Comput Optim Appl
DOI 10.1007/s10589-014-9644-1

A survey on multi-objective evolutionary algorithms for many-objective problems

Christian von Lücken · Benjamín Barán · Carlos Brizuela

Received: 26 April 2013

Abstract Multi-objective evolutionary algorithms (MOEAs) are well-suited for solving several complex multi-objective problems with two or three objectives. However, as the number of conflicting objectives increases, the performance of most MOEAs is severely deteriorated. How to improve MOEAs' performance when solving many-objective problems, i.e. problems with four or more conflicting objectives, is an important issue since a large number of this type of problems exists in science and engineering; thus, several researchers have proposed different alternatives. This paper presents a review of the use of MOEAs in many-objective problems describing the evolution of the field, the methods that were developed, as well as the main findings and open questions that need to be answered in order to continue shaping the field.

Keywords Multi-objective optimization problems · Many-objective optimization · Multi-objective evolutionary algorithms

1 Introduction and motivation

In multi-objective optimization problems (MOPs), conflicts among objectives usually prevent from having a single optimal solution but rather a set of trade-off solutions,

C. von Lücken (✉)
Facultad Politécnica, Universidad Nacional de Asunción, San Lorenzo, Paraguay
e-mail: clucken@pol.una.py

B. Barán
Universidad Nacional de Asunción, San Lorenzo, Paraguay
e-mail: bbaran@pol.una.py

C. Brizuela
CISESE, Km 107 Carretera Tijuana-Ensenada, 22860 Ensenada, B.C., Mexico
e-mail: cbrizuel@cisece.mx

Published online: 20 February 2014

called the Pareto optimal set. It is said that a solution Pareto dominates, or simply dominates, another solution if it is not worse in any objective and it is strictly better in at least one objective. Solutions in the Pareto set are non-dominated by any other solution of the feasible solution space [17,22].

In many MOPs, it is difficult to obtain the complete and exact Pareto set, and consequently, an approximation to it is welcome. To this aim, multi-objective evolutionary algorithms (MOEAs) have proven to be well-suited for complex MOPs with two or three objectives [17]. These algorithms cope with multi-objective problems by simulating the basic principles of the evolutionary process on a set of individuals (solutions), i.e. an evolutionary population, by means of the so-called evolutionary operators (fitness assignment, selection, crossover, mutation and elitism) [17,22]. In general, MOEAs differ on the fitness assignment method, but most of them are part of a family, called Pareto-based, which use the Pareto dominance concept as the foundation to discriminate solutions to guide their search [17].

In the last few years, several researchers have pointed out convergence difficulties of MOEAs when they are used for solving many-objective problems, i.e. problems having 4 or more objectives [19,33,41,57,58,83,84]. In case of Pareto-based MOEAs, these difficulties are intrinsically related to the fact that as the number of objectives increases, the proportion of non-dominated elements in the population grows, being increasingly difficult to discriminate among solutions using only the dominance relation [22,33]. Also, several MOEAs are based on data structures and subroutines with complexities that grow exponentially in the number of objectives [19]. Furthermore, the number of solutions required for approximating the entire Pareto front increases exponentially with the dimensionality of the objective space [49,52].

Once a MOEA obtains a Pareto set approximation, it is supposed that a decision maker selects one solution. At this stage, visualization of solution alternatives becomes very important. Even though some methods have been proposed to this aim [34,75,79,81,101], there is still a lack of a simple intuitive way to represent solutions in objective space with four or more objectives.

In recent years, methods for improving the performance of MOEAs in many-objective problems have received much interest since problems of this type abound in science and engineering [34]. In order to study these methods, this work classifies them into two groups: (i) methods using alternative preference relations and (ii) methods transforming the original many-objective problem into a related one. Methods in each of these groups can be sub-classified according to the key concepts in which they are based; thus, yielding the general taxonomy shown in Fig. 1.

Methods that use preference relations compare solutions, one objective at a time, taking into account additional information such as the number of objectives for which one solution is better than another [31,33], the size of improvement [33,95], or the number of subspaces in which a given solution remains as non-dominated [29]. Also, some works proposed relations using fuzzy membership functions to represent preferences [33,62]. Therefore, methods based on preference relation can be further classified into crisp and fuzzy-based methods.

On the other hand, methods transforming the original problem can be classified into: (i) based on scalarization functions, (ii) indicator based, (iii) based on dimensional reduction techniques, and (iv) based on space partitioning. There are two types

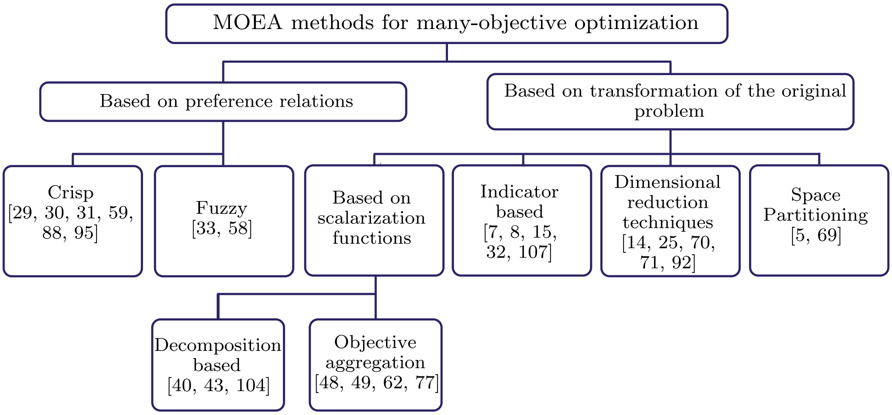
MOEA methods for many-objective optimization
Based on preference relations
Based on transformation of the original problem
Crisp [29, 30, 31, 59, 88, 95]
Fuzzy [33, 58]
Based on scalarization functions
Indicator based [7, 8, 15, 32, 107]
Dimensional reduction techniques [14, 25, 70, 71, 92]
Space Partitioning [5, 69]
Decomposition based [40, 43, 104]
Objective aggregation [48, 49, 62, 77]

Fig. 1 A taxonomy of MOEAs methods for many-objective optimization

of MOEAs using scalarization functions: decomposition, and objective aggregation based. Decomposition methods split the original problem into a collection of scalar functions to be simultaneously optimized [41,43,104]; whereas, objective aggregation methods combine groups of objectives to transform the original many-objective problem into one with a smaller number of objectives [64,76]. Indicator based methods evaluate solutions using a given scalar performance metric (indicator) [6,106]; thus, an indicator based MOEA transforms the many-objective optimization problem into the problem of optimizing an indicator. Finally, in case of space partitioning approaches, they work by considering an alternative subset of objectives at different iterations of the optimization process [5]; this way, these methods transform the original problem into a number of related subproblems.

Although several surveys and books on MOEAs are available [17,22], as far as the authors know, there is no other survey paper except for [49] in 2008 specifically devoted to MOEAs in many-objective problems. Therefore, this work provides an updated and comprehensive coverage of the literature on MOEAs for many-objective problems, presenting the most relevant issues in the field, proposed methods, as well as, experimental and theoretical findings; moreover, it points out several open research questions.

This paper is organized as follows: Sect. 2 presents key multi-objective optimization concepts; Sect. 3 reviews some works regarding two relevant issues of MOEAs in many-objective optimization: the influence of dimension and visualization; Sect. 4 presents methods that use preference relations, while Sect. 5 presents methods based on transforming the original problem into a related one; Sect. 6 presents a summary of experimental works on many-objective optimization using MOEAs. Finally, Sect. 7 presents the conclusions of this work.

2 Multi-objective optimization concepts

In this work, without loss of generality, we consider multi-objective optimization problems in a minimization context; thus, a MOP is defined as follows [22]:

Definition 1 *Multi-objective optimization problem:* Let $\mathcal{F}$ be a set of m objective functions $\{f_1, \ldots, f_m\}$, $f_i : \mathbb{R}^n \Rightarrow \mathbb{R}$, a MOP is defined as:

$$\begin{aligned} \text{Minimize } \ \mathbf{y} = \mathbf{F}(\mathbf{x}) &= (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})) \\ \mathbf{x} &= (x_1, \ldots, x_n) \in \mathcal{X} \subseteq \mathbb{R}^n \\ \mathbf{y} &= (y_1, \ldots, y_m) \in \mathcal{Y} \subseteq \mathbb{R}^m, \end{aligned} \tag{1}$$

subject to

$$\mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), \ldots, g_k(\mathbf{x})) \leq \mathbf{0} \tag{2}$$

$$x_i^{(L)} \leq x_i \leq x_i^{(U)} \quad \forall i \in \{1, \ldots, n\}, \tag{3}$$

$\mathbf{x}$ is a vector of n decision variables, while $\mathbf{y}$ represents an m-dimensional objective vector. Constraints 3 represents $2n$ variable bounds that help to define the decision variable space or decision space $\mathcal{X}$. Objective functions constitute a multi-dimensional space called the objective space, termed as $\mathcal{Y}$. Vector $\mathbf{g}$ is composed of k constraint functions which shape the feasible region. Solutions that do not satisfy constraint functions and/or variable bounds are called infeasible solutions, while solutions that meet all constraints in (3) and (2) are feasible solutions. The set of all feasible solutions $\mathcal{X}_f$ is known as the feasible region. The domain of each f_i is $\mathcal{X}_f$.

For each solution $\mathbf{x} \in \mathcal{X}_f$ there exists a point $\mathbf{y}$ in the objective space. Thus, $\mathcal{X}_f$ defines the feasible objective space $\mathcal{Y}_f$:

$$\mathcal{Y}_f = \mathbf{F}(\mathcal{X}_f) = \bigcup_{\mathbf{x} \in \mathcal{X}_f} \{\mathbf{F}(\mathbf{x})\}. \tag{4}$$

As previously stated, most MOEAs use Pareto dominance in order to compare solutions. In case of many-objective problems, some MOEAs use this concept over a subset of objectives of the original MOP to be solved [12,29,70]. To cover these cases, in a minimization context, the following definition of weak Pareto dominance over a set of objectives is used [12]:

Definition 2 *Weak Pareto dominance relation:* The weak Pareto dominance relation over a set of objectives $\mathcal{F}' \subseteq \mathcal{F}$ is defined as:

$$\preceq_{\mathcal{F}'} = \{(\mathbf{x}, \mathbf{x}') | \mathbf{x}, \mathbf{x}' \in \mathcal{X}_f \wedge \forall f_i \in \mathcal{F}', f_i(\mathbf{x}) \leq f_i(\mathbf{x}')\}. \tag{5}$$

If $(\mathbf{x}, \mathbf{x}') \in \preceq_{\mathcal{F}'}$, it is said that solution $\mathbf{x}$ weakly dominates solution $\mathbf{x}'$ over $\mathcal{F}'$, also denoted by $\mathbf{x} \preceq_{\mathcal{F}'} \mathbf{x}'$. If $(\mathbf{x}, \mathbf{x}') \notin \preceq_{\mathcal{F}'}$, it is said that solution $\mathbf{x}'$ is weakly non-dominated regarding $\mathbf{x}$ over objective set $\mathcal{F}'$, denoted by $\mathbf{x} \npreceq_{\mathcal{F}'} \mathbf{x}'$. To lighten notation, if there is not ambiguity, in case that $\mathcal{F}' = \mathcal{F}$ and $\mathbf{x} \preceq_{\mathcal{F}'} \mathbf{x}'$, it is simply said that $\mathbf{x}$ weakly dominates $\mathbf{x}'$, denoted as $\mathbf{x} \preceq \mathbf{x}'$.

Definition 3 *Pareto dominance relation:* The Pareto dominance relation over a set of objectives $\mathcal{F}' \subseteq \mathcal{F}$ is defined as:

$$\prec_{\mathcal{F}'}=\{(\mathbf{x},\mathbf{x}')|\mathbf{x},\mathbf{x}'\in\mathcal{X}_f \wedge \forall f_i \in \mathcal{F}',\, f_i(\mathbf{x})\leq f_i(\mathbf{x}') \wedge \exists f_j \in \mathcal{F}', f_j(\mathbf{x}) < f_j(\mathbf{x}')\}. \tag{6}$$

If $(\mathbf{x},\mathbf{x}') \in\prec_{\mathcal{F}'}$, it is said that solution $\mathbf{x}$ dominates solution $\mathbf{x}'$ over $\mathcal{F}'$, denoted by $\mathbf{x} \prec_{\mathcal{F}'} \mathbf{x}'$, while if $(\mathbf{x},\mathbf{x}') \notin\prec_{\mathcal{F}'}$ it is said that solution $\mathbf{x}'$ is non-dominated regarding $\mathbf{x}$ over objective set $\mathcal{F}'$, denoted by $\mathbf{x} \not\prec_{\mathcal{F}'} \mathbf{x}'$. In case that $\mathcal{F}' = \mathcal{F}$ and $\mathbf{x} \prec_{\mathcal{F}'} \mathbf{x}'$, it is simply said that $\mathbf{x}$ dominates $\mathbf{x}'$, denoted as $\mathbf{x} \prec \mathbf{x}'$.

Definition 4 *Pareto optimality:* A solution $\mathbf{x} \in \mathcal{X}_f$ is said to be non-dominated considering objectives $\mathcal{F}' \subseteq \mathcal{F}$ regarding a set $\Omega \subseteq \mathcal{X}_f$, if and only if $\nexists\mathbf{x}' \in \Omega$ for which $\mathbf{x}' \prec_{\mathcal{F}'} \mathbf{x}$. If $\mathbf{x}$ is non-dominated regarding $\mathcal{X}_f$ considering $\mathcal{F}'$, it is called a Pareto optimal solution for the given subspace of objectives, while, if $\mathcal{F}' = \mathcal{F}$ it is said that it is a Pareto optimal solution of the problem or simply a Pareto optimal solution.

Pareto optimal solutions form the so-called Pareto set, defined as follows:

Definition 5 *Pareto set:* For a given MOP with a set of objectives $\mathcal{F}$, the Pareto set considering objectives $\mathcal{F}' \subseteq \mathcal{F}$ is defined as:

$$\mathcal{P}^{\star}_{\mathcal{F}'} = \{\mathbf{x} \in \mathcal{X}_f | \nexists \mathbf{x}' \in \mathcal{X}_f \text{ such as } \mathbf{x}' \prec_{\mathcal{F}'} \mathbf{x}\}. \tag{7}$$

The corresponding vectors of $\mathcal{P}^{\star}_{\mathcal{F}'}$ in the objective space defined by $\mathcal{F}'$ form the Pareto front, termed as $\mathcal{PF}^{\star}_{\mathcal{F}'}$. When $\mathcal{F}' = \mathcal{F}$, the sets $\mathcal{P}^{\star}_{\mathcal{F}}$ and $\mathcal{PF}^{\star}_{\mathcal{F}}$ are called the Pareto set and the Pareto front of the problem, respectively.

To the best of our knowledge, the term many-objective problem was used for the first time in [33] to denote multi-objective problems with more than three objectives, which is the general accepted meaning; therefore, it is how this term is used in this work. However, as it was pointed out in [37], according to [97] the true dimension of a many-objective problem must be measured in terms of objective interrelations, we believe that further research in many-objective optimization must develop a formal definition for many-objective problem taking into account this consideration. Provide such definition is out of the scope of this work. The next section introduces other relevant issues regarding the use of MOEAs in many-objective optimization.

3 Relevant issues of using MOEAs in many-objective optimization

3.1 Influence of dimension on the problem hardness for MOEAs

In 1978, Bentley et al. [10] derived a recurrence for computing the average number of non-dominated m-dimensional vectors in a given set of size N, for a fixed m, under the assumption that the magnitude of the components are distributed independently, and that, for each component, the magnitudes chosen for each vector are distinct. Then, solving the recurrence, it is shown that the expected number of non-dominated vectors is bounded by $O((logN)^{m-1})$.

Just with the appearance of the first Pareto-based MOEAs, some authors noted that these methods may not obtain satisfactory solutions on problems having many conflicting objectives due to the increase in the number of non-dominated solutions, however, they did not provide further analysis [35]. Later, in 2001, Deb [22] presented an empirical study that counts the number of non-dominated solutions that exist in one million random populations generated for different combinations of population size and number of objectives. The experimental results showed how, increasing the number of objective functions causes that many solutions belong to the set of non-dominated solutions. Thus, [22] concluded that Pareto-based MOEAs are not able to provide the required selection pressure towards better solutions in order to conduct an efficient evolutionary search and, even elitism may become difficult.

The work of Farina and Amato [33] provided the following general expression for the expected growing of the domain space proportion e containing the points that the Pareto dominance classifies as non-comparable to a given one:

$$e = \frac{2^m - 2}{2^m}. \tag{8}$$

Thus, according to Eq. 8, for a given solution, as the number of objectives grows the proportion of non-comparable solutions regarding that solution tends to one. This way, [33] explained that Pareto dominance may be inadequate to discriminate among solutions in many-objective problems.

Teytaud et al. [97] compared the lower bound for the computation time of a certain class of MOEAs against the upper bound for a pure random search. Analysed MOEAs were those based only on binary Pareto dominance to compare solutions. The work considers a family of multi-objective problems having smooth Pareto sets, fitting the Lipschitz inequality on the Pareto front and for which a reduction in the number of its objectives is not possible. For these problems, the computation time was defined as the count of the objective function evaluations and dominance comparisons necessary for obtaining, with a given probability, an approximated Pareto set with a specified precision for the Haussdorf distance. The analysis in [97] showed that, for the considered problems, when the number of objectives increases the upper bound of the computation time of a pure random search is very close to the lower bound of the studied class of MOEAs; therefore, these MOEAs are (at best) equivalent to random search when the number of objectives is large.

Contrarily to the idea that increasing objective dimensionality makes a problem harder, Brockhoff et al. [14] developed a theoretical runtime analysis of a multi-objective optimizer showing that a given plateau function can become both harder and easier with an additional objective. Consequently, augmenting the number of objectives can both speed up or slow down a MOEA depending on the problem and the chosen objectives, this way, just counting objectives is not enough to measure the hardness of a many-objective problem.

Recently, Schütze et al. [93] analysed the influence of the number of objectives on the hardness of a continuous multi-objective optimization problem in terms of convergence. In [93], authors proposed a set of quadratic test functions to simplify determining the distance of a solution to the Pareto set and to the Pareto front. Using theoretical and empirical observations, [93] concluded that adding an objective to these problems can make them harder; however, the difference is not significant. These findings were used to analyse scalability issues pointed out by several researchers on more complex problems, suggesting that the following questions have to be considered for an efficient numerical treatment of many-objective problems with evolutionary algorithms (EA) [93]: (i) how to determine which solution to keep and which to discard in order to converge toward the Pareto set, (ii) how to increase the probability to improve an individual and (iii) how to deal with the multimodality of a MOP.

As can be noted, while the first works on the hardness of many-objective problems have paid attention to the number of objectives and the ratio of non-dominated solutions, recent works are mostly concerned with the interrelationships of objectives. In spite of great advances, the general question on how the dimension and interactions among objectives affect the hardness for different types of problems, as well as, how to predict the effect of including additional objectives on the performance of a MOEA, are still open. Even more, there is no commonly accepted formal definition of many-objective problems considering both, the number of objectives and the interrelationships among them.

3.2 Visualization in evolutionary many-objective optimization

In general when a population-based algorithm such as a MOEA finishes an optimization run, a decision maker is required to select a particular solution from a set of several possible alternatives. With more than two objectives, visualization of objectives trade-off is not trivial, and it gets more difficult with an increasing number of objectives [73].

Methods to visualize a set of solutions of a many-objective problem may be grouped in three classes. In the first class of methods, objectives are shown in groups of two or three at a time; an example of this type of visualization methods is the scatterplot matrix [22] that shows trade-off between paired objectives. Showing objectives in pairs may become impractical when the number of objectives is large, and may preclude decision maker to visualize the existing relations in the whole set of objectives.

The second class of visualization methods is composed of those showing all objectives at a time as: the parallel-coordinates plot, bar charts, star coordinate systems, petal diagrams and pentagonal representation [73]. Among the aforementioned alternatives, the parallel-coordinates plot [44] may be considered the most applied visualization technique in many-objective problem solving research using MOEAs [34,68,89]. The parallel coordinates plot is a 2D graph, where objective labels are in the horizontal axis and normalized objective values for each objective are in the vertical axis. Using parallel-coordinates plot, it may be possible to heuristically visualize conflict or harmony relations among objectives [34].

Heatmaps are another option to visualize solutions sets of many-objective problems, simultaneously considering the whole set of objectives [81]. A heatmap is a two-dimensional array where each row represents a solution and each column represents a parameter or objective. The colours of cells represent the value. In order to improve visual analysis using heatmaps, Pryke et al. [81] have proposed hierarchical clustering techniques to keep together solutions with similar objective values. More recently, Walker et al. [101] presented a method for reordering a heatmap according to a rank of vectors using a similarity measure based on spectral analysis.

The last group of visualization paradigms presented here considers the characteristics of the solution set to reduce the number of objectives. As an example, Obayashi and Sasaki [79] proposed a two-stage procedure in which the first stage used an unsupervised neural network model, called self organizing maps, to project m-dimensional solutions in a low-dimensional space, and, in a second stage, it uses a hierarchical clustering to facilitate quantitative analysis. Another work that used a two-stage dimensional reduction visualization procedure is [61]. In this case, the first stage maps the solution set to a Pareto set into a 2-dimensional space, while the second stage is the mapping of non-dominated solutions.

Despite of the aforementioned existing alternatives, there is still a lack of intuitive and simple techniques for visualizing many-objective trade-offs, becoming visualization an interesting research topic.

4 MOEAs for many-objective problems based on preference relations

4.1 Crisp alternatives

4.1.1 $(1-k)$*-dominance relation*

In order to refine the ranking of solutions in many-objective problems, Farina and Amato [33] presented the $(1-k)$-dominance relation. This relation is based on counting the number of objectives in which a given solution $\mathbf{x}$ is better, equal or worse than another solution $\mathbf{x}'$. The $(1-k)$-dominance is defined as follows:

Definition 6 $(1-k)$*-dominance*: Let $\mathbf{x}$ and $\mathbf{x}' \in \mathcal{X}_f$, $0 \leq k \leq 1$, $\mathbf{y} = \mathbf{F}(\mathbf{x})$, $\mathbf{y}' = \mathbf{F}(\mathbf{x}')$, then $\mathbf{x}$ $(1-k)$-dominates $\mathbf{x}'$, denoted $\mathbf{x} \prec_{(1-k)} \mathbf{x}'$, iff :

$$n_e(\mathbf{y}, \mathbf{y}') < m \quad \text{and} \quad n_b(\mathbf{y}, \mathbf{y}') \geq \frac{m - n_e(\mathbf{y}, \mathbf{y}')}{k+1}, \tag{9}$$

where

$$n_b(\mathbf{y}, \mathbf{y}') = |\{y_i |\, y_i < y_i', \forall i \in [1, m]\}| \tag{10}$$

$$n_e(\mathbf{y}, \mathbf{y}') = |\{y_i |\, y_i = y_i', \forall i \in [1, m]\}|. \tag{11}$$

The $(1-k)$-dominance serves to propose a new concept of optima, called k-optimality, and its corresponding k-optimal set [33]:

Definition 7 *k-optimality:* A solution $\mathbf{x}$ is k-optimal regarding $\Omega \subseteq \mathcal{X}_f$ iff $\nexists \mathbf{x}' \in \Omega$ such that $\mathbf{x}'$ $(1-k)$-dominates $\mathbf{x}$. The k-optimal set and k-optimal front are defined as the set of k-optimal solutions in the definition domain and objective space, respectively.

When k is zero, the $(1-k)$-dominance relation is equivalent to the dominance relation (Definition 4). Different values of k, in general, provide different subsets of Pareto optimal solutions corresponding to higher degrees of optimality. In order to

Table 1 $n_e(\mathbf{y}, \mathbf{y}')$ values for Example 1 of $(1-k)$-dominance

n_e (row, colum)	**A**	**B**	**C**	**D**	**E**
$\mathbf{A} = (1, 4, 2, 3)$	4	0	1	0	0
$\mathbf{B} = (4, 3, 4, 2)$	0	4	0	0	0
$\mathbf{C} = (2, 4, 1, 7)$	1	0	4	1	1
$\mathbf{D} = (2, 6, 5, 1)$	0	0	1	4	0
$\mathbf{E} = (5, 2, 1, 6)$	0	0	1	0	4

Table 2 $n_b(\mathbf{y}, \mathbf{y}')$ values for Example 1 of $(1-k)$-dominance

n_b (row, colum)	**A**	**B**	**C**	**D**	**E**
$\mathbf{A} = (1, 4, 2, 3)$	0	2	2	3	2
$\mathbf{B} = (4, 3, 4, 2)$	2	0	2	2	2
$\mathbf{C} = (2, 4, 1, 7)$	1	2	0	2	1
$\mathbf{D} = (2, 6, 5, 1)$	1	2	1	0	2
$\mathbf{E} = (5, 2, 1, 6)$	2	2	2	2	0

Table 3 $(m - n_e(\mathbf{y}, \mathbf{y}')/(k+1)$ results for Example 1 considering $k = 0.5$

	A	**B**	**C**	**D**	**E**
A	0.00	2.67	2.00	2.67	2.67
B	2.67	0.00	2.67	2.67	2.67
C	2.00	2.67	0.00	2.00	2.00
D	2.67	2.67	2.00	0.00	2.67
E	2.67	2.67	2.00	2.67	0.00

rank solutions, the $(1-k)$-dominance relation can be used instead of the dominance relation in the non-dominance sorting procedure [22]. The non-dominance sorting procedure classifies solutions by dominance levels. It first determines the set of non-dominated solutions in the population to be classified, this set is called the first front or level of non-dominance, termed as $\mathcal{PF}_1$, then, elements in $\mathcal{PF}_1$ are excluded from the population to obtain the next non-domination front $\mathcal{PF}_2$, and so on; the procedure continues iteratively until all population elements are classified. The following example illustrates the use of the $(1-k)$-dominance relation to rank solutions.

Example 1 Let us consider A, B, C, D, and E solutions of a multi-objective minimization problem such that $\mathbf{F}(A) = \mathbf{A} = (1, 4, 2, 3)$, $\mathbf{F}(B) = \mathbf{B} = (4, 3, 4, 2)$, $\mathbf{F}(C) = \mathbf{C} = (2, 4, 1, 7)$, $\mathbf{F}(D) = \mathbf{D} = (2, 6, 5, 1)$, and $\mathbf{F}(E) = \mathbf{E} = (5, 2, 1, 6)$. If they are classified by non-dominance rank [22], all of them will be in the first rank, i.e. they are all non-dominated; however, using the $(1-k)$-dominance relation, a different rank of solution may be provided. Considering $k = 0.5$, Tables 1, 2, and 3 show the corresponding values for $n_e(\mathbf{y}, \mathbf{y}')$, $n_b(\mathbf{y}, \mathbf{y}')$, and $(m - n_e(\mathbf{y}, \mathbf{y}')/(k+1)$, respectively. These values serve to evaluate Definition 6 to determine the 0.5-dominance relation among solutions in this example, which is shown in Table 4. Thus, according to Table 4 there are the following 0.5-dominance relations: $A \prec_{0.5} C$, $A \prec_{0.5} D$, $C \prec_{0.5} D$, $E \prec_{0.5} C$. In this case, A, B, and E are not 0.5-dominated by any other solution considered in this example; therefore, they

Table 4 0.5-dominance relation table for Example 1

$\prec_{0.5}$	**A**	**B**	**C**	**D**	**E**
A	0	0	1	1	0
B	0	0	0	0	0
C	0	0	0	1	0
D	0	0	0	0	0
E	0	0	1	0	0

receive a rank 1. Between the remaining solutions, as $C \prec_{0.5} D$, C receives a rank 2, and D has rank 3.

In order to test their definitions, [33] used simple discrete examples as well as continuous analytical cases showing the validity of their definitions and how they fit with the common knowledge-based reasoning.

4.1.2 Satisfiability class ordering based on the favour relation

Drechsler et al. [31] used an algorithm previously proposed in [30] to solve a many-objective problem. This algorithm uses the *favour relation* to compare solutions to each other and a method called Satisfiability Class Ordering (SCO) to sort solutions. The favour relation is defined as follows:

Definition 8 *Favour relation:* Given $\mathbf{x}, \mathbf{x}' \in \mathcal{X}_f$, it is said that $\mathbf{x}$ is favoured to $\mathbf{x}'$, denoted as $\mathbf{x} \prec_{favour} \mathbf{x}'$, if and only if

$$n_b(\mathbf{F}(\mathbf{x}), \mathbf{F}(\mathbf{x}')) > n_b(\mathbf{F}(\mathbf{x}'), \mathbf{F}(\mathbf{x})), \qquad (12)$$

where n_b is defined as in Eq. 10.

Since the favour relation is not transitive, the SCO method is used to determine a partial ordering among solutions. The SCO starts by obtaining a relation graph of the solutions to be sorted. In the favour relation graph, nodes represent solutions in objective space and directed edges exist for all pairs of solutions where one is favoured over the other. If there are cycles, nodes are collapsed into a single node. Then, on the basis of the resulting collapsed favour relation graph, a Strongly Connected Components algorithm based on Deep First Search [18] determines a partial ordering between nodes. Finally, solutions that were collapsed on the same node receive the same rank. In the algorithm introduced in [30], at each generation, solutions are classified using satisfiability class ordering, then, the best half of the population is copied to the next iteration without modification while the other half is replaced by new elements.

The following example shows how the SCO ranking method improves classification of non-dominated solutions.

Example 2 Using the same set of vectors of Example 1 and their corresponding n_b values in Table 2, Table 5 shows the favour relation matrix for the five considered solutions. Rows correspond to the left hand side of the relation, while columns are for the right hand side of the relation, thus a value of 1 in the intersection of row **A** and

Table 5 Favour relation [33] between pairs of solutions

$\prec_{favour}$	**A**	**B**	**C**	**D**	**E**
$\mathbf{A} = (1, 4, 2, 3)$	0	0	1	1	0
$\mathbf{B} = (4, 3, 4, 2)$	0	0	0	0	0
$\mathbf{C} = (2, 4, 1, 7)$	0	0	0	1	0
$\mathbf{D} = (2, 6, 5, 1)$	0	0	0	0	0
$\mathbf{E} = (5, 2, 1, 6)$	0	0	1	0	0

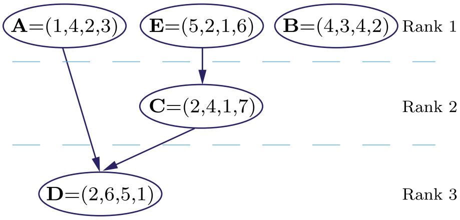

Fig. 2 Favour relation graph and SCO ranking of solutions

column **C** indicates that **A** $\prec_{favour}$ **B**. Considering the aforementioned relation matrix, Fig. 2 shows the favour relation graph and the ranking produced by SCO algorithm [18]. In this example using SCO, solutions A, B, and E have rank 1, solution C has rank 2, while solution D has rank 3.

In [31], the SCO ranking based on favour relation was tested on two instances of a heuristic learning problem for VLSI CAD requiring simultaneous optimization of 6 and 7 objectives, respectively. The results, for small instances, where directly compared with those obtained by an exact method, showing a similar quality. Also, for instances where it is not possible to compute the exact results, the favour-based relation method outperforms an EA using a weighted sum, in 50% of the considered test cases.

4.1.3 Satisfiability class ordering based on ϵ-preferred relation

Based on the favour relation [30,31], Sülflow et al. [95] defined a relation called ϵ-preferred. In this relation, comparison between two solutions is based on counting the number of objectives in which one solution exceeds the other by a predefined threshold. In case of a tie, the favour relation would be used to define which one is better. To build up the ϵ-preferred relation, the following additional relation is defined:

Definition 9 *ϵ-exceed relation:* Let $\mathbf{x}$ and $\mathbf{x}' \in \mathcal{X}_f$, $\mathbf{y} = \mathbf{F}(\mathbf{x})$ and $\mathbf{y}' = \mathbf{F}(\mathbf{x}')$, given a specified vector of limit values $\epsilon = (\epsilon_1, \ldots, \epsilon_m)$, it is said that $\mathbf{x}$ ϵ-exceed $\mathbf{x}'$, denoted as $\mathbf{x} \prec_{\epsilon\text{-}exceed} \mathbf{x}'$ iff:

$$|\{i|y_i < y_i' \wedge |y_i - y_i'| > \epsilon_i, \forall i \in [1, m]\}| > |\{i|y_i' < y_i \wedge |y_i' - y_i| > \epsilon_i, \forall i \in [1, m]\}|. \tag{13}$$

Definitions 8 and 9 are combined in the relation ϵ-preferred as follows [95]:

Table 6 ϵ-preferred relation [95] between pairs of solutions

$\prec_{\epsilon\text{-}pref}$	**A**	**B**	**C**	**D**	**E**
$\mathbf{A} = (1, 4, 2, 3)$	0	1	1	1	1
$\mathbf{B} = (4, 3, 4, 2)$	0	0	0	0	0
$\mathbf{C} = (2, 4, 1, 7)$	0	1	0	1	0
$\mathbf{D} = (2, 6, 5, 1)$	0	0	0	0	0
$\mathbf{E} = (5, 2, 1, 6)$	0	0	1	0	0

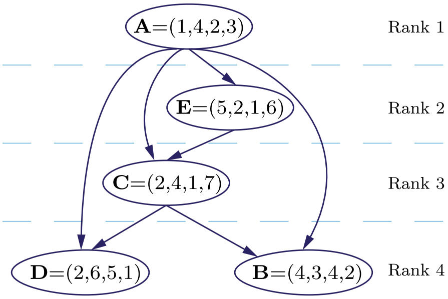

Fig. 3 ϵ-preferred relation graph and SCO ranking of solutions

Definition 10 *ϵ-preferred relation*: given two solutions $\mathbf{x}$ and $\mathbf{x}' \in \Omega$, it is said that $\mathbf{x}$ is ϵ-preferred to $\mathbf{x}'$, denoted as $\mathbf{x} \prec_{\epsilon\text{-}pref} \mathbf{x}'$, iff:

$$\mathbf{x} \prec_{\epsilon\text{-}exceed} \mathbf{x}' \vee (\mathbf{x}' \not\prec_{\epsilon\text{-}exceed} \mathbf{x} \wedge \mathbf{x} \prec_{favour} \mathbf{x}'). \tag{14}$$

In [95], the ϵ-preferred relation replaced the favour relation in the algorithmic framework used in [31]. The following is an example of the ranking of solutions produced by using the ϵ-preferred relation.

Example 3 Taking into account the same set of solutions used in Example 2, Table 6 shows the ϵ-preferred relation for a value of $\epsilon_i = 1.5$ for all $i \in [1, m]$, while Fig. 3 shows the corresponding ϵ-preferred relation graph and the different rank levels. As can be noted, the ranking procedure classifies A in the best rank, E in rank two, C in the third rank and finally, B and D receive a rank value of four.

The ranking method based on the ϵ-preferred was compared in [95] to the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [24], the method called Dominates [56], and the method presented in [31] with the favour relation. As a test problem, a nurse rostering problem with 25 objectives was used. According to the reported results, ϵ-preferred produces better quality solutions and behaves more robustly than the other compared options.

4.1.4 Preference ordering

In order to improve the ranking stage of the NSGA-II [24], di Pierro et al. [28,29] presented two ranking procedures based on the definitions of order of efficiency, and efficiency degree, denoted as PO_k and $PO_{k,z}$, respectively. The efficiency of order k and the order of efficiency are defined as follows:

Definition 11 *Efficiency of order k and order of efficiency*: a solution $\mathbf{x}$ is efficient of order k if it is Pareto optimal in the $\binom{m}{k}$ subspaces of the objective space taking into account only k out of m objectives at a time. The order of efficiency of a solution $\mathbf{x}$, denoted by $K(\mathbf{x})$ is the minimum value k for which $\mathbf{x}$ is efficient.

The PO_k ranking procedure combines Definition 11 with the non-dominance sorting procedure [22] to classify solutions. In PO_k, once all solutions are classified by non-dominance, the order of efficiency is calculated for each solution in $\mathcal{PF}_1$ and the minimum of such values, denoted as K_{min}, is obtained. Later, the PO_k procedure assigns to every solution $\mathbf{x}$ in $\mathcal{PF}_1$ a rank value equals to:

$$R_{PO_k}(\mathbf{x}) = K(\mathbf{x}) - K_{min} + 1. \tag{15}$$

According to Eq. 15, the best elements in $\mathcal{PF}_1$ will receive a rank 1 and the others will follow. Solutions not in $\mathcal{PF}_1$ obtains a rank equals to the worst assigned rank for solutions in the first front plus their corresponding dominance rank. The set of solutions receiving a rank 1 by the PO_k ranking procedure is termed $\mathcal{PF}_1^\star$.

The $PO_{k,z}$ ranking procedure proposed in [29] refines PO_k for those cases in which the set $\mathcal{PF}_1^\star$ of solutions having the best order of efficiency $k^\star \neq m$ posses more than one element, i.e. $|\mathcal{PF}_1^\star| > 1$. $PO_{k,z}$ ranking strategy is based on the following definition:

Definition 12 *Efficiency of order k with degree z*: a solution $\mathbf{x}$ is called efficient of order k with degree z if x is non-dominated by any member of the Pareto set for exactly z out of the possible $\binom{m}{k}$ k-element subsets of the objectives.

Once PO_k is applied, procedure $PO_{k,z}$ uses Definition 12 to compute the degree of efficiency of order $k^\star - 1$ for all $\mathbf{x} \in \mathcal{PF}_1{}^\star$. Thus, the rank value computed by $PO_{k,z}$ for elements $\mathbf{x} \in \mathcal{PF}_1{}^\star$ is:

$$R_{PO_{k,z}}(\mathbf{x}) = Z_{max}^{k^\star-1} - Z^{k^\star-1}(\mathbf{x}) + 1, \tag{16}$$

where $Z^{k^\star-1}(\mathbf{x})$ is the degree of efficiency of solution $\mathbf{x}$ and $Z_{max}^{k^\star-1}$ is the maximum degree of efficiency of order $k^\star - 1$. Finally, to those solutions not in $\mathcal{PF}_1{}^\star$ the final rank is obtained by adding $Z_{max}^{k^\star-1}$ to their current PO_k rank.

Example 4 Consider the objective vectors in previous examples: $\mathbf{A} = (1, 4, 2, 3)$, $\mathbf{B} = (4, 3, 4, 2)$, $\mathbf{C} = (2, 4, 1, 7)$, $\mathbf{D} = (2, 6, 5, 1)$, and $\mathbf{E} = (5, 2, 1, 6)$, they are efficient of order 4 since all of them are non-dominated. Taking into account three objectives at a time there are $\binom{4}{3} = 4$ possible subspaces, Table 7 shows the dominance relations that hold in the different 3-objective subspaces. Solutions C and D are not efficient of order 3 since they are dominated in at least one 3-objective subspace, while solutions A, B, and E are efficient of order 3 since they are non-dominated by any vector for all combination of three objectives. However, considering 2 objectives at a time, A, B, and E are dominated in at least one 2-objective subspace. Therefore, the values of $K(A)$, $K(B)$, and $K(E)$ are equal to 3, while $K(C)$, and $K(D)$ are 4 and the minimum order of efficiency for the considered solutions is $k^\star = 3$. Using Eq. 15,

Table 7 Dominated solutions in 3-objective subspaces

Subspace	Dominance relations
(f_1, f_2, f_3)	$A \prec D$
(f_1, f_2, f_4)	$E, A \prec C$
(f_1, f_3, f_4)	–
(f_2, f_3, f_4)	$E \prec C$

Table 8 Solutions that dominates others in each 2-objective subspaces

Subspace	**A**	**B**	**E**
(f_1, f_2)	–	–	–
(f_1, f_3)	–	$A \prec B$	–
(f_1, f_4)	–	–	$A, B \prec E$
(f_2, f_3)	$E \prec A$	$E \prec B$	–
(f_2, f_4)	$B \prec A$	–	–
(f_3, f_4)	–	–	–

the PO_k procedure assigns a rank of 1 for A, B, and E, while, C and D receive a rank of 2.

To improve the ranking of solutions, the $PO_{k,z}$ procedure calculates the degree of efficiency of order 2 for solutions with the best order of efficiency $\mathcal{PF}_1^\star = \{A, B, E\}$. To determine these values, Table 8 shows which solution dominates each other in the existing 2-objective subspaces. As can be noted, **A** and **B** are not dominated by any solution in 4 out of 6 subspaces, while **E** is not dominated by another solution in 5 out of 6 subspaces; thus, $Z^2(A) = 4$, $Z^2(B) = 4$, $Z^2(E) = 5$. In this way, according to Eq. 16, the rank assigned by the $PO_{k,z}$ procedure for E is 1, for A and B is 2. The $PO_{k,z}$ rank for solutions C and D is 7 as result of adding the maximum degree of efficiency or order 2, $Z^2_{max} = 5$, to their previous PO_k rank value.

To support the viability of the preference order definition as alternative to Pareto dominance, [29] replaced the ranking procedure of the NSGA-II with PO_k and $PO_{k,z}$ to compare the three methods on DTLZ1, DTLZ2, DTLZ3, and DTLZ5 test functions [25], considering 4, 5, 6, 7, and 8 objectives. The experimental evaluation was produced using five metrics: Generational distance (GD) [98], Hypervolume [108], Diversity Metric 1 and 2 (DM1 and DM2) [57] and Coverage [109, 111]. The results in [29] showed that, according to the evaluated metrics, the PO_k and $PO_{k,z}$ methods are able to attain a better convergence to the true Pareto set than the original NSGA-II for the considered problems; however, they are less effective than the original NSGA-II in maintaining a good diversity of solutions over the whole Pareto surface.

4.1.5 $-\epsilon$-DOM ranking

Köppen and Yoshida [60] proposed four procedures to replace the NSGA-II [24] crowding distance assignment for many-objective optimization problems. In NSGA-II, the crowding distance serves as a secondary selection metric to promote diversity among solutions in different fronts; however, as in a many-objective problem almost

Table 9 Values for *mepsd* and $-\epsilon$-DOM rank for solutions in the Example 5

	Pairwise *mepsd* values					$-\epsilon$-DOM	
	(1, 4, 2, 3)	(4,3,4,2)	(2,4,1,7)	(2,6,5,1)	(5,2,1,6)	Value	Rank
A = (1,4,2,3)	–	3	4	3	3	3	1
B = (4,3,4,2)	1	–	5	3	4	1	3
C = (2,4,1,7)	1	3	–	4	3	1	3
D = (2,6,5,1)	2	2	6	–	5	2	2
E = (5,2,1,6)	2	3	2	4	–	2	2

all solutions are classified in the same front (the first) crowding distance becomes a main selection criterion guiding the evolutionary search. An experimental comparison among these four procedures showed that the so-called $-\epsilon$-DOM provided the best trade-off for the metrics and problems considered in [60]; therefore it is the one presented here.

In [60] an auxiliary function, called *mepsd*, is used as foundation to define the $-\epsilon$-DOM rank. Such function, receives a pair of solutions $(\mathbf{x}, \mathbf{x}')$ and returns the smallest value, which, if subtracted from $\mathbf{F}(\mathbf{x}')$ makes $\mathbf{x}'$ weakly dominate $\mathbf{x}$. The $-\epsilon$-DOM rank of solution $\mathbf{x}$ is formally defined as follows:

Definition 13 *$-\epsilon$-DOM rank:* The $-\epsilon$-DOM rank of a solution $\mathbf{x}$ regarding a population P is:

$$-\epsilon\text{-DOM}(\mathbf{x}) = \min_{\mathbf{x}' \in P}\{mepsd(\mathbf{F}(\mathbf{x}), \mathbf{F}(\mathbf{x}'))\}, \tag{17}$$

where

$$mepsd(\mathbf{F}(\mathbf{x}), \mathbf{F}(\mathbf{x}')) = \begin{cases} 0 \quad \text{if } f_i(\mathbf{x}') \leq f_i(\mathbf{x}) \\ \max_{i=1}^{m}\{f_i(\mathbf{x}') - f_i(\mathbf{x}) : f_i(\mathbf{x}) < f_i(\mathbf{x}')\}. \end{cases} \tag{18}$$

As can be noted in Definition 13, a higher $-\epsilon$-DOM value for a solution $\mathbf{x} \in P$ indicates a greater effort for another solution in P to dominate $\mathbf{x}$.

Example 5 Table 9 shows pairwise *mepsd* values and $-\epsilon$-DOM ranking values using the same set of solutions of previous examples. The *mepsd* value in the row corresponding to $\mathbf{A} = (1, 4, 2, 3)$ and column $\mathbf{B} = (4, 3, 3, 2)$, indicates that 3 is the smallest value that have to be subtracted from each element of $\mathbf{B}$ for B to dominate A; i.e. $mepsd(\mathbf{A}, \mathbf{B}) = \max\{4 - 1, 0, 4 - 2, 0\} = 3$, thus, if $\mathbf{B} = \mathbf{B} - (3, 3, 3, 3)$, then $B \prec A$. The last two columns in Table 9 indicates the $-\epsilon$-DOM values for solutions in the current example (minimum *mepsd* value) and the corresponding rank. In this case $-\epsilon\text{-DOM}(A) = 3$, $-\epsilon\text{-DOM}(B) = 1$, $-\epsilon\text{-DOM}(C) = 1$, $-\epsilon\text{-DOM}(D) = 2$, $-\epsilon\text{-DOM}(E) = 2$. The best (greater) $-\epsilon$-DOM value is for solution A that therefore is in rank 1, rank 2 is for solutions D and E, and finally rank 3 is for solutions B and C.

The four alternatives to the NSGA-II crowding distance method proposed in [60] were evaluated in the same work using DTLZ2, DTLZ3, DTLZ6 [25], and the Pareto-Box problem [62] with 2, 8, and 15 objectives. For comparison purposes, Convergence [26,62] and Coverage [109,111] metrics were used. The experiments in [62] concluded that the new methods provide a better convergence than the standard crowding procedure, and the $-\epsilon$-DOM also provides a good Pareto front coverage. Also, [60] stated that it may be beneficial to use different ranking procedures as the optimization run executes.

4.1.6 Expansion dominance relation

In order to induce an appropriate ranking of solutions, Sato et al. [87] proposed a method that expands or contracts the dominance area by moving the location of each solution in the objective space. Using an user-defined parameter $S = [S_1, ..., S_m]$, the expansion relation can be defined as follows:

Definition 14 *Expansion dominance relation:* It is said that a solution $\mathbf{x}$ dominates another solution $\mathbf{x}'$ under the expansion relation iff

$$\forall i : f_i'(\mathbf{x}) \leq f_i'(\mathbf{x}') \;\; \wedge \;\; \exists i \;\; \text{such that } f_i'(\mathbf{x}) < f_i'(\mathbf{x}'), \tag{19}$$

where

$$f_i'(\mathbf{x}) = \frac{r(F(\mathbf{x})) \cdot sin(\omega_i + S_i \cdot \pi)}{sin(S_i \cdot \pi)} \quad \forall \; i \in [1, m], \tag{20}$$

$r(F(\mathbf{x}))$ is a norm of $\mathbf{F}(\mathbf{x})$, ω_i is the angle between $\mathbf{F}(\mathbf{x})$ and a vector with $f_i(\mathbf{x})$ as the only non zero component.

If $S_i = 0.5$, the expansion relation is equivalent to dominance. If $S_i < 0.5$, then, $f_i'(\mathbf{x}) < f_i(\mathbf{x})$ and the dominance area is reduced. On the other hand, if $S_i > 0.5$, the dominance area is increased. Expansion of the dominance area produces a finer ranking of solutions, which may improve selection.

Once all solutions in objective space are moved to their new location, using Definition 14, the classical Pareto dominance rank [22], briefly described in Sect. 4.1.1 may be used to classify solutions using these new objective values. An example of this method is presented in what follows.

Example 6 Let us consider again the same set of solutions of the previous examples. Using $S_i = 0.25$ for $i \in [1, m]$ Table 10 shows the corresponding $\mathbf{F}'$ values and the new ranking of solutions. These S_i values corresponds to the maximum expansion of the dominance area [87]. Thus, to obtain the first component of $\mathbf{F}'(A)$, we calculate $r(F(A))$ as $\sqrt{1^2 + 4^2 + 2^2 + 3^2} = 5.477$, then $w_i = arcos((1 \cdot 1)/(\sqrt{1 \cdot 1} \cdot 5.477) = 1.39$, finally using Eq. 20 we obtain a value of 6.39. Once all $\mathbf{F}'(x)$ values are calculated, these values are used to determine the non-dominated ranking of solutions. In this case, the best rank is for solution A, which is non-dominated, in the second rank are solutions B, C and E which are dominated by A, finally, the last rank is for solution D, dominated by B and D.

Table 10 $\mathbf{F}'$ values and ranking for the Example 6 of the expansion relation

Solution	$\mathbf{F}(x)$	$\mathbf{F}'(x)$	Rank
A	$(1, 4, 2, 3)$	$(6.39,\ 7.74,\ 7.10,\ 7.58)$	1
B	$(4, 3, 4, 2)$	$(9.39,\ 9.00,\ 9.39,\ 8.40)$	2
C	$(2, 4, 1, 7)$	$(10.12,\ 11.35,\ 9.31,\ 11.58)$	2
D	$(2, 6, 5, 1)$	$(9.87,\ 11.48,\ 11.40,\ 9.06)$	3
E	$(5, 2, 1, 6)$	$(11.40,\ 9.87,\ 9.06,\ 11.48)$	2

In [11], Branke et al. proposed the Guided Multi-Objective Evolutionary Algorithm (G-MOEA) which, as the method proposed by Sato et al. [87], is based on modifying the dominated region of each solution. However, in G-MOEA the decision maker must specify trade-off values for each pair of objectives to define a set of auxiliary linear objective functions that are used for producing the dominance modification. Thus, although both methods have a similar basic idea, the number of trade-off relations the user must specify results in a severe impediment to scale the G-MOEA to a large number of objectives in many-objective problems.

Sato et al. [87] used the knapsack problem [108] with 2, 3, 4, and 5 objectives to compare the NSGA-II [24] to its modified version with the expansion relation. The results of both methods were evaluated considering the following metrics: Spread [22], Hypervolume [108], and Inverse Generational Distance (IGD) [98]. Authors concluded that the expansion relation may be useful in combination with other methods since the relation can improve either convergence or diversity, but not both simultaneously. In a more recent work [71], modified NSGA-II with the expansion relation has shown a remarkable performance to obtain solutions on the knee region of DTLZ problems [25] when compared to other relations such as the average ranking [9], the preference order [29], and the favour relation [30].

4.2 Fuzzy alternatives

4.2.1 $(1 - k_F)$*-dominance relation*

In order to extend $(1-k)$-dominance relation (Definition 6) to a fuzzy domain, Farina and Amato [33] proposed to fuzzify the count of objectives that may be considered better, equal, or worse between two solutions by determining membership functions μ_b^i, μ_e^i, and μ_w^i for each objective function i. Thus, the fuzzy $(1 - k_F)$-dominance is defined as follows:

Definition 15 $(1 - k_F)$-*dominance*: Let $\mathbf{x}$ and $\mathbf{x}' \in \mathcal{X}_f$, $0 \leq k_F \leq 1$ and $\mathbf{y} = \mathbf{F}(\mathbf{x})$, $\mathbf{y}' = \mathbf{F}(\mathbf{x}')$, then $\mathbf{x}$ $(1 - k_F)$-dominates $\mathbf{x}'$ iff:

$$n_e^F(\mathbf{y}, \mathbf{y}') < m \quad \text{and} \quad n_b^F(\mathbf{y}, \mathbf{y}') \geq \frac{m - n_e^F(\mathbf{y}, \mathbf{y}')}{k_F + 1}, \tag{21}$$

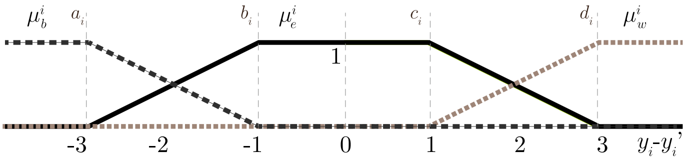

Fig. 4 Fuzzy membership functions μ_e^F, μ_b^F, μ_w^F for Example 7

Table 11 $n_e(\mathbf{y}, \mathbf{y}')$ values for Example 7 of the $(1 - k_F)$-dominance

n_e^F	**A**	**B**	**C**	**D**	**E**
A	4	2.5	3	2	1.5
B	2.5	4	1.5	2.5	2
C	3	1.5	4	1.5	2.5
D	2	2.5	1.5	4	0
E	1.5	2	2.5	0	4

where

$$n_b^F(\mathbf{y}, \mathbf{y}') = \sum_{i=1}^{m} \mu_b^i \left(y_i - y_i'\right) \tag{22}$$

$$n_e^F(\mathbf{y}, \mathbf{y}') = \sum_{i=1}^{m} \mu_e^i \left(y_i - y_i'\right) \tag{23}$$

$$n_w^F(\mathbf{y}, \mathbf{y}') = \sum_{i=1}^{m} \mu_w^i \left(y_i - y_i'\right) \tag{24}$$

$$n_b^F + n_e^F + n_w^F = m. \tag{25}$$

Depending on the selected membership function it may be necessary to provide additional parameters. For example, for the trapezoidal membership function [33], the decision maker must provide, for each objective i, a 4-tuple (a_i, b_i, c_i, d_i) that defines the shape of the trapezoid (see Fig. 4), i.e. the range of values for which the difference between objective values can be considered meaningless, or not.

Using Definition 15, k_F-optimality is formally defined as:

Definition 16 *k_F-optimality:* Vector $\mathbf{x}$ is k_F-optimal if and only if $\nexists\, \mathbf{x}' \in \Omega$ such that $\mathbf{x}'$ $(1 - k_F)$-dominances $\mathbf{x}$. The k_F-optimal set and k_F-optimal front are defined as the set of k_F-optimal solutions in function's domain and its corresponding image in the objective space, respectively.

Example 7 In order to classify the same set of solutions of all previous examples with objective vectors: $\mathbf{A} = (1, 4, 2, 3)$, $\mathbf{B} = (4, 3, 4, 2)$, $\mathbf{C} = (2, 4, 1, 7)$, $\mathbf{D} = (2, 6, 5, 1)$ and $\mathbf{E} = (5, 2, 1, 6)$ by k_F-dominance, let us consider, for all objectives, trapezoidal membership functions μ_b^i, μ_f^i and μ_w^i with parameters $(a_i, b_i, c_i, d_i) = (-3, -1, 1, 3)$, as shown in Fig. 4. Tables 11 and 12 show values for n_e^F and n_b^F for

Table 12 $n_b^F(\mathbf{y}, \mathbf{y}')$ values for Example 7 of the $(1 - k_F)$-dominance

n_b^F	**A**	**B**	**C**	**D**	**E**
A	0	1.5	1	1.5	2
B	0	0	1	1	1
C	0	1.5	0	1.5	1
D	0.5	0.5	1	0	2
E	0.5	1	0.5	2	0

Table 13 $(m - n_e^F(\mathbf{y}, \mathbf{y}'))/(k_F + 1)$ values for the Example 7, $k_F = 0.5$

	A	**B**	**C**	**D**	**E**
A	0	1	0.67	1.33	1.67
B	1	0	1.67	1	1.33
C	0.67	1.67	0	1.67	1
D	1.3	1	1.67	0	2.67
E	1.67	1.33	1	2.67	0

Table 14 0.5_F-dominance relation table for Example 7

	A	**B**	**C**	**D**	**E**
A	0	1	1	1	1
B	0	0	0	1	0
C	0	0	0	0	1
D	0	0	0	0	0
E	0	0	0	0	0

the different possible pairs of solutions, using these values, Table 13 shows the results of $(m - n_e^F(\mathbf{y}, \mathbf{y}'))/(k_F + 1)$ for $k_F = 0.5$. Then, with the aforementioned values, it is possible to apply the Definition 15 to each pair of solutions producing the 0.5_F-dominance relation table shown in Table 14 . Thus, replacing the Pareto dominance relation by the 0.5_F-dominance relation in the non-dominated sorting procedure [22], the ranking of solutions is: **A** is in rank 1, **B** and **C** are in rank 2, and vectors **D** and **E** are in rank 3.

4.2.2 Fuzzy-dominance-driven genetic algorithm

In [62], Köppen et al. proposed the Fuzzy-Dominance-Driven Genetic Algorithm (FDD-GA), an algorithm based on a fuzzy extension of the Pareto dominance relation to calculate dominance degrees among each pair of solutions in the population. In [62] two different dominance degrees between solutions $\mathbf{x}$ and $\mathbf{x}'$ are defined by functions μ_a and μ_p, as follows:

Definition 17 *Fuzzy Pareto dominance relation*: let $\mathbf{x}$ and $\mathbf{x}'$ be solutions in $\mathcal{X}_f$, $\mathbf{y} = \mathbf{F}(\mathbf{x})$, $\mathbf{y}' = \mathbf{F}(\mathbf{x}')$, it is said that $\mathbf{x}$ dominates $\mathbf{x}'$ by degree μ_a, denoted $\mathbf{x} \prec_{\mu_a} \mathbf{x}'$, where

$$\mu_a(\mathbf{y}, \mathbf{y}') = \frac{\prod_{i=1, y_i, y_i' \neq 0}^{m} \min(y_i, y_i')}{\prod_{i=1, y_i \neq 0}^{m} y_i} \tag{26}$$

Table 15 μ_P and r_P values for solutions considered in Example 8

μ_P	**A**	**B**	**C**	**D**	**E**	r_P	Rank
$\mathbf{A} = (1, 4, 2, 3)$	–	0.125	0.214	0.133	0.100	0.214	1
$\mathbf{B} = (4, 3, 4, 2)$	0.500	–	0.214	0.400	0.267	0.500	4
$\mathbf{C} = (2, 4, 1, 7)$	0.500	0.125	–	0.133	0.400	0.500	4
$\mathbf{D} = (2, 6, 5, 1)$	0.333	0.250	0.143	–	0.067	0.333	2
$\mathbf{E} = (5, 2, 1, 6)$	0.250	0.167	0.429	0.067	–	0.429	3

and that $\mathbf{x}$ is dominated by $\mathbf{x}'$ at degree μ_p, denoted $\mathbf{x} \prec_{\mu_p} \mathbf{x}'$, where

$$\mu_p(\mathbf{y}, \mathbf{y}') = \frac{\prod_{i=1, y_i, y_i' \neq 0}^{m} \min(y_i, y_i')}{\prod_{i=1, y_i' \neq 0}^{m} y_i'}. \tag{27}$$

In case of having an y_i or y_i' equal to 0, the corresponding index in the products is excluded from the numerator and denominator.

Note that if $\mathbf{x} \preceq \mathbf{x}'$, the elements to multiply in the numerator are from $\mathbf{y}$ since the corresponding elements in $\mathbf{y}'$ are greater than or equal to $\mathbf{y}$; therefore, $\mu_a(\mathbf{y}, \mathbf{y}') = 1$ while $\mu_p(\mathbf{y}', \mathbf{y}) \leq 1$.

The fuzzy rank of a given solution $\mathbf{x}$ in a population P is the maximum degree of being dominated by any other element, the smaller it is the better [62]:

$$r_P(\mathbf{x}) = \max_{\mathbf{x}' \in \mathbf{P}, \{\mathbf{x} \neq \mathbf{x}'\}} \mu_p(\mathbf{F}(\mathbf{x}), \mathbf{F}(\mathbf{x}')). \tag{28}$$

Example 8 Table 15 presents μ_p and r_P values for the same set of solutions and objective values considered in previous examples. The value 0.125 corresponding to $\mu_p(\mathbf{F}(A), \mathbf{F}(B))$ is calculated by dividing $1 \cdot 3 \cdot 2 \cdot 2$ by $4 \cdot 3 \cdot 4 \cdot 2$. As can be noted, in this case, A is considered as the best, D is in the second position, E in the third, while solutions B and C are in the worst rank.

In [62] the usefulness of the FDD-GA approach was demonstrated by means of an experimental performance comparison against the NSGA-II [24] for the Pareto-Box problem with 20 objectives.

4.3 Summary of MOEAs for many-objective problems based on preference relations

Methods based on preference relations consider the growing proportion of non-dominated solutions as the principal challenge in solving many-objective optimization problems. Therefore, in general, surveyed relations use additional information to provide a finer ranking of solutions. The relation to be used depends on the goal of the optimization process. Some of these relations are parameter free and can be used to gain insight into the characteristics of the problem at hand in the early stages of the optimization process or implemented into a posteriori optimization method, i.e.

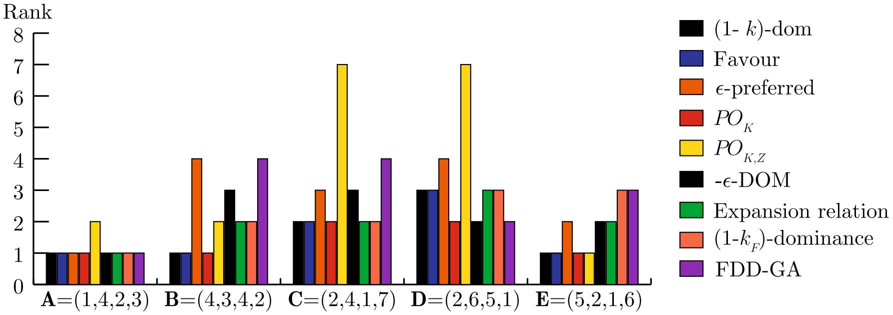

Fig. 5 Ranking of example solutions for the surveyed relations

to select a solution after the generation of the Pareto set approximation [72]. However, relations such as the ϵ-preferred [95], the expansion dominance [87], and the $(1 - k_F)$-dominance [33] require the decision maker to define more problem-specific parameters to guide the search towards a region of his interest; therefore, these relations are best suited for being implemented into interactive approaches. Studying how the different settings of these parameters influence the search, may serve to develop techniques that automatically adjust those values as evolution proceeds, as well as interactive frameworks in order to apply these algorithms to real life many-objective optimization problems.

Throughout this section, examples were used to show how each studied preference relation compares a set of five non-dominated solutions. Using only the dominance relation, all of these solutions are equally good; however, using alternative relations they can be discriminated. In order to visualize the difference between the Pareto optimality and the optimality induced by the surveyed relations, Fig. 5 shows the ranks that solutions receive in each example. As could be noted, since each alternative relation represents a different preference, a given non-dominated solution may be classified differently for some of the considered relations. In turn, by classifying solutions differently, each preference relation possibly guides the evolutionary process towards a different region of the search space; therefore, comparing the results of methods based on different preferences require the development of methodologies incorporating additional considerations [71].

Moreover, some relations provide a finer ranking of solutions than others. Increasing the selection pressure may improve convergence; however, it also may lead to a decrease in the diversity of solutions and other undesirable effects. Analysing how biases introduced by preference relations affect the trade-off between convergence and diversity in different types of problems would be an interesting topic for further research. It may be particularly interesting to determine the features of the problems for which a given additional information (as the count of the number of best objective values) conducts the search toward solutions in the global Pareto set or to get stuck at local optima.

An advantage of using alternatives to Pareto dominance relation is that they are relatively easy to implement into existing (Pareto-based) MOEAs. In cases such as the k-dominance, the $(1 - k_F)$-dominance [33] and the expansion relation [87] they can be implemented in the NSGA-II [24] framework by only replacing the Pareto dominance by the new relation, whereas other parts of the algorithm remains without modifications. Since the favour relation [31] and the ϵ-preferred relation [95] are not transitive, they require to replace the fast non-dominance sorting procedure by the SCO. Also, preference ordering methods [29] completely replace the original Pareto dominance ranking. In case of the FDD-GA, once each solution was compared to each other, there is no need of an additional sorting procedure to get the fuzzy rank of a solution since it is the maximum value achieved in comparisons. Finally, implementing the ϵ-DOM within the NSGA-II only affects the crowding assignment procedure. Despite several works propose modifications of existing MOEAs, there is a lack of studies analysing which are the characteristics that make a specific MOEA the best choice to incorporate a new ranking, based on a given preference relation. Also, since different relations may share the same implementation framework, it may be useful to analyse how to combine them at different moments of the evolutionary process to improve the search.

In general, procedures to rank solutions using the relations studied in this section can be divided into two stages: first, solutions are compared to each other, one objective at a time; second, using the comparison results a partial ordering of solutions is produced. In most cases, the first stage have the same computational cost of the fast non-dominated sorting procedure [24], which is $O(m|P|^2)$, with $|P|$ representing the population size and m the number of objectives. However, the preference ordering methods [29] require to compare solutions for different subspaces of objectives until a differentiation may be produced; there is no known efficient method to calculate this relation; thus, it becomes impractical for a large number of objectives. According to [22], as the population size increases, the proportion of non-dominated solutions decreases; therefore, reducing the time complexity of methods that classify solutions may be a research topic in order to handle larger population sizes for many-objective problems. Also, it must be noted that, for some relations, it is possible to obtain a contradictory comparison result if two solutions are only compared with each other or if they are compared by using their rank regarding the whole population. In Table 15, for example, it could be noted that if FDD-GA only compares solutions C with E, C will be considered to be better, whereas regarding the rank produced considering the five solutions E is better than C. This behaviour must be taken into account when using a relation without a complete rank of solutions, for instance in a tournament selection framework.

5 MOEAs for many-objective problems based on transformations of the original problem

5.1 Scalarization-based methods: objective aggregation approaches

5.1.1 Objective aggregation using a desirability index

In several cases, objectives of a MOP can be grouped by the decision maker into different categories. For these problems, Kruisselbrink et al. [64] proposed the use of desirability functions to combine groups of objectives in order to recast the original many-objective optimization problem into an optimization problem with a moderate number of objectives. Basically, a desirability function serves to associate a quality criteria to a value ranging from 0, in case the value is unacceptable, to 1, in case the value fits a desired target. Desirability functions can be combined according to predefined categories obtaining a single quality value for each category, named as its desirability index. Thus, the set of all desirability indexes forms the new multi-objective problem to be optimized. Moreover, [64] explains how to map problems consisting of a small number of objectives m' and a number of fuzzy or soft constraints k into a corresponding many-objective problem with $m = m' + k$ objectives, by mapping both (objectives and constraints) into desirability functions.

To test the objective aggregation method using the desirability index, [64] implemented an adapted version of the EA presented in [63] that use the NSGA-II [24] selection method and a single objective genetic algorithm to solve an automated drug design problem where the aim is to find estrogen receptor antagonists. The considered problem has three objectives and six constraints that were mapped to desirability functions to obtain a many-objective problem with nine objectives. Then, two categories were defined: desirability functions corresponding to the original objectives and desirability functions corresponding to constraints. These categories served to define three aggregation alternatives: (i) aggregation of desirability functions of all objectives and aggregation of all constraints modelled by desirability functions; (ii) aggregation of desirability functions corresponding to constraints but not those corresponding to objectives; and (iii) aggregation of all desirability functions in one desirability index. The results obtained by executions of these three aggregation alternatives were compared using a visual approach and a Monte-Carlo approximation of the Hypervolume metric. Kruisselbrink et al. [64] concluded that the use of desirability index is a good alternative to reduce the number of objectives in many-objective problems. The downside of using a desirability index is that it yields a much smaller spread of the solutions with respect to the Pareto front than would be obtained by applying Pareto optimization [64].

5.1.2 Objective aggregation using correlation between objectives

Murata and Taki [76] presented a weighted-sum approach to reduce a many-objective problem with m objectives into a corresponding m'-objective one that can be solved using a MOEA. Different from [64], in [76] the objectives to be combined are not specified by the decision maker but determined by the objective function correlations.

The method proposed in [76] receives as input the number of desired objectives m' in which the original many-objective problem will be combined, and the number of generations when such combination of objectives will be held. Then, after the specified number of generations occurs, a set of solutions are sampled for recalculating correlations between objectives. These correlations are used to form m' groups of objectives, trying to maximize the average correlation of the objectives in each group. Once the objectives groups are obtained, each group is used to calculate, for each solution, a fitness value corresponding to the average sum of objectives in the corresponding group.

This way, the many-objective problem is transformed into m' objective function that will be used by a MOEA for searching for solutions.

For experimental comparison, in [76] objective aggregation using correlation between objectives was included into the NSGA-II [24] framework. Results of the original and modified versions of the NSGA-II were compared on knapsack instances [108] with 10 and 40 objectives using as performance measures the normalized sum and range of objectives [49]. The obtained results showed that the NSGA-II with the proposed aggregation approach can find better non-dominated solutions than the original NSGA-II version.

5.2 Scalarization-based methods: decomposition approaches

5.2.1 Multi-objective evolutionary algorithm based on decomposition (MOEA/D)

Zhang and Li [104] proposed the Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D) which, similarly to an approach previously presented in [77], is a decomposition based method that uses a scalarization function with a different weight vector for each individual. This way, the original problem is decomposed into a collection of scalar optimization problems, i.e. one problem for each individual in the evolutionary population.

MOEA/D requires the user to define a set of uniformly distributed weight vectors W^j, satisfying certain conditions [104]. For each W^j, the Euclidean distance is used to determine a neighbourhood $B(W^j)$ of T weight vectors, where T is also provided by the user. Since each weight vector W^j is associated with a single individual $\mathbf{x}^j$ of the population, $B(W^j)$ define a corresponding neighbourhood of solutions. In MOEA/D, selection and crossover are executed considering each neighbourhood. Each solution is evaluated using a scalarization function using the weight vector of the corresponding solution. Zhang and Li [104], evaluated three scalarization alternatives for MOEA/D: weighted sum, weighted Tchebycheff and boundary intersection approach. Detailed explanations of scalarization function alternatives, as well as the MOEA/D, can not be included here for space constraint reasons, and we refer the reader to [50, 104].

In Ishibuchi et al. [50], MOEA/D with weighted sum [72] was favourably evaluated on many-objective problems with convex Pareto fronts; however, MOEA/D using the weighted Tchebycheff method [72] takes advantage in non-convex Pareto Fronts; thus, since the choice of the appropriate scalarization function to use in MOEA/D depends on the shape of the Pareto front, Ishibuchi et al. [50] proposed to automatically alternate between weighted sum and weighted Tchebycheff.

The method to alternate between scalarization methods assumes that, by using a weighted sum, several individuals in a given neighbourhood have the same vector objective value searching in non-convex areas. Therefore, the procedure considers that if there are at least J solutions with identical vector objective value in a given neighbourhood, the weighted Tchebycheff method must be used, otherwise, the weighted sum is recommended. The value of J is defined by the decision maker.

In [50], Hypervolume metric was used to compare the performance of three scalarization methods for MOEA/D: weighted Tchebycheff, weighted sum and the method proposed in [50] in the knapsack problem [108] and its modified non-convex version for 2, 4, and 6 objectives with 500 items. The experiments in [50] showed that the weighted sum approach worked well for many-objectives, the Tchebycheff method for non-convex Pareto Fronts, while the method proposed in [50] provides intermediate results.

Before the method proposed in [104], Ishibuchi and Murata [45] proposed a decomposition method called Multi-objective Genetic Local Search (MOGLS), which was further improved by Jaszkiewicz [54]. Basically, MOGLS generates at each iteration a random weight vector to evaluate the current population and uses an external population to store non-dominated solutions. In [104], MOEA/D was shown to produce better approximations than MOGLS on a knapsack problem with four objectives. However, in [47] MOGLS has shown that it may work well on many-objective problems. Performance comparison between MOGLS and MOEA/D is indicated in [104] as an interesting research topic.

5.2.2 Multiple single objective Pareto sampling

Hughes [41] used multiple single objective Pareto sampling (MSOPS) [40] for solving many-objective optimization problems. In the MSOPS, a set of T weights vectors $\{W^1, \ldots, W^T\}$ is used to evaluate each solution using a weighted min-max method. Thus, each solution $\mathbf{x}$ has T scores, one for each target weight vector. In [41], weight vectors, also called target vectors, must be specified a-priori by the user.

The weighted min-max score solution $\mathbf{x}$ and target vector W^j, denoted as $S^j(\mathbf{x})$ is calculated using the following equation:

$$S^j(\mathbf{x}) = \max_{i=1}^{m} \; W_i^j \cdot f_i(\mathbf{x}) \tag{29}$$

where W_i^j is the ith component of the weight vector W^j.

The scores of all solutions are stored in rows of a matrix $\mathbf{S}$ of size $|P| \times T$ ($|P|$: population size); thus, column j of $\mathbf{S}$, denoted S^j, corresponds to target vector score W^j. Each column S^j is sorted in ascending order to produce a ranking of solutions R^j for the corresponding target vector j; thus, a rank of 1 is for the best solution, while the worst solution has a rank equal to the size of the population being classified. Rank values are stored in a matrix $\mathbf{R}$, which serves to obtain the final ranking of the population, where the best solution is considered to be the one having most scores in rank 1, and so on.

Hughes [41] compared the Hypervolume [108] performance of the MSOPS and the NSGA-II [24] on two many-objective problems defined in [99] with four and six objectives. The results in [41] showed that MSOPS outperformed NSGA-II for the considered metric and problems.

Later, in [43], Hughes presented MSOPS-II within two extensions to the original MSOPS. The first extension was a method that uses the current population as input to generate a new set of target vectors while the optimization run proceeds. The second extension was the reduction of the time complexity of MSOPS fitness assignment method by considering as the fitness of a population member the best score obtained after scaling the results of each target vector.

In [43], MSOPS-II was compared to the original MSOPS and a random search procedure using a 2-objective problem defined in [96], and a new problem defined in [43] with 5 objectives. To evaluate the obtained results, Hughes [43] used the Hypervolume [108] and the Multi-Objective Equivalent Random Search (MOERS) [42] metrics. The experimental results showed that the MSOPS-II ranking method may be effectively applied as a general purpose algorithm in many-objective optimization requiring a minimal initial configuration.

5.3 MOEAs based on dimensional reduction techniques

5.3.1 Principal component analysis based reduction methods

There are problems that have a low-dimensional Pareto front although many objectives are in the original problem. It means that, for two randomly picked solutions, a conflict between two given objectives may exist, however the same objectives may not conflict with another near the Pareto front. Deb and Saxena [23] combined the NSGA-II [24] with an objective reduction technique based on principal component analysis (PCA) to deal with those many-objective problems that have a low-dimensional Pareto front. In [23], NSGA-II executes a given number of iterations obtaining an approximate Pareto front considering the initial set of objectives $\mathcal{F}'_0 = \mathcal{F}$, then, the intermediate outcome is passed to the PCA to get a new set of objectives $\mathcal{F}'_1 \subseteq \mathcal{F}'_0$ to be used in the next NSGA-II iterations.

The PCA-based reduction method stores the objective values of the population in a matrix $\mathbf{D}$ of size $m \times |P|$ where each row r_i represents the values of the objective function f_i, while each column represents the objective values of a solution. The values of $\mathbf{D}$ are used to obtain the standardized $\mathbf{X}$ matrix, i.e. a data set having zero as its centroid. Standardized $\mathbf{X}$ matrix can be obtained by subtracting the mean from each objective value. Then, matrix $\mathbf{X}$ is used to calculate the covariance matrix $\mathbf{V}$ and the correlation matrix $\mathbf{R}$.

Latter, eigenvectors and eigenvalues of $\mathbf{R}\mathbf{R}^T$ are calculated. Eigenvectors are considered as the principal components (PCs). The eigenvector with the largest eigenvalue corresponds to the first PC, the second largest eigenvalue is the second PC, and so on. The first component of a PC (a given eigenvector) denotes the contribution of the first objective towards this vector, the second denotes the contribution of the second objective, and so on. The most negative and most positive elements of a given PC are its two most important conflicting objectives. Thus, [23] uses a predefined threshold cut (TC) to examine PCs in an increasing order until the cumulative contribution of analysed PCs exceeds TC.

To make the PCA-based method viable, [23] proposed an additional procedure that selects, for the first PC, those objectives contributing the most positive and most negative. Then, the other PCs are considered in order. Details of this method can be found in [23].

To validate the PCA-based method, Deb and Saxena [23] experimented on a modified version of the DTLZ5 problem [25] that let them specify the desired dimensionality of the Pareto front. Experimental results showed that the PCA-NSGA-II can be effectively used to solve redundant and large-dimensional problems. Also, the method PCA-NSGA-II uses a fraction of the computation time that the standard NSGA-II needs to make a good estimation of the Pareto front for some tested problems. However, the method has shown difficulties to find the correct combination of objectives in problems having a large dimensional Pareto front.

On the basis of [23,90] proposed two new non-linear dimensionality reduction algorithms for evolutionary multi-objective optimization. One of these methods is based on the correntropy PCA concept [103], while the other implements maximum variance unfolding principle [102]. In [90] an experimental comparison showed that the new proposed approaches outperform the basic PCA-based procedure both in terms of accuracy and computation time when solving up to 50-objective DTLZ2 and DTLZ5 problems.

5.3.2 Greedy reduction

Brockhoff and Zitzler [12] presented a new notion of conflict among objectives based on the dominance structure, according to which two set of objectives $\mathcal{F}_1$ and $\mathcal{F}_2 \subseteq \mathcal{F}$ are considered as conflicting if $\preceq_{\mathcal{F}_1} \neq \preceq_{\mathcal{F}_2}$. A reduction in the number of objectives may change the dominance relation, in this sense there is an error. In order to measure such an error, [12] defined the δ-conflict relation between objectives, as follows:

Definition 18 *ϵ-dominance relation*: let a set of objectives $\mathcal{F}' \subseteq \mathcal{F}$, the ϵ-dominance relation over $\mathcal{F}'$ is defined as:

$$\preceq^{\epsilon}_{\mathcal{F}'} = \{(\mathbf{x}, \mathbf{x}') | \mathbf{x}, \mathbf{x}' \in \mathcal{X} \wedge \forall f_i \in \mathcal{F}',\ f_i(\mathbf{x}) - \epsilon \leq f_i(\mathbf{x}')\} \tag{30}$$

Definition 19 Let $\mathcal{F}_1, \mathcal{F}_2 \subseteq \mathcal{F}$ be two objective sets, it is said that:

- $\mathcal{F}_1$ is δ-nonconflicting with $\mathcal{F}_2$ iff $(\preceq_{\mathcal{F}_1} \subseteq \preceq^{\delta}_{\mathcal{F}_2}) \wedge (\preceq_{\mathcal{F}_2} \subseteq \preceq^{\delta}_{\mathcal{F}_1})$
- $\mathcal{F}_1$ is δ-conflicting with $\mathcal{F}_2$ iff ($\mathcal{F}_1$ is not δ-nonconflicting with $\mathcal{F}_2$)

A δ-minimal objective set was defined as a subset of $\mathcal{F}$ that can not be further reduced without changing the associated dominance structure with an error of at most δ. A set $\mathcal{F}$ of objectives is called δ-redundant if and only if there exists a set $\mathcal{F}' \subseteq \mathcal{F}$ that is δ-minimal with regard to $\mathcal{F}$.

Then, [12] proposed an exact and a greedy algorithm to solve the problem of finding the minimum objective subset corresponding to a given error (δ-MOSS) as well as the problem of finding an objective subset of size k with the minimum possible error (k-EMOSS). As both of these problems are $\mathcal{NP}$-hard, the exact method is only applicable to small size problems. The greedy algorithm is based on iteratively constructing an objective subset $\mathcal{F}'$ that is δ-nonconflicting with $\mathcal{F}$.

The objective reduction approach proposed in [12] was compared with results of IBEA on 9 instances of a knapsack problem [108] and instances of DTLZ2, DTLZ5, and DTLZ7 [27]. An analysis of experimental results showed that, in almost all considered problems, a reduction in the number of objectives is possible without changing the dominance structure. Moreover, the exact algorithm yields smaller objective subsets than the greedy algorithm, while the running times are considerably smaller for the greedy algorithm. Finally, it was experimentally shown that the PCA-based method [23] produces objective sets having larger errors in the dominance structure than those computed by the greedy algorithm proposed in [12].

5.3.3 Objective set reduction based on unsupervised feature selection

In [70], the reduction technique originally developed in [74] was integrated in a MOEA. The basic idea of the reduction method is similar to the one of [23], i.e. non-dominated solutions obtained by a MOEA are iteratively used to estimate a correlation matrix indicating conflicts between each pair of objectives and determining the most conflicting objectives of a problem in order to reduce the set of objectives. Based on the aforementioned idea, [70] presented an algorithm that finds the minimum subset of non-redundant objectives with the minimum possible error and another algorithm that obtain the minimum set of k non-redundant objectives that yields to the minimum error. These two algorithms only differ in the stopping criteria and input parameters, i.e. in one case the desired number of objectives is specified while in the other case it is calculated.

After computing the correlation matrix of a set of non-dominated solutions, the reduction technique operates following three basic steps [70]:

1. Use the conflict between objectives as distance in order to divide the objective set into homogeneous neighborhoods of size q around each objective.
2. Select the most compact neighborhood. That is, the neighborhood with the minimum distance to its qth neighbor.
3. Retain the center of that neighborhood and discard its q neighbors (the objectives with least conflict in the current set). In this process, the distance to the qth neighbor can be thought of as the error committed by removing the q neighbouring objectives.

According to the specific case, the procedure iterates steps 2 and 3 while the number of desired objectives does not reach the specified k value or while there are not more considered neighborhoods.

Objective reduction techniques proposed in [70] were incorporated into the NSGA-II [24], and compared against the PCA-based reduction method [23] and the reduction method of [15]. Three different problems were considered in the experiments: a variation of the DTLZ5 with 3, 5, and 10 objectives [23], a variation of the DTLZ2 [15] and the knapsack problem [108] with 10 and 20 objectives. Metrics used to evaluate the outcomes were the IGD [98] and δ-error [12]. The experiment results showed that the proposed methods are competitive compared to the algorithms described in [15,23].

In order to eliminate the need of the additional parameters that must be provided for the algorithm presented in [70], a modified version of the reduction technique is proposed in [53].

5.4 Indicator-based MOEAs

Indicator-based MOEAs use a quality indicator to assign fitness to solutions; thus, these algorithms transform the original many-objective problem into a single objective one, i.e. the problem of optimizing the given indicator. An unary indicator takes one set of non-dominated solutions and returns a real number related with a given performance criteria; whereas, binary quality indicators serve to compare the relative quality of two sets of non-dominated solutions.

In [106], Zitzler and Künzli proposed the Indicator-based Evolutionary Algorithm (IBEA) as a general framework for incorporating indicator-based search into evolutionary algorithms. In that framework, solutions are compared in pairs using an arbitrary binary indicator; however, it requires that these indicators be dominance preserving, i.e. a solution will not be evaluated as better than other that dominates it [106].

In IBEA, mating selection is performed by using binary tournaments; also, an environmental selection procedure iteratively removes the worst individual from the population and updates the fitness values of the remaining individuals. In [106], authors proposed two IBEA variants based on the following indicators: the additive ϵ-indicator ($I_{\epsilon+}$) [111], and the I_{HD} indicator, which is based on the Hypervolume concept [111]. These indicators can be computed in linear time regarding the dimension of the objective space when comparing two solutions at a time. Both variants of the IBEA were compared against other methods in several many-objective test problems [6,37,89,88,100]. From these works, it is worth noting results in [100], were both IBEA variants converged to the Pareto Front of the DTLZ2 problem with up to 6 objectives [27]; while only the IBEA with $I_{\epsilon+}$ was able to converge in the case of the DTLZ1 problem using the same number of objectives. However, both failed to produce a suitable distribution of solutions on DTLZ1. Sato et al. [88], when using IBEA for the knapsack problem also found that they achieve a high convergence, but low diversity diversity.

Emmerich et al. [32] proposed the $\mathcal{S}$-metric Selection-EMOA (SMS-EMOA), a steady-state EA that aims to maximize the Hypervolume, i.e. the $\mathcal{S}$-metric [111]. In this case, only one new individual is generated at each iteration by means of random variation operators. In the SMS-EMOA, the non-dominated sorting procedure serves as the primary selection criterion, whereas Hypervolume is the secondary criterion applied to the last front. The new generated individual is incorporated into the population if replacing a member increases the Hypervolume covered by the population. SMS-EMOA has shown good results for two and three objective problems [32,78]; however, since algorithms that exactly calculate the Hypervolume for a solution set have exponential running times in the number of objectives, the use of the Hypervolume metric becomes a severe drawback for scaling the SMS-EMOA to many-objective problems.

Basseur and Zitzler [8] proposed several techniques to integrate uncertainties within the IBEA with the $I_{\epsilon+}$ indicator. In that work, cases where a stochastic process determines the objective function values are considered; thus, each solution is associated with a probability distribution over the objective space. For these cases, [8] proposed a method that computes the exact expected value of $I_{\epsilon+}$; however, to apply the idea on the IBEA framework they also propose several methods to approximate such expected value. These methods were empirically investigated on various problems, providing satisfactory results with an increasing number of objectives.

The computational load of the Hypervolume calculation, in general, prevents to use this indicator in many-objective problems. Thus, Brockhoff and Zitzler [13] proposed the Simple Indicator Based Evolutionary Algorithm (SIBEA) as a framework to combine objective reduction strategies and Hypervolume-based search. Using this framework, [13] studied the objective reduction techniques δ-MOSS and k-EMOSS proposed in [12] in problems up to nine objectives. Experimental analysis in [13] showed that, considering the same computation time, SIBEA without reduction techniques can be improved regarding its convergence by combining it with k-EMOSS. This result is explained due to savings in the computation time of the Hypervolume computation caused by the objective reduction which allows evaluating more solutions within the same time. The possible improvement that could be obtained by objective reduction is limited due to the sensitivity of the Hypervolume computation with respect to the number of objectives, also, to be viable, problems must be reduced to less than 6 objectives. Also, [13] showed that SIBEA obtain better Hypervolume values that NSGA-II [24] and SPEA2 [110].

In order to deal with the computational effort required for Hypervolume calculation, instead of reducing the number of objectives, Bader and Zitzler [6] proposed the use of a Monte Carlo algorithm to approximate the exact Hypervolume values. Based on this idea, [6] presented the Hypervolume Estimation Algorithm for Multi-objective Optimization (Hype). An important characteristic of the method in [6] is that the accuracy of the approximation and available computing resources can be trade-off. Hype is similar to other EAs but uses the concept of environmental selection to create a new population from the best solutions in the union set of the parent and offspring populations and estimate the Hypervolume value by sampling solutions in different fronts. In [6], two variants of Hype were compared against several state-of-the-art MOEAs on 17 test problems with up to 50 objectives. The quality of the outcomes of the different compared methods were assessed using exact Hypervolume calculation in problems with less than 6 objective, otherwise Hypervolume values were approximated by Monte Carlo sampling [7]. For the experimental settings used in [6], Hype showed to be very competitive, reaching the best Hypervolume value for most of the cases.

5.5 MOEAs based on space partitioning

In [5], Aguirre and Tanaka presented the ϵR-EMO (ϵ Ranking-Evolutionary Multi-objective Optimizer). In this algorithm, the main idea is to alternate between one iteration considering the whole set of objectives, and a set of iterations taking into account a different subspace each time.

The ϵR-EMO was developed over the basis of the NSGA-II [24]. Thus, it starts by generating an initial empty population P and a random population Q. Then, an iterative procedure takes into account all objectives of the many-objective problem to classify solutions in $(P \cup Q)$ using the fast non-dominated sorting procedure. Classified solutions are stored in an additional memory $\mathcal{M}$, and crowding distance is assigned to elements in $\mathcal{M}$. Thereafter, the objective space is partitioned into a set of

m_s non-overlapping subspaces with the same dimension. Next, the original P and Q populations are emptied. The ϵR-EMO continues by evolving $\mathcal{M}$ one generation for each subspace s by using the ϵ-ranking procedure [4]. A truncated set of solutions in $\mathcal{M}$ is used to obtain a new parent population P_s that is used to generate Q_s. At the end of each subspace iteration, solutions in P_s and Q_s are joined to solutions in P and Q, respectively. The procedure continues until a stop criterion is reached. This way, search uses different combinations of reduced sets of objectives at each generation.

Previous to its execution, the ϵR-EMO requires to set up how the subspaces will be sampled. In [5], three strategies were analysed: random, shift and fixed. Also, the ϵR-EMO requires the decision maker to set up the number of solutions to be considered at each partition, as well as the number of generations before create a new partition. In [53], the original ϵR-EMO was modified in order to automatically determine those required parameters. In this case, the number of solutions in each partition is defined according to the contribution of each subspace to the total conflict of the problem, whereas a convergence detection method is used to detect were a new partition is needed.

In [5] the ϵR-EMO method was tested for the MNK-landscapes [3] considering 4 to 10 objectives. As comparison metrics, the work uses the Hypervolume and Coverage measures [108]. According to the experimental results, the convergence and diversity of the solutions found are improved with the ϵR-EMO in all cases. Sato et al. [88] compared a method similar to the one presented in [5] with a random sampling of objectives, which is called in that work the Partial Pareto Dominance MOEA (PPD-MOEA), against the IBEA [106], an expansion relation method [87], the NSGA-II [24], and the MSOPS [41] for the 0/1 knapsack problem with 4, 6, 8, and 10 objectives. Also, a variation of the PPD-MOEA using the expansion relation instead the Pareto dominance was considered. In [88], the PPD-MOEA achieved higher diversity than IBEA, MSOPS and the expansion relation method, as well as better convergence than the original NSGA-II.

5.6 Summary of MOEAs for many-objective problems based on transformations of the original problem

As it was previously pointed out, difficulties in solving many-objective optimization problems using MOEAs include: (i) the growing proportion of non-dominated solutions, (ii) the computational cost, and (iii) visualization of results. Methods based on preference relations focus on the first of these difficulties by providing improved ranking schemes (see Sect. 4); on the other hand, methods based on transformations of the original problem also considers, with a varying extent, the other two issues.

Aggregation approaches are among the simplest techniques that could be used to transform a many-objective problem into a multi-objective or single objective optimization problem. In general, these methods require less computational load than other alternatives. Furthermore, considering combinations of objectives, aggregation based approaches facilitate visualization and decision making. However, the main disadvantage of aggregation based MOEAs is that their search ability depends on the choice of the scalarizing function as well as on the selection of objectives to be combined.

Both of these issues relate with the characteristics of the problem being optimized, therefore, represent a severe drawback for the robustness of these techniques. In fact, for the method based on desirability functions [64], the decision maker has to define a priori the objectives to be combined. Thus, two significant aspects in the application of aggregation based MOEAs for further research are: how to identify the objectives to combine, and which scalarization function to use.

The decomposition approaches presented in this work vary in the number of scalarization functions used to determine the fitness of a given individual. The MOEA/D [104] uses a different scalar function for each element in the evolutionary population, whereas, the MSOPS [40] considers a set of scalar functions for each population element. Similarly to aggregation based approaches, the performance of decomposition methods for a given problem depends on the choice of the scalarizing functions. Also, the effectiveness of decomposition methods depends on the process used to generate the weight vectors and target values. As pointed out in [50], during the execution of a MOEA based on decomposition, different scalarization functions may be necessary. Thus, in order to improve the robustness of these approaches, there is a need of further research on the adaptive determination of the most efficient function for different problems, as well as, the settings of weights and target vectors. Besides, in case of MSOPS, the number of target vectors needed to approximate the Pareto front for a given problem depends on the number of objectives and population size; thus, determining the adequate number of target vectors is also an important research point.

A significant disadvantage of indicator-based methods is the large execution time of the quality indicator calculation, especially for Hypervolume based methods. To alleviate this impediment, the following methods were proposed: (i) combining indicator-based search with objective reduction techniques [13], and (ii) the use of approximation indicator values instead of an exact calculation [6]. Analysing other alternatives to improve computational cost of these algorithms is clearly an interesting research topic to be explored.

Dimensional reduction techniques can be combined with other alternatives in order to alleviate difficulties arising in many-objective optimization such as computational cost and visualization. In fact, the main advantage of this approach is that by reducing a many-objective optimization problem into a multi-objective one, it can be solved with any existing MOEA. However, in problems with minimal set of objectives the method is not viable. There are several objective reduction techniques that can be used, thus determining which of them fits the best for a given class of problems is a promising research area. In the studied cases, elimination of objectives depends on the current solutions in the population. Evolutionary population in most MOEAs evolves from a random sampling of the search space through an approximation to the Pareto set, thus, the moment in which the objective reduction executes a different set of objectives may be selected. Therefore, determining the criteria to execute the objective reduction method becomes an issue.

Finally, despite the promissory results presented in [5,53,88], there are few works studying space partitioning approaches. These methods appear to be very attractive for being applied in combination with preference relations as in [88]. Also, they could be implemented along with indicator based methods providing a simple way to reduce the execution load in these methods by sampling objectives.

6 Experimental studies on evolutionary many-objective optimization

Experimental approaches in evolutionary many-objective optimization were typically applied to: (i) evaluate the performance of existing MOEAs, and (ii) validate new algorithm proposals. Besides the question to be answered or the hypothesis to be tested, there are three key experimental design elements that can be identified: the set of problems to be considered, the chosen metrics, and the selected algorithms.

The key characteristics of the experiments conducted to validate the reviewed methods were already presented at the time each method to solve many-objective problems was analysed. Therefore, in what follows, Subsect. 6.1 describes some works evaluating the performance of existing MOEAs, while Subsect. 6.2 summarises several experimental woks reported throughout this paper enriched with additional references and comments.

6.1 Performance evaluation of existing MOEAs in many-objective optimization

In [25], Deb et al. presented the DTLZ problem suite, the first set of functions specifically designed to study scalability issues of MOEAs. DTLZ is composed of seven test problems, named as DTLZ1 to DTLZ7, which are scalable in both parameter and objective dimension. Each of these problems represents different characteristics (spherical, linear, discontinuous, and degenerate) of the Pareto-optimal front. It is worth noting here that, according to [39], despite DTLZ5 and DTLZ6 are both claimed to be problems with degenerate Pareto optimal fronts, this is untrue for instances with four or more objectives.

To the best of our knowledge, Khare et al. [57] were the first to use the DTLZ suite [25] to compare the performance of three MOEAs in a many-objective context. The algorithms evaluated in [57] were: the NSGA-II [24], the SPEA2 (Strength Pareto Evolutionary Algorithm II) [110], and the PESA (Pareto Enveloped-based Selection Algorithm) [20]. Khare's work showed that, with many objectives, PESA is able to find a good approximation to the global Pareto set but with a poor distribution, whereas SPEA2 and NSGA-II have the opposite characteristics; however, no algorithm outperformed the others for all performance metrics considered in that work. Thus, [57] concluded that results of performance evaluation for two or three objectives can not be generalized to higher dimensions.

To have a better insight into the NSGA-II's [24] behaviour at a component level, Purshouse and Fleming [83,84] used proximity maps to represent its performance using several selection and variation operators for DTLZ2 with 3, 6, and 12 objectives, concluding that classical settings for recombination become increasingly inappropriate as the number of objectives increases. Also, confirming Khare et al. [57] results, in [83,84] it is shown that the NSGA-II performs very well with a small number of objectives but increasingly worse as the problem scales in the number of objectives.

To analyse the efficiency of MOEAs' search in discrete spaces as a function of the number of objectives, Knowles and Corne [58] derived an analytical expression for the expected coverage [111] of a random search procedure which they used to compare with the experimental results obtained by Pareto Envelope-based Selection

Algorithm II (PESA-II) [21]. The comparison was done across 10 instances of the NK-landscapes problem [55] with 2, 5, and 10 objectives using different inter-objective correlations. The main conclusion reached in [58] is that 10 objectives are enough to deteriorate the performance of PESA-II below the performance of random search; thus, if preference information is unavailable, random search becomes a valid alternative to an evolutionary algorithm (EA) for problems with more than 10 objectives. In spite of the methodological differences between the work of Teytaud [97] (described in Subsect. 3.1) and [58], in both cases it is demonstrated that as the number of objectives increases, the random search becomes a serious competitor to traditional Pareto-based MOEAs.

In [100] Wagner et al. compared eight MOEAs representing Pareto, aggregation, and indicator-based approaches using the DTLZ1 and DTLZ2 test functions with 3 to 6 objectives. The Pareto-based MOEAs selected for comparison were NSGA-II [24], SPEA2 [110], and ϵ-MOEA [67]; aggregation approaches were the MSOPS and the Repeated Single Objective (RSO) [41]; while indicator-based methods were two variants of the Indicator-based Evolutionary Algorithms (IBEA) [106] and the SMS-EMOA ($\mathcal{S}$-metric Selection-EMOA) [32]. Evaluation of obtained results was performed using the Hypervolume metric [107] and the convergence measure [26]. As conclusion, [100] showed that, for several MOEAs, the selection bias towards extreme solutions is an important factor that impede the search progression in many-objective spaces. Also, performance evaluation showed that some modern Pareto-based MOEA, as the ϵ-MOEA, may effectively obtain good metric values in problems having a number of objectives greater than four.

Ishibuchi et al. [48] analysed approaches developed to improve the convergence property of NSGA-II [24] for knapsack problems [108] with 2, 4, 6, and 8 objectives. The evaluated methods were: a modification of the NSGA-II crowding distance, alternative ranking methods [95], modification of Pareto dominance [46,87], as well as a hybridization with a local search method [54]. The outcomes of the different methods were evaluated using the following set of metrics defined in [49]: maximum sum of the objective values (MaxSum), the sum of the maximum objective values (SumMax), and the sum of the range of the objective values of each objective (Range). Experimental results in [48] showed that hybridization with local search significantly improves the diversity and slightly improves the convergence, while the other compared methods improve convergence; however, with a severe side-effect on the diversity of solutions. Thus, according to [48], local search may be an useful alternative to improve the performance of MOEAs for many-objective problems.

López Jaimes and Coello Coello [71] proposed a framework to compare preference relations to deal with many-objective problems considering the ability of such relations to converge towards the Pareto Front knee. Using that framework, [71] empirically studied preference relations in many-objective problems considering: the average and maximum ranking methods [9], the favour relation [31], the preference order relation [29], and the expansion relation [87]. These relations were implemented over a simple MOEA [19] to solve DTLZ2 to DTLZ7 problems [27]. Experimental results were evaluated using the generational distance (GD), the IGD [98], and the distribution of Tchebycheff distance. As a conclusion, [71] noted that some relations contribute to a quick convergence to the Pareto front, but they promote the generation of solutions far from the knee regions.

In [39], Huband et al. presented the Walking Fish Group (WFG) toolkit, which allow researchers to define scalable, multi-objective test problems with Pareto fronts of different geometries, parameter dependencies, modality, bias and fitness landscapes. Also, using the WFG toolkit [39] the authors proposed a set of nine test functions covering a wide rank of problem attributes. Using four of these test functions, Purshouse et al. [85] compared different ordering methods in a simple $(\mu + \lambda)$ elitist framework without recombination: an adaptation of a co-evolutionary approach based on goal vectors [69], the non-dominated sorting and crowding scheme from NSGA-II [24], a combination of average ranking [19], and a sharing scheme based on the Epanechnikov kernel [35]. In [85] it was found that the co-evolutionary approach has a better performance for the Hypervolume metric [108] when compared to the other tested methods.

The work of Ishibuchi et al. [51] compared the performance of the MOEA/D [104], the SPEA2 [110], and the NSGA-II [24] using the Hypervolume metric. The conclusion they found is that highly correlated objectives severely degrade the performance of the MOEA/D which is based on a scalarizing function, while they have almost no negative effects on the performance of the SPEA2 and the NSGA-II, which are Pareto-based.

Very recently, Hadka and Reed [37] presented a diagnostic framework to evaluate how MOEA operators, parameters and interactions among these factors influence the success or failure of MOEAs in many-objective optimization. In that work, authors considered 33 instances of 18 unconstrained real valued test problems to compare the NSGA-II [24], the SPEA2 [110], the ϵ-MOEA [67], the ϵ-NSGA-II [59], a version of the MOEA/D [105], the IBEA [106], the differential evolution method introduced in [65] (GDE3), the Multi-Objective Particle Swarm Optimizer proposed in [94] (OMOPSO), and the Borg MOEA [36]. The problems considered in [37] were: the ones used during the CEC 2009 competition [105] as well as the DTLZ 1-4 and 7, while the considered metrics were Hypervolume, GD [98] and the additive ϵ-indicator (I_{ϵ^+}) [111]. Also, authors propose a set of metrics for statistically sampled ensembles of approximation sets in order to measure the best achieved value, probability of attainment, efficiency, and controllability. The Sobol variance decomposition [86] was used to identify MOEA's key parameters and the multivariate relations among its control parameters. The experiments performed in [37] showed the necessity of diversity-maintaining archives when MOEAs are applied to many-objective problems and the usefulness of the proposed evaluation framework to take into account the impacts of different operators and parameters.

6.2 Summary of experimental works in many-objective evolutionary optimization

In order to provide a general overview of the experimental research in many-objective evolutionary optimization, Table 17 summarises the key characteristics of several reviewed works enriched with additional references. Using this information, Table 16 details the references corresponding to problems used in more than two works.

Table 16 References by problem considering the 36 works in Table 17

Problem	References	Total
MNK-landscapes [3]	[5,80]	2
Modified DTLZ5 [23]	[53,70,92]	3
WFG problems [39]	[6,85,92]	3
Modified DTLZ2 [15]	[23,70,52]	3
knapsack [108]	[6,49–53,70,76,87,89]	10
DTLZ problems [25]	[1,6,28,29,37,38,57,60, 66,71,84,92,100,112]	14
Detailed references for DTLZ problems		
DTLZ1	[6,28,29,37,57,66,92,100,112]	9
DTLZ2	[1,6,28,29,37,38,57,60, 66,71,84,92,100,112]	14
DTLZ3	[6,28,29,37,38,57,60,66,71,92]	10
DTLZ4	[6,28,37,66,71,92]	6
DTLZ5	[6,28,29,66,71]	5
DTLZ6	[6,28,57,60,66,71,112]	7
DTLZ7	[6,28,37,71]	4

As Table 16 indicates, 14 out 36 works consider a DTLZ problem. Also, detailing the number of papers using DTLZ problems, it could be noted that the DTLZ2 is the most studied problem. Moreover, several experiments [12,23,37,53,70,52] used modifications of two DTLZ functions (DTLZ2 and DTLZ5).

Other test suites used for scalability analysis are the WFG toolkit [39], the CEC 2009 competition problems [105], and a set of problems based on a framework for continuous many-objective test problems with arbitrarily prescribed Pareto set shapes described in [91]; the last two in only one work. Also, some combinatorial test problems like the knapsack problem [108], the multi-objective travelling salesperson problem (MTSP) and the multi-objective single-machine job-shop problem (MJSP) were considered. The MTSP and MJSP where only used in [58], whereas, as Table 16 indicates there are 10 references of Table 17 using the knapsack problem. Besides, some real life many-objective optimization problems have been used in the referenced literature [31,64,95].

Figure 6 counts the maximum number of objectives that were considered by the works reported in Table 17. The figure shows that in 21 out of 36 works the research has been done with 10 objectives or less, and only in 3 reported papers the number of objectives is greater than 25. The largest number of considered objectives was 50 in [6,66], thus, there is a lack of study cases with instances in the range of hundreds of objective functions.

Table 17 takes into account more than two dozens of different performance metrics used to evaluate the performance of MOEAs in many-objective optimization. As can be noted, most of these metrics are the same than those used in multi-objective problems with 2 or 3 objectives. Besides, Ishibuchi et al. [49] proposed three metrics to measure convergence and diversity of solution sets in a many-objective context: maximum sum of the objective values (MaxSum), the sum of the maximum objective values (SumMax), and the sum of the range of the objective values of each objective (Range).

Table 17 Summary of some experimental works related to MOEAs

References	Test prob.	# Objs.	Analysed methods	Metrics
[31]	Heuristic learning for VLSI CAD [30]	6	Algorithm introduced in [30] with dominance and favour relations, EA with weighted sum, exact method, greedy method	Comparison with exact method, execution time
[57]	DTLZ(1-3, 6) [25]	2, 3, 4, 6, 8	NSGA-II [24], PESA [20], SPEA2 [110]	Distance to the global Pareto front, diversity metric 1 and 2 (DM1 and DM2) [57]
[23]	Modified DTLZ5 [23]	10	NSGA-II [24], proposed PCA-based NSGA-II [23]	Convergence [26]
[62]	Pareto-box [62]	20	Random, FDD-GA [62], NSGA-II [24]	Convergence to the problem single optimal point [62]
[41]	Problem defined in [41]	2, 4, 6	NSGA-II [24], MSOPS [40], RSO [41], Weighted min–max	Hypervolume [108]
[28]	DTLZ(1-7) [25]	4, 5, 6, 7, 8	NSGA-II [24], NSGA-II with preference ordering [28]	GD [98], Hypervolume [108], DM1 and DM2 [57] and Coverage [109,111]
[87]	Knapsack [108]	2, 3, 4, 5	NSGA-II [24], NSGA-II with expansion relation [87]	Spread [22], Hypervolume [108], IGD [98]
[95]	Nurse rostering	25	NSGA-II [24], Dominates [56], the method presented in [31] using favour and ϵ-preferred relations	Average of best weighted sum values obtained in different runs
[60]	Pareto-box [62], DTLZ(2,3,6)[25]	2, 8, 15	NSGA-II [24] with substitute secondary distances defined in [60]	Convergence [26,62], Coverage [109,111]
[58]	NK-landscapes [55]	2, 5, 10	PESA-II [21], Random	Coverage [109,111]
[100]	DTLZ(1, 2) [25]	3, 4, 5, 6	NSGA-II [24], SPEA2 [110], ϵ-MOEA [67], MSOPS [40], RSO [41], IBEA [106], SMS-EMOA [32]	Hypervolume [108], Convergence [26]
[66]	DTLZ(1-6) [25]	2–10, 15, 20, 25, 30, 35, 40, 45, 50	Modified (sorting based on ranking-dominance) and original Generalized Differential Evolution 3 (GDE3) [65]	Convergence [26]
[43]	Problem defined in [43]	5	MSOPS [40], MSOPS-II [43] and variations	Hypervolume [108], MOERS [42]
[84]	DTLZ2 [25]	3, 6, 12	NSGA-II [24]	Proximity indicator [83], Spread [22]

Table 17 continued

References	Test prob.	# Objs.	Analysed methods	Metrics
[29]	DTLZ(1-3, 5) [25]	4, 5, 6, 7, 8	NSGA-II [24], NSGA-II with preference ordering ranking	GD [98], Hypervolume [108], DM1 and DM2 [57], coverage [109, 111]
[19]	MTSP, MJSP	5, 10, 15, 20	Simple MOEA [19] with different ranking methods: the average and summed ratio ranking methods [9], the favour relation [31], the preference order relation [29], random selection, sum of objectives ranking	Relative entropy [19], coverage [108]
[112]	DTLZ(1-2, 6) [25]	3, 4, 5, 6, 7, 8, 9	Dynamic MOEA [112], IBEA [106], MSOPS [40], MSOPS-II [43], NSGA-II [24]	Convergence [26], Hypervolume [108]
[70]	Modified DTLZ5 [23] and modified DTLZ2 [15], knapsack [108]	3, 5, 10, 20	Modifications of NSGA-II [24] with different objective reduction techniques [70, 23, 15]	IGD [98], δ-error [12]
[49]	Knapsack [108]	2, 4, 6, 8, 10, 12, 14, 16, 18, 20	NSGA-II [24]	Number of non-dominated solutions, MaxSum, SumMax, range [49]
[52]	Modified DTLZ2 [15], knapsack	4,6,8,10	NSGA-II [24], modified NSGA-II proposed in [70]	ϵ-indicator ($I_{\epsilon+}$) [111]
[5]	MNK-landscapes [3]	4–10	NSGA-II [24], modified NSGA-II with new partition of objective space method [5]	Hypervolume [108], Coverage [109, 111]
[50]	Knapsack [108], modified knapsack [50]	2, 4, 5	MOEA/D [104], modified MOEA/D	Hypervolume [108]
[1]	DTLZ2 [25]	6, 8, 12	NSGA-II [24], modified NSGA-II with new diversity management operator [1]	Dominated distance metric (DD-Metric) [109], coverage [109, 111]
[64]	Drug design problem	9	EA with two approaches for aggregation of objective using desirability index using NSGA-II [24] selection scheme	Hypervolume [108]
[76]	Knapsack [108]	10, 40	NSGA-II [24], modified NSGA-II with new objective aggregation using correlation between objectives [76]	Normalized sum and range of objectives [49]

Table 17 continued

References	Test prob.	# Objs.	Analysed methods	Metrics
[71]	DTLZ(2-7) [25]	3, 5, 10, 15	Basic MOEA [19] combined with: average and maximum ranking methods [9], the favour relation [31], the preference order relation [29], and the expansion relation [87]	GD, IGD [98] and the distribution of Tchebycheff distance
[51]	Knapsack [108]	2, 4, 6	MOEA/D [104], SPEA2 [110], NSGA-II [24]	Hypervolume [108]
[85]	WFG (2, 4, 6, 8), k = 12, l = 20 [39]	2, 7, 13	Random, a $(\mu + \lambda)$ co-evolutionary approach using three sorting methods	Dominance rank, Hypervolume [108]
[53]	Modified DTLZ5 [23], knapsack [108]	4-15	NSGA-II [24], modified NSGA-II with different partition strategies [53]	GD [98], ϵ-indicator $(I_{\epsilon}+)$ [111], Hypervolume [108]
[91]	Problems based on a new test generator framework	5	NSGA-II [24], MOEA/D [104]	IGD [98]
[80]	MNK-landscapes	2–10	NSGA-II [24], Multi-objective random one-bit climbers [2] with tabu moves and adaptive ϵ-ranking	SumMax [49], Hypervolume [108], Coverage [109,111]
[6]	DTLZ(1-7) [25], WFG [39], Knapsack [108]	2, 3, 5, 7, 10, 25, 50	Hype [6], SPEA2 [110], NSGA-II [24], IBEA [106], the method proposed in [7]	Hypervolume [108]
[92]	DTLZ(1-4) [25], modified DTLZ5 [23], WFG3 [39]	5,10,15,20,25	NSGA-II [24], MOEA/D [104]	Convergence [92], diversity [83]
[89]	Knapsack [108]	2, 4, 6, 8, 10	SPEA2 [110], NSGA-II [24], IBEA [106], MSOPS [41] and modifications with local recombination	Hypervolume [108]
[37]	CEC 2009 competition problems [105], DTLZ(1-4, 7) [25]	2–8	NSGA-II [24], SPEA2 [110], ϵ-MOEA [67], ϵ-NSGA-II [59], a version of the MOEA/D [105], IBEA [106], GDE3 [65], OMOPSO [94], the Borg MOEA [36]	Hypervolume [108], GD [98], ϵ-indicator $(I_{\epsilon}+)$ [111]
[38]	DTLZ(2, 3) [25]	3, 5, 10	Fuzzy-based NSGA-II, NSGA-II [24], SPEA2 [110]	GD, IGD, Spacing [98]

Fig. 6 Maximum number of considered objectives by reference

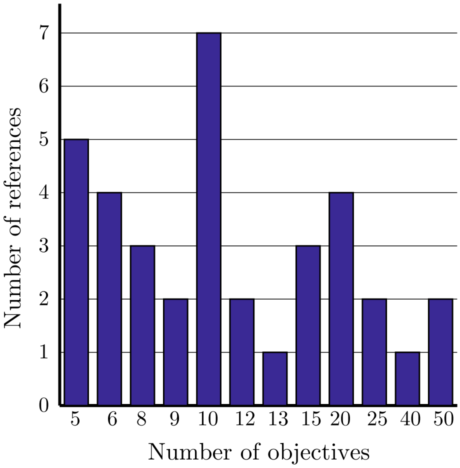

Fig. 7 Number of references by metric

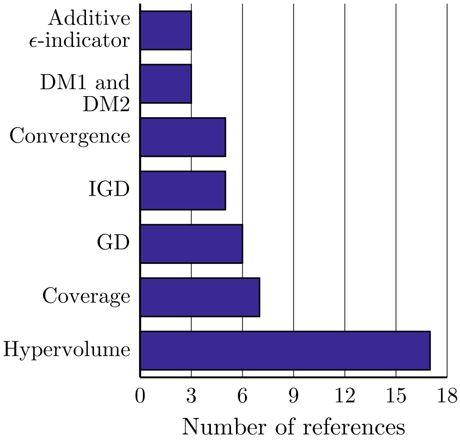

Figure 7 shows the number of references for metrics that were used in at least three works in Table 17. As the figure indicates, the Hypervolume [108] metric is the most popular metric and was used in 17 out of 36 articles summarized in Table 17. However, as explained in [6], Hypervolume calculation requires a high computational effort that exponentially grows with the number of objectives, thus for an online performance assessments having a large number of objectives and trials, the Hypervolume may not be applicable. Alternatively, Hypervolume approximations using Monte Carlo sampling have been proposed and tested in problems up to 50 objectives [6].

Several methods try to introduce some sort of preference information to guide the search; however, metrics used to evaluate algorithms do not consider these preference settings. In [71], it was pointed out that the generally accepted assumption is that the most interesting solutions are those in the knee region, thus it may be interesting to develop metrics that can evaluate how well these points are sampled. An important related question is how to effectively evaluate those methods using dimensional reduc-

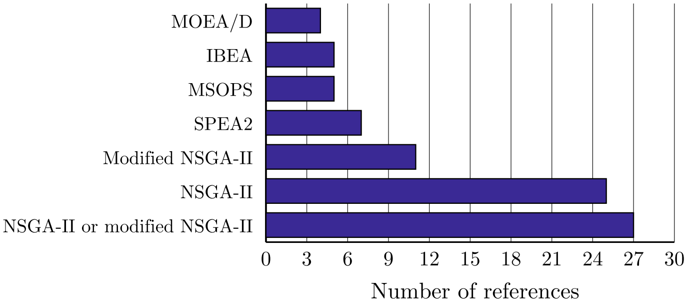

Fig. 8 Number of references by MOEA

tion techniques. In this case both, problems and metrics must be proposed in order to develop meaningful research works.

Regarding algorithms considered for the experiments, Fig. 8 shows the references for MOEAs that have been used in at least 4 works reported in Table 17. This figure indicates that 27 out of 36 papers in Table 17 use the original or a modified version of the NSGA-II [24], from these works, 25 use the original NSGA-II, while 11 consider a modified version of that algorithm. Other MOEAs that where used in at least 4 references consigned in Table 17 are: SPEA2 [110], MSOPS [40], IBEA [106], and MOEA/D [104]. As Coello pointed out [16]: there are few new evolutionary methods in the design of MOEAs, and almost in all cases new algorithms are presented as modifications of existing MOEAs, specially the NSGA-II; however, much remains to be done regarding algorithm design and new approaches for many-objective problems are expected to be developed.

To finalize this section, the main conclusions regarding the performance analysis of MOEAs for many-objective problems presented throughout this work can be summarized as follows:

- The proportion of local non-dominated solutions in a random sample grows as the number of objectives increases, becoming difficult for Pareto-based MOEAs to discrimitate among solutions [22] (see Sect. 3.1).
- Algorithm performance conclusions can not be generalized as it changes with the number of objectives, i.e. parameter settings that produce good results with a small number of objectives may become inappropriate as the number of objectives increases [57,83,84].
- For a given class of many-objective problems, the convergence rate of all Pareto-based multi-objective algorithms is not much better than a pure random search [97] (see Sect. 3.1).
- If preference information is unavailable, random search may be a good alternative to an EA for problems with more than 10 objectives [58].
- Adding objectives to a given optimization problem does not necessarily affects negatively the computation effort required to generate the set of Pareto optimal solutions [14].
- Some modern Pareto-based MOEA may effectively cope with problems having four to six objectives [100].

- For some MOEAs, addition of correlated objectives may severely degrade their performance, whereas for others it had almost no negative effect [51].
- For a given class of test problems, the addition of an objective makes them not significantly difficult [93] (see Sect. 3.1).

7 Conclusion

This article presented a comprehensive up-to-date survey of many-objective optimization using MOEAs. Several methods, as well as the main achievements of experimental and theoretical works, have been reported and summarized in order to help the interested reader in understanding the state-of-the-art in the field as well as to find new research opportunities. Throughout the paper, and according to the number of reviewed works, it was shown that many-objective optimization using MOEAs is an active research area, having multiple challenges that need to be addressed regarding scalability analysis (Sect. 3.1), visualization (Sect. 3.2), algorithm design (Sects. 4.3 and 5.6), and experimental algorithm evaluation (Sect. 6.2).

Scalability analysis shows that, besides the number of objectives, interactions among them must be considered to evaluate the hardness of a many-objective problem [14,93]. Despite great advances, additional research is needed to improve the understanding of the difficulties MOEAs face when the number of problem objectives increases. In particular, we consider that further works must develop theoretical frameworks for measuring the hardness of the problem, clarifying the links between the number of objectives and interactions among them. First studies on this direction are [82], proposing a model to study the relationships among objectives, and [12], developing a notion of conflict among objectives based on the dominance structure.

Some preference relations let the decision maker specify additional parameters to guide the search towards a given region of his/her interest. For these relations, further research is necessary to get an improved knowledge on the effects of different parameter settings. Such knowledge is the basis for further development of techniques that adaptively adjust the search as the optimization run proceeds, as well as, improved interactive frameworks for solving real world many-objective problems.

To produce a fair experimental comparison among the outcomes of algorithms based on preference relations, analysis must take into account that, in general, each of these relations represents a different search criterion exploring a different region of the Pareto Front; therefore, comparing among solution sets provided by algorithms based on different preference relations require additional considerations. An analysis on this issue presented in [71], however, further development is needed.

In general, two main properties are used to evaluate the quality of a Pareto Front approximation: convergence and diversity; and several criteria have been developed to evaluate these factors for a moderate number of objectives. Section 6.2 showed that some of these metrics were also used to evaluate solutions for many-objective problems provided by MOEAs. The Hypervolume is considered to be able to evaluate both quality criteria [22], and was the most used metric considering the experimental works summarized in Table 17; nevertheless, the computation cost limits its use to problems with few objectives and alternatives must be developed. Also, the literature review let us note the need of theoretical and empirical assessments of the available criteria as the number of objectives increases.

Considering the above discussions, from the authors stance, one of the most urgent need in the field is the proposal and development of new performance measures as fundamental tools for algorithm design and solution appraisal. Focusing only on global proximity and global diversity may cover only a small set of algorithms. Different approaches were designed to take into account alternative search criteria and must be evaluated accordingly; for instance, to evaluate a method based on the $(1-k)$-dominance relation (see Sect. 4.1.1) a metric measuring the ability of that method to obtain Pareto set approximations composed by solutions having a large number of better objectives must be considered. Also, in general, it makes no sense to compare an algorithm whose aim is to find an approximation of the whole Pareto Front to an algorithm that focuses on producing solutions close to a specified region of interest.

For problems having two or three objectives, visualizing the Pareto Front may be used as an additional tool for performance assessment and decision making. Particularly, visualization is useful to provide an idea of the two main concerns regarding diversity: extend and spread. However, as explained in Sect. 3.2, despite several visualization alternatives have been proposed, there is still a lack of intuitive and simple techniques for visualizing many-objective trade-offs. Thus, in order to improve the assessment of Pareto Front approximations in many-objective problems, new metrics ideally must provide information to alleviate difficulties caused by the weaknesses of the visualization support, such as the spread of solutions in particular regions or near to specified reference points.

Several MOEAs, such as the NSGA-II, are designed within a diversity preservation mechanism. Because obtaining a good diversity in many-objective problems with a large proportion of non-dominated solutions is not a difficult task, and several experimental works [48,71,87,88] show that there is a trade-off between convergence and the extension of the covered Pareto front, the role of these diversity preservation mechanism in many-objective EAs must be evaluated.

As it was discussed in Sect. 4.3, an advantage of methods based on preferences is that they are relatively easy to implement into the same underlying MOEA. Also, as it was previously stated, preference relations may sample different space regions; therefore, an alternative to obtain a good relation among convergence and diversity into a broader range of problems may be to explore combinations of preference relations.

In general, research works regarding alternative preference relations in many-objective optimization consider the proportion of non-dominated solutions as the main issue to solve many-objective problems. Thus, these methods try to improve the algorithm capacity to differentiate among solutions. As discussed in Sect. 3.1, scalability studies have found contradictory results that suggest that the proportion of non-dominated solutions is not enough to define the hardness of a problem. Some methods based on transformations of the original problem also consider other problem characteristics, mainly the number of objectives and their interrelations (see Sect. 5.6). For instance, interrelations among objectives are the basis to eliminate objectives in objective reduction techniques, whereas, in some aggregation based approaches interrelations determine the objectives to be aggregated.

Objective reduction techniques consider that, in some problems, the number of objectives can be reduced to have a multi-objective problem by eliminating a number of objectives without affecting in a great extent the quality of the obtained approximation. The applicability of such techniques is limited to problems having a moderate number of conflicting objectives. Additionally, objective reduction processes are based on the information that they can obtain from the population, thus, depending on how the populations are sampled to obtain the reduction. On the other hand, indicator-based methods, specially those based on indicators having an expensive computational cost, are difficult to scale to problems having a number of objectives greater than 10. Combination of different approaches may be an interesting option to overcome the difficulties that each approach individually has. However, there is a relatively small number of works regarding this issue such as [13], which combined dimensional reduction with indicator based methods, and [88], which used the expansion relation within a decomposition approach.

As it was pointed out, more than a half of the references use DTLZ problems. Also, there are few experimental works dealing with problems with more that 25 objectives. Defining the characteristics of many-objective problems that make them different from multi-objective problems, as well as new metrics and comparison frameworks specifically designed to measure the performance of MOEAs into many-objective problems become necessary and urgent. Furthermore, development of new test problem generators facilitating the specification of desired interrelations among objectives will become an important contribution in order to improve the evaluation of many-objective methods.

References

1. Adra, S.F., Fleming, P.J.: A diversity management operator for evolutionary many-objective optimisation. In: Proceedings of the 5th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2009). Lecture Notes in Computer Science, vol. 5467, pp. 81–94. Springer, Berlin (2009)
2. Aguirre, H., Tanaka, K.: Effects of elitism and population climbing on multiobjective MNK-landscapes. In: 2004 IEEE Congress on Evolutionary Computation (2004). doi:10.1109/CEC.2004.1330891
3. Aguirre, H., Tanaka, K.: Insights on properties of multiobjective MNK-landscapes. In: 2004 IEEE Congress on Evolutionary Computation (2004). doi:10.1109/CEC.2004.1330857
4. Aguirre, H., Tanaka, K.: Robust optimization by ϵ-ranking on high dimensional objective spaces. In: Simulated Evolution and Learning. Springer, New York (2008). doi:10.1007/978-3-540-70928-2_54
5. Aguirre, H., Tanaka, K.: Many-objective optimization by space partitioning and adaptive ϵ-ranking on MNK-landscapes. In: Proceedings of the 5th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2009). Lecture Notes in Computer Science, vol. 5467, pp. 407–422. Springer, Berlin (2009)
6. Bader, J., Zitzler, E.: Hype: an algorithm for fast hypervolume-based many-objective optimization. Evol. Comput. (2011). doi:10.1162/EVCO_a_00009
7. Bader, J., Deb, K., Zitzler, E.: Faster hypervolume-based search using Monte Carlo sampling. In: Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems, pp. 313–326 (2010)
8. Basseur, M., Zitzler, E.: A preliminary study on handling uncertainty in indicator-based multiobjective optimization. In: Applications of Evolutionary Computing. Lecture Notes in Computer Science, vol. 3907, pp. 727–739. Springer, New York (2006). doi:10.1007/11732242_71

9. Bentley, P.J., Wakefield, J.P.: Finding acceptable solutions in the Pareto-optimal range using multi-objective genetic algorithms. In: Soft Computing in Engineering Design and Manufacturing. Part 5, pp. 231–240. Springer, London (1997)
10. Bentley, J., Kung, H., Schkolnick, M., Thompson, C.: On the average number of maxima in a set of vectors and applications. J. ACM **25**(4), 536–543 (1978)
11. Branke, J., Kaußler, T., Schmeck, H.: Guidance in evolutionary multi-objective optimization. Adv. Eng. Softw. **32**, 499–507 (2001)
12. Brockhoff, D., Zitzler, E.: Are all objectives necessary? On dimensionality reduction in evolutionary multiobjective optimization. In: Parallel Problem Solving from Nature (PPSN IX). Lecture Notes in Computer Science, vol. 4193, pp. 533–542. Springer, Berlin (2006)
13. Brockhoff, D., Zitzler, E.: Improving hypervolume-based multiobjective evolutionary algorithms by using objective reduction methods. In: 2007 IEEE Congress on Evolutionary Computation (2007). doi:10.1109/CEC.2007.4424730
14. Brockhoff, D., Friedrich, T., Hebbinghaus, N., Klein, C., Neumann, F., Zitzler, E.: Do additional objectives make a problem harder? In: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO '07), pp. 765–772. ACM Press, New York (2007)
15. Brockhoff, D., Saxena, K., Deb, K., Zitzler, E.: On handling a large number of objectives a posteriori and during optimization. In: Knowles, J., Corne, D., Deb, K. (eds.) Multi-Objective Problem Solving from Nature: From Concepts to Applications, pp. 377–403. Springer, Berlin (2008)
16. Coello Coello, C.: Evolutionary multi-objective optimization: some current research trends and topics that remain to be explored. Frontiers Comput. Sci. China **3**(1), 18–30 (2009)
17. Coello Coello, C., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd edn. Springer, New York (2007)
18. Cormen, T., Leiserson, C., Rivest, R.: Introduction to Algorithms. The MIT Press, Cambridge (1990)
19. Corne, D.W., Knowles, J.D.: Techniques for highly multiobjective optimisation: some nondominated points are better than others. In: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO '07), pp. 773–780. ACM Press, New York (2007)
20. Corne, D.W., Knowles, J.D., Oates, M.J.: The Pareto envelope-based selection algorithm for multiobjective optimization. In: Parallel Problem Solving from Nature (PPSN VI). Lecture Notes in Computer Science, vol. 1917, pp. 839–848. Springer, Berlin (2000)
21. Corne, D.W., Jerram, N.R., Knowles, J.D., Oates, M.J.: PESA-II: Region-based selection in evolutionary multiobjective optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001), pp. 283–290. Morgan Kaufmann Publishers, San Francisco (2001)
22. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. Wiley, Chichester (2001)
23. Deb, K., Saxena, D.: On finding Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. Kangal report 2005011, Indian Institute of Technology (2005)
24. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. (2002). doi:10.1109/4235.996017
25. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: Proceedings of the 2002 Congress on Evolutionary Computation (2002). doi:10.1109/CEC.2002.1007032
26. Deb, K., Mohan, M., Mishra, S.: Towards a quick computation of well-spread Pareto-optimal solutions. In: Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2003). Lecture Notes in Computer Science, vol. 2632, pp. 222–236. Springer, Berlin (2003)
27. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. In: Jain, L., Wu, X., Abraham, A., Jain, L., Goldberg, R. (eds.) Evolutionary Multiobjective Optimization, Advanced Information and Knowledge Processing, pp. 105–145. Springer, Berlin (2005). doi:10.1007/1-84628-137-7_6
28. di Pierro, F.: Many-objective evolutionary algorithms and applications to water resources engineering. Ph.D. thesis, School of Engineering, Computer Science and Mathematics, UK (2006)
29. di Pierro, F., Khu, S.T., Savić, D.A.: An investigation on preference order ranking scheme for multiobjective evolutionary optimization. IEEE Trans. Evol. Comput. (2007). doi:10.1109/TEVC.2006.876362
30. Drechsler, N., Drechsler, R., Becker, B.: Multi-objected optimization in evolutionary algorithms using satisfyability classes. In: Reusch, B. (ed.) International Conference on Computational Intelligence,

Theory and Applications, 6th Fuzzy Days. Lecture Notes in Computer Science, vol. 1625, pp. 108–117. Springer, Dortmund (1999)

31. Drechsler, N., Drechsler, R., Becker, B.: Multi-objective optimisation based on relation favour. In: First International Conference on Evolutionary Multi-Criterion Optimization. Lecture Notes in Computer Science, vol. 1993, pp. 154–166. Springer, Berlin (2001)
32. Emmerich, M., Beume, N., Naujoks, B.: An EMO algorithm using the hypervolume measure as selection criterion. In: Proceedings of the 3rd International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005). Lecture Notes in Computer Science, vol. 3410, pp. 62–76. Springer, Berlin (2005). doi:10.1007/978-3-540-31880-4_5
33. Farina, M., Amato, P.: On the optimal solution definition for many-criteria optimization problems. In: Proceedings of the NAFIPS-FLINT International Conference 2002 (2002). doi:10.1109/NAFIPS.2002.1018061
34. Fleming, P., Purshouse, R.C., Lygoe, R.J.: Many-objective optimization: an engineering design perspective. In: Proceedings of the 3rd International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005). Lecture Notes in Computer Science, vol. 3410, pp. 14–32. Springer, Berlin (2005)
35. Fonseca, C.M., Fleming, P.J.: An overview of evolutionary algorithms in multiobjective optimization. Evol. Comput. (1995). doi:10.1162/evco.1995.3.1.1
36. Hadka, D., Reed, P.: Borg: an auto-adaptive many-objective evolutionary computing framework. Evol. Comput. (2012). doi:10.1162/EVCO_a_00075
37. Hadka, D., Reed, P.: Diagnostic assessment of search controls and failure modes in many-objective evolutionary optimization. Evol. Comput. (2012). doi:10.1162/EVCO_a_00053
38. He, Z., Yen, G.G.: A new fitness evaluation method based on fuzzy logic in multiobjective evolutionary algorithms. In: 2012 IEEE Congress on Evolutionary Computation (2012). doi:10.1109/CEC.2012.6256534
39. Huband, S., Hingston, P., Barone, L., While, L.: A review of multiobjective test problems and a scalable test problem toolkit. IEEE Trans. Evol. Comput. (2006). doi:10.1109/TEVC.2005.861417
40. Hughes, E.J.: Multiple single objective Pareto sampling. In: 2003 IEEE Congress on Evolutionary Computation (2003). doi:10.1109/CEC.2003.1299427
41. Hughes, E.J.: Evolutionary many-objective optimisation: many once or one many? In: 2005 IEEE Congress on Evolutionary Computation (2005). doi:10.1109/CEC.2005.1554688
42. Hughes, E.J.: Multi-objective equivalent random search. In: Parallel Problem Solving from Nature (PPSN IX), vol. 4193, pp. 463–472 (2006)
43. Hughes, E.J.: MSOPS-II: a general-purpose many-objective optimiser. In: 2007 IEEE Congress on Evolutionary Computation (2007). doi:10.1109/CEC.2007.4424985
44. Inselberg, A., Dimsdale, B.: Parallel coordinates: a tool for visualizing multi-dimensional geometry. In: Proceedings of the First Conference on Visualization '90 (VIS '90), pp. 361–378. IEEE Press, Los Alamitos (1990)
45. Ishibuchi, H., Murata, T.: A multi-objective genetic local search algorithm and its application to flowshop scheduling. IEEE Trans. Syst. Man Cybern. C **28**(3), 392–403 (1998). doi:10.1109/5326.704576
46. Ishibuchi, H., Nojima, Y.: Optimization of scalarizing functions through evolutionary multiobjective optimization. In: Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007). Lecture Notes in Computer Science, vol. 4403, pp. 51–65. Springer, Berlin (2007)
47. Ishibuchi, H., Hitotsuyanagi, Y., Nojima, Y.: Scalability of multiobjective genetic local search to many-objective problems: knapsack problem case studies. In: 2008 IEEE Congress on Evolutionary Computation (2008). doi:10.1109/CEC.2008.4631283
48. Ishibuchi, H., Tsukamoto, N., Hitotsuyanagi, Y., Nojima, Y.: Effectiveness of scalability improvement attempts on the performance of NSGA-II for many-objective problems. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation (GECCO '08), pp. 649–656. ACM Press, New York (2008)
49. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Evolutionary many-objective optimization: a short review. In: 2008 IEEE Congress on Evolutionary Computation (2008). doi:10.1109/CEC.2008.4631121
50. Ishibuchi, H., Sakane, Y., Tsukamoto, N., Nojima, Y.: Adaptation of scalarizing functions in MOEA/D: an adaptive scalarizing function-based multiobjective evolutionary algorithm. In: Proceedings of the

5th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2009). Lecture Notes in Computer Science, vol. 5467, pp. 438–452. Springer, Berlin (2009)

51. Ishibuchi, H., Hitotsuyanagi, Y., Ohyanagi, H., Nojima, Y.: Effects of the existence of highly correlated objectives on the behavior of MOEA/D. In: Proceedings of the 6th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2011). Lecture Notes in Computer Science, vol. 6576, pp. 166–181. Springer, Berlin (2011)
52. Jaimes, López: A., Coello Coello, C., Urías Barrientos, J.E.: Online objective reduction to deal with many-objective problems. In: Proceedings of the 5th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2009). Lecture Notes in Computer Science, vol. 5467, pp. 423–437. Springer, Berlin (2009)
53. Jaimes, López: A., Coello Coello, C., Aguirre, H., Tanaka, K.: Adaptive objective space partitioning using conflict information for many-objective optimization. In: Proceedings of the 6th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2011). Lecture Notes in Computer Science, vol. 6576, pp. 151–165. Springer, Berlin (2011)
54. Jaszkiewicz, A.: On the performance of multiple-objective genetic local search on the 0/1 knapsack problem—a comparative experiment. IEEE Trans. Evol. Comput. (2002). doi:10.1109/TEVC.2002.802873
55. Kauffman, S., Weinberger, E.: The NK model of rugged fitness landscapes and its application to maturation of the immune response. J. Theor. Biol. **141**(2), 211–245 (1989)
56. Keijzer, M.: Scientific discovery using genetic programming. Ph.D. thesis, Technical University of Denmark, Denmark (2001)
57. Khare, V., Yao, X., Deb, K.: Performance scaling of multi-objective evolutionary algorithms. In: Proceedings of the 2nd International Conference on Evolutionary Multi-Criterion Optimization (EMO 2003). Lecture Notes in Computer Science, vol. 2632, pp. 376–390. Springer, Berlin (2003)
58. Knowles, J., Corne, D.: Quantifying the effects of objective space dimension. In: Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007). Lecture Notes in Computer Science, vol. 4403, pp. 757–771. Springer, Berlin (2007)
59. Kollat, J.B., Reed, P.M.: Comparing state-of-the-art evolutionary multi-objective algorithms for long-term groundwater monitoring design. Adv. Water Resour. **29**(6), 792–807 (2006)
60. Köppen, M., Yoshida, K.: Substitute distance assignments in NSGA-II for handling many-objective optimization problems. In: Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007). Lecture Notes in Computer Science, vol. 4403, pp. 727–741. Springer, Berlin (2007)
61. Köppen, M., Yoshida, K.: Visualization of Pareto-sets in evolutionary multi-objective optimization. In: 7th International Conference on Hybrid Intelligent Systems, 2007 (HIS 2007), pp. 156–161. IEEE (2007)
62. Köppen, M., Vicente-Garcia, R., Nickolay, B.: Fuzzy-Pareto-dominance and its application in evolutionary multi-objective optimization. In: Proceedings of the 3rd International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005). Lecture Notes in Computer Science, vol. 3410, pp. 399–412. Springer, Berlin (2005)
63. Kruisselbrink, J.W., Bäck, T., IJzerman, A.P., van der Horst, E.: Evolutionary algorithms for automated drug design towards target molecule properties. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation (GECCO ’08), pp. 1555–1562. ACM Press, New York (2008)
64. Kruisselbrink, J.W., Emmerich, M.T., Bäck, T., Bender, A.: IJzerman, A.P., van der Horst, E.: Combining aggregation with Pareto optimization: a case study in evolutionary molecular design. In: Proceedings of the 5th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2009). Lecture Notes in Computer Science, vol. 5467, pp. 453–467. Springer, Berlin (2009)
65. Kukkonen, S., Lampinen, J.: GDE3: The third evolution step of generalized differential evolution. In: 2005 IEEE Congress on Evolutionary Computation (2005). doi:10.1109/CEC.2005.1554717
66. Kukkonen, S., Lampinen, J.: Ranking-dominance and many-objective optimization. In: 2007 IEEE Congress on Evolutionary Computation (2007). doi:10.1109/CEC.2007.4424990
67. Laumanns, M., Thiele, L., Deb, K., Zitzler, E.: Combining convergence and diversity in evolutionary multi-objective optimization. Evol. Comput. (2002). doi:10.1162/106365602760234108
68. Li, M., Zheng, J., Shen, R., Li, K., Yuan, Q.: A grid-based fitness strategy for evolutionary many-objective optimization. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation (GECCO ’10), pp. 463–470. ACM Press, New York (2010). doi:10.1145/1830483.1830570

69. Lohn, J.D., Kraus, W.F., Haith, G.L.: Comparing a coevolutionary genetic algorithm for multiobjective optimization. In: Proceedings of the 2002 Congress on Evolutionary Computation (2002). doi:10.1109/CEC.2002.1004406
70. López Jaimes, A., Coello Coello, C., Chakraborty, D.: Objective reduction using a feature selection technique. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation (GECCO '08), pp. 673–680. ACM Press, New York (2008)
71. López Jaimes, A., Coello Coello, C.: Study of preference relations in many-objective optimization. In: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation (GECCO '09), pp. 611–618. ACM Press, New York (2009)
72. Miettinen, K.: Nonlinear Multiobjective Optimization. Springer, New York (1999)
73. Miettinen, K.: Graphical illustration of Pareto optimal solutions. In: Multi-Objective Programming and Goal Programming: Theory and Applications, pp. 197–202. Springer, Berlin (2003)
74. Mitra, P., Murthy, C., Pal, S.: Unsupervised feature selection using feature similarity. IEEE Trans. Pattern Anal. Mach. Intell. **24**(3), 301–312 (2002)
75. Mostaghim, S., Branke, J., Schmeck, H.: Multi-objective particle swarm optimization on computer grids. In: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO '07), pp. 869–875. ACM Press, New York (2007)
76. Murata, T., Taki, A.: Many-objective optimization for knapsack problems using correlation-based weighted sum approach. In: Proceedings of the 5th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2009). Lecture Notes in Computer Science, vol. 5467, pp. 468–480. Springer, Berlin (2009)
77. Murata, T., Ishibuchi, H., Gen, M.: Specification of genetic search directions in cellular multi-objective genetic algorithms. In: First International Conference on Evolutionary Multi-Criterion Optimization. Lecture Notes in Computer Science, vol. 1993, pp. 82–95. Springer, Berlin (2001). doi:10.1007/3-540-44719-9_6
78. Naujoks, B., Beume, N., Emmerich, M.: Multi-objective optimization using S-metric selection: application to three-dimensional solution spaces. In: 2005 IEEE Congress on Evolutionary Computation (2005). doi:10.1109/CEC.2005.1554838
79. Obayashi, S., Sasaki, D.: Visualization and data mining of Pareto solutions using self-organizing map. In: Proceedings of the 2nd International Conference on Evolutionary Multi-Criterion Optimization (EMO 2003). Lecture Notes in Computer Science, vol. 2632, pp. 796–809. Springer, Berlin (2003)
80. Pasia, J., Aguirre, H., Tanaka, K.: Improved random one-bit climbers with adaptive ϵ-ranking and tabu moves for many-objective optimization. In: Proceedings of the 6th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2011). Lecture Notes in Computer Science, vol. 6576, pp. 182–196. Springer, Berlin (2011)
81. Pryke, A., Mostaghim, S., Nazemi, A.: Heatmap visualization of population based multi objective algorithms. In: Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007). Lecture Notes in Computer Science, vol. 4403, pp. 361–375. Springer, Berlin (2007)
82. Purshouse, R., Fleming, P.J.: Conflict, harmony, and independence: relationships in evolutionary multi-criterion optimisation. In: Proceedings of the 2nd International Conference on Evolutionary Multi-Criterion Optimization (EMO 2003). Lecture Notes in Computer Science, vol. 2632, pp. 16–30. Springer, Berlin (2003)
83. Purshouse, R.C., Fleming, P.J.: Evolutionary multi-objective optimisation: an exploratory analysis. In: 2003 IEEE Congress on Evolutionary Computation (2003). doi:10.1109/CEC.2003.1299927
84. Purshouse, R.C., Fleming, P.J.: On the evolutionary optimization of many conflicting objectives. IEEE Trans. Evol. Comput. (2007). doi:10.1109/4235.797969
85. Purshouse, R., Jalba, C., Fleming, P.: Preference-driven co-evolutionary algorithms show promise for many-objective optimisation. In: Proceedings of the 6th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2011). Lecture Notes in Computer Science, vol. 6576, pp. 136–150. Springer, Berlin (2011)
86. Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., Tarantola, S.: Global Sensitivity Analysis: The Primer. Wiley-Interscience, Hoboken (2008)
87. Sato, H., Aguirre, H.E., Tanaka, K.: Controlling dominance area of solutions and its impact on the performance of MOEAs. In: Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007). Lecture Notes in Computer Science, vol. 4403, pp. 5–20. Springer, Berlin (2007)

88. Sato, H., Aguirre, H.E., Tanaka, K.: Pareto partial dominance MOEA and hybrid archiving strategy included CDAS in many-objective optimization. In: 2010 IEEE Congress on Evolutionary Computation (2010). doi:10.1109/CEC.2010.5586247
89. Sato, H., Aguirre, H., Tanaka, K.: Variable space diversity, crossover and mutation in MOEA solving many-objective knapsack problems. Ann. Math. Artif. Intell. 1–28 (2012). doi:10.1007/s10472-012-9293-y
90. Saxena, D.K., Deb, K.: Non-linear dimensionality reduction procedures for certain large-dimensional multi-objective optimization problems: employing correntropy and a novel maximum variance unfolding. In: Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007). Lecture Notes in Computer Science, vol. 4403, pp. 772–787. Springer, Berlin (2007)
91. Saxena, D., Zhang, Q., Duro, J., Tiwari, A.: Framework for many-objective test problems with both simple and complicated Pareto-set shapes. In: Proceedings of the 6th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2011). Lecture Notes in Computer Science, vol. 6576, pp. 197–211. Springer, Berlin (2011)
92. Saxena, D., Duro, J., Tiwari, A., Deb, K., Zhang, Q.: Objective reduction in many-objective optimization: Linear and nonlinear algorithms. IEEE Trans. Evol. Comput. (2012). doi:10.1109/TEVC.2012.2185847
93. Schütze, O., Lara, A.: Coello Coello, C.: On the influence of the number of objectives on the hardness of a multiobjective optimization problem. IEEE Trans. Evol. Comput. (2011). doi:10.1109/TEVC.2010.2064321
94. Sierra, M.R.: Coello Coello, C.A.: Improving PSO-based multi-objective optimization using crowding, mutation and ϵ-dominance. In: Proceedings of the 3rd International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005). Lecture Notes in Computer Science, vol. 3410, pp. 505–519. Springer, Berlin (2005)
95. Sülflow, A., Drechsler, N., Drechsler, R.: Robust multi-objective optimization in high dimensional spaces. In: Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007). Lecture Notes in Computer Science, vol. 4403, pp. 715–726. Springer, Berlin (2007)
96. Tanaka, M., Watanabe, H., Furukawa, Y., Tanino, T.: GA-based decision support system for multicriteria optimization. In: Proceedings of the International Conference on Systems. Man, and Cybernetics, vol. 2, pp. 1556–1561. IEEE, Piscataway (1995)
97. Teytaud, O.: On the hardness of offline multi-objective optimization. Evol. Comput. (2007). doi:10.1162/evco.2007.15.4.475
98. Veldhuizen, D.A.V.: Multiobjective evolutionary algorithms: classifications, analyses, and new innovations. Ph.D. thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB (1999)
99. Veldhuizen, D.A.V., Lamont, G.B.: Multiobjective evolutionary algorithm research: a history and analysis. Technical report TR-98-03, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB (1998)
100. Wagner, T., Beume, N., Naujoks, B.: Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In: Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007). Lecture Notes in Computer Science, vol. 4403, pp. 742–756. Springer, Berlin (2007)
101. Walker, D., Fieldsend, J., Everson, R.: Visualising many-objective populations. In: Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference Companion (GECCO Companion '12) (2012). doi:10.1145/2330784.2330853
102. Weinberger, K.Q., Saul, L.K.: Unsupervised learning of image manifolds by semidefinite programming. Int. J. Comput. Vis. **70**(1), 77–90 (2006)
103. Xu, J.W., Pokharel, P., Paiva, A., Principe, J.: Nonlinear component analysis based on correntropy. In: International Joint Conference on Neural Networks, 2006 (IJCNN '06), pp. 1851–1855 (2006)
104. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. (2007). doi:10.1109/TEVC.2007.892759
105. Zhang, Q., Liu, W., Li, H.: The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In: 2009 IEEE Congress on Evolutionary Computation (2009). doi:10.1109/CEC.2009.4982949

106. Zitzler, E., Künzli, S.: Indicator-based selection in multiobjective search. In: Parallel Problem Solving from Nature (PPSN VIII). Lecture Notes in Computer Science, vol. 3242, pp. 832–842. Springer, Berlin (2004)
107. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms—a comparative study. In: Parallel Problem Solving from Nature (PPSN V), pp. 292–301. Springer, Amsterdam (1998)
108. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Trans. Evol. Comput. (1999). doi:10.1109/4235.797969
109. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: empirical results. Evol. Comput. (2000). doi:10.1162/106365600568202
110. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength Pareto evolutionary algorithm. Technical report 103, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology Zurich (2001)
111. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans. Evol. Comput. (2003). doi:10.1109/TEVC.2003.810758
112. Zou, X., Chen, Y., Liu, M., Kang, L.: A new evolutionary algorithm for solving many-objective optimization problems. IEEE Trans. Syst. Man Cybern. B **38**(5), 1402–1412 (2008)