Information Sciences 530 (2020) 22–42

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A new approach to generate pattern-efficient sets of non-dominated vectors for multi-objective optimization

Bogdana Stanojević [a,*], Fred Glover [b]

[a] Mathematical Institute of the Serbian Academy of Sciences and Arts, Kneza Mihaila 36, Belgrade 11001, Serbia
[b] College of Engineering & Applied Science, University of Colorado, Boulder, USA

ARTICLE INFO

Article history:
Received 30 October 2018
Revised 28 January 2020
Accepted 15 April 2020
Available online 28 April 2020

Keywords:
Multiple objectives
Efficient frontier
Pattern efficiency
Diversified representations

ABSTRACT

Pareto optimality is the fundamental construct employed to determine whether a given solution to a multi-criteria mathematical optimization model is preferred to another solution. In this paper we describe an approach (Pattern Efficient Set Algorithm – PESA) to generating a pattern-efficient set of non-dominated vectors to a multi-objective optimization problem. Our approach incorporates an optimization model designed to yield certain non-dominated vectors that can fill gaps between already generated non-dominated vectors, providing a way to deal with the adjacency of generated non-dominated vectors and to quantify the gaps between them. We also propose a pseudo-randomized variant of PESA (rPESA) that randomly generates hypothetical bounds for the objective functions and uses them in the optimization model.

To test our approach we selected ten problems from the literature, including bi-objective, 3-objective, 5-objective and 10-objective test instances with non-convex, disconnected or continuous Pareto. The inverted generational distance (IGD) and the hyper-volume (HV) are used as performance metrics to measure the quality of the obtained approximations. We also present graphically the numerical results from applying our method.

1. Introduction

The efficient frontier associated with Pareto efficiency has found special application in multi-objective (multicriteria) decision making, where it is used to classify trade-offs among decisions according to their effect on different organizational or business goals. Pareto optimality is the fundamental construct employed to determine whether a given solution to a mathematical optimization model is preferred to another solution. Non-linear multiple objective optimization has been widely studied, and several surveys can be found in the literature (see for instance [12,14,19]).

Some of the more salient studies of multiple objective optimization problems are briefly noted as background for the approach studied here. Das and Dennis [4] addressed the challenge of minimizing weighted sums of objectives when generating a Pareto set, identifying pitfalls and providing several examples of convex Pareto curves, together with showing that an uneven distribution of weights can correspond to an even spread of points on those Pareto curves. Ray et al. [16] developed an evolutionary algorithm for generic multiple objective design optimization problems, founding their approach on non-

* Corresponding author.
E-mail addresses: bgdnpop@mi.sanu.ac.rs (B. Stanojević), glover@colorado.edu (F. Glover).

https://doi.org/10.1016/j.ins.2020.04.040
0020-0255/

dominance and disclosing an ability to avoid the use of scaling and aggregation that generally negatively affect conventional penalty function methods for constraint handling.

A decomposition-based evolutionary algorithm was presented by Asafuddoula et al[1]. that generates uniformly distributed reference points using systematic sampling, maintaining a fair balance between convergence and diversity by using two independent distance measures. They proposed a simple preemptive distance comparison scheme for association and used an adaptive epsilon formulation to deal with the constraints. Chand and Wagner [3] presented an overview of recent developments in many-objective optimization; and suggested some research directions in that field. Zitzler et al. [25] provided a systematic comparison of various evolutionary approaches to multiple objective optimization proposing six carefully chosen test problems, each involving a feature that is known to cause difficulty in the evolutionary optimization process. We use one of their instances to illustrate our approach, described subsequently.

A non-dominated sorting-based multiple objective evolutionary algorithm called NSGA-II, was developed by Deb et al. [5] that uses a fast non-dominated sorting procedure, an elitist strategy, and a simple and efficient constraint-handling method. A multiple objective evolutionary algorithm was proposed by Zhang and Li [22] based on decomposition (MOEA/D), which decomposed the multiple objective optimization problem into a number of scalar optimization sub-problems, and optimized them simultaneously. Since then many variants of MOEA/D have been developed (see for instance MOEA/D-DE [11], MOEA/D-AWA [15], and MOEA/D-UD [24]). A method for adjusting weight vectors for bi-objective optimization problems with discontinuous Pareto front (MOEA/D-ABD) was recently proposed in [21].

The widespread relevance of multi-objective optimization to practical decision making in business and government makes it a central focus of our current study. Within this setting we disclose that classical Pareto efficiency analysis harbors two key limitations that create a need for a better way to evaluate decision trade-offs. The first limitation involves a reliance on assumptions that are often unrealistic in practical settings, causing the approach to exclude solutions that are highly relevant to decision analysis. The second involves the presence of a "blind spot" that makes the approach oblivious to the issue of generating solutions that are diverse in certain critical ways.

As emphasized in [19], any approach aiming to generate a subset of the Pareto front must control the number of the derived non-dominated solutions. In weighted sum approaches, moreover, it is essential to choose the parameters intelligently from one iteration to another, in order to obtain a set of diversified non-dominated solutions. We aim to generate non-dominated vectors that populate the criterion space in a representative manner without wide gaps between them. Preliminary results of our research were presented in [18].

The main contributions of this paper are three-fold: (i) we propose a novel approach to solve multi- and many-objective optimization problems that is able to derive varied types of solutions; (ii) our advanced formulation optimizes a weighted sum (parametric) aggregation of the objective functions plus a scalar variable multiplied by a constant, under additional constraints consisting of hypothetical bounds on the objective functions - keeping in this way the simplicity of the weighted sum technique and also enabling the derivation of non-supported non-dominated points; (iii) our approach provides a methodology for choosing the model parameters such that the generated non-dominated points cover the Pareto front in a pattern-efficient manner.

As a caveat regarding our findings, in the process of our empirical testing we encountered a limitation in the current state-of-the-art for solving optimization problems that affected the outcomes of some of our experiments. In practice, as pointed out in [13], only locally Pareto optimal solutions are available as a basis for running experiments of proposals for solving multi-objective problems unless special conditions, such as convexity, are fulfilled. At present, global solvers often yield locally optimal solutions that are not globally optimal. As a result, although the methodology introduced in this paper theoretically yields non-dominated vectors, we found during experimentation on non-convex instances that the global algorithm we used as a subroutine became trapped in some sub-optimal Pareto solutions. This weakness can be noticed in some graphic representations and in some numerical results accompanying our experimentation. Despite this, our overall experimental findings confirm the soundness of our theoretical results.

The paper is organized as follows. Section 2 briefly describes the multiple objective optimization problem and introduces basic terminology. This section additionally discusses some limitations of the popular weighted sum scalarization technique for deriving non-dominated vectors. We propose advanced formulations for generating non-supported non-dominated vectors in Section 3, providing a description of parameter settings for our proposed optimization model, and giving a way to compute the gaps between adjacent non-dominated vectors using the generalized volume of a $j-$simplex in R^n. Our solution algorithm is introduced in Section 4, and numerical results are presented in Section 5. Finally, Section 6 offers concluding remarks.

2. Problem formulation

We consider the system of m optimization problems given by

$$\max_{x\in X}(f_i(x))_{i\in M}, \tag{1}$$

where $M = \{1, \ldots, m\}$. The vector x is $n-$dimensional with components represented by x_j, $j \in N = \{1, \ldots, n\}$ and the solution space X can be convex or non-convex. In a variety of optimization problems of interest, X may be written in the form

$$X = \left\{x \in R^n | Ax \le b,\ g(x) \le d,\ x_j \text{ discrete},\ j \in D\right\}, \tag{2}$$

respectively identifying linear constraints, nonlinear constraints and discrete (e.g. integer) variable conditions. When the nonlinear constraints and discrete variable conditions are absent, X is a constraint set of the type customarily encountered in linear programming, and when in addition the function $f_i(x)$ is linear, the corresponding $i-$th maximization problem is a linear program.

The multi-objective optimization problem consists in finding solutions that solve the system given in Eq. (2) in a special sense, accounting for the need to balance trade-offs between the objectives $f_i(x)$, given that it is highly unlikely that a single x vector in X will optimize each of the objectives simultaneously.

2.1. Terminology

Let $F(x) = (f_1(x), \ldots, f_m(x))$ represent the m-vector of objective function values for x. Let Y denote the **criterion set**, i.e. the image of the **feasible set** X in the criterion space. A solution x' is called feasible if $x' \in X$, and x' is said to dominate another feasible solution x'' if $F(x') \geq F(x'')$ (hence if $f_i(x') \geq f_i(x'')$ for all $i \in M$). Under such conditions, x' strictly dominates x'' if in addition $f_i(x') > f_i(x'')$ for at least one $i \in M$. A feasible solution is **non-dominated** if there is no feasible solution that strictly dominates it. The condition of being non-dominated expresses the condition in which a feasible solution cannot be made better relative to one of the objectives of Eq. (2) without being made worse for another. We will use the term **efficient solution** with the same meaning as **non-dominated solution**. Let $\left(f_1(x'), \ldots, f_m(x')\right)$ be called **non-dominated vector** if and only if x' is a non-dominated solution. Following [14], we will call **efficient frontier**, or **Pareto front** (PF) the set of all non-dominated vectors.

A solution $x' \in X$ is called marginal if and only if it optimizes one objective function over the feasible set X. Each criterion $f_i, i = 1, \ldots, m$ has its own marginal solutions, and at least one marginal solution that is non-dominated. For the i-th criterion one non-dominated marginal solution can be obtained by using the lexicographic method with the ith criterion in the first position, and the others in an arbitrary order.

By this means, we can generate non-dominated solutions directly by selecting any given vector c satisfying $c_i > 0$, $i \in M$ and $\sum_{i \in M} c_i = 1$; and solving the following optimization problem (called **weighted sum scalarization (WSS) problem**) over the space of solutions X

$$\max \left\{ \sum_{i \in M} c_i f_i(x) | x \in X \right\}. \tag{3}$$

The efficient solutions x obtained by solving Eq. (3) with $c_i > 0$, $i \in M$, are called **supported efficient solutions**. The vector $y = F(x)$ is then called a **supported non-dominated vector**. In other words, the supported non-dominated vectors are the non-dominated vectors of the convex cover of the criterion set. For more details concerning the terminology of multi-objective optimization we refer the reader to Ehrgott [8].

2.2. Limitation of the classical scalarization approaches

Under circumstances where the classical assumptions regarding X and the functions $f_i(x)$ are satisfied, each efficient solution is also a supported efficient solution; the objective function values of efficient solutions lie on a boundary of a convex set, and the Pareto front can be visualized in a convenient manner.

A possible way to obtain a representation of the Pareto front is to vary the weights c_i, $i \in M$ and solve the corresponding WSS problems described in Eq. (3). Though it is sometimes difficult to set the weights to obtain a non-dominated vector in a particular region of the criterion space, the classical WSS approach has proved valuable in many applications, and is highly useful in analyzing trade-offs among such solutions in multi-objective optimization. Consequently, it becomes important to consider how to derive an appropriate analog of this approach under the more general conditions encountered in real world settings where the classical assumptions typically fail to apply.

The relevance of this challenge is underscored by the fact that the classical structure of the set of the supported non-dominated vectors changes radically in the absence of convexity, or more accurately, loses its relevance. Even in classical settings, a practical representation of an efficient frontier is often not easy to create, and under a variety of circumstances can be highly misleading. The curves constructed by economists are typically based on the supposition that all relevant solutions are known, at least in principle (which is to say, the distribution of their objective function values can be inferred and graphed) - a supposition that is generally not valid.

Example 1. To emphasize the difficulties caused by the non-convexity of the Pareto front for weighted-sum based algorithms, we consider the bi-objective problem

$$\begin{array}{ll} \max & f_1(x) = x_1^2, \\ \max & f_2(x) = x_2^2, \\ \text{s.t.} & x_1 + x_2 \leq 1, \\ & x_1, x_2 \geq 0. \end{array} \tag{4}$$

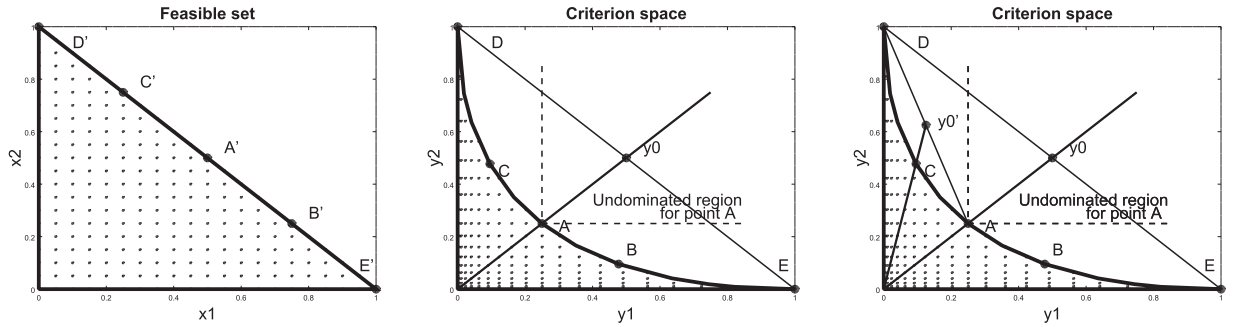

Fig. 1. Graphical representation of the feasible set (to the left), the criterion space and a few non-dominated vectors on the Pareto frontier of Problem (4) (right). The vectors D and E are supported non-dominated vectors, while A, B and C are non-supported non-dominated vectors.

The feasible and criterion sets are shaded in Fig. 1. Although the feasible set is convex, the functions x_1^2 and x_2^2 are convex rather than concave (as is required in the case of maximization). As a result, the problem has only two supported efficient solutions: $\bar{x}_1 = (1, 0)$ (point D) and $\bar{x}_2 = (0, 1)$ (point E).

Fig. 1 (middle and right) also shows the region consisting of points that are non-dominated relative to the point *A*. It may be noted that this region ($y_1 \geq 1/4, y_2 \geq 1/4$) is the region that results by shifting the origin to the position of the point A.

This example shows that the reliance on positive *c* vectors in a WSS problem (Eq. (3)) is not sufficient to identify all non-dominated solutions. It may even be that such vectors will only identify a small subset of the efficient solutions, in this case, two out of an infinite number. Consequently, it is desirable to identify formulations that preserve the ability to identify supported non-dominated solutions, but that also make it possible to identify a much larger range of non-dominated solutions.

The Tchebycheff method is one of the well known parametric scalarization methods that uses both a reference point z^* and weights w_i for the objectives. It is proved that by choosing appropriate parameters, it is possible to find any Pareto optimal solution (supported or unsupported) by solving the corresponding weighted Tchebycheff problem

$$\min_{x \in X} \max_{i \in M} w_i\left(z_i^* - f_i(x)\right).$$

For some parameters the method may yield a weakly Pareto optimal solution to the original problem, but this disadvantage can be overcome by using an augmented weighted Tchebycheff method [12]. Many versions of the weighted Tchebycheff method can be found in the literature and they are widely used in developing evolutionary algorithms based on decomposition.

Some methods for solving multi-objective optimization problems were analyzed in [9], disclosing some advantages of using a decomposition-based method (particularly the Tchebysheff scalarizing function) over Pareto-based methods together with the assumptions needed to achieve a good functionality. The authors also proposed a methodology to obtain optimal scalarizing functions for a given problem, subject to prior assumptions on the Pareto front geometry. According to [12], the weakness of the Tchebycheff method is the burden of computation and the assumptions needed at each iteration.

The advanced formulation that we introduce in the next section has the useful property of permitting a strategy that iteratively discovers progressively diverse (pattern-efficient) solutions which can model the structure of the Pareto front. In this way, being able to generate non-supported non-dominated points, our approach overcomes limitations of the widely used WSS method. At the same time, we retain the computational simplicity of WSS method, and additionally provide a methodology to tune the parameters that assures a pattern-efficient cover of the Pareto front, thereby also improving the Tchebysheff scalarization method.

3. Advanced formulations for generating non-dominated vectors

The key to our advanced formulations consists of introducing a scheme that penalizes violations of hypothetical lower bounds on *y* vectors. The lower bounds are hypothetical (tentative) because there is no way to know their correct value at the outset, barring special knowledge about the objective functions and the space of feasible solutions. However, the formulations have the useful property of permitting a strategy that applies them iteratively to discover progressively better lower bound estimates, and allowing the search in those parts of the space that need to be searched to find diverse (pattern-efficient) solutions.

3.1. The targeted directional model

Building on ideas first sketched in Glover [10], we propose the following Targeted Directional Model (TDM) as given in Eq. (5) as alternative to the WSS model.

$$\max\left\{\sum_{i\in M} c_i f_i(x) + p_0\lambda + q_0\sum_{i\in M} c_i\left(f_i(x) - \lambda y_i^0\right) | x \in X,\ f_i(x) \geq \lambda y_i^0,\ i \in M,\ \lambda \geq 0\right\} \tag{5}$$

Here p_0 and q_0 are scalar constants; λ is a scalar variable; c, y^0 are $m-$dimensional constants; and all functions f_i, $i \in M$ are normalized.

Further on we analyze the effect of solving Eq. (5) in the particular cases $p_0 = 0$, $q_0 = 0$, and $p_0 > 0$, $q_0 = 0$. Finally, we address the most general case $p_0 > 0$, $q_0 > 0$.

Solving TDM (Eq. (5)) with $p_0 = 0$ and $q_0 = 0$ yields only supported non-dominated points, since the reduced model corresponds to the WSS method.

We proceed to explain what can be achieved solving TDM with $p_0 > 0$ and $q_0 = 0$. The hypothetical bound y^0 on vectors $y \in Y$ is used as a direction vector for a line from the origin which passes through y^0. The model seeks the best point y on this line relative to the dominance criterion. At the same time we seek the maximum value of a scalar variable λ such that the point $y = \lambda y^o$ lies in Y. This implies that y dominates all other points on the line and is itself (necessarily) non-dominated by any point on the line that lies in Y. It is nevertheless possible that y is dominated by some y in Y that does not lie on the line. For this reason, we make use of a positive scalar p_0 as an incentive coefficient for making λ as large as possible, and add the term $p_0\lambda$ to the classic objective function of the WSS method.

A graphical illustration is given in Fig. 1. Suppose we have obtained the two extreme points D and E that respectively optimize $f_1(x)$ and $f_2(x)$, giving the corresponding objective function values of $\bar{y}_1 = (1, 0)$, $\bar{y}_2 = (0, 1)$. To find a non-dominated solution that fills the gap between D and E vectors, assume we generate the hypothetical bound y^0 by means of the convex combination that weights each of D and E equally, to give $y^0 = (1/2, 1/2)$. The line from the origin through the point y^0 shown in Fig. 1 (middle) is the line that gives rise to the variable λ in formulation (5) and identifies the non-dominated point A on this line. Similarly, to find a non-dominated solution that fills the gap between D and A (Fig. 1 (right)), we generate $y^{0\prime}$ as a convex combination of D and A, and then obtain the non-dominated C vector on the line that connects the origin and $y^{0\prime}$.

We now turn to the general case $p_0 > 0$, $q_0 > 0$. Employing the non-negative scalar q_0, we add a further term to the objective with the idea of not only maximizing λ (by means of the incentive coefficient p_0) but of pushing the solution farther in the direction of the efficient frontier, if the maximization of λ does not already achieve this, by maximizing the sum of the values $f_i(x) - \lambda y_i^0$ over $i \in M$. Thus the new term will reinforce the role of the term $cF(x)$ in driving the solution to lie on the frontier. The rationale for this additional maximization component is that the way to retain the focus on "filling a gap" provided by the target y_i^0 and to compatibly achieve a non-dominated solution is to push the $f_i(x)$ values beyond their lower bounds of λy_i^0. The term $f_i(x) - \lambda y_i^0$ thus comes into play because it represents the amount of the increase of $f_i(x)$ beyond its lower bound. (Note that all the terms $f_i(x) - \lambda y_i^0$ are non-negative by the constraints of Eq. (5).) A natural variant of this objective is to introduce the c_i components of the vector c into the new objective function term to multiply the entire quantity $f_i(x) - \lambda y_i^0$ and not just the quantity $f_i(x)$.

The justification for the modelgiven in Eq. (5) is provided by Theorem 1 that applies to a more general type of formulation. To state the theorem, we refer to a system of the form $Ay \geq G(z)$, $z \in Z$, where A is a nonzero matrix satisfying $A \geq 0$, $G(z)$ is a vector valued mapping whose output is conformable with Ay, and Z is nonempty.

Theorem 1. *For any function $g(z)$ and for $c > 0$, an optimal solution to the* model *given in Eq. (6)*

$$\max\{cy + g(z) | y \in Y, Ay \geq G(z), z \in Z\} \tag{6}$$

yields a non-dominated y vector.

Proof. Let y', z' denote an optimal solution to Eq. (6). Then y' is optimal for the instance of Eq. (6) where z' replaces z. Suppose y^* is a vector that strictly dominates y. Then $Ay^* \geq Ay' \geq G(z')$ and hence y^*, z' is feasible for Eq. (6). In addition, for c positive, $cy^* > cy'$ which contradicts the optimality of y', z'. □

For an additional useful outcome, we replace $g(z)$, $z \in Z$ by $p_0\lambda$, where p_0 is constant scalar, and λ is a positive scalar variable. Correspondingly, we replace A by the unit matrix and replace $G(z)$ by λy^0, where y^0 is a constant vector conformable with y. Then the following Corollary 1 can be derived from Theorem 1.

Corollary 1. *For $p_0 \geq 0$ and $c > 0$, an optimal solution tothe* model *given in Eq. (7)*

$$\max\left\{cy + p_0\lambda | y \in Y, y \geq \lambda y^0, \lambda \geq 0\right\} \tag{7}$$

yields a non-dominated y vector.

Observing the model defined by Eq. (5) in the criterion space (i.e. using y_i instead of $f_i(x)$ and $y \in Y$ instead of $x \in X$) with its coefficients in unified form, i.e. as

$$\max \left\{ (1+q_0)\sum_{i\in M} c_i y_i + \left(p_0 - q_0 \sum_{i\in M} c_i y_i^0 \right) \lambda \,|\, y \in Y, y_i \geq \lambda y_i^0, i \in M, \lambda \geq 0 \right\}, \tag{8}$$

we may conclude that Corrolary 1, with c replaced by $(1+q_0)c$ and p_0 replaced by $(p_0 - q_0 \sum_{i\in M} c_i y_i^0)$ applies to Model (5). Therefore an optimal solution to Eq. (5) yields a non-dominated y vector to the original roblem described by Eq. (1).

3.2. Parameters settings

The c vector used in Eq. (5) is generated using the same weights that are used to create the target vector y^0. To begin, we assume we have generated the extreme vectors that maximize each function $f_i(x)$ separately, simultaneously using small weights to maximize the sum of the remaining functions given the maximization of a given $f_i(x)$. Note that the determination of weights is done more readily by considering the normalized functions $f_i(x)$, since these are more commensurate in value.

The performance of the proposed algorithm essentially depends on the values set for p_0 and q_0. To create a solution that fills a gap as effectively as possible, we choose p_0 to be sufficiently large in relation to q_0, i.e. $(p_0 - q_0 \sum_{i\in M} c_i y_i^0) \geq 0$, to assure that the goal of maximizing λ contributes more to the objective than the goal of maximizing $\sum_{i\in M}(f_i(x) - \lambda y_i^0)$. However, an optimal solution to the formulation will lie on the efficient frontier even without such a relationship between p_o and q_0.

To determine p_0 and q_0, we make use of a term $U_0 = \sum_{i\in M} c_i$. We have chosen to set $q_0 = 1 + multq$, where $multq$ is a multiple selected from the interval from 0 to 1. (Suggested values for initial testing are 0.1, 0.2 and 1.) Finally, we set $p_0 = multp \cdot U_0 \cdot (1+q_0)$ where $multp$ is chosen to be one of the multiples 5, 10, 20, 50, 100. Note that these settings imply that the coefficient of λ in Eq. (8) is positive, thus obliging λ to increase as much as possible.

These of course are simply values suggested for experimentation. If the value $multp = 5$ or 10 works well, there is no need to look at other values. The optimal solutions for Eq. (5) are assured to lie on the efficient frontier, whether or not the term $cF(x)$ is included in the formulation. Depending on the values of the c_i coefficients, the value of p_0 may need to be increased in relation to q_0 to assure the priority of maximizing λ.

3.3. Adjacency and gap measuring

To generate a pattern efficient set of non-dominated vectors to the problem given in Eq. (1), we need to measure somehow the gaps between already generated non-dominated vectors. In fact, our algorithm keeps information about "places" (gaps) where new non-dominated vectors are desired to be generated by measuring the generalized volumes of such gaps and ordering them accordingly.

For the case $m = 2$, which is very particular, we generate a new non-dominated vector between two adjacent non-dominated vectors already generated. In Fig. 1 (middle) which has only two non-dominated points D and E we identify the single gap between them and measure it as the length of the line segment DE. After generating the point A we identify two gaps, one between D and A, and another one between E and A. Their measure is the same due to the symmetry. But, after generating C there are three gaps: between D and C, C and A, and A and E. The length AE is obviously greater than CA and AE, so the next gap that will be filled is between A and E. The relation of adjacency for $m = 2$ is evident, since the non-dominated vectors can be lexicographically ordered.

The case for $m = 3$, is shown in Fig. 2 (left). Considering that A, B and C are already generated, the gap between them will be measured as the area of the triangle ABC. The relation of adjacency for $m = 3$ is determined as follows. In our algorithm, considering that the points A, B and C are adjacent, every new non-dominated point generated with their help becomes adjacent separately to A and B, to A and C, and to B and C.

For the case $m = 4$, considering that A, B, C and D are already generated (Fig. 2 (right)) the gap between them will be measured as the volume of the tetrahedron $ABCD$. Similarly, every new non-dominated point generated with the help of A, B, C and D becomes separately adjacent to each triple A, B, C; A, B, D; A, C, D; and B, C, D.

For the general case, we define the adjacency of m non-dominated vectors in a recursive way: when a new non-dominated vector y^{new} is generated with the help of m non-dominated vectors $y^1, \ldots, y^m$, it becomes adjacent (separately) to each group of $m-1$ vectors out of those initial m. To decide which gap to cover first, we need to measure the generalized volume of each gap and keep them ordered, according to their volumes.

Elementary shapes from 1D, 2D and 3D geometry, i.e. segments, triangles and tetrahedrons respectively are generalized by $j-$simplexes in R^n. We compute the generalized volumes of such shapes in the algorithm introduced in the next section. For this, we use the Cayley-Menger determinant (Sommerville [17]) identified in Eq. (9), to compute the generalized volume of a $j-$simplex S in R^n formed with $j+1$ vertexes $v_1, v_2, \ldots, v_{j+1} \in R^n$:

$$V_j^2(S) = \frac{(-1)^{j+1}}{2^j (j!)^2} \det\left(\widehat{B}\right), \tag{9}$$

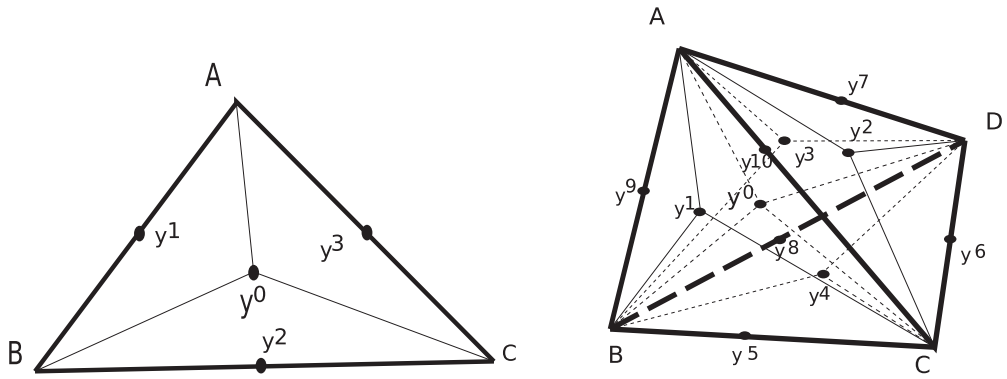

Fig. 2. Graphical illustration for $j-$simplexes in R^3 and R^4. Triangle ABC to the left is a $2-$simplex in R^3, and the tetrahedron $ABCD$ to the right is a $3-$simplex in R^4. The segment AB to the left is a $1-$simplex in R^3 The face ABC of the tetrahedron to the right is a $2-$simplex in R^4.

where $\widehat{B}$ is a $(j+2)\times(j+2)$ matrix obtained from the matrix $B=(\beta_{ik})_{i=1,\ldots,j+1}^{k=1,\ldots,j+1}$, $\beta_{ik}=||v_i-v_k||_2^2$, bordered with a top row $(0,1,1,\ldots,1)$ and a left column $(0,1,1,\ldots,1)^T$.

In the general case of non-degeneracy of the problem described in Eq. (1) (i.e. the number of the objective functions is $m\geq 3$ and the number of distinct marginal non-dominated vectors is also m) we must compute the generalized volume of an $(m-1)-$simplex in R^m.

4. The solution algorithm

In this section we present our **P**attern**E**fficient**S**et**A**lgorithm (PESA) that provides a pattern-efficient set of non-dominated vectors to a multi-objective optimization problem. In the initialization step, the algorithm finds m non-dominated vectors (Step 1); qualifies them to be adjacent; and records the gap between them (Step 2). Then, iteratively (Step 4), the algorithm identifies the greatest gap (Sub-step (a)); discards that gap by solving single-objective optimization problems and generating new non-dominated vectors within it; establishes the new adjacent neighbors, quantifies and records the gaps between them as described in the previous section (Sub-steps (b) and (c)).

Before discarding a gap the algorithm constructs new vectors that are used as hypothetical bounds on the objective functions. For instance, Fig. 2 (left) shows which vectors will be used one-by-one in TDM to generate new non-dominated vectors within the gap ABC of a 3-objective optimization problem, where first y^0 is computed as the center of gravity of the triangle ABC, and then y^1, y^2 and y^3 are computed as the centers of gravity for the triangle's sides. Fig. 2 (right) shows which vectors will be used to generate new non-dominated vectors within the gap $ABCD$ of a 4-objective optimization problem: here, y^0 is the center of gravity of the tetrahedron $ABCD$, y^1, y^2, y^3 and y^4 are centers of gravity for the tetrahedron's faces, and y^5, y^6, y^7, y^8, y^9 and y^{10} are centers of gravity for the tetrahedron's sides.

Data describing an instance of the multiple objective problem given in Eq. (1) together with proper values for the additional parameters needed in Eq. (5) are inputs for the solution algorithm. Note that in Eq. (5) we refer to an objective function vector y^0 without specifying a corresponding solution x^0. We are interested in a situation where y^0 is generated directly as a specific convex combination of certain y vectors in Y, without generating a point x^0 that yields $y^0=F(x^0)$. In fact, y^0 may not belong to Y, and hence there may be no x^0 in X that produces y^0. By strict rules y^0 will be produced for the goal of filling in gaps in the current set of generated non-dominated vectors to give a diversified coverage of the Y space.

The output of the algorithm is the list G of the non-dominated vectors generated. List L contains the problems that are waiting to be solved. Each problem recorded in L contains a sequence of m adjacent non-dominated vectors y, the coefficients c used to derive them, and the generalized volume V of the $(m-1)-$simplex in R^m formed with the vectors Y. List L is maintained ordered with respect to the decreasing values V. The algorithm stops either after the maximum number of iterations is reached or when the list of problems L becomes empty.

PESA is summarized below.

Step 1. (Initial stage) Define m coefficient vectors $c^1,\ldots,c^m$ as follows. The coefficient c^i is a vector of size m containing a value close to 1 in the $i-$th position, and values close to 0 in remaining positions. For each $i=1,\ldots,m$, c^i in Eq. (3) is used to produce the $i-$th non-dominated vector y^i. In this way we generate m non-dominated vectors as close as possible to m non-dominated marginal solutions, one for each criterion.

Step 2. Save problem $P_0(V,y^1,\ldots,y^m,c^1,\ldots,c^m)$ in the list (L) of problems prepared to be solved further. The argument V is the value of the generalized volume of the $m-$simplex defined by the vectors $y^1,\ldots,y^m$. (Details about computing the generalized volume are given in Section 3.3.)

Step 3. Set the maximum number of iterations *maxit*.

Step 4. For $it = \overline{1, maxit}$ do:
If L is not empty, then
(a) Load P_0, the first problem from the list L, and compute the corresponding parameters p_0 and q_0.
(b) For $i = \overline{2, m}$ do:
For each distinct combination of i out of m indexes, collected in the set M_i, do:
- Set $y^0 = \left(\sum_{k \in M_i} y^k\right)/i$ and $c^0 = \left(\sum_{k \in M_i} c^k\right)/i$.
- Use Model (5) to obtain a new non-dominated vector y^{new}.
- If $y^{new} \notin G$, then
  - add y^{new} to G; and
  - add m new problems to L, as it follows. The j-th new problem, $j \in M$, is obtained from Problem P_0 loaded from L, replacing the vector y^j by y^{new}, c^j by c^0, and computing the generalized volume V of the new m-simplex.

(c) Remove the first problem from the list L;
(d) Order the list L with respect to the generalized volume V associated with each problem.
Otherwise, break.

Each new non-dominated vector y^{new} is obtained as a solution to an optimization problem that is defined with respect to a hypothetical bound on the values of the objective functions y^0 for which no x^0 such that $y^0 = F\left(x^0\right)$ is specified. The presence of x^0 is not essential from a theoretical point of view, since our algorithm is able to provide a diversified coverage of the PF. On the other hand, the presence of x^0 may improve the running time of our algorithm for large scale problems, if it is used as starting solution in the optimization step. That is why we introduce a pseudo-randomization strategy that randomly generates feasible solutions x^0 that are used in solving Model (5) to obtain a new non-dominated vector y^{new} from the y^0 that is equal to $F(x^0)$.

The new procedure (randomized PESA – rPESA) is summarized below.

Step 1. Set the desired number of non-dominated vectorsnr.
Step 2. For $it = \overline{1, nr}$ do:
- Generate randomly a feasible solution x^0 and compute the corresponding $y^0 = F\left(x^0\right)$.
- Use y^0 in Eq. (5) to obtain the non-dominated vector y^{new}. Add y^{new} to the list of generated non-dominated vectors.

The use of rPESA instead of PESA is also recommended when the multiple objective optimization problem is degenerate in the sense of having fewer marginal efficient solutions than the number of the objectives.

In what follows we analyze the computational complexity of PESA and rPESA. We use k to denote the maximum number of iterations; $s(n)$ for the computational complexity of the algorithm for solving non-linear optimization problem with n decision variables; p for the number of constraints in the original problems; and m for the number of objective functions. The initial phase of PESA needs at most m^2 calls of the optimization algorithm yielding the complexity $m^2 s(n)$. The second step gives $O(m^3)$ complexity for computing the determinant involved in the computation of the generalized volume V. Step 4 has complexity

$$O(k \cdot (n + 2^m(m + s(n) + m \cdot (m + \log(k \cdot m))))). \tag{10}$$

In further detail, in Step 4, k corresponds to the main loop, n corresponds to the computation of parameters p_0 and q_0, and 2^m corresponds to the second loop (in sub-step (b)) that comes from computing all combinations of m points. Inside the second loop we set y^0 and c^0 with complexity $O(m)$, perform one optimization with complexity $s(n)$, and insert m problems in a list of maximum $2^m k$ elements. Insertion in the heap will then cost $O\left(\log\left(2^m \cdot k \cdot m^3\right)\right) = O(m \cdot \log(2) + \log(k) + 3 \cdot \log(m))$. One insertion will therefore cost $O(m + \log(k) + \log(m))$. Computing inside the expression given in Eq. (10), i.e. replacing $m + m^2 + m \cdot \log(m)$ by m^2 and $n + 2^m s(n)$ by $2^m s(n)$, we obtain the complexity of PESA to be $O\left(k \cdot 2^m\left(m^2 + m \cdot \log(k) + s(n)\right)\right)$. Considering k to be constant, the complexity of PESA is $O\left(2^m\left(m^2 + s(n)\right)\right)$.

The complexity of rPESA is thus $O(k \cdot (n \cdot p + s(n) + \log(k)))$, since each iteration generates a feasible point (complexity equal to $n \cdot p$), solves one optimization problem (complexity $s(n)$), and inserts the obtained non-dominated point in a list with at most k elements (complexity $\log(k)$). Considering again that k is constant we further conclude that the complexity of rPESA is $O(n \cdot p + s(n))$. Keeping in mind that the number of constraints is considerably less than the number of variables, we see that rPESA has the same complexity as the optimization algorithm used to solve the single objective optimization problems.

5. Numerical results

In this section, ten problems are selected from the literature to test the proposed algorithm, including bi-, 3-, 5- and 10-objective test instances with convex, non-convex, disconnected or continuous PF in our experiments.

The experiments were conducted on an Intel® Core™ i5 at 1.60GHz and 7.5GB RAM. All the codes are written in Octave version 4.2.2, configured for "x86_64-pc-linux-gnu". For optimization we used Octave's *sqp* global optimization function.

If not stated otherwise, for each instance solved by rPESA the reported results include the mean value and standard deviation obtained from 30 runs of the algorithm.

We use two performance metrics whose values are able to describe both the convergence of the approach and diversity of the obtained approximations: the inverted generational distance and the hyper-volume, as defined in Section 5.1.

5.1. Performance metrics

A wide variety of performance metrics are used in the literature to measure the convergence of approaches used or the diversity of the generated approximations compared to a set of reference points from the source sets. To measure the performance of our approach in finding a pattern efficient approximation to the Pareto front we use two performance metrics: the inverted generational distance (IGD), and the hyper-volume (HV). As observed in [3], the quality indicators IGD and HV can be used to measure both the spread of the generated non-dominated vectors and the convergence of the approximation to the true Pareto-front.

Given a set of uniformly distributed points along the Pareto front (P^*) and a set approximating the Pareto front (A), $IGD(A, P^*)$ represents the average distance from P^* to A, and is defined as:

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

where $d(v, A)$ is the minimum Euclidean distance between v and the points in A. A small value of $IGD(A, P^*)$ can be reached only when the set A is very close to the PF and does not miss any part of the whole PF.

The hyper-volume calculates the volume dominated by the solution set A with respect to a reference point. Generally, the reference point is derived from P^* by collecting the worst values of the objective functions over the points in P^*.

5.2. Reference Pareto fronts and algorithms used for comparison

The non-dominated vectors belonging to the reference Pareto fronts of the test instances were obtained using the JMetal framework for multi-objective optimization with meta-heuristics [7] and can be found at the web address "http://jmetal.sourceforge.net/problems.html".

The algorithm NSGA-II [5], and a few variants of MOEA/D [22] are used to provide the following comparisons: MOEA/D-DE, MOEA/D-UD1, MOEA/D-AWA, and MOEA/D-UD2. We did not run those algorithms on our computer, but used the numerical results reported in [24], which were obtained by the software Matlab in a personal computer with Intel Core i5, 2.30 GHz CPU processor and 3 GB RAM on Windows XP OS.

As previously noted, the methodology introduced in this paper theoretically yields non-dominated vectors. However, in practice, during experiments on non-convex instances, in some cases the optimization solver became trapped in a locally Pareto optimal solution and failed to provide a global optimum. This weakness of the solver we relied on (which is shared by other non-convex optimization solvers currently available) can be noticed in some of our graphic representations and also in some of the reported numerical results. Despite this, our overall numerical outcomes confirm the soundness of the underlying theory, and indicate that our approach performs quite well compared to other algorithms from the literature on the chosen test problems.

5.3. Experimental results for the test functions with continuous PF

The selected test functions with continuous PF contain ZDT1, ZDT2 from [25], UF1, UF7, UF8 from the multi-objective test suit for the CEC 2009 special session and competition [23], and DTLZ2 from [6]. We solved ZDT1, ZDT2, UF1, UF7 and UF8 using PESA, and DTLZ2 using both PESA and rPESA. Details of these functions are shown below.

- ZDT1 ($m = 2,\ n = 30$):

$$\begin{aligned} &\min && f_1(x) = x_1, \\ &\min && f_2(x) = g(x) \cdot h(f_1(x), g(x)), \\ &\text{where} && g(x) = 1 + \tfrac{9}{29} \textstyle\sum_{i=2}^{30} x_i, \\ & && h(f_1(x), g(x)) = 1 - \sqrt{\tfrac{f_1(x)}{g(x)}}, \\ & && x \in [0, 1]^n. \end{aligned}$$

- ZDT2 ($m = 2,\ n = 30$):

$$\begin{aligned} &\min && f_1(x) = x_1, \\ &\min && f_2(x) = g(x) \cdot h(f_1(x), g(x)), \\ &\text{where} && g(x) = 1 + \tfrac{9}{29} \textstyle\sum_{i=2}^{30} x_i, \\ & && h(f_1(x), g(x)) = 1 - \left(\tfrac{f_1(x)}{g(x)}\right)^2, \\ & && x \in [0, 1]^n. \end{aligned}$$

- UF1 ($m = 2,\ n = 30$):

$$\begin{aligned}
&\min \quad f_1(x) = x_1 + \frac{2}{|J_1|}\sum_{j\in J_1} y_j^2,\\
&\min \quad f_2(x) = 1 - \sqrt{x_1} + \frac{2}{|J_2|}\sum_{j\in J_2} y_j^2,\\
&\text{where} \quad J_1 = \{j | j \text{ is odd and } 2 \le j \le n\},\\
&\qquad\quad J_2 = \{j | j \text{ is even and } 2 \le j \le n\}\\
&\qquad\quad y_j = x_j - \sin\left(6\pi x_1 + \tfrac{j\pi}{n}\right), 2 \le j \le n,\\
&\qquad\quad x \in [0,1] \times [-1,1]^{n-1}.
\end{aligned}$$

- UF7 ($m = 2,\ n = 3$) and ($m = 2,\ n = 30$):

$$\begin{aligned}
&\min \quad f_1(x) = \sqrt[5]{x_1} + \frac{2}{|J_1|}\sum_{j\in J_1} y_j^2,\\
&\min \quad f_2(x) = 1 - \sqrt[5]{x_1} + \frac{2}{|J_2|}\sum_{j\in J_2} y_j^2,\\
&\text{where} \quad J_1 = \{j | j \text{ is odd and } 2 \le j \le n\},\\
&\qquad\quad J_2 = \{j | j \text{ is even and } 2 \le j \le n\}\\
&\qquad\quad y_j = x_j - \sin\left(6\pi x_1 + \tfrac{j\pi}{n}\right), 2 \le j \le n,\\
&\qquad\quad x \in [0,1] \times [-1,1]^{n-1}.
\end{aligned}$$

- UF8 ($m = 3,\ n = 30$):

$$\begin{aligned}
&\min \quad f_1(x) = \cos(0.5x_1\pi)\cos(0.5x_2\pi) + \frac{2}{|J_1|}\sum_{j\in J_1} y_j^2,\\
&\min \quad f_2(x) = \cos(0.5x_1\pi)\sin(0.5x_2\pi) + \frac{2}{|J_2|}\sum_{j\in J_2} y_j^2,\\
&\min \quad f_3(x) = \sin(0.5x_1\pi) + \frac{2}{|J_3|}\sum_{j\in J_3} y_j^2,\\
&\text{where} \quad J_1 = \{j | j-1 \text{ is multiple of 3, and } 3 \le j \le n\},\\
&\qquad\quad J_2 = \{j | j-2 \text{ is multiple of 3, and } 3 \le j \le n\},\\
&\qquad\quad J_3 = \{j | j \text{ is multiple of 3, and } 3 \le j \le n\},\\
&\qquad\quad y_j = x_j - 2x_2\sin\left(2\pi x_1 + \tfrac{j\pi}{n}\right), 3 \le j \le n,\\
&\qquad\quad x \in [0,1]^2 \times [-1,1]^{n-2}.
\end{aligned}$$

- DTLZ2 ($m = 3,\ n = 3$), ($m = 3,\ n = 12$), ($m = 5,\ n = 14$) and ($m = 10,\ n = 19$):

$$\begin{aligned}
&\min \quad f_1(x) = (1 + g(x_M))\cos\left(\tfrac{x_1\pi}{2}\right)\ldots\cos\left(\tfrac{x_{m-2}\pi}{2}\right)\cos\left(\tfrac{x_{m-1}\pi}{2}\right),\\
&\min \quad f_2(x) = (1 + g(x_M))\cos\left(\tfrac{x_1\pi}{2}\right)\ldots\cos\left(\tfrac{x_{m-2}\pi}{2}\right)\sin\left(\tfrac{x_{m-1}\pi}{2}\right),\\
&\min \quad f_2(x) = (1 + g(x_M))\cos\left(\tfrac{x_1\pi}{2}\right)\ldots\sin\left(\tfrac{x_{m-2}\pi}{2}\right),\\
&\vdots \qquad\quad \vdots\\
&\min \quad f_3(x) = (1 + g(x_M))\sin\left(\tfrac{x_1\pi}{2}\right),\\
&\text{where} \quad g(x_M) = \sum_{i=m}^{n}(x_i - 0.5)^2, x_M = (x_m, \ldots, x_n),\\
&\qquad\quad x \in [0,1]^n.
\end{aligned}$$

The generated approximations for the Pareto fronts of the test instances ZDT1, ZDT2, UF1, UF7 and UF8 are shown in Figs. 3 and 4. Table 1 presents the IGD-metric values of the compared algorithms on the same test instances.

The numerical results for DTLZ2 with $m = 3$ and $n = 12$ are graphically represented in Fig. 5. The PF approximations were obtained by running both PESA and rPESA.

Fig. 6 shows the parallel coordinate plots of the PF approximations obtained by running PESA on 5-objective DTLZ2, rPESA on 5-objective DTLZ2, and rPESA on 10-objective DTLZ2. In order to make the plots clearer, the generated vectors that do not belong to the PF (obtained when the solver failed to provide a global optimal solution) were removed from the representation.

5.4. Experimental results for the test functions with discontinuous PF

The selected test functions with discontinuous PF contain TNK from [20], DTLZ7 from [6] and ZDT3 from [25]. We solved TNK and the bi-objective DTLZ7 using PESA, and the 3-objective DTLZ7 using rPESA. The details of these functions are shown below.

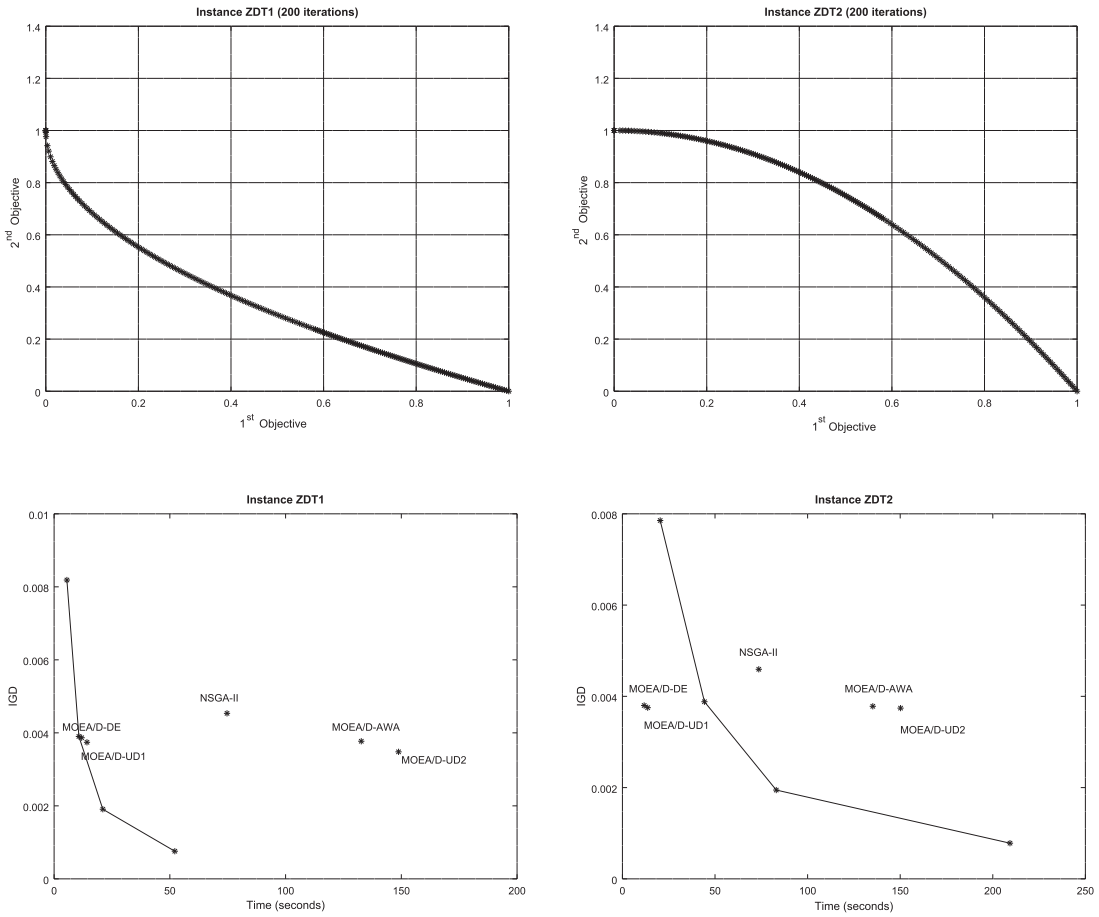

Fig. 3. Generated Pareto frontiers obtained by running PESA on ZDT1 and ZDT2 respectively (at the top); and the metric values IGD with respect to time for the same instances (on the bottom). The values related to the algorithms MOEA/D-DE, MOEA/D-UD1, NSGA-II, MOEA/D-AWA, and MOEA/D-UD2 are obtained from [24].

Table 1
IGD values for ZDT1, ZDT2, UF1, UF7, UF8 and DTLZ2 ($m = 3,\ n = 12$) obtained by various algorithms. For all algorithms except PESA, the reported values are obtained form [24].

Algorithm		ZDT1	ZDT2	UF1	UF7	UF8	DTLZ2
MOEA/D-DE	IGD	3.866e-3	3.800e-3	7.094e-2	5.324e-2	1.449e-1	3.762e-2
	Time	11.9	11.7	19.4	18.8	35.3	20.2
MOEA/D-UD1	IGD	3.741e-3	3.752e-3	6.815e-2	4.847e-2	1.373e-1	3.562e-2
	Time	14.2	13.6	20.6	20.7	55.3	35.2
NSGA-II	IGD	4.5343e-3	4.593e-3	6.626e-2	4.341e-2	1.707e-1	4.310e-2
	Time	74.7	73.6	90.9	89.5	396.5	374.1
MOEA/D-AWA	IGD	3.770e-3	3.782e-3	6.819e-2	4.357e-2	1.221e-1	3.384e-2
	Time	132.7	135.2	161.9	145.0	721.7	619.0
MOEA/D-UD2	IGD	3.480e-3	3.742e-3	6.620e-2	4.338e-2	1.201e-1	3.275e-2
	Time	148.8	150.2	173.3	156.5	733.6	643.8
PESA	IGD	1.907e-3	7.817e-4	3.899e-2	4.089e-2	1.171e-1	7.063e-1
	Time	20.3	46.4	352.5	869.7	655.1	274.4

- TNK ($m = 2,\ n = 2$):

$$
\begin{aligned}
\min \quad & f_1(x) = x_1, \\
\min \quad & f_2(x) = x_2, \\
\text{s.t.} \quad & -x_1^2 - x_2^2 + 1 + 0.1\cos\left(16\arctan\left(\tfrac{x_1}{x_2}\right)\right) \leq 0, \\
& (x_1 - 0.5)^2 + (x_2 - 0.5)^2 - 0.5 \leq 0, \\
& x \in [0, \pi]^2.
\end{aligned}
$$

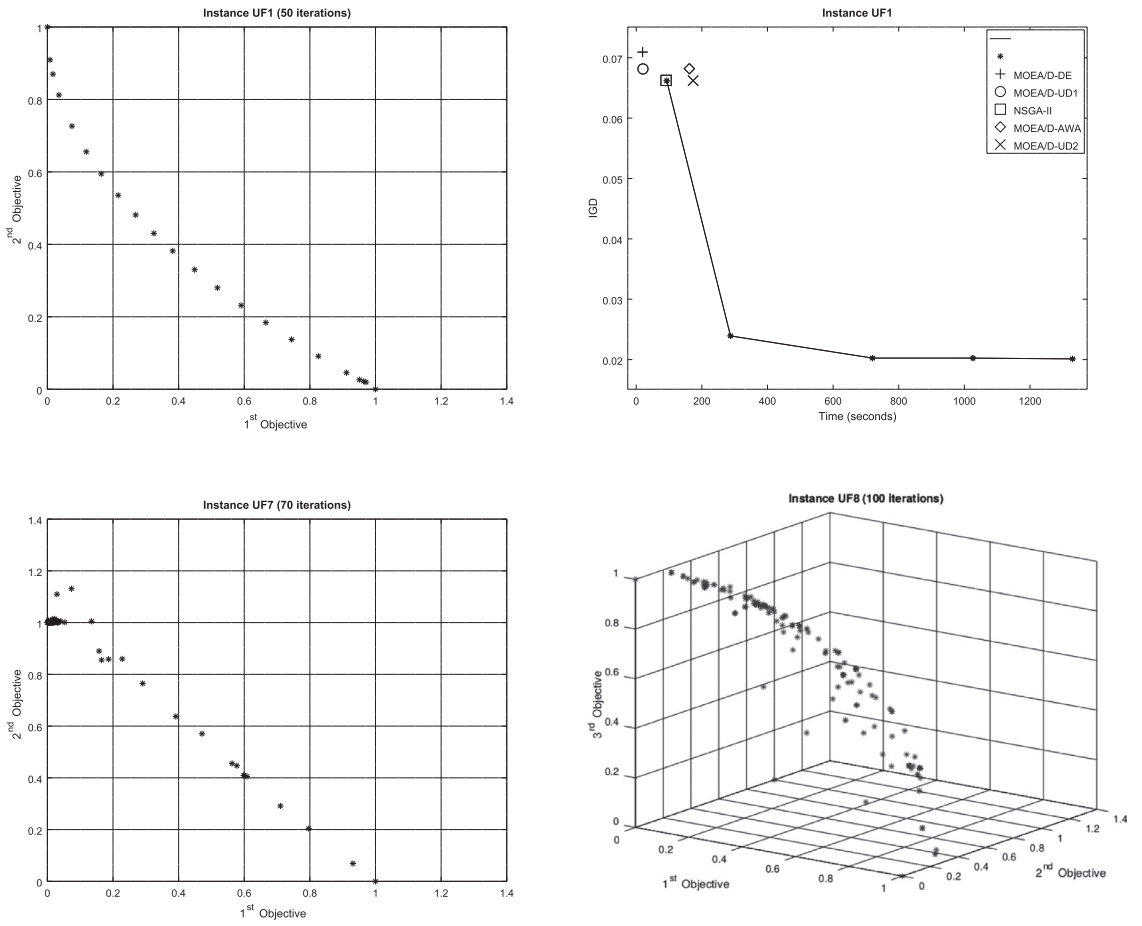

Fig. 4. Generated Pareto frontiers obtained by running PESA on UF1, UF7 and UF8 respectively. The metric values IGD with respect to time for UF1 are also shown. The values related to the algorithms MOEA/D-DE, MOEA/D-UD1, NSGA-II, MOEA/D-AWA, and MOEA/D-UD2 are obtained from [24].

- DTLZ7 $(m = 2,\ n = 2)$ and $(m = 3,\ n = 3)$:

$$
\begin{aligned}
&\min \quad f_1(x) = x_1,\\
&\min \quad f_2(x) = x_2,\\
&\vdots \quad\quad \vdots\\
&\min \quad f_{m-1}(x) = x_{m-1},\\
&\min \quad f_m(x) = (1 + g(x_M))h(f_1, \ldots, f_{m-1}, g),\\
&\text{where} \quad g(x_M) = 1 + \frac{9}{|x_M|}\sum_{i=m}^{n} x_i,\\
&\qquad\quad h(f_1, \ldots, f_{m-1}, g) = m - \sum_{i=1}^{m-1}\left(\frac{f_i(1+\sin(3\pi f_i))}{1+g}\right),\\
&\qquad\quad x_M = (x_m, \ldots, x_n),\\
&\qquad\quad x \in [0,1]^n.
\end{aligned}
$$

- ZDT3 $(m = 2,\ n = 30)$:

$$
\begin{aligned}
&\min \quad f_1(x) = x_1,\\
&\min \quad f_2(x) = g(x) \cdot h(f_1(x), g(x)),\\
&\text{where} \quad g(x) = 1 + \tfrac{9}{29}\textstyle\sum_{i=2}^{30} x_i,\\
&\qquad\quad h(f_1(x), g(x)) = 1 - \sqrt{\tfrac{f_1(x)}{g(x)}} - \left(\tfrac{f_1(x)}{g(x)}\right)\sin(10\pi f_1),\\
&\qquad\quad x \in [0,1]^{30}.
\end{aligned}
$$

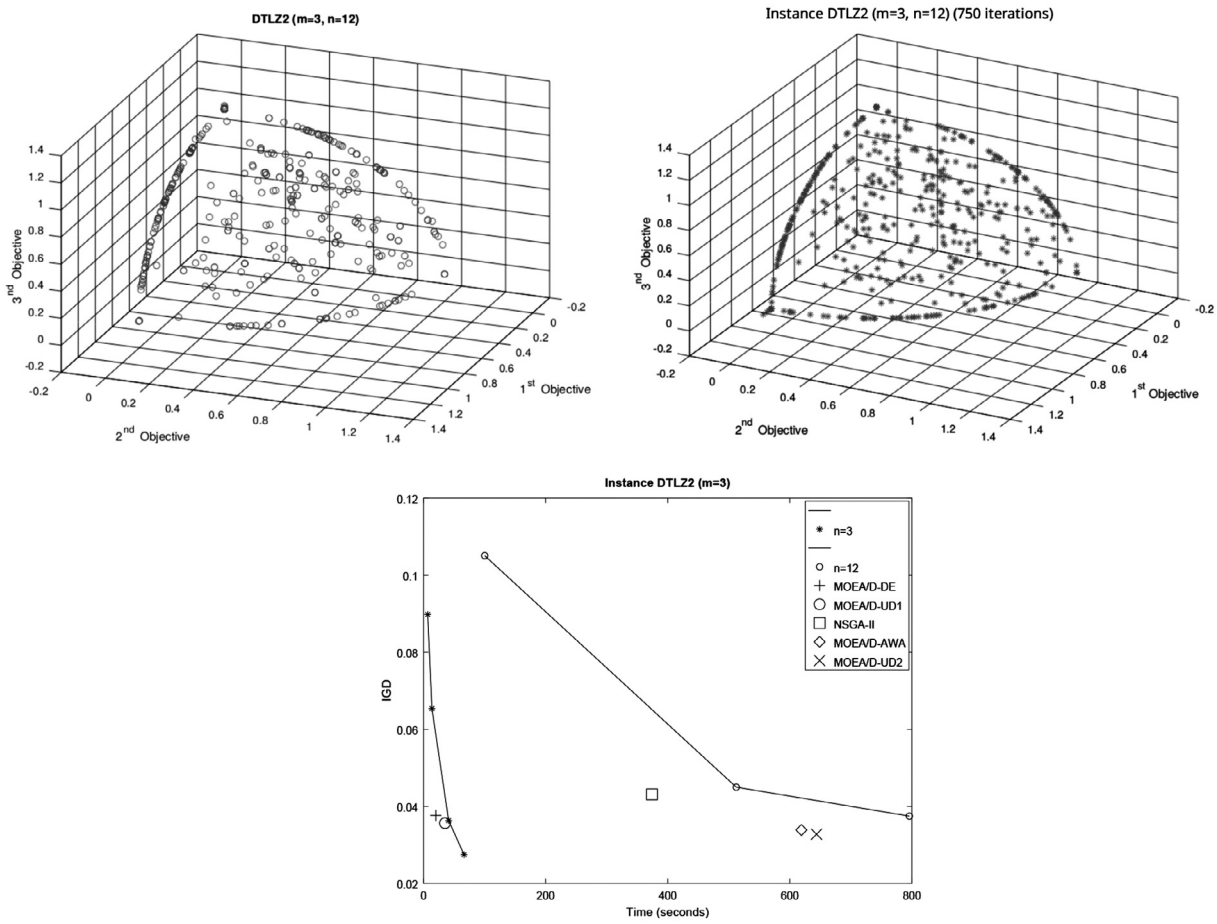

Fig. 5. Generated non-dominated vectors obtained by running PESA (to the left, in 131 seconds), and rPESA (to the right, in 102 seconds) respectively on DTLZ2 ($m = 3,\ n = 12$). The metric values IGD and HV with respect to time are also included. The values related to the algorithms MOEA/D-DE, MOEA/D-UD1, NSGA-II, MOEA/D-AWA, and MOEA/D-UD2 are obtained from [24].

Table 2
Influence of the multipliers used for computing p_0 and q_0 on the IGD values obtained for running PESA (500 iterations) on DTLZ7 ($m = 2, n = 2$).

IGD	0.05	0.1	0.2	0.5	1
5	4.5584e-02	1.9383e-02	2.1171e-02	2.1069e-02	3.0131e-02
10	4.6259e-02	4.6245e-02	4.6209e-02	4.6100e-02	4.5908e-02
20	2.8980e-02	2.8971e-02	2.8995e-02	15.7740e-02	15.7520e-02
50	3.8491e-02	3.8490e-02	3.8487e-02	3.4309e-02	3.7246e-02
75	6.8152e-02	6.8140e-02	6.8126e-02	6.8086e-02	3.2610e-02
100	3.4546e-02	4.3876e-02	4.3873e-02	4.3869e-02	4.3854e-02

The graphic representations of the generated sets are presented in Figs. 7 and 8 giving the non-dominated vectors. These figures show a good spread of the obtained non-dominated vectors over the true Pareto front.

The IGD and HV values for the obtained approximations are reported in Tables 4, 5 and 6. Since no reference set of non-dominated vectors for TNK is available in the literature, the IGD and HV values for this instance are omitted.

Some experiments on the influence of the parameters p_0 and q_0 on IGD and HV values obtained for running PESA are shown in Tables 2 and 3. The values of the multipliers *multp* (used for computing p_0) and *multq* (used for computing q_0) for the instance DTLZ7 ($m = 2, n = 2$) are listed in the first column and first row respectively. A graphical representation of the same experiments can be seen in Fig. 9: IGD values, HV values, and the number of generated non-dominated points are

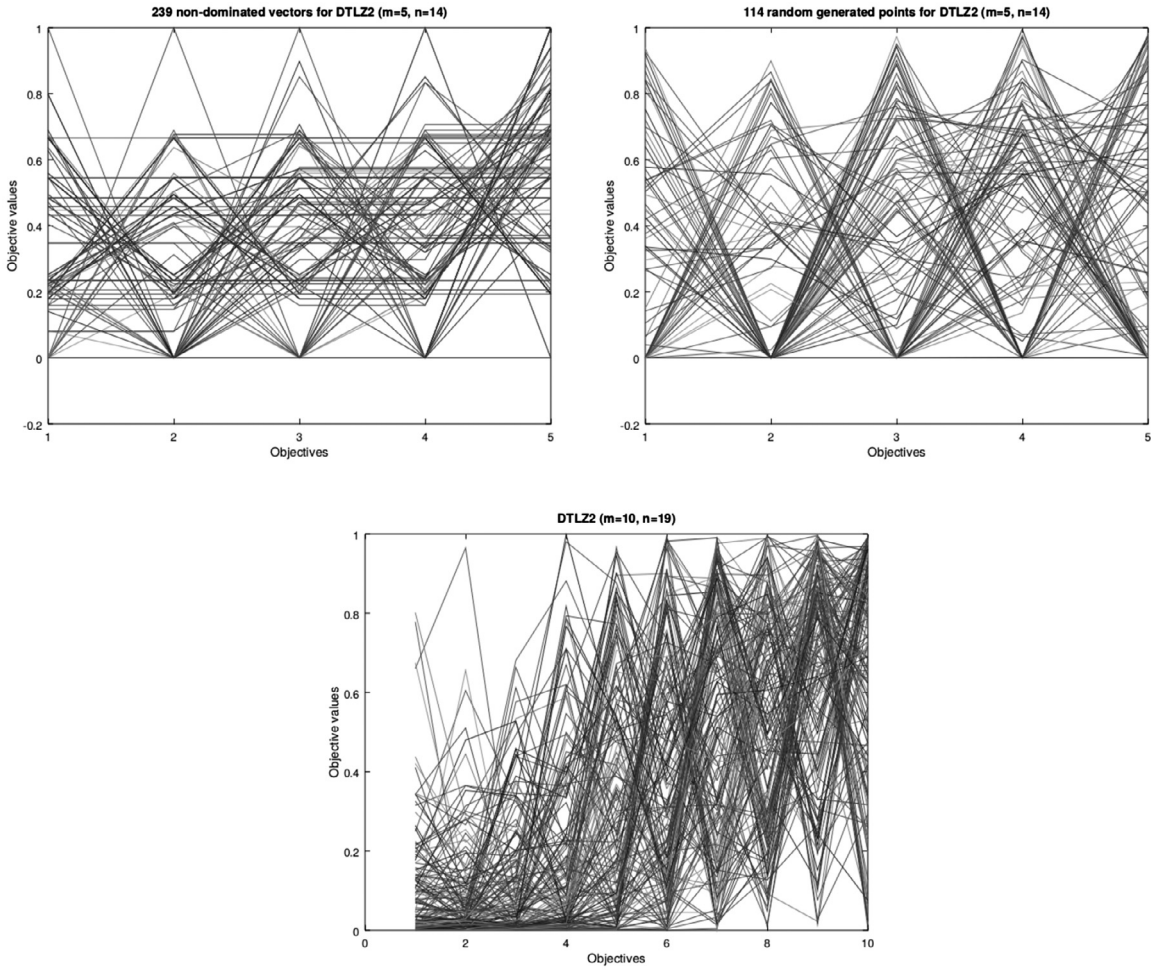

Fig. 6. Parallel coordinates plots for 239 non-dominated vectors obtained by running PESA (151 seconds) on DTLZ2 ($m = 5,\ n = 14$), 114 non-dominated vectors obtained by running rPESA (502 seconds) on DTLZ2 ($m = 5,\ n = 14$), and 165 non-dominated vectors obtained by running rPESA (226 seconds) on DTLZ2 ($m = 10,\ n = 19$). For similar representations we refer the reader to [2].

Table 3
Influence of the multipliers used for computing p_0 and q_0 on the HV values obtained for running PESA (500 iterations) on DTLZ7 ($m = 2, n = 2$).

HV	0.05	0.1	0.2	0.5	1
5	5.4156e-01	5.5378e-01	5.5216e-01	5.5226e-01	5.4846e-01
10	5.4126e-01	5.4126e-01	5.4128e-01	5.4132e-01	5.4140e-01
20	5.4785e-01	5.4785e-01	5.4785e-01	4.8034e-01	4.8042e-01
50	5.4272e-01	5.4272e-01	5.4273e-01	5.4453e-01	5.4313e-01
75	5.3000e-01	5.3001e-01	5.3001e-01	5.3003e-01	5.4529e-01
100	5.4446e-01	5.4003e-01	5.4003e-01	5.4003e-01	5.4004e-01

represented separately for four different values of the multiplier *multq*. Since the values of parameters p_0 and q_0 depend on the vector of coefficients c that varies from iteration to iteration, we provided the values for the multipliers *mult*p and *multq* that were kept constant during each run of PESA.

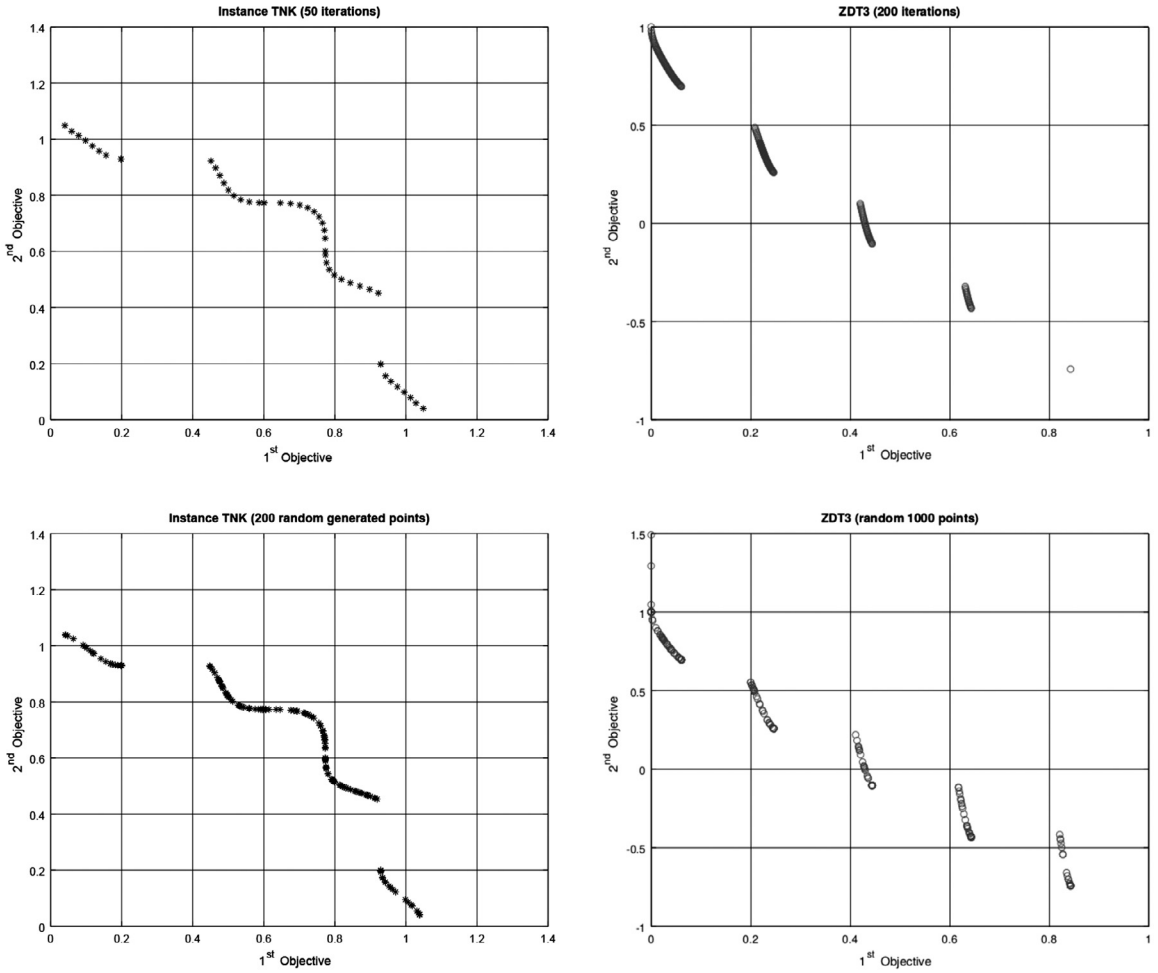

Fig. 7. Generated Pareto frontiers for Instances TNK and ZDT3 obtained by running PESA (top) and rPESA (bottom) respectively.

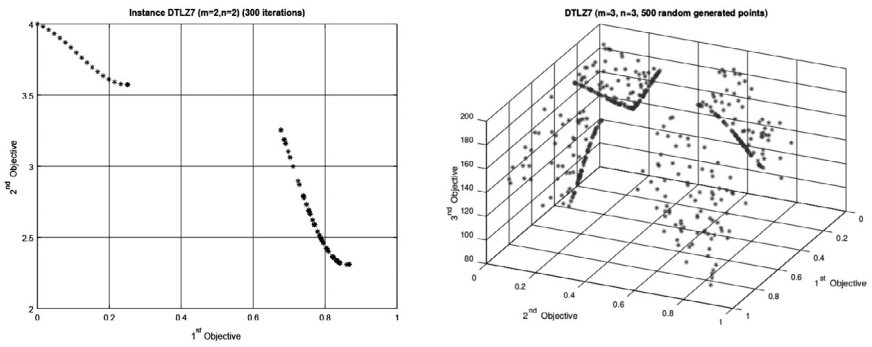

Fig. 8. Generated Pareto frontiers for DTLZ7 ($m = 2$, $n = 2$) obtained by running PESA, and for DTLZ7 ($m = 3$, $n = 3$) obtained by running rPESA.

Table 4
Numerical results obtained by running PESA. IGD and HV values are missing for those instances for which no set of reference points is available in the literature. The quality of the obtained solutions can be concluded from their graphical representations.

Instance	IGD	HV	Time (s)	# points	# *sqp* calls
TNK	-	-	7.2	189	200
DTLZ2 ($m = 3, n = 12$)	7.063e–1	1.3243e+0	274.4	414	600
DTLZ2 ($m = 5, n = 14$)	–	–	151.0	239	250
DTLZ7 ($m = 2, n = 2$)	4.389e–2	5.4003e–1	41.7	1000	1000
UF1 ($m = 2, n = 30$)	1.173e–2	8.6174e–1	254.3	50	50
UF7 ($m = 2, n = 3$)	5.384e–3	7.0232e–1	28.9	70	70
UF7 ($m = 2, n = 30$)	4.089e–2	6.3948e–1	869.7	70	70
UF8 ($m = 3, n = 30$)	8.562e–2	6.9051e–1	3345.1	100	100
ZDT1 ($m = 2, n = 30$)	1.907e–3	8.7411e–1	20.3	200	200
ZDT2 ($m = 2, n = 30$)	7.818e–4	5.4229e–1	46.4	500	500
ZDT3 ($m = 2, n = 30$)	2.837e–2	1.0002e+0	178.1	200	200
Water Resource Management Problem	1.997e+5	6.3625e+24	9.9	1510	2000

Table 5
Best results obtained by running rPESA. IGD and HV values are missing for those instances for which no set of reference points is available in the literature. The quality of the obtained solutions can be concluded from their graphical representations.

Instance	IGD	HV	Time (s)	# points
TNK	-	-	6.6	200
DTLZ2 ($m = 3, n = 3$)	2.723e–2	7.6161e–1	2.536e+1	500
DTLZ2 ($m = 3, n = 12$)	2.340e–2	7.8519e–1	3.454e+2	1500
DTLZ2 ($m = 5, n = 14$)	–	–	1.020e+2	114
DTLZ2 ($m = 10, n = 19$)	–	–	5.004e+2	165
DTLZ7 ($m = 3, n = 3$)	4.275e–2	1.6732e+0	1.900e+1	500
ZDT3 ($m = 2, n = 30$)	9.0572e–3	1.0085e+0	8.935e+3	1650
Water Resource Management Problem	3.367e+3	7.9152e+24	1.291e+2	2000

Table 6
Mean values and standard deviations obtained by running rPESA.

Instance		IGD	HV	Time (s)	# points
DTLZ2 ($m = 3, n = 3$)	mean	2.770e–2	7.6502e–1	2.183e+1	500
	std.dev.	3.451e–4	2.8847e–3	2.337e+0	
DTLZ2 ($m = 3, n = 12$)	mean	3.748e–2	7.7388e–1	1.257e+2	750
	std.dev.	1.725e–3	1.0095e–3	3.559e+0	
DTLZ7 ($m = 3, n = 3$)	mean	4.593e–2	1.6714e+0	1.903e+1	500
	std.dev.	3.030e–3	7.1188e–3	4.203e–1	
ZDT3 ($m = 2, n = 30$)	mean	1.2366e-2	1.0068e+0	1.837e+3	1000
	std.dev.	9.416e-4	5.5287e-4	7.092e+1	
Water Resource	mean	3.657e+3	7.9104e+24	4.355e+1	2000
Management Problem	std.dev.	3.101e+2	6.0521e+21	6.415e–1	

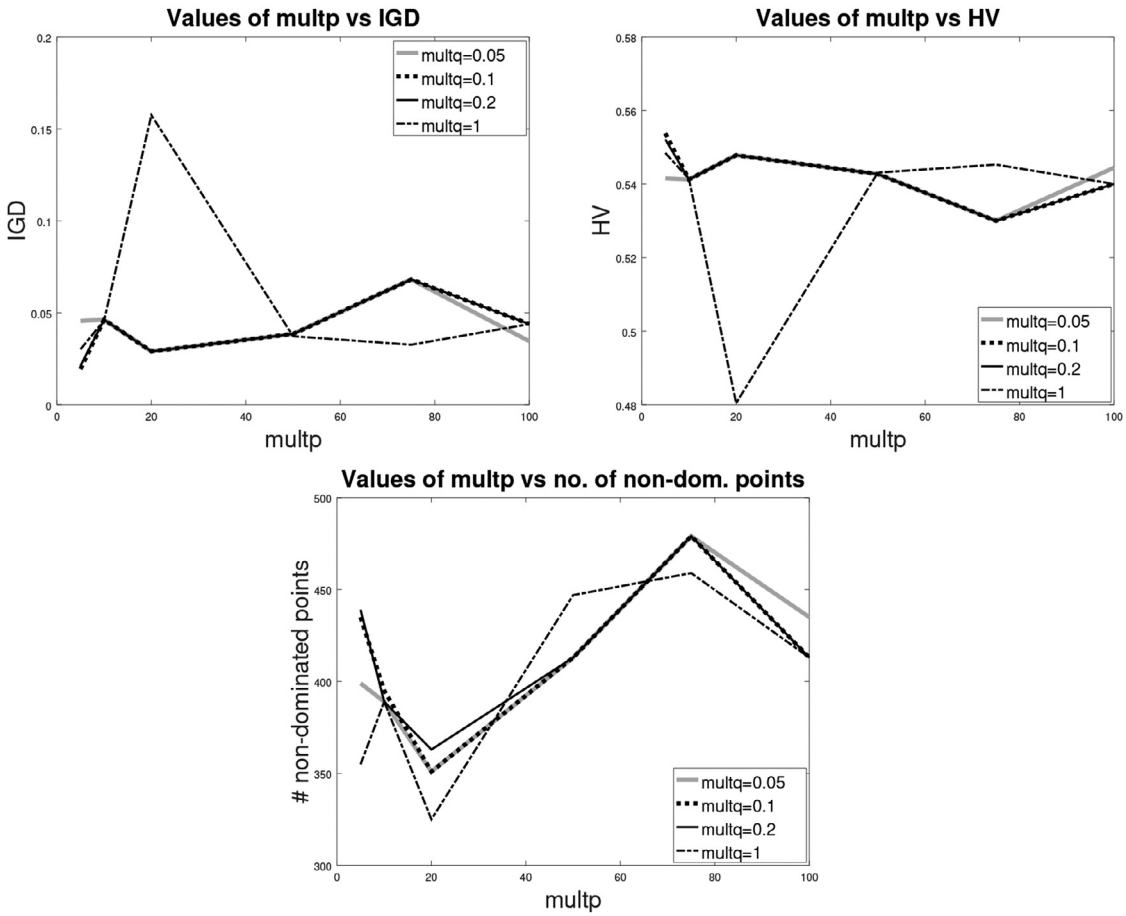

Fig. 9. Graphical representation of the multiplier *multp* versus IGD values, HV values, and the number of generated non-dominated points obtained for running PESA (500 iterations) on DTLZ7 ($m = 2, n = 2$).

5.5. Experimental results for the water resource management problem

The Water Resource Management Problem descibed by Eq. (11) with five objective functions and three variables is introduced in [16] and recalled in [1].

$$
\begin{aligned}
\min \quad & f_1(x_1,x_2,x_3) = 106780.37(x_2+x_3) + 61704.67,\\
\min \quad & f_2(x_1,x_2,x_3) = 3000x_1,\\
\min \quad & f_3(x_1,x_2,x_3) = \frac{(305700)2289x_2}{[(0.06)2289]^{0.65}},\\
\min \quad & f_4(x_1,x_2,x_3) = (250)2289e^{-39.75x_2+9.9x_3+2.74},\\
\min \quad & f_5(x_1,x_2,x_3) = 25\left(\frac{1.39}{x_1x_2} + 4940x_3 - 80\right),\\
\text{s.t.} \quad & g_1(x_1,x_2,x_3) = \frac{0.00139}{x_1x_2} + 4.94x_3 - 0.08 \le 1,\\
& g_2(x_1,x_2,x_3) = \frac{0.000306}{x_1x_2} + 1.082x_3 - 0.0986 \le 1,\\
& g_3(x_1,x_2,x_3) = \frac{12.307}{x_1x_2} + 49408.24x_3 + 4051.02 \le 50000,\\
& g_4(x_1,x_2,x_3) = \frac{2.098}{x_1x_2} + 8046.33x_3 - 696.71 \le 16000,\\
& g_5(x_1,x_2,x_3) = \frac{2.138}{x_1x_2} + 7883.39x_3 - 705.04 \le 10000,\\
& g_6(x_1,x_2,x_3) = \frac{0.417}{x_1x_2} + 1721.26x_3 - 136.54 \le 2000,\\
& g_7(x_1,x_2,x_3) = \frac{0.164}{x_1x_2} + 631.13x_3 - 54.48 \le 550,\\
& 0.01 \le x_1 \le 0.45,\\
& 0.01 \le x_2 \le 0.10,\\
& 0.01 \le x_3 \le 0.10.
\end{aligned}
\tag{11}
$$

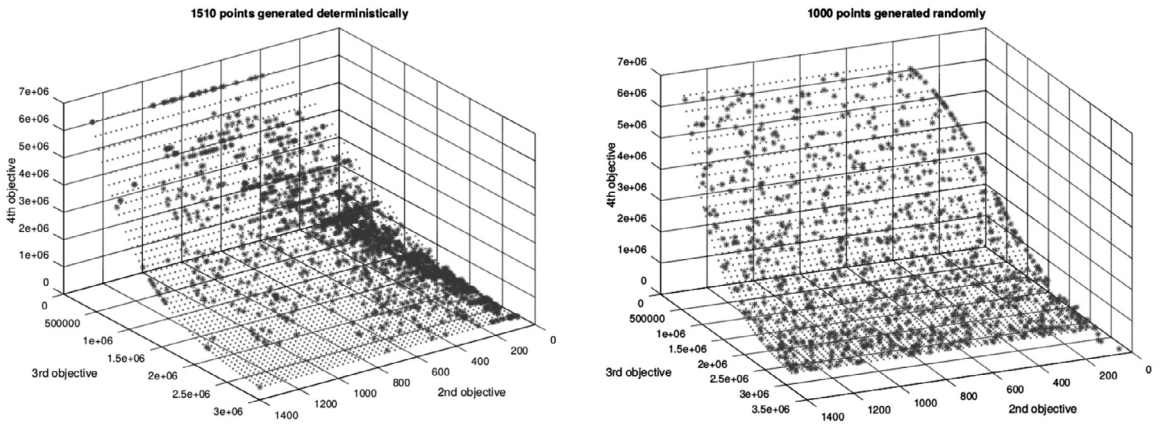

Fig. 10. The representation of the non-dominated vectors (1510 obtained deterministically using PESA, and 1000 obtained from randomly chosen starting points using rPESA) generated for Water Resource Management Problem (11), giving the values of the 2nd, 3rd and 4th objective functions. The reference points are shown using small dots (Â), while the vectors from the generated approximations are shown using stars (*).

Observing some particularities, we may eliminate one objective function (due to its correlation with another one) and one variable (due to monotonicity). In this way, all constraints acquire the same form. Thus, we focus on solving the following reduced problem (Eq. (12)) and extend the resulting solutions to provide solutions to the problem given in Eq. (11) in order to compare our results to those from the literature.

$$\begin{aligned}
&\min && f_1(x_1,x_2,x_3) = 3000x_1,\\
&\min && f_2(x_1,x_2,x_3) = 28534690\cdot x_2,\\
&\min && f_3(x_1,x_2,x_3) = 572250e^{-39.75x_2+2.839},\\
&\min && f_4(x_1,x_2,x_3) = \tfrac{34.75}{x_1x_2} - 765,\\
&\text{s.t.} && x_1x_2 \geq 0.0013487,\\
&&& 0.01 \leq x_1 \leq 0.45,\\
&&& 0.01 \leq x_2 \leq 0.10.
\end{aligned} \tag{12}$$

When 1000 iterations are allowed in PESA, the algorithm generates 1510 points and stops because the list of problems becomes empty. These points are presented together with the reference points found in the literature in Fig. 10 (to the left). Fig. 10 (to the right) represents 1000 non-dominated vectors generated using rPESA. Both representations give the values of the 2nd, 3rd and 4th objective functions at the generated efficient solutions.

In Fig. 11 we represent the parallel coordinate plots of 1510 non-dominated vectors generated using PESA, 1000 non-dominated vectors generated using rPESA, and 2429 reference non-dominated vectors obtained from the literature. We compute the values of the IGD and HV metrics to evaluate the generated approximations and report them in Tables 4, 5 and 6. It can be noticed that rPESA gets smaller IGD values and greater HV values in less time than PESA, due to a better performance of the optimization solver when a starting feasible solution is known.

In both Figs. 10 and 11 one can note that the upper values of the second objective function on PF are not quite well covered by the vectors from the approximation obtained from running PESA. The reason for this is the degeneracy encountered, in the sense that the problem has fewer marginal efficient solutions than the number of the objectives. On the other hand, the vectors from the approximation obtained by running rPESA have a good distribution along the PF.

5.6. Summary

To summarize our experimental results we report the following information: IGD and HV values to evaluate the quality of the approximations; time in seconds and number of generated non-dominated vectors to evaluate the effort needed to obtain the approximations; and references to the graphic representations included in the paper.

Tables 4 and 5 report the best results obtained by running PESA and rPESA respectively, while Table 6 reports the mean values and their standard deviations for IGD, HV, and the time consumed by running rPESA 30 times with fixed values for the parameters that were tuned separately for each instance.

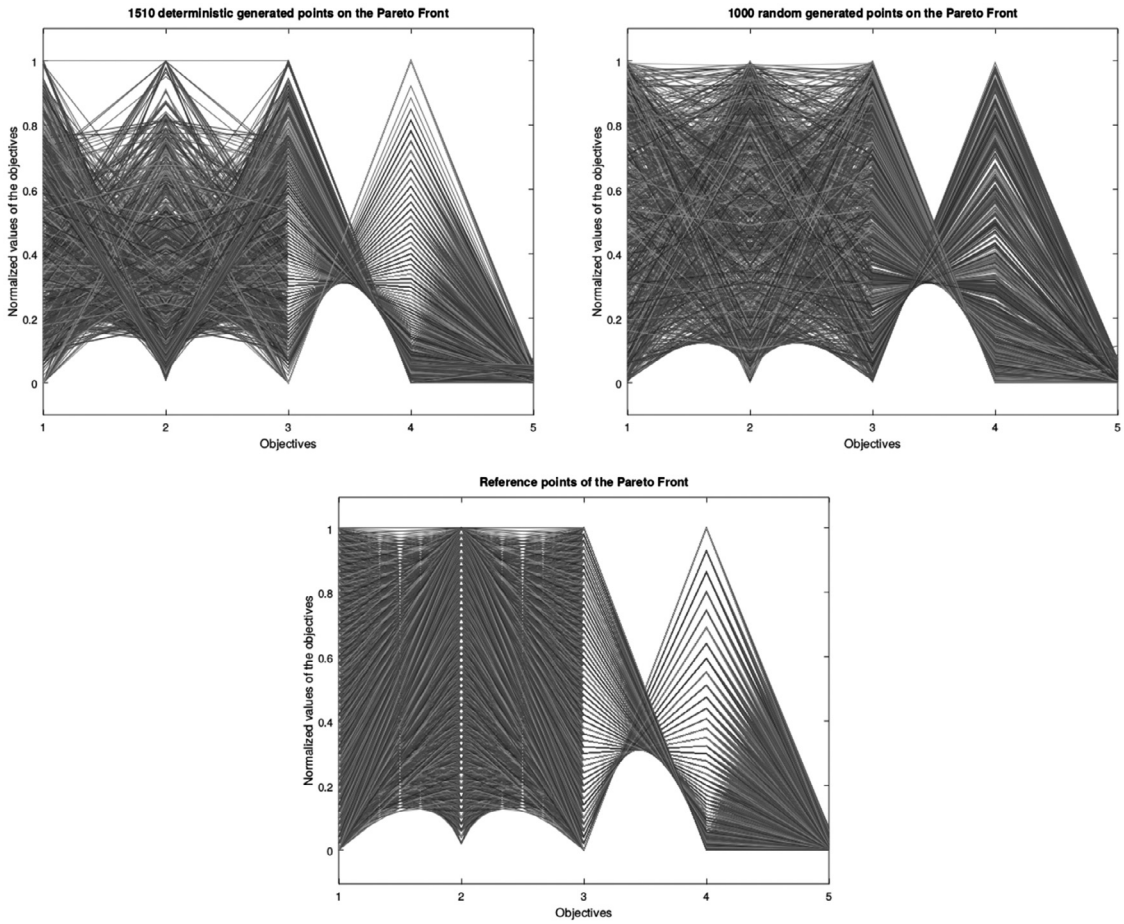

Fig. 11. Parallel coordinate plots for Water Resource Management Problem (11) (1510 non-dominated vectors obtained by PESA; 1000 non-dominated vectors obtained by rPESA; and 2429 reference vectors from the literature).

6. Concluding remarks

To carry out an effective multiple criteria decision analysis, it is essential to be able to populate the criterion space in a representative (pattern-efficient) manner, without producing knots or clumps of non-dominated vectors with wide gaps between them.

Our design for finding a pattern-efficient set of non-dominated vectors employs a single-objective optimization model to derive non-dominated vectors that can fill gaps between already generated non-dominated vectors.

We have observed that attempts to create a practical representation of an efficient frontier have often been founded on dubious assumptions and the claims made for their relevance can be highly misleading. Curves constructed by economists typically suppose that all relevant solutions are known, at least in principle (which is to say, the distribution of their objective function values can be inferred and graphed) - a supposition that is generally not valid.

Consequently, it becomes important to consider how to derive a formulation for multi-objective optimization that applies under more general conditions encountered in real world settings. The significance of this challenge is underscored by the fact that the classical structure of the set of the supported non-dominated vectors changes radically in the absence of convexity and loses its relevance.

The advanced formulation that we introduce has the useful property of permitting a strategy that iteratively discovers progressively diverse (pattern-efficient) solutions which can model the structure of the Pareto front. In this way, by its ability to generate non-supported non-dominated points, our approach overcomes limitations of approaches like the WSS method widely used in practice.

At the same time, we retain the computational simplicity of the WSS method, and additionally provide a methodology to tune the parameters that assures a pattern-efficient cover of the Pareto front.

In particular, we describe a way to deal with the adjacency of non-dominated vectors, quantifying the gaps between them, and proposing a solution algorithm (PESA) for generating these vectors. We also introduce a pseudo-randomized

variant of PESA (rPESA) that generates hypothetical bounds for the objective functions used in the optimization model. This approach improved the running time of our initial algorithm and reduced the number of failures of the global optimization solver.

Ten problems from the literature with diverse characteristics were chosen to test our proposed approach, including bi-objective, 3-objective, 5-objective and 10-objective test instances with convex, non-convex, disconnected or continuous Pareto fronts. The inverted generational distance and the hyper-volume performance metrics were used to describe the convergence of our approach and the diversity of the approximations obtained. Numerical results obtained from applying our method were also presented graphically.

Our overall experiments showed that our solution algorithm succeeds in covering the Pareto front of the instances obtained from the literature in a pattern-efficient manner, and confirm the soundness of our theoretical results. A refined tuning of the algorithm' s parameters and the use of a more effective global optimization solver as a subroutine may be expected to produce still better outcomes.

Declaration of Competing Interest

The authors declare that there is no conflict of interest.

CRediT authorship contribution statement

Bogdana Stanojević: Methodology, Software, Investigation, Validation, Writing - review & editing, Visualization. **Fred Glover:** Conceptualization, Methodology, Supervision, Writing - review & editing.

Acknowledgments

This research was partially supported by the Serbian Ministry of Education, Science and Technological Development through Mathematical Institute of the Serbian Academy of Sciences and Arts.

References

[1] M. Asafuddoula, T. Ray, R. Sarker, A decomposition based evolutionary algorithm for many objective optimization with systematic sampling and adaptive epsilon control, in: R.C. Purshouse, P.J. Fleming, C.M. Fonseca, S. Greco, J. Shaw (Eds.), Evolutionary Multi-Criterion Optimization, Springer, Berlin, Heidelberg, 2013, pp. 413–427.
[2] X. Cai, H. Sun, Z. Fan, A diversity indicator based on reference vectors for many-objective optimization, Inf. Sci. (Ny) 430–431 (2018) 467–486, doi:10.1016/j.ins.2017.11.051.
[3] S. Chand, M. Wagner, Evolutionary many-objective optimization: A quick-start guide, Surveys in Operations Research and Management Science 20 (2015) 35–42.
[4] I. Das, J. Dennis, A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems, Structural optimization 14 (1) (1997) 63–69, doi:10.1007/BF01197559.
[5] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-ii, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[6] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable Test Problems for Evolutionary Multiobjective Optimization, Springer, London, pp. 105–145.
[7] J. Durillo, A. Nebro, E. Alba, The jmetal framework for multi-objective optimization: Design and architecture, in: CEC 2010, Barcelona, Spain, 2010, pp. 4138–4325.
[8] M. Ehrgott, Multicriteria optimization, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
[9] I. Giagkiozis, P. Fleming, Methods for multi-objective optimization: an analysis, Inf. Sci. (Ny) 293 (2015) 338–350.
[10] F. Glover, Pattern-Efficient Frontiers for Multi-objective optimization and Implications for Visualization., Research Report, OptTek Systems, 2015.
[11] H. Li, Q. Zhang, Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II, IEEE Trans. Evol. Comput. 13 (2) (2009) 284–302, doi:10.1109/TEVC.2008.925798.
[12] K. Miettinen, Nonlinear Multiobjective Optimization, Kluwer Academic Publishers, Boston, 1999.
[13] K. Miettinen, J. Hakanen, D. Podkopaev, Interactive nonlinear multiobjective optimization methods, in: S. Greco, M. Ehrgott, J.R. Figueira (Eds.), Multiple Criteria Decision Analysis, Springer, 2016, pp. 927–976.
[14] P.M. Pardalos, A. Žilinskas, J. Žilinskas, Non-Convex Multi-Objective Optimization, Springer, New York, 2017.
[15] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, J. Wu, Moea/d with adaptive weight adjustment, Evol. Comput. 22 (2) (2014) 231–264.
[16] T. Ray, K. Tai, K.C. Seow, Multiobjective design optimization by an evolutionary algorithm, Eng. Optim. 33 (4) (2001) 399–424, doi:10.1080/03052150108940926.
[17] D.M.Y. Sommerville, An Introduction to the Geometry of n Dimensions, Dover, New York, 1958.
[18] B. Stanojević, F. Glover, On finding a pattern-efficient set of non-dominated vectors to a multi-objective optimization problem, in: Proceedings of XIII Balkan Conference on Operational Research, The Mathematical Institute of the Serbian Academy of Sciences and Arts, 2018, pp. 235–242.
[19] E.-G. Talbi, M. Basseur, A.J. Nebro, E. Alba, Multi-objective optimization using metaheuristics: non-standard algorithms, International Transactions in Operational Research 19 (1–2) (2012) 283–305.
[20] M. Tanaka, H. Watanabe, Y. Furukawa, T. Tanino, GA-based decision support system for multicriteria optimization, in: Proceedings of the 1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century (Cat. No.95CH3576-7), Vol. 2, Department of Information Technology, Okayama University, Japan, IEEE, New York, NY, USA, 1995, pp. 1556–1561.
[21] C. Zhang, K.C. Tan, L.H. Lee, L. Gao, Adjust weight vectors in MOEA/D for bi-objective optimization problems with discontinuous pareto fronts, Soft. Comput. 22 (12) (2002) 3997–4012.
[22] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731, doi:10.1109/TEVC.2007.892759.
[23] Q. Zhang, A. Zhou, S. Zhao, P.N. Suganthan, W. Liu, S. Tiwari, Multiobjective optimization test instances for the CEC 2009 special session and competition, School of Computer Science and Electrical Engineering, University of Essex, Technical Report CES-487 (2009).
[24] Y. Zhang, R. Yang, J. Zuo, X. Jing, Enhancing moea/d with uniform population initialization, weight vector design and adjustment using uniform design, J. Syst. Eng. Electron. 26 (5) (2015) 1010–1022.
[25] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, Evol. Comput. 8 (2) (2000) 173–195.

Bogdana Stanojević graduated Mathematics and Computer Science specialization at "Transilvania" University of Braşov, Romania in 1995. She obtained her doctoral degree in Mathematics in 2003 from the Romanian Academy. Currently she is research associate professor at Mathematical Institute of the Serbian Academy of Sciences and Arts, Belgrade, Serbia. Her research interests include different aspects of fuzzy optimization, multiple objective optimization, fractional programming and mathematical fundamentals of computers.

Fred Glover is Chief Technology Officer of OptTek Systems, Inc., in charge of algorithmic design and strategic planning initiatives, and Distinguished University Professor, Emeritus, in the University of Colorado School of Engineering and Leeds School of Business. He has authored or co-authored more than 500 published articles and eight books in the fields of mathematical optimization, computer science and artificial intelligence. He is an elected member of the U. S. National Academy of Engineering and is the recipient of the von Neumann Theory Prize, the highest honor of the INFORMS Society, as well as numerous other awards and honorary fellowships.