Engineering Optimization

ISSN: 0305-215X (Print) 1029-0273 (Online) Journal homepage: http://www.tandfonline.com/loi/geno20

Directed search domain: a method for even generation of the Pareto frontier in multiobjective optimization

Tohid Erfani & Sergei V. Utyuzhnikov

To cite this article: Tohid Erfani & Sergei V. Utyuzhnikov (2011) Directed search domain: a method for even generation of the Pareto frontier in multiobjective optimization, Engineering Optimization, 43:5, 467-484, DOI: 10.1080/0305215X.2010.497185

To link to this article: http://dx.doi.org/10.1080/0305215X.2010.497185

Engineering Optimization
Vol. 43, No. 5, May 2011, 467–484

Taylor & Francis
Taylor & Francis Group

Directed search domain: a method for even generation of the Pareto frontier in multiobjective optimization

Tohid Erfani and Sergei V. Utyuzhnikov*

School of Mechanical, Aerospace and Civil Engineering, The University of Manchester, Sackville Street, M60 1QD, UK

(*Received 31 October 2009; final version received 10 May 2010*)

Optimization is one of the most important and challenging parts of any engineering design. In real-world design, multiobjective optimization with constraints has to be considered. The optimal solution in this case is not unique because the objectives can contradict each other. Therefore, a set of optimal solutions, which forms the Pareto frontier, should be considered. There are many algorithms to generate a Pareto set. However, only a few of them are potentially capable of providing an evenly distributed set of solutions. This property is especially important in real-life design because a decision maker is usually able to analyse only a very limited number of solutions. The main objective of this article is to develop and give detailed description of an algorithm that is able to generate an evenly distributed Pareto set in a general formulation. The approach is based on shrinking a search domain to generate a Pareto optimal solution in a selected area on the Pareto frontier. The effectiveness of the algorithm is demonstrated by a number of challenging test cases. For the first time, some of these test cases are successfully solved via a classical approach.

Keywords: multiobjective optimization; Pareto solution; Pareto set; directed search domain

1. Introduction

A key challenge in real-life design is simultaneously to optimize different objectives. A decision-maker (DM) has to take into account different criteria such as low cost, manufacturability, long life and good performance, which cannot be satisfied at the same time. In fact, it is only possible to consider a trade-off among all (or almost all) criteria. The task becomes even more complicated because of additional constraints, which always exist in practice.

Mathematically, the trade-off analysis can be formulated as a vector nonlinear optimization problem with constraints. Generally speaking, the solution of such a problem is not unique. It is natural to exclude from the consideration any design solution that can be improved without deterioration of any discipline and violation of the constraints; in other words, a solution that can be improved without any trade-off. This leads to the notion of a Pareto optimal solution (Miettinen 1999). Each Pareto point is a solution of the multiobjective optimization (MOO) problem. A designer selects the ultimate solution among the Pareto set on the basis of additional requirements, which may be subjective. In general, it is desirable to have a sufficient number of

*Corresponding author. Email: s.utyuzhnikov@manchester.ac.uk

ISSN 0305-215X print/ISSN 1029-0273 online

DOI: 10.1080/0305215X.2010.497185
http://www.informaworld.com

Pareto points to represent the entire Pareto frontier in the objective space. In real design, the DM is able to consider only a few possible solutions (Pareto points). In such a context, it is important to have a well-spread distribution of Pareto points to obtain maximum information on the Pareto surface at a minimum (computational) cost.

Despite the existence of many numerical methods for vector nonlinear optimization, there are few approaches potentially suitable for real-design applications. This is because the problem in question is usually very time-consuming. In many practical multidisciplinary optimization applications, the design cycle includes time-consuming and expensive computations at each discipline. For example, in the aerospace industry, aerodynamics and stress analysis have to be considered, apart from other objectives. Thus, to take into account their mutual influences, interdisciplinary iterations are required.

In general, optimization methods can be split into two principle categories: classical (preference-based) methods and evolutionary algorithms. The classical methods usually use deterministic approaches, whereas evolutionary ones are based on stochastic algorithms. It goes without saying that such a division is not strict and the combination of classical and evolutionary methods is also possible. In classical MOO methods, vector optimization approaches are often reduced to the minimization of an aggregate objective function (AOF) (preference function), which includes a combination of objective (cost) functions. On the other hand, evolutionary methods consider all objective functions for optimization simultaneously. They start by generating a random initial population. Using an iterative procedure, the current population is updated and the next population is created by using some operators, namely: selection, mutation and crossover. By setting some stopping criteria such as time limit, constraint tolerance or fitness (objective) value, the iterative procedure reaches to the end (Michalewicz 1996, Deb 2001). Evolutionary algorithms have a number of clear advantages over classical approaches. They are not sensitive to non-smoothness of objective functions and are efficient in finding a global extremum. Therefore, they are quite popular nowadays. However, in evolutionary methods, there is no guarantee for capturing an optimum solution. In addition, a huge number of solutions are to be considered to generate an even set of optimal solutions. In real design, eventually most of them become redundant. Moreover, evolutionary-based algorithms are time-consuming in multiobjective optimization.

In this article, a classical algorithm that provides a well-distributed representation of the entire Pareto surface is considered. To seek the Pareto frontier, a strategy is described based on a Directed Search Domain (DSD) algorithm, which was first suggested and applied for the modification of the Physical Programming method (Utyuzhnikov *et al*. 2005, 2009). The main idea of DSD is to shrink a search domain to obtain a Pareto solution in a selected area of objective space. A well-spread distribution of the selected search domains should provide a quasi-even Pareto set.

The rest of the article is organized as follows. In Section 2, a short survey of the existing classical methods is given. Then, in Section 3, the main notions of multiobjective optimization are described. The principles and the steps of the DSD algorithm are explained in Section 4. The efficiency and effectiveness of DSD is demonstrated in Section 5, in which the approach is tested on two- and three-dimensional tasks. Some of these test cases are challenging for the existing classical methods (Shukla and Deb 2007). By considering these examples, it is demonstrated that the DSD algorithm can successfully be applied to them.

2. The existing classical generating methods

As noted above, in classical methods, the MOO problem is reduced to the minimization of the combination of objective functions or AOF. The simplest and often used AOF is a linear (weighted) combination of the objective functions (Miettinen 1999). This approach has many drawbacks,

mainly related to an uncertainty in the weighting coefficients. It may require many iterations to find the combination of the weights leading to a solution that corresponds to the DM's expectations (Messac 2000). Furthermore, it is well known that this method can generate only the convex part of a Pareto surface (Koski 1985, Athan and Papalambros 1996, Das and Dennis 1997, Messac *et al.* 2000b) while real-life problems often result in non-convex Pareto frontiers. This drawback can be avoided by using either a more complex consideration of the AOF (Athan and Papalambros 1996) or the weighted Tchebychev method (Miettinen 1999). However, the weights remain to be unknown functions of the objectives (Messac 2000).

Das and Dennis (1997) showed that an even spread of weights in the AOF does not necessarily result in an even distribution of the solutions on the Pareto frontier. Further, the most successful methods for evenly generating the entire Pareto frontier in the general multidimensional formulation are reviewed.

The Normal Boundary Intersection (NBI) method was developed by Das and Dennis (1998) and Das (1999). The method has a clear geometrical interpretation. It is based on the well-known fact that a Pareto surface is related to the boundary of a feasible domain towards the minimization of objective functions (Miettien 1999). First, so-called anchor points are obtained in the objective space. An anchor point corresponds to the optimal value of one and only one objective function in the feasible domain. Thus, n objective functions provide up to n anchor points. Second, the utopia plane passing through the anchor points is obtained. The Pareto surface is then constructed by the intersection of lines, which are normal to the utopia plane, and the boundary of the feasible objective domain. A single optimization problem with respect to one of the objective functions is solved along each line. An even distribution of Pareto points is provided by an even distribution of lines orthogonal to the utopia plane. The method, which seems to generate both non-Pareto and locally Pareto solutions, requires a filtering procedure (Messac *et al.* 2003, Shukla and Deb 2007). In addition, the NBI method might be non-robust since the feasible domain is reduced to a line.

The Normal Constraint (NC) method by Messac *et al.* (2003) and Messac and Mattson (2004) represents a modification of the NBI approach. The single optimization problem, used in the NC, is based only on inequality constraints. This modification makes the method more flexible and stable. Both NC and NBI methods may fail to generate Pareto solutions over the entire Pareto frontier in a multidimensional case (Messac and Mattson 2004). The modification of the NC (Messac and Mattson 2004) partially eliminates this drawback. However, both methods may generate non-Pareto and locally Pareto solutions (although the NC is less likely to do this – see Messac and Mattson 2004). Shukla and Deb (2007) and Utyuzhnikov *et al.* (2009) show that both the NC and NBI methods can be inefficient because of the significant number of redundant solutions. In the example given by Utyuzhnikov *et al.* (2009), 66 points on the utopia plane lead to only 24 Pareto solutions. Another example can be found in Shukla and Deb (2007). Both methods can encounter significant difficulties in the case of a disconnected frontier (Shukla and Deb 2007). In particular, they are not always able to find the entire Pareto surface. Meanwhile, one can note that recent modifications of the NC method (Fantini 2007, Sanchis *et al.* 2008) are able to improve these approaches.

The Physical Programming (PP) method was suggested by Messac (1996). This method also generates Pareto points on both convex and non-convex Pareto frontiers (Messac and Mattson 2002). The method does not use any weight coefficients and allows one to take into account the DM experience immediately. In the PP, the designer assigns each objective to one of the four categories or class-functions. The optimization is based on minimization of an AOF determined by the preference functions (class-functions). The algorithm given by Messac and Mattson (2002) is able to generate an evenly distributed Pareto set. However, it contains a few free parameters, the optimal choice of which requires preliminary information on the location of the Pareto frontier.

Utyuzhnikov *et al.* (2005, 2009) suggested modifications to the PP method to make it possibly simpler and more flexible for practical applications. A simpler structure for the class-functions is suggested. The class-functions are generalized to shrink the search domain to a 'hypercone' and make its location in the objective space better. This is critical for generating an even set of the Pareto frontier. The proposed modification combines the advantages of the PP, NBI and NC methods. One of the main advantages of the approach (Utyuzhnikov *et al.* 2009, Utyuzhnikov 2010) is that it does not provide non-Pareto solutions while local Pareto solutions may be easily recognized and removed. Utyuzhnikov *et al.* (2009) show that the modified PP is able to generate a quasi-even Pareto set in the general formulation. It is mathematically proven that the method is able to capture the entire Pareto frontier by conducting the search domain in the objective space. The approach is based on the DSD algorithm described in this article. In the current article, the use of the PP method is avoided to make the DSD approach simpler for practical implementation.

3. Multiobjective optimization problems

Consider n objective functions. The n objectives form a space called *objective space* $\mathcal{Z} \subseteq \mathbb{R}^n$. A design variable is represented by a vector in a decision space $\mathcal{D} \subseteq \mathbb{R}^m$. The set $\mathcal{D}^* \subseteq \mathcal{D}$ of the elements satisfying all the constraints is called a $feasible\ set$ or $feasible\ space$. For each $x \in \mathcal{D}^*$ there exists a point in $\mathcal{Z}$ corresponding to mapping $\mathbb{R}^m \to \mathbb{R}^n$. Hence, the feasible objective space, $\mathcal{Z}^*$, is the image of $\mathcal{D}^*$, *i.e.* $\mathcal{Z}^*=\{z = \mathcal{F}(x)|x \in \mathcal{D}^*\}$.

The generic form of any multiobjective optimization is given by

$$\begin{aligned} &\text{Min } \mathcal{F} = \{F_1(x), F_2(x), \ldots, F_n(x)\}, \\ &\text{subject to } x \in \mathcal{D}^*. \end{aligned} \tag{1}$$

A point corresponding to the minimum of all objective functions is usually situated outside the feasible space $\mathcal{D}^*$. Therefore, a set of solutions called the $Pareto\ optimal$ set is considered according to the following definition.

DEFINITION 3.1 – PARETO OPTIMALITY *Vector $\mathbf{x}^* \in \mathcal{D}^*$ is called a Pareto solution to problem* (1) *iff $\nexists\, \mathbf{x}^{**}$ such that $F_i(\mathbf{x}^{**}) \leq F_i(\mathbf{x}^*)$ for any $i = 1, \ldots, n$ and $\exists\, j\ (1 \leq j \leq n) : F_j(\mathbf{x}^{**}) < F_j(\mathbf{x}^*)$.*

Vector $\mathbf{x}^*$ is called a *local Pareto* solution if the above condition holds in the ϵ-vicinity of $\mathbf{x}^*$. If $\mathbf{x}^*$ is a Pareto solution, it is said that $\mathbf{x}^*$ is *not dominated* by any other feasible solution.

In the objective space $\mathcal{Z}$, Pareto solutions lie on the boundary of $\mathcal{Z}^*$ called the *Pareto frontier*. The above definition gives the best trade-off solutions to the multiobjective optimization problem (1). For further consideration, the following definitions are also needed.

DEFINITION 3.2 – ANCHOR POINT *In the objective space $\mathcal{Z}$, an anchor point μ_i represents the minimum of an ith objective function subject to the constraints.*

The above definition does not always provide a unique anchor point. To avoid this problem, a lexicographic-based prioritization is introduced as follows (Utyuzhnikov *et al.* 2005, 2009).

DEFINITION 3.3 – MODIFIED ANCHOR POINT *In the feasible objective space $\mathcal{Z}^*$, the anchor point μ_i of an ith objective function is determined in the circular order:* $\min\ F_i, \min\ F_{i+1}, \ldots, \min\ F_n, \min\ F_1, \min\ F_2, \ldots, \min\ F_{i-1}$. *Any successive minimization problem:* $\min\ F_{i+l}$ $(1 \leq i + l \leq n,\ l \neq 0)$, *is to be considered only on the set $\{A_{l-1}\}$, which is the solution to the previous minimization problem, and only if the set $\{A_{l-1}\}$ contains more than one element.*

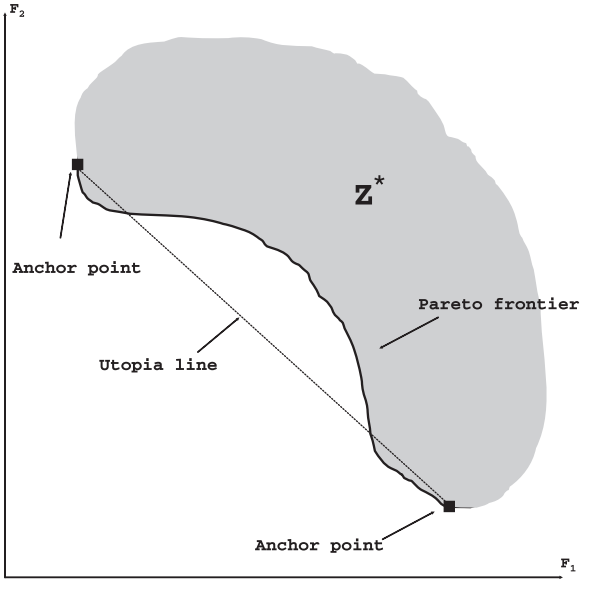

Figure 1. Generic utopia plane and anchor points in objective space.

DEFINITION 3.4 – UTOPIA HYPERPLANE *A hyperplane that contains all the anchor points is called the utopia hyperplane.*

All of the above definitions are demonstrated in Figure 1.

4. Directed search domain algorithm

Further, a Pareto set is generated by the use of the DSD approach. This approach exploits the transformation technique of Utyuzhnikov *et al*. (2009). The algorithm contains six major steps illustrated below.

It is to be noted that, in the algorithm, it is assumed that an even distribution of the reference points leads to a well enough distributed Pareto set. This is not the case if the normal to the utopia plane is almost orthogonal to the normal to the Pareto surface. A similar problem also arises for the NBI, NC and PP methods. This can be overcome by an adaptive generation of the reference points. However, this question is beyond the scope of this article.

4.1. *Major steps of DSD algorithm*

Step 0 (Pre-processor scaling). In the general formulation, to avoid undesirable severe skewing of the search domain in the algorithm, the objective functions should be preliminarily scaled (Miettinen 1999):

$$F_i^{\text{sc}} = \frac{F_i - \mu_i}{F_{i,\max} - \mu_i}. \tag{2}$$

Here, $F_{i,\max} = \max_{x \in \mathcal{D}^*} F_i(x)$. This value can be found by solving a single objective optimization problem. Alternatively, using the DM knowledge, $F_{i,\max}$ can merely be estimated because its exact value is not necessary. Overall, if it is known that the objectives vary almost in the same range, this step can be ignored.

Step 1 (Modified anchor points). In this step, the anchor points according to Definition 3.3 are to be calculated.

Step 2 (Utopia hyperplane calculation). A hyperplane P containing all the anchor points is considered. Its equation is given by (see *e.g.* Hoffman and Kunze 2002)

$$P = \sum_{i=1}^{n} \alpha_i \mu_i,$$
$$\sum_{j=1}^{n} \alpha_j = 1. \tag{3}$$

The interior of the utopia polygon formed by the anchor points is determined by

$$0 \leq \alpha_i \leq 1 \quad (i = 1, \ldots, n).$$

Step 3 (Reference points on utopia hyperplane). By varying the coefficients α_i in Equation (3) uniformly, it is possible to generate evenly distributed points on the utopia hyperplane, further denoted by set $\mathcal{M}$ (Messac and Mattson 2004).

Step 4 (Aggregate objective function). In this step, an AOF is formed. Obviously, the AOF cannot be arbitrary. Requirements for an admissible AOF can be found in Messac *et al.* (2000a), according to whom an admissible AOF must be a locally coordinatewise increasing function of the objective functions. By definition, a function is a coordinatewise increasing function if it is a strict monotonically increasing function with respect to each argument (Steuer 1986).

In this article, the AOF is introduced as a sum of objective functions. To form the search domain, each objective function is restricted from the top. A reference point $M \in \mathcal{M}$ with coordinates $(M_1, M_2, \ldots, M_n)$ determines the upper bound for F_i $(i = 1, \ldots, n)$. Thus, the following problem is formulated:

$$\begin{aligned} \text{Min} \quad & \sum_{i=1}^{n} F_i(x), \\ \text{subject to} \quad & F_i \leq M_i, \\ & x \in \mathcal{D}^*. \end{aligned} \tag{4}$$

For $n = 2$, this step is illustrated in Figure 2.

To solve problem (4), any minimizer algorithm can be used, *e.g.* a gradient based method if the AOF is smooth enough. The formulated single-objective minimization problem (4) should be solved for each reference point $M \in \mathcal{M}$ generated at step 3.

Step 4.1 (Shrinking search domain). A distribution of reference points on the utopia hyperplane leads to different search domains. As a result, one can expect different Pareto solutions to be generated. However, as shown by Utyuzhnikov *et al.* (2009), this is not always the case. Although the search domain is limited for each case, it is relatively large. This may lead to generating redundant solutions for different optimization runs. To avoid this problem, the search domain is shrunk around a given direction. To realize this, following Utyuzhnikov *et al.* (2009), a new coordinate system $\mathbf{a}_i$ is introduced with the origin at a reference point M and the axes forming angle γ_c with an arbitrary unit vector $\mathbf{l}$. Thus, a hypercone with its vertex at point M and surface aligned with the edges of the search box is obtained. It is natural to choose $\mathbf{l}$ parallel to the normal

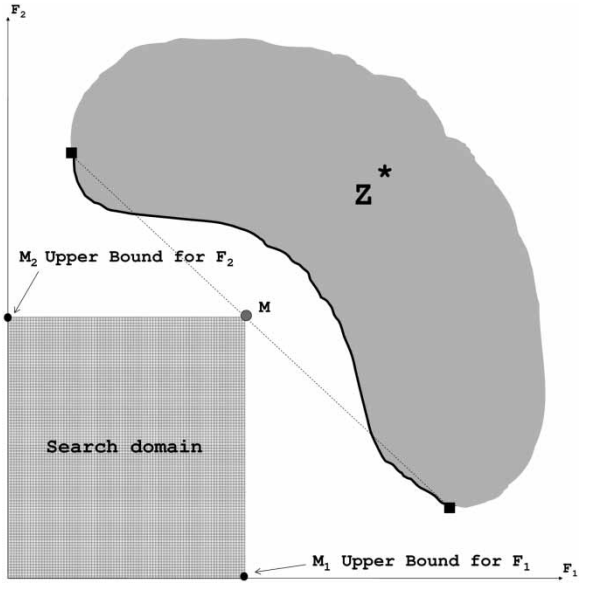

Figure 2. Search domain in the two-dimensional case.

of the utopia hyperplane (Figure 3). Thereby, it is required to find an affine transformation:

$$\hat{F}_i = \sum_{j=1}^{n} F_j B_{ji} \quad (\forall i = 1, \ldots, n),$$
$$B = A^{-1}, \tag{5}$$

where A is the matrix of transformation from the Cartesian coordinate system to the new local coordinate system:

$$\mathbf{a}_i = \sum_{j=1}^{n} A_{ij} \mathbf{e}_j \quad (\forall i = 1, \ldots, n).$$

In the Cartesian coordinate system, a unit vector $\mathbf{l}_0 = (l_0, l_0, \ldots, l_0)^{\mathrm{T}}$ is introduced, where $l_0 \equiv \cos \gamma_0 = 1/\sqrt{n}$.

It can be shown (Utyuzhnikov *et al*. 2009) that

$$A = \frac{\sin \gamma_c}{\sin \gamma_0} R^{\mathrm{T}} + \frac{\sin(\gamma_0 - \gamma_c)}{\sin \gamma_0} E. \tag{6}$$

Here, the rotation matrix R maps vector $\mathbf{l}_0$ onto the vector $\mathbf{l}$:

$$R\mathbf{l}_0 = \mathbf{l}, \tag{7}$$

and $\|E_{ij}\| = \|l_j\|$ $(i, j = 1, \ldots, n)$. Matrix R is given by

$$R = DTD^{\mathrm{T}}, \tag{8}$$

where D is an orthogonal matrix, which can be obtained by the Gram-Schmidt orthogonalization procedure from, for example, the normal vector to the utopia plane $\mathbf{n}$, vector $\mathbf{l}_0$ and basis vectors $\mathbf{e}_3, \mathbf{e}_4, \ldots, \mathbf{e}_n$ (see Appendix A):

$$D = \text{Gram–Schmidt } (\mathbf{l}_0, \mathbf{n}, \mathbf{e}_3, \mathbf{e}_4, \ldots, \mathbf{e}_n).$$

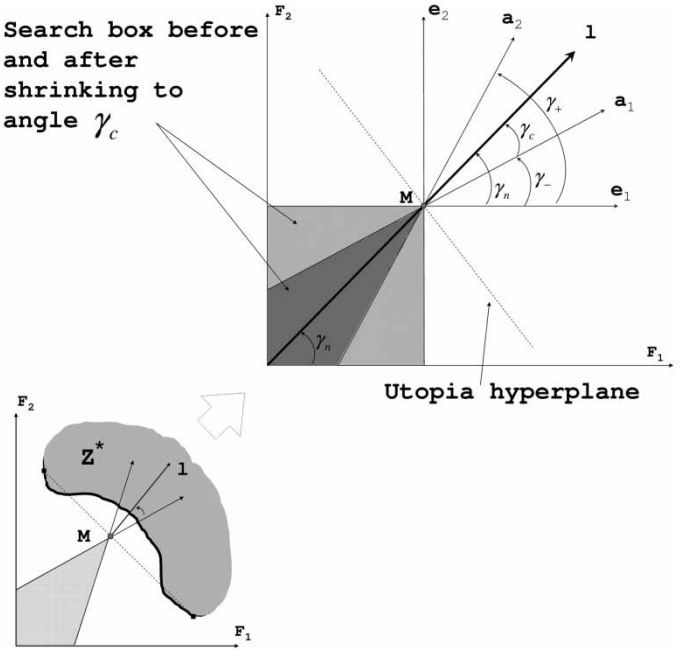

Figure 3. Shrinking search domain.

In Equation (8), matrix T describes an elementary rotation in the hyperplane created by the vectors $\mathbf{n}$ and $\mathbf{l}_0$:

$$T = \begin{pmatrix} (\mathbf{l}_0, \mathbf{n}) & -\sqrt{1-(\mathbf{l}_0, \mathbf{n})} & 0 & \cdots & 0 \\ \sqrt{1-(\mathbf{l}_0, \mathbf{n})} & (\mathbf{l}_0, \mathbf{n}) & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}. \tag{9}$$

The angles γ_c and γ_n used in Equation (6) are shown in Figure 3. It should be noted that $\gamma_n = \gamma_0$ if $\mathbf{l} = \mathbf{l}_0$. In the general case, $\mathbf{l}_0$ is mapped onto $\mathbf{l}$ by matrix R (Equation 7).

In the two-dimensional case, as an example: $n = 2$, $\gamma_0 = 45°$ and it is easy to show that

$$A = \begin{pmatrix} \cos\gamma_- & -\sin\gamma_- \\ \cos\gamma_+ & \sin\gamma_+ \end{pmatrix},$$

where $\gamma_+ = \gamma_n + \gamma_c$, $\gamma_- = \gamma_n - \gamma_c$.

As can be seen, it is possible to shrink the search domain by the use of transformation (5). Thus, the constraints of problem (4) can be replaced by the following:

$$\hat{F}_i \le \sum_{j=1}^{n} M_j B_{ji}, \quad (i = 1, \ldots, n). \tag{10}$$

It is to be noted that, in Equation (5), matrix A is to be inverted to find matrix B. However, this needs to be done only once because matrix A is the same for all the search domains. Therefore, the DSD algorithm is not time-consuming even in a multi-dimensional case.

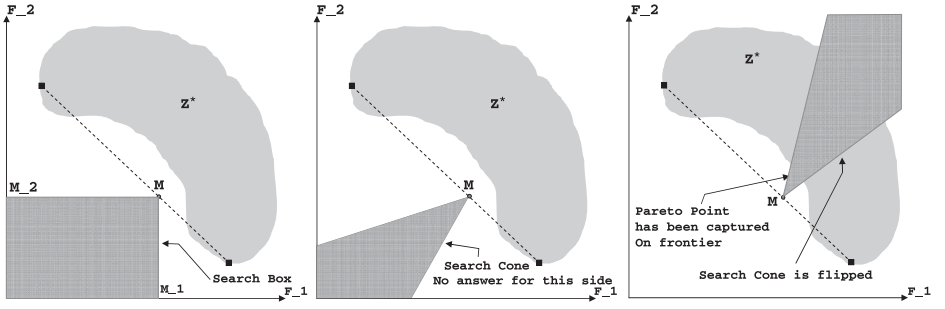

Figure 4. Schematic illustration of the DSD algorithm for the generic two-dimensional case.

Step 4.2 (Flipping search domain). Consider the case of a non-convex boundary as illustrated in Figure 4. It is possible not to have any feasible solution in the search domain including the reference point M. In this case, the search domain should be flipped to the opposite side of the utopia hyperplane to capture the points on the Pareto frontier. This can be done by changing the upper bound of each objective function to its lower bound using the current coordinates of the point M as follows:

$$\hat{F}_i \geq \sum_{j=1}^{n} M_j B_{ji} \quad (i = 1, \ldots, n). \tag{11}$$

Step 4.3 (Rotating search domain). In some cases, it is possible that the orthogonal projection of the utopia hyperplane does not fully cover the entire Pareto surface (*e.g.* Figure 5).

Thus, the search cones, whose axes are perpendicular to the utopia hyperplane, are not able to capture the entire Pareto frontier. To overcome this problem, the search cone is rotated if the current point M is located on the edge of the utopia polygon. To realize this, a vector s is introduced as the outer normal to the edge of the polygon on the utopia plane (Figure 6):

$$\boldsymbol{s}_i = \frac{\boldsymbol{v}_{i-1} + \beta_i \boldsymbol{v}_i}{|\boldsymbol{v}_{i-1} + \beta_i \boldsymbol{v}_i|}, \quad \beta_i = -\frac{(\boldsymbol{v}_{i-1}, \boldsymbol{v}_i)}{(\boldsymbol{v}_i, \boldsymbol{v}_i)}, \tag{12}$$

where $\boldsymbol{v}_i = \boldsymbol{\mu}_i - \boldsymbol{\mu}_{i-1}$ are the boundaries of the polygon.

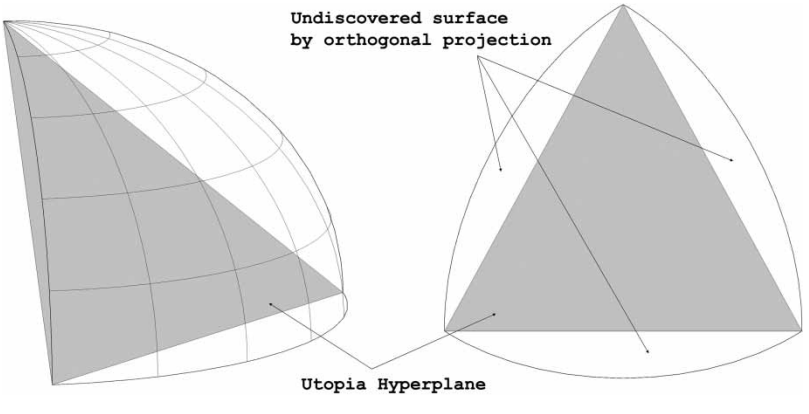

Figure 5. Orthogonal projection cannot capture the entire Pareto frontier.

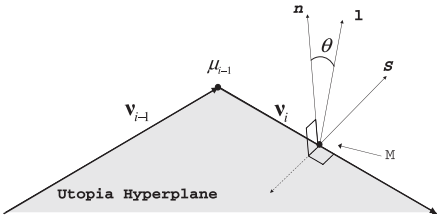

Figure 6. Rotation of vector **l**.

Consequently, **l** is replaced by

$$\mathbf{l} = -\cos\theta\mathbf{n} + \sin\theta\mathbf{s}_i, \quad 0 < \theta < \pi/2. \tag{13}$$

Changing θ from 0 to $\pi/2$, rotation of the cone is possible from its normal direction to the direction lying on the outer part of the polygon (Utyuzhnikov *et al*. 2009). The rotation continues until no new solution is capturable.

Figure 7 shows this step for a generic three-dimensional case.

Step 5 (Filtering procedure). Because of generating local Pareto points in some cases, the *contact* theorem (Vincent and Grantham 1981) is used to remove these points from the set of

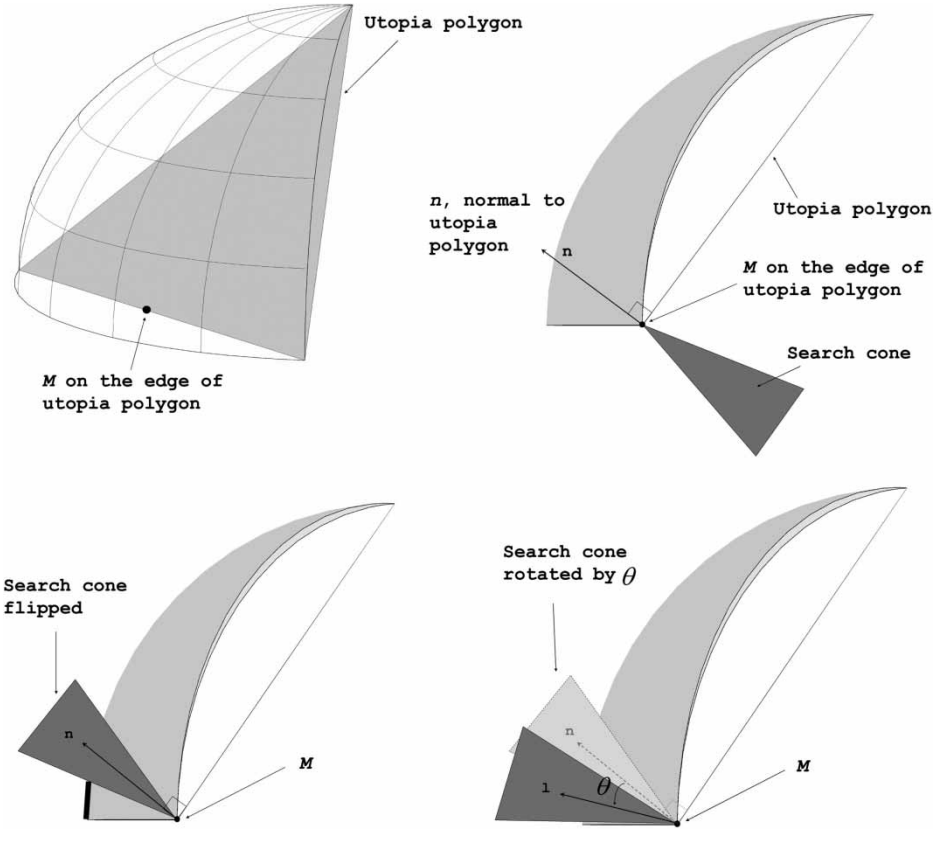

Figure 7. Rotation of search cone on the edge of the utopia polygon.

Figure 8. DSD filtering procedure.

solutions. The filtering procedure is based on the search domain of problem (4), in which point M is situated at each generated Pareto-solution candidate denoted by P. If the intersection of the search domain with the set $\mathcal{D}^*$ is not empty, then the point P is not a Pareto solution. This algorithm is applied at each Pareto candidate. It is clear that if a Pareto candidate is dominated by another Pareto candidate, then the former one must be removed. A pseudo code for DSD filtering is shown in Algorithm 1 and is depicted in Figure 8.

Algorithm 1 DSD filtering pseudo code.

Set **PS** *the set of all Pareto candidates*
$i \leftarrow 1$
for all $i = 1, \ldots, |PS|$ **do**
  $P \leftarrow ParetoCandidate_i$
  if $(Cone^P_{90^\circ} \cap \mathcal{D}^* \neq \emptyset)$ **then**
    $PS = PS - ParetoCandidate_i$
  end if
  $i = i + 1$
end for

4.2. *DSD algorithm sketch*

Overall, the sketch of the DSD approach is formulated in Algorithm 2 and for a two-dimensional case is illustrated in Figure 4 with an arbitrary reference point M.

5. The test cases

The effectiveness of the DSD approach is validated by five test cases, of which three examples are three-dimensional and two cases are two-dimensional. The test cases include both convex and non-convex Pareto frontiers with different peculiarities. In particular, they contain a discontinuous Pareto frontier, a non-uniform density of feasible solutions, and the degeneration of the utopia plane to a lower dimension. Each of these test cases represents a real challenge for classical methods. For example, Shukla and Deb (2007) concluded that test cases ZDT6 and DTLZ5 cannot be tackled by any existing classical method. Test case Comet is considered by Deb *et al.*

Algorithm 2 DSD algorithm

0. Scale the objective functions.
1. Find the modified anchor points.
2. Find the utopia hyperplane.
3. Generate evenly distributed reference points (M) on the utopia hyperplane assigned to the set $\mathcal{M}$.
4. Consider the following constraint:

$$F_i \le M_i \quad \forall i. \qquad (*)$$

5. Shrink the constraint $(*)$ (search domain) by an angle γ: $0 < \gamma \le 45^\circ$ using

$$\hat{F}_i \le \sum_{j=1}^{n} M_j B_{ji} \quad \forall i.$$

for all $M \in \mathcal{M}$ **do**
6. Solve the following problem:

$$\begin{aligned} \text{Min} \quad & \sum_{i=1}^{n} F_i(x), \\ \text{s.t.} \quad & \hat{F}_i \le \sum_{j=1}^{n} M_j B_{ji} \quad \forall i, \\ & x \in \mathcal{D}^*. \end{aligned}$$

6.1. if no solution is found **then**
flip the search box to the opposite side of the utopia hyperplane.
6.2. if the M is on the edge of the hyperplane **then**
rotate the search box by $0 < \theta < \pi/2$ to capture undiscovered regions.
end for
7. Apply the filtering algorithm to eliminate local Pareto solutions.

(2005) to be a very time-consuming problem for classical methods since it leads to a considerable number of redundant solutions.

One of the important factors in judging the efficiency of an optimization algorithm is the number of objective function evaluations. In the case of the DSD algorithm, this factor can be evaluated as follows:

$$N_o = \sum_{n=1}^{P_s} k_n,$$

where P_s is the number of Pareto solutions and k_n is the number of iterations required to solve a single objective problem. The value of k_n, of course, depends on the optimizer used, the initial approximation and requirements to the accuracy of the solution. In the presented study, the coefficient k_n was no greater than ten due to a relatively high convergence speed of the Newton method exploited and a good enough initial approximation.

In order to evaluate quantitatively how well a Pareto set is spread, a coefficient of evenness, suggested by Utyuzhnikov *et al.* (2009), is introduced as follows.

Introduce first the Riemann metric:

$$\mathrm{d}\boldsymbol{r}^2 = \sum_{i=1}^{n-1}\sum_{j=1}^{n-1} g_{ij}\,\mathrm{d}\boldsymbol{x}^i\,\mathrm{d}\boldsymbol{x}^j, \tag{14}$$

where $\{\boldsymbol{x}^i\}$ $(i = 1, \ldots, n-1)$ is a coordinate system on the Pareto surface and g is the metric tensor. Then, the coefficient of evenness can be determined as

$$E = \frac{\max_i \min_j r_{ij}}{\min_i \min_j r_{ij}} \quad (\forall i = 1, \ldots, P_\mathrm{s}\ i \neq j). \tag{15}$$

The coefficient E represents the ratio between the maximal possible distance from any Pareto point to its nearest point and the minimal one. Thus, r_{ij} denotes the distance between Pareto solutions i and j $(i \neq j)$ in metric (14). It is clear that, in the case of a completely even set, $E = 1$.

The scaling procedure (Step 0) is not used in any of the test cases considered below.

5.1. *Two-dimensional test cases*

TNK. In this test case introduced by Tanaka *et al.* (1995), a discontinuous Pareto frontier is considered.

$$\begin{aligned}
&\text{Minimize} \quad (x_1, x_2),\\
&\text{s.t.} \quad g_1(x) = x_1^2 + x_2^2 - 1 - 0.1\cos\left(16\arctan\frac{x_1}{x_2}\right) \geq 0,\\
&\qquad\quad g_2(x) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5,\\
&\qquad\quad 0 \leq x_i \leq \pi \quad (i = 1, 2).
\end{aligned}$$

In the test problem in question, the decision space coincides with the objective space. The boundary of the feasible space is non-convex and, as noted above, the Pareto frontier is discontinuous. This is due to the fact that both constraints should be satisfied. In Figure 9, the generated Pareto frontier is represented by black dots with anchor points calculated as (1.1, 0) and (0, 1.1). For this test case, the coefficient of evenness E equals 1.9, and the number of objective function evaluations N_o is equal to 314 for 100 Pareto solutions. It should be noted that, because of nonlinear constraints, many optimization algorithms face difficulty in finding a good spread of solutions across the discontinuous Pareto sets (Deb 2001).

ZDT6. This test case was introduced by Shukla and Deb (2007) as follows:

$$\begin{aligned}
&\text{Minimize} \quad (F_1(x), F_2(x)),\\
&\text{s.t.} \quad\ 0 \leq x_i \leq 1 \quad (i = 1, 2, \ldots, 10),\\
&\text{where}\\
&\qquad F_1(x) = 1 - \exp(-4x_1)\sin^6(4\pi x_1),\\
&\qquad F_2(x) = g(x)\left[1 - \left(\frac{F_1(x)}{g(x)}\right)^2\right],\\
&\qquad g(x) = 1 + 9\left(\sum_{i=2}^{10} x_i^2/(n-1)\right)^{1/4}.
\end{aligned}$$

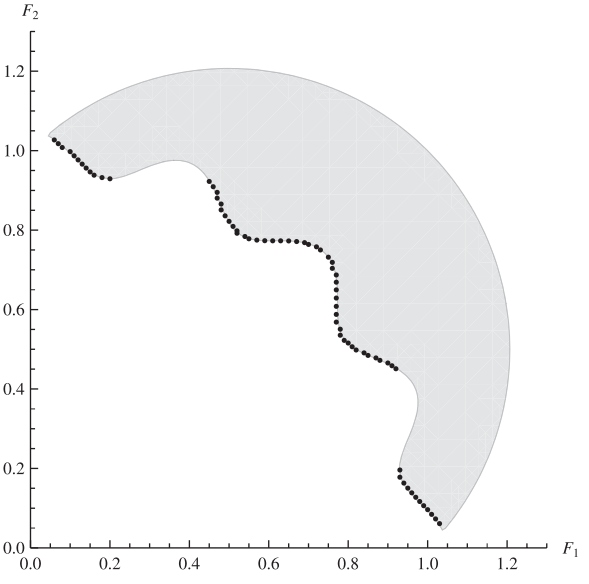

Figure 9. Pareto front generation for the TNK test case.

As explained by Deb (2001) and Shukla and Deb (2007), the adverse (non-uniform) density of solutions on the Pareto frontier creates a real problem for any generating method. Shukla and Deb (2007) claimed that no classical method (including NC and NBI) is capable of finding the Pareto frontier for this test case. The DSD algorithm works reasonably well for this test case as shown in Figure 10. The generated Pareto set is characterized by $E = 1.87$, $N_o = 675$ and $P_s = 100$. The anchor points for this test case correspond to (0.38,0.88) and (1,0). This test problem also demonstrates the operation of the algorithm in generating a non-convex frontier.

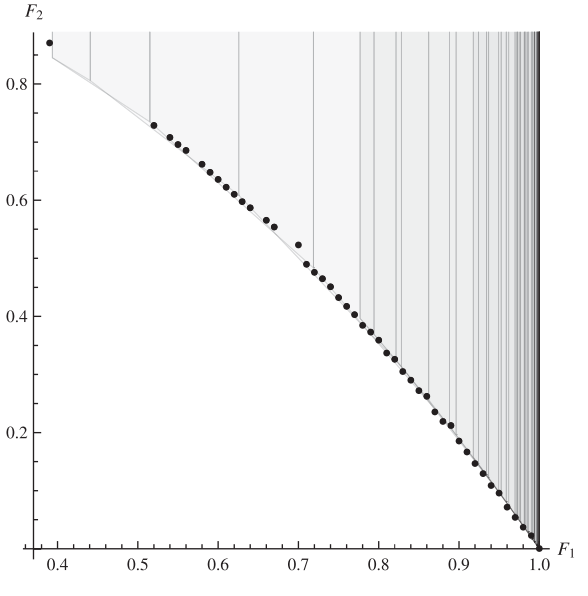

Figure 10. Pareto front generation for the ZDT6 test case.

5.2. *Three-dimensional test cases*

DTLZ2. A relatively simple three-dimensional test case, introduced by Deb *et al*. (2005) is considered. As shown by Utyuzhnikov *et al*. (2009) for a similar test case, the NBI and NC methods generate a significant number of redundant solutions.

$$
\begin{aligned}
&\text{Minimize} \quad (F_1(x), F_2(x), F_3(x)),\\
&\text{s.t.} \qquad 0 \le x_i \le 1 \quad (i = 1, 2, 3),\\
&\qquad \text{where}\\
&\qquad F_1(x) = (1 + g(x_3)) \cos(x_1 \pi/2) \cos(x_2 \pi/2),\\
&\qquad F_2(x) = (1 + g(x_3)) \cos(x_1 \pi/2) \sin(x_2 \pi/2),\\
&\qquad F_3(x) = (1 + g(x_3)) \sin(x_1 \pi/2),\\
&\qquad g(x_3) = (x_3 - 0.5)^2.
\end{aligned}
$$

In this test case, three anchor points, (1, 0, 0), (0, 1, 0) and (0, 0, 1), are calculated. It appears that not the entire orthogonal projection of the Pareto surface onto the utopia plane is situated in the triangle created by the anchor points. To resolve this problem, the rotation of the search domain is applied on the edges of the utopia triangle. This allows the entire Pareto surface shown in Figure 11 to be captured. As can be seen, 82 solution points are generated corresponding to 55 reference points on the utopia plane with the number of objective function evaluations equal to $N_o = 288$ ($P_s = 82$). This leads to $E = 1.41$ (See Equation 15).

DTLZ5. In this test case, introduced by Deb *et al*. (2005), the number of anchor points is less than the number of objective functions.

$$
\begin{aligned}
&\text{Minimize} \quad (F_1(x), F_2(x), F_3(x)),\\
&\text{s.t.} \qquad 0 \le x_i \le 1 \quad (i = 1, 2, 3),\\
&\qquad \text{where}\\
&\qquad F_1(x) = (1 + g(x_3)) \cos(\theta_1) \cos(\theta_2),\\
&\qquad F_2(x) = (1 + g(x_3)) \cos(\theta_1) \sin(\theta_2),\\
&\qquad F_3(x) = (1 + g(x_3)) \sin(\theta_1),\\
&\qquad g(x_3) = (x_3 - 0.5)^2,\\
&\qquad \theta_1 = \frac{\pi}{2}(x_1),\\
&\qquad \theta_2 = \frac{\pi}{4(1 + g(r))}(1 + 2g(x_3)x_2).
\end{aligned}
$$

A specific characteristic of this example is that the utopia plane degenerates to a utopia line in three-dimensional space. As quoted by Shukla and Deb (2007), all classical algorithms fail in capturing the Pareto frontier for this test case. According to the modified definition of the anchor point, two anchor points are obtained: (0.71, 0.71, 0) and (0, 0, 1). The DSD approach yields a Pareto set as shown in Figure 12 with $E = 1.43$. The objective functions are called $N_o = 22$ times in this test case for the whole Pareto set with $P_s = 10$.

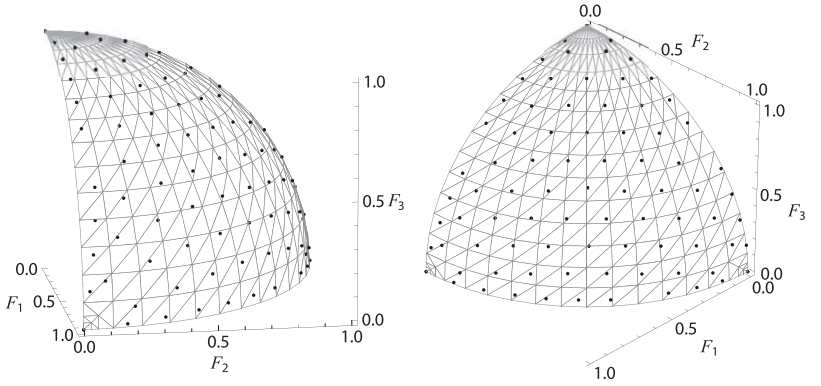

Figure 11. Pareto front generation for the DTLZ2 test case.

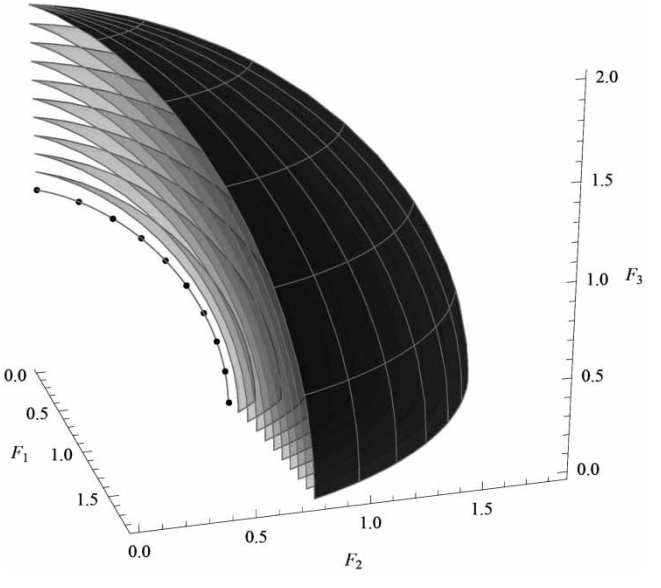

Figure 12. Pareto front generation for the DTLZ5 test case.

Comet. In the next test case, introduced by Deb *et al*. (2005), the Pareto frontier sharply shrinks.

$$
\begin{aligned}
&\text{Minimize } (F_1(x), F_2(x), F_3(x)),\\
&\text{s.t.} \quad 1 \le x_1 \le 3.5,\\
&\qquad -2 \le x_1 \le 2,\\
&\qquad \text{where}\\
&\qquad F_1(x) = (1 + g(x_3))(x_1^3 x_2^2 - 10x_1 - 4x_2),\\
&\qquad F_2(x) = (1 + g(x_3))(x_1^3 x_2^2 - 10x_1 + 4x_2),\\
&\qquad F_3(x) = 3(1 + g(x_3))x_1^2,\\
&\qquad g(x_3) = x_3.
\end{aligned}
$$

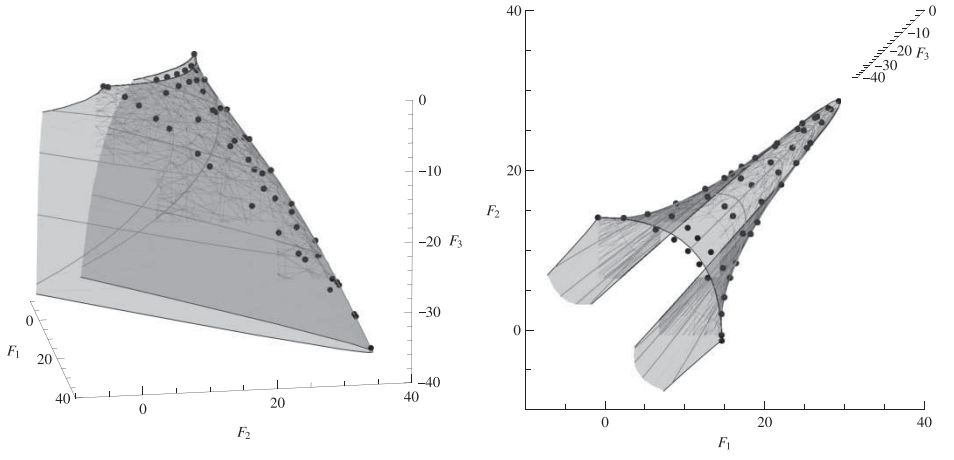

Figure 13. Pareto front generation for the Comet test case: front view (left); top view (right).

Generating a well-distributed Pareto set on the entire highly variable surface is a challenge for the existing methods. As illustrated by Deb *et al.* (2005), the use of classical generating methods is very time-consuming and leads to solving a significantly large number of redundant single-objective optimization problems. Yet the DSD algorithm is effective enough in this case. The Pareto set is represented in Figure 13 and characterized by $E = 1.49$, $N_o = 611$ and $P_s = 100$. The anchor points have coordinates: $(-14, 2, 3)$, $(-37, -36, 35)$ and $(2, -14, 3)$.

6. Conclusion

The DSD algorithm has been represented in this article together with an efficient filtering procedure for higher-dimension problems. The method provides an efficient way for quasi-evenly generating a Pareto set in a quite arbitrary multidimensional formulation without *a priori* knowledge from the DM. The approach is based on shrinking a search domain, the orientation of which can be easily controlled in space. The method is formulated in a way which may be suitable for practical implementation. The proposed approach requires the inversion of only one matrix, which makes the algorithm computationally efficient. The approach has been successfully tested on five different cases with specific peculiarities. The results obtained may demonstrate the method's competency in generating all parts of the Pareto frontier even if the anchor points are not unique and they number less than the objective functions. It is worth noting that the DSD algorithm can be relatively easily parallelized.

Acknowledgements

Tohid Erfani gratefully acknowledges the research scholarship awarded by the School of Mechanical, Aerospace and Civil Engineering, University of Manchester. The authors are listed in alphabetical order.

References

Athan, T. and Papalambros, P., 1996. A note on weighted criteria methods for compromise solutions in multi-objective optimization. *Engineering Optimization*, 27 (2), 155–176.

Das, I., 1999. An improved technique for choosing parameters for Pareto surface generation using normal-boundary intersection. *In: Proceedings of the third world congress of structural and multidisciplinary optimization (WCSMO-3)*, 17–21 May 1999, University of Buffalo, Amherst, NY. Berlin: Springer.

Das, I. and Dennis, J., 1997. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural and Multidisciplinary Optimization*, 14 (1), 63–69.
Das, I. and Dennis, J., 1998. Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8, 631.
Deb, K., 2001. *Multi-objective optimization using evolutionary algorithms*. New York: Wiley.
Deb, K., *et al.*, 2005. Scalable test problems for evolutionary multi-objective optimization. *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*. Berlin: Springer, 105–145.
Fantini, P., 2007. Effective multiobjective MDO for conceptual design – an aircraft design perspective. Thesis (PhD). Cranfield University, UK.
Hoffman, K. and Kunze, R., 2002. *Linear algebra*. Englewood Cliffs, NJ: Prentice Hall.
Koski, J., 1985. Defectiveness of weighting method in multicriterion optimization of structures. *Communications in Applied Numerical Methods*, 1 (6), 333–337.
Messac, A., 1996. Physical programming: effective optimization for computational design. *AIAA Journal*, 34 (1), 149–158.
Messac, A., 2000. From dubious construction of objective functions to the application of physical programming. *AIAA Journal*, 38 (1), 155–163.
Messac, A., Ismail-Yahaya, A., and Mattson, C., 2003. The normalized normal constraint method for generating the Pareto frontier. *Structural and Multidisciplinary Optimization*, 25 (2), 86–98.
Messac, A. and Mattson, C., 2002. Generating well-distributed sets of Pareto points for engineering design using physical programming. *Optimization and Engineering*, 3 (4), 431–450.
Messac, A. and Mattson, C., 2004. Normal constraint method with guarantee of even representation of complete Pareto frontier. *AIAA Journal*, 42 (10), 2101–2111.
Messac, A., Puemi-Sukam, C., and Melachrinoudis, E., 2000a. Aggregate objective functions and Pareto frontiers: required relationships and practical implications. *Optimization and Engineering*, 1 (2), 171–188.
Messac, A., *et al.*, 2000b. Ability of objective functions to generate points on nonconvex Pareto frontiers. *AIAA Journal*, 38 (6), 1084–1091.
Michalewicz, Z., 1996. *Genetic algorithms + data structures = evolution programs*. Berlin: Springer-Verlag.
Miettinen, K., 1999. *Nonlinear multiobjective optimization*. Berlin: Springer-Verlag.
Sanchis, J., *et al.*, 2008. A new perspective on multiobjective optimization by enhanced normalized normal constraint method. *Structural and Multidisciplinary Optimization*, 36 (5), 537–546.
Shukla, P. and Deb, K., 2007. On finding multiple Pareto-optimal solutions using classical and evolutionary generating methods. *European Journal of Operational Research*, 181 (3), 1630–1652.
Steuer, R., 1986. *Multiple criteria optimization: theory, computation, and application*. New York: Wiley.
Tanaka, M., *et al.*, 1995. GA-based decision support system for multicriteria optimization. *In*: *Proceedings of the IEEE international conference on systems, man and cybernetics: intelligent systems for the 21st century*, Vol. 2. 22–25 October, Vancouver, Canada. New York: IEEE Press.
Utyuzhnikov, S., 2010. Multi-objective optimization: quasi-even generation of Pareto frontier and its local approximation. *In*: J. Varela and S. Acuna, eds. *Handbook of optimization theory: decision analysis and application*. New York: Nova Science.
Utyuzhnikov, S., Fantini, P., and Guenov, M., 2005. Numerical method for generating the entire Pareto frontier in multiobjective optimization. *Proceedings of Eurogen'2005*, 12–14 September 2005, Munich, Germany.
Utyuzhnikov, S., Fantini, P., and Guenov, M., 2009. A method for generating a well-distributed Pareto set in nonlinear multiobjective optimization. *Journal of Computational and Applied Mathematics*, 223 (2), 820–841.
Vincent, T. and Grantham, W., 1981. *Optimality in parametric systems*. New York: Wiley.

Appendix A. Calculation of matrix D

According to the Gram–Schmidt procedure (Hoffman and Kunze 2002), the matrix $D = (\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n)$ can be calculated as follows:

$$D = \text{Gram-Schmidt}(\mathbf{l}_0, \mathbf{n}, \mathbf{e}_3, \mathbf{e}_4, \ldots, \mathbf{e}_n),$$

$$\begin{aligned}
\mathbf{d}_1 &= \mathbf{l}_0, \\
\mathbf{d}_2 &= \frac{\mathbf{n} - (\mathbf{n}\cdot\mathbf{l}_0)\mathbf{l}_0}{\sqrt{1+(\mathbf{n}\cdot\mathbf{l}_0)^2}}, \\
\mathbf{d}_k &= \frac{\mathbf{e}_k - \sum_{i=1}^{k-1}(\mathbf{e}_k\cdot\mathbf{d}_i)\mathbf{d}_i}{\sqrt{1+\sum_{i=1}^{k-1}(\mathbf{e}_k\cdot\mathbf{d}_i)^2}}, \quad (k=3,\ldots,n),
\end{aligned}$$

where $(\cdot)$ denotes the dot product. In addition, $\mathbf{e}_k = (\delta_{1k}, \delta_{2k}, \ldots, \delta_{nk})$ where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$ $(i, j = 1, \ldots, n)$.