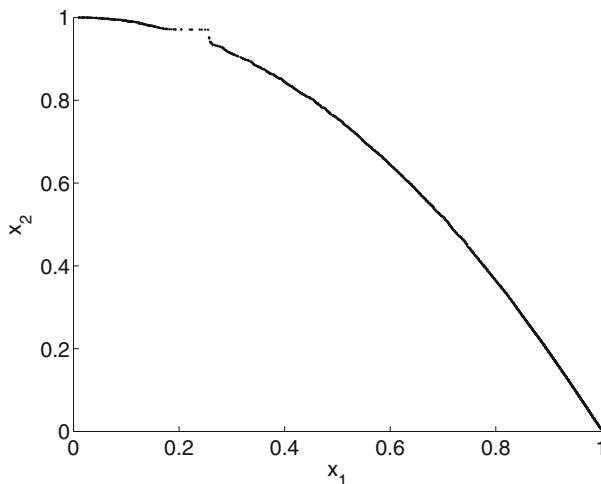


Fig. 32 Pareto front of all nondominated solutions obtained by the "diverge" method with $\alpha = 0.1$ for the F9 problem in 30 runs

algorithm the interpretation is "worse". If the median for a given algorithm is higher than that for the standard MOEA/D algorithm the interpretation is "significant" if the corresponding p-value is not larger than 0.05 and "insignificant" otherwise.

Table 12 contains a summary of statistical tests performed on the results obtained by each of the algorithms on each of the test problems (presented in detail in Tables 3, 4, 5, 6, 7, 8, 9, 10, and 11). In this table the sign "+" is used to denote that a given algorithm performed significantly better than the standard MOEA/D algorithm. The sign "#" is used to denote that a given algorithm performed better than the standard MOEA/D

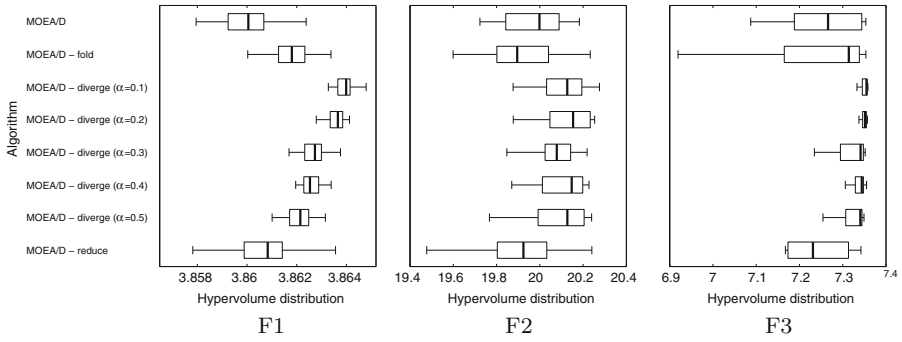


Fig. 33 Distribution of the hypervolume obtained at the last (500th) generation by each of the algorithms for the F1–F3 test problems

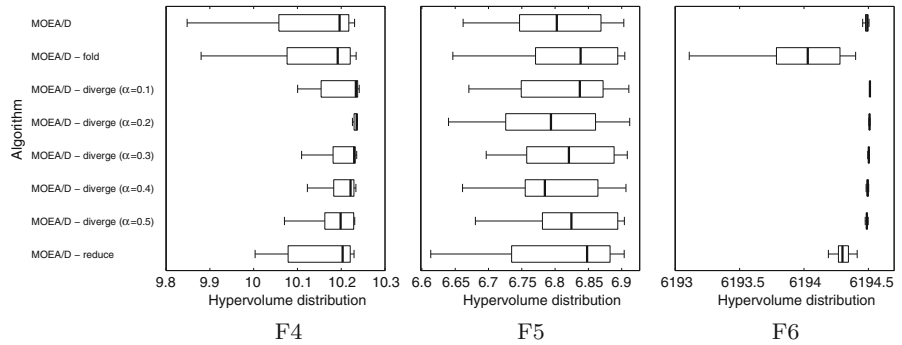


Fig. 34 Distribution of the hypervolume obtained at the last (500th) generation by each of the algorithms for the F4–F6 test problems

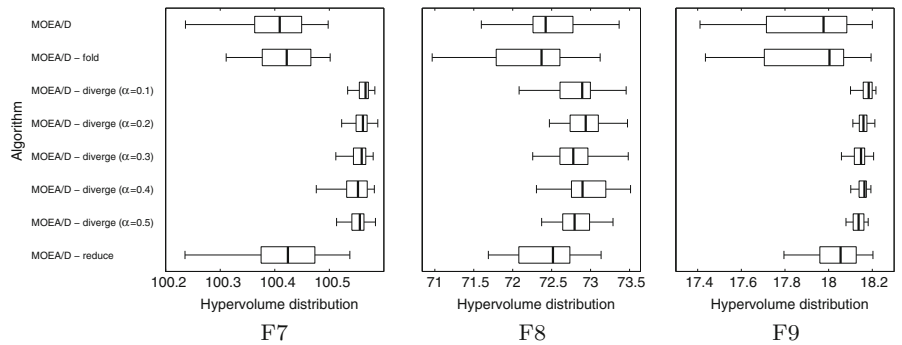


Fig. 35 Distribution of the hypervolume obtained at the last (500th) generation by each of the algorithms for the F7–F9 test problems

Table 3 Median hypervolume values obtained at the last (500th) generation by each of the algorithms for the F1 test problem

Algorithm	Median	Comparison to MOEA/D	
		p value	Interpretation
MOEA/D	3.8601	–	–
MOEA/D—fold	3.8618	3.4053e–005	Significant
MOEA/D—diverge ($\alpha = 0.1$)	3.8640	1.7344e–006	Significant
MOEA/D—diverge ($\alpha = 0.2$)	3.8637	1.7344e–006	Significant
MOEA/D—diverge ($\alpha = 0.3$)	3.8627	1.7344e–006	Significant
MOEA/D—diverge ($\alpha = 0.4$)	3.8625	1.7344e–006	Significant
MOEA/D—diverge ($\alpha = 0.5$)	3.8621	1.7344e–006	Significant
MOEA/D—reduce	3.8608	0.042767	Significant

Table 4 Median hypervolume values obtained at the last (500th) generation by each of the algorithms for the F2 test problem

Algorithm	Median	Comparison to MOEA/D	
		p value	Interpretation
MOEA/D	19.9983	–	–
MOEA/D—fold	19.8955	0.22888	Worse
MOEA/D—diverge ($\alpha = 0.1$)	20.1269	0.00024118	Significant
MOEA/D—diverge ($\alpha = 0.2$)	20.1537	8.9187e–005	Significant
MOEA/D—diverge ($\alpha = 0.3$)	20.0784	0.0049916	Significant
MOEA/D—diverge ($\alpha = 0.4$)	20.1473	0.00013595	Significant
MOEA/D—diverge ($\alpha = 0.5$)	20.1273	0.0011138	Significant
MOEA/D—reduce	19.9243	0.33886	Worse

Table 5 Median hypervolume values obtained at the last (500th) generation by each of the algorithms for the F3 test problem

Algorithm	Median	Comparison to MOEA/D	
		p value	Interpretation
MOEA/D	7.2661	–	–
MOEA/D—fold	7.3140	0.37094	Insignificant
MOEA/D—diverge ($\alpha = 0.1$)	7.3550	1.1265e–005	Significant
MOEA/D—diverge ($\alpha = 0.2$)	7.3518	6.9838e–006	Significant
MOEA/D—diverge ($\alpha = 0.3$)	7.3412	0.0043896	Significant
MOEA/D—diverge ($\alpha = 0.4$)	7.3439	0.0003065	Significant
MOEA/D—diverge ($\alpha = 0.5$)	7.3406	0.0018326	Significant
MOEA/D—reduce	7.2312	0.36004	Worse

Table 6 Median hypervolume values obtained at the last (500th) generation by each of the algorithms for the F4 test problem

Algorithm	Median	Comparison to MOEA/D	
		p value	Interpretation
MOEA/D	10.1963	–	–
MOEA/D—fold	10.1920	0.58571	Worse
MOEA/D—diverge ($\alpha = 0.1$)	10.2334	0.0043896	Significant
MOEA/D—diverge ($\alpha = 0.2$)	10.2353	3.1123e–005	Significant
MOEA/D—diverge ($\alpha = 0.3$)	10.2295	0.0082167	Significant
MOEA/D—diverge ($\alpha = 0.4$)	10.2213	0.0011973	Significant
MOEA/D—diverge ($\alpha = 0.5$)	10.1986	0.00066392	Significant
MOEA/D—reduce	10.2032	0.57165	Insignificant

Table 7 Median hypervolume values obtained at the last (500th) generation by each of the algorithms for the F5 test problem

Algorithm	Median	Comparison to MOEA/D	
		p value	Interpretation
MOEA/D	6.8026	–	–
MOEA/D—fold	6.8383	0.50383	Insignificant
MOEA/D—diverge ($\alpha = 0.1$)	6.8371	0.70356	Insignificant
MOEA/D—diverge ($\alpha = 0.2$)	6.7938	0.68836	Worse
MOEA/D—diverge ($\alpha = 0.3$)	6.8208	0.74987	Insignificant
MOEA/D—diverge ($\alpha = 0.4$)	6.7846	0.74987	Worse
MOEA/D—diverge ($\alpha = 0.5$)	6.8244	0.41653	Insignificant
MOEA/D—reduce	6.8481	0.55774	Insignificant

Table 8 Median hypervolume values obtained at the last (500th) generation by each of the algorithms for the F6 test problem

Algorithm	Median	Comparison to MOEA/D	
		p value	Interpretation
MOEA/D	6,194.4912	–	–
MOEA/D—fold	6,194.0290	1.7344e–006	Worse
MOEA/D—diverge ($\alpha = 0.1$)	6,194.5119	1.7344e–006	Significant
MOEA/D—diverge ($\alpha = 0.2$)	6,194.5090	1.7344e–006	Significant
MOEA/D—diverge ($\alpha = 0.3$)	6,194.5032	5.2165e–006	Significant
MOEA/D—diverge ($\alpha = 0.4$)	6,194.4949	0.0082167	Significant
MOEA/D—diverge ($\alpha = 0.5$)	6,194.4882	0.90993	Worse
MOEA/D—reduce	6,194.2985	1.7344e–006	Worse

Table 9 Median hypervolume values obtained at the last (500th) generation by each of the algorithms for the F7 test problem

Algorithm	Median	Comparison to MOEA/D	
		p value	Interpretation
MOEA/D	100.4091	–	–
MOEA/D—fold	100.4219	0.76552	Insignificant
MOEA/D—diverge ($\alpha = 0.1$)	100.5664	1.7344e–006	Significant
MOEA/D—diverge ($\alpha = 0.2$)	100.5616	2.8434e–005	Significant
MOEA/D—diverge ($\alpha = 0.3$)	100.5594	1.9209e–006	Significant
MOEA/D—diverge ($\alpha = 0.4$)	100.5525	1.7344e–006	Significant
MOEA/D—diverge ($\alpha = 0.5$)	100.5559	2.3534e–006	Significant
MOEA/D—reduce	100.4240	0.11561	Insignificant

Table 10 Median hypervolume values obtained at the last (500th) generation by each of the algorithms for the F8 test problem

Algorithm	Median	Comparison to MOEA/D	
		p value	Interpretation
MOEA/D	72.4230	–	–
MOEA/D—fold	72.3700	0.027029	Worse
MOEA/D—diverge ($\alpha = 0.1$)	72.8925	0.0064242	Significant
MOEA/D—diverge ($\alpha = 0.2$)	72.9374	0.00061564	Significant
MOEA/D—diverge ($\alpha = 0.3$)	72.7774	0.010444	Significant
MOEA/D—diverge ($\alpha = 0.4$)	72.8966	0.0001057	Significant
MOEA/D—diverge ($\alpha = 0.5$)	72.7943	0.0031618	Significant
MOEA/D—reduce	72.5166	0.45281	Insignificant

Table 11 Median hypervolume values obtained at the last (500th) generation by each of the algorithms for the F9 test problem

Algorithm	Median	Comparison to MOEA/D	
		p value	Interpretation
MOEA/D	17.9766	–	–
MOEA/D—fold	18.0039	0.76552	Insignificant
MOEA/D—diverge ($\alpha = 0.1$)	18.1837	1.3601e–005	Significant
MOEA/D—diverge ($\alpha = 0.2$)	18.1596	4.8603e–005	Significant
MOEA/D—diverge ($\alpha = 0.3$)	18.1485	4.0715e–005	Significant
MOEA/D—diverge ($\alpha = 0.4$)	18.1640	1.2381e–005	Significant
MOEA/D—diverge ($\alpha = 0.5$)	18.1373	6.8923e–005	Significant
MOEA/D—reduce	18.0549	0.075213	Insignificant

Table 12 The summary of the results of statistical tests performed for each of the algorithms working on each of the test problems

Algorithm	F1	F2	F3	F4	F5	F6	F7	F8	F9
MOEA/D—fold	+	–	#	–	#	–	#	–	#
MOEA/D—diverge ($\alpha = 0.1$)	+	+	+	+	#	+	+	+	+
MOEA/D—diverge ($\alpha = 0.2$)	+	+	+	+	–	+	+	+	+
MOEA/D—diverge ($\alpha = 0.3$)	+	+	+	+	#	+	+	+	+
MOEA/D—diverge ($\alpha = 0.4$)	+	+	+	+	–	+	+	+	+
MOEA/D—diverge ($\alpha = 0.5$)	+	+	+	+	#	–	+	+	+
MOEA/D—reduce	+	–	+	#	#	–	#	#	#

+ Significantly better than the standard MOEA/D algorithm

Non-significantly better than the standard MOEA/D algorithm

– Worse than the standard MOEA/D algorithm

Table 13 Median values of the Inverted Generational Distance (IGD) indicator obtained at the last (500th) generation by each of the tested algorithms for the F1–F3 test problems

Algorithm	F1	F2	F3
MOEA/D	0.575×10^{-3}	2.108×10^{-2}	0.794×10^{-2}
MOEA/D—fold	0.547×10^{-3}	2.382×10^{-2}	0.895×10^{-2}
MOEA/D—diverge ($\alpha = 0.1$)	0.795×10^{-3}	2.749×10^{-2}	0.631×10^{-2}
MOEA/D—diverge ($\alpha = 0.2$)	0.930×10^{-3}	3.267×10^{-2}	0.726×10^{-2}
MOEA/D—diverge ($\alpha = 0.3$)	1.143×10^{-3}	3.817×10^{-2}	0.985×10^{-2}
MOEA/D—diverge ($\alpha = 0.4$)	1.277×10^{-3}	4.165×10^{-2}	1.074×10^{-2}
MOEA/D—diverge ($\alpha = 0.5$)	1.483×10^{-3}	4.575×10^{-2}	1.286×10^{-2}
MOEA/D—reduce	0.579×10^{-3}	2.385×10^{-2}	0.896×10^{-2}

Table 14 Median values of the Inverted Generational Distance (IGD) indicator obtained at the last (500th) generation by each of the tested algorithms for the F4–F6 test problems

Algorithm	F4	F5	F6
MOEA/D	1.004×10^{-2}	1.269×10^{-2}	0.669×10^{-2}
MOEA/D—fold	0.839×10^{-2}	1.219×10^{-2}	2.316×10^{-2}
MOEA/D—diverge ($\alpha = 0.1$)	0.483×10^{-2}	1.219×10^{-2}	0.791×10^{-2}
MOEA/D—diverge ($\alpha = 0.2$)	0.604×10^{-2}	1.318×10^{-2}	0.940×10^{-2}
MOEA/D—diverge ($\alpha = 0.3$)	0.788×10^{-2}	1.478×10^{-2}	1.056×10^{-2}
MOEA/D—diverge ($\alpha = 0.4$)	0.864×10^{-2}	1.587×10^{-2}	1.211×10^{-2}
MOEA/D—diverge ($\alpha = 0.5$)	1.017×10^{-2}	1.632×10^{-2}	1.326×10^{-2}
MOEA/D—reduce	0.685×10^{-2}	1.185×10^{-2}	1.423×10^{-2}

Table 15 Median values of the Inverted Generational Distance (IGD) indicator obtained at the last (500th) generation by each of the tested algorithms for the F6–F9 test problems

Algorithm	F7	F8	F9
MOEA/D	2.226×10^{-3}	4.624×10^{-2}	2.871×10^{-2}
MOEA/D—fold	2.138×10^{-3}	6.184×10^{-2}	3.067×10^{-2}
MOEA/D—diverge ($\alpha = 0.1$)	2.673×10^{-3}	5.439×10^{-2}	3.400×10^{-2}
MOEA/D—diverge ($\alpha = 0.2$)	4.518×10^{-3}	6.122×10^{-2}	4.501×10^{-2}
MOEA/D—diverge ($\alpha = 0.3$)	4.143×10^{-3}	5.419×10^{-2}	4.529×10^{-2}
MOEA/D—diverge ($\alpha = 0.4$)	6.171×10^{-3}	6.568×10^{-2}	4.146×10^{-2}
MOEA/D—diverge ($\alpha = 0.5$)	5.773×10^{-3}	7.313×10^{-2}	5.213×10^{-2}
MOEA/D—reduce	2.725×10^{-3}	4.552×10^{-2}	2.751×10^{-2}

algorithm but no statistical significance was achieved. The sign “–” is used to denote that a given algorithm performed worse than the standard MOEA/D algorithm.

The algorithm based on the “diverge” method is parameterized by the parameter α which controls the divergence of the weight vectors. In the experiments the value of this parameter was set to $\alpha = 0.1, 0.2, 0.3, 0.4$ and 0.5 . Among these values the value of $\alpha = 0.2$ yielded the best results in the case of test problems F2, F4 and F8. For all the other problems the “diverge” method produced the best results for $\alpha = 0.1$.

The evaluation using the Inverted Generational Distance (IGD) indicator is less conclusive than the one performed using the hypervolume indicator. The values of the IGD obtained in the experiments are presented in Tables 13, 14 and 15. The best (lowest) value for each test problem is marked in bold. The original MOEA/D algorithm obtained the best IGD value for F2 and F6 test problems. In the remaining cases the versions of the algorithm that are proposed in this paper obtained better results. The versions that produced better results than the original MOEA/D algorithm were “fold”, “reduce” and “diverge” with $\alpha = 0.1$. However, these versions produced best results for different test problems. Therefore it is hard to choose the best performing algorithm with respect to the IGD indicator.

6 Conclusion

In this paper three approaches to increasing the selective pressure to the outside of the Pareto front were proposed: the “fold”, “reduce” and “diverge” methods. The investigated methods change the way in which weight vectors affect the working of the multiobjective evolutionary algorithm based on decomposition (MOEA/D). The motivation for increasing the outwards selective pressure is that in the standard version of the algorithm the specimens which are placed near the edges of the Pareto front can only exchange information with specimens that direct their search to the inside of the Pareto front. Such mechanism can be expected to decrease the capability of the algorithm to extend the area of the search.

In order to validate the proposed methods experiments on nine test problems F1–F9 were performed. These test problems were specifically proposed in [17] as benchmarks for testing the MOEA/D algorithm on problems with complicated Pareto sets. The quality of Pareto fronts found by the tested algorithms was evaluated using the hypervolume indicator. The results of the experiments show that the algorithm based on the “diverge” method in which weight vectors are pointed outwards was able to significantly improve the optimization results for all the test problems except F5. In the case of the 3-dimensional problem F6 the improvement was obtained for the values of the parameter $\alpha < 0.5$. For all the problems the best results for the “diverge” method were obtained when the parameter α was set either to 0.1 or to 0.2. A possible explanation of these results is that high values of the α parameter cause the algorithm to put too much pressure on extending the search instead of improving the objectives. Other algorithms, namely “fold” and “reduce” were able to improve the results for some of the test problems but without statistical significance. The only exception is the F1 problem for which all the proposed methods performed significantly better than the standard MOEA/D algorithm. To the contrary, on the F5 problem no significant improvement has been observed, even though some of the tested methods produced better results than the standard MOEA/D algorithm. Distributions of hypervolume values presented in Fig. 34 show that for the F5 problem the quality of results produced by all the algorithms varies greatly between runs. This makes it hard to determine if the observed differences in median values are caused by a better working of a particular algorithm or by coincidental variation of the performance of the search. It is worth noticing that also the authors of the MOEA/D algorithm observed in [17] that this algorithm performs poorly on the test problem F5.

In general, the proposed approach based on extending the selective pressure outwards seems to improve the quality of the results. On most of the test problems the improvement obtained by the “diverge” method is statistically significant at the confidence level 0.05. The “fold” and “reduce” methods performed worse than the “diverge” method with the only exception of the F5 test problem for which the “reduce” method performed best. This suggests that it is not enough to only increase the probability of choosing the outermost weight vectors in the selection process. There seems to be a significant benefit in explicitly directing the search towards the outside of the Pareto front.

Further work may include hybridization of the proposed approach with other optimization algorithms and local search methods. For example, hybrid approaches presented in [1, 14, 16, 26] could be combined with the weight generation scheme proposed in this paper. Local search procedures work on solutions already generated by the main evolutionary algorithm and usually are executed as a separate subroutine. Therefore, it is possible to choose independently which weight generation scheme to use and what kind of solution improvement procedure to employ for a given problem.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

1. Al Moubayed, N., Petrovski, A., McCall, J.: *D²MOPSO*: multi-objective particle swarm optimizer based on decomposition and dominance. In: Hao, J.K., Middendorf, M. (eds.) *Evolutionary Computation in Combinatorial Optimization*, Lecture Notes in Computer Science, vol. 7245, pp. 75–86. Springer, Berlin, Heidelberg (2012)
2. Branke, J., Deb, K., Miettinen, K., Slowinski, R. (eds.): *Multiobjective Optimization, Interactive and Evolutionary Approaches*. Lecture Notes in Computer Science, vol. 5252. Springer, Berlin (2008)
3. Cai, K., Zhang, J., Zhou, C., Cao, X., Tang, K.: Using computational intelligence for large scale air route networks design. *Appl. Soft Comput.* **12**(9), 2790–2800 (2012)
4. Deb, K.: *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, New York, NY (2001)
5. Deb, K., Agarwal, R.: Simulated binary crossover for continuous search space. *Complex Syst.* **9**(2), 115–148 (1995)
6. Deb, K., Goyal, M.: A combined genetic adaptive search (GeneAS) for engineering design. *Comput. Sci. Inform.* **26**, 30–45 (1996)
7. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evolut. Comput.* **6**, 182–197 (2002)
8. Derrac, J., Garca, S., Molina, D., Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evolut. Comput.* **1**(1), 3–18 (2011)
9. Ding, D., Wang, H., Wang, G.: Evolutionary computation of multi-band antenna using multi-objective evolutionary algorithm based on decomposition. In: *Proceedings of the Second International Conference on Information Computing and Applications*, Springer, Berlin, Heidelberg, ICICA'11, pp. 383–390 (2011)
10. Fleischer, M.: The measure of pareto optima. applications to multi-objective metaheuristics. In: *Evolutionary Multi-Criterion Optimization*. Second International Conference, EMO 2003, Springer, Berlin, Heidelberg, EMO'03, pp. 519–533 (2003)
11. Hamdan, M.: On the disruption level of polynomial mutation for evolutionary multi-objective optimization algorithms. *Comput. Inform.* **29**, 783–800 (2010)
12. Ishibuchi, H., Sakane, Y., Tsukamoto, N., Nojima, Y.: Adaptation of scalarizing functions in MOEA/D: An adaptive scalarizing function-based multiobjective evolutionary algorithm. In: Ehr Gott, M., Fonseca, C., Gandibleux, X., Hao, J.K., Sevaux, M. (eds.) *Evolutionary Multi-Criterion Optimization*, Lecture Notes in Computer Science, vol. 5467, pp. 438–452. Springer, Berlin, Heidelberg (2009)
13. Ishibuchi, H., Sakane, Y., Tsukamoto, N., Nojima, Y.: Simultaneous use of different scalarizing functions in MOEA/D. *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, pp. 519–526. ACM, New York, NY (2010)
14. Ke, L., Zhang, Q., Battiti, R.: Multiobjective combinatorial optimization by using decomposition and ant colony. *Tech. rep* (2010)
15. Konstantinidis, A., Yang, K.: Multi-objective energy-efficient dense deployment in wireless sensor networks using a hybrid problem-specific MOEA/D. *Appl. Soft Comput.* **11**(6), 4117–4134 (2011)
16. Li, H., Landa-Silva, D.: An adaptive evolutionary multi-objective approach based on simulated annealing. *Evolut. Comput.* **19**(4), 561–595 (2011)
17. Li, H., Zhang, Q.: Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. *IEEE Trans. Evolut. Comput.* **13**(2), 284–302 (2009)
18. Michalak, K.: The effects of asymmetric neighborhood assignment in the MOEA/D algorithm. *Appl. Soft Comput.* **25**, 97–106 (2014)
19. Miettinen, K.: *Nonlinear Multiobjective Optimization*, International Series in Operations Research and Management Science, vol. 12. Kluwer Academic Publishers, Dordrecht (1999)
20. Ott, L., Longnecker, M.: *An Introduction to Statistical Methods and Data Analysis*. Brooks/Cole Cengage Learning, Belmont, CA (2010)
21. Peng, W., Zhang, Q., Li, H.: Comparison between MOEA/D and NSGA-II on the multi-objective travelling salesman problem. In: Goh, C.K., Ong, Y.S., Tan, K.C. (eds.) *Multi-Objective Memetic Algorithms*, 1st edn, pp. 309–324. Springer Publishing Company, Incorporated, Berlin (2009)
22. Price, K., Storn, R.M., Lampinen, J.A.: *Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)*. Springer, Secaucus, NJ, USA (2005)
23. Srinivas, N., Deb, K.: Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolut. Comput.* **2**, 221–248 (1994)

24. Waldock, A., Corne, D.: Multiple objective optimisation applied to route planning. In: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA, GECCO '11, pp. 1827–1834 (2011)
25. Wilcoxon, F.: Individual comparisons by ranking methods. *Biom. Bull.* **1**(6), 80–83 (1945)
26. Zapotecas Martinez, S., Coello Coello, C.A.: A hybridization of MOEA/D with the nonlinear simplex search algorithm. In: 2013 IEEE Symposium on, Computational Intelligence in Multi-Criteria Decision-Making (MCDM), pp. 48–55 (2013)
27. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evolut. Comput.* **11**(6), 712–731 (2007)
28. Zhang, Q., Liu, W., Li, H.: The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In: 2009. CEC '09. IEEE Congress on, Evolutionary Computation, pp. 203–208 (2009a)
29. Zhang, Q., Zhou, A., Zhao, S., Suganthan, P.N., Liu, W., Tiwari, S.: Multiobjective optimization test instances for the CEC2009 special session and competition. Tech. Rep. CES-487, The School of Computer Science and Electronic Engineering, University of Essex (2009b)
30. Zhou, A., Qu, B.Y., Li, H., Zhao, S.Z., Suganthan, P.N., Zhang, Q.: Multiobjective evolutionary algorithms: a survey of the state of the art. *Swarm Evolut. Comput.* **1**(1), 32–49 (2011)
31. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms - a comparative case study. In: Eiben, A.E., Back, T., Schoenauer, M., Schwefel, H.P. (eds.) *Parallel Problem Solving from Nature—PPSN V*, 5th International Conference, Amsterdam, The Netherlands, September 27–30, 1998, Proceedings, Springer, Lecture Notes in Computer Science, vol. 1498, pp. 292–304 (1998)
32. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Trans. Evolut. Comput.* **3**(4), 257–271 (1999)
33. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: improving the strength Pareto evolutionary algorithm for multiobjective optimization. In: Giannakoglou, K., et al. (eds.) *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*, International Center for Numerical Methods in Engineering (CIMNE), pp. 95–100 (2002a)
34. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Trans. Evolut. Comput.* **7**, 117–132 (2002b)