Accepted Manuscript

A method for generating a well-distributed Pareto set in nonlinear multiobjective optimization

S.V. Utyuzhnikov, P. Fantini, M.D. Guenov

Please cite this article as: S.V. Utyuzhnikov, P. Fantini, M.D. Guenov, A method for generating a well-distributed Pareto set in nonlinear multiobjective optimization, *Journal of Computational and Applied Mathematics* (2008), doi:10.1016/j.cam.2008.03.011

A Method for Generating a Well-Distributed Pareto Set in Nonlinear Multiobjective Optimization

S.V. Utyuzhnikov[1,2], P. Fantini[1], and M.D. Guenov[1]

[1]*Department Aerospace Engineering, School of Engineering Cranfield University,*

Cranfield, Bedfordshire, MK43 0AL, UK

e-mail: m.d.guenov@cranfield.ac.uk

[2]*School of Mechanical, Aerospace & Civil Engineering,*

University of Manchester, Sackville Street, M60 1QD

e-mail: s.utyuzhnikov@manchester.ac.uk

Abstract

Presented is a method for generating a well-distributed Pareto set in nonlinear multiobjective optimization. The approach shares conceptual similarity with the Physical Programming based method, the Normal-Boundary Intersection and the Normal Constraint methods, in its systematic approach investigating the objective space in order to obtain a well-distributed Pareto set. The proposed approach is based on the generalization of the class-functions which allows the orientation of the search domain to be conducted in the objective space. It is shown that the proposed modification allows the method to generate an even representation of the entire Pareto surface. The generation is performed for both convex and non-convex Pareto frontiers. A simple algorithm has been proposed to remove local Pareto solutions. The suggested approach

has been verified by several test cases, including the generation of both convex and concave Pareto frontiers.

MSC: 65K05, 90C29, 90C30.

Key words: Multiobjective optimization, Pareto solution, Pareto set, physical programming.

1. Introduction

In real industrial Multidisciplinary Design and Optimization (MDO) setting many different criteria such as high performance, low through-life cost and manufacturability have to be satisfied simultaneously. Finding a reasonable compromise is not a trivial problem. It requires the generation of a sufficiently large number of representative solutions in order to perform trade-off studies.

Pareto optimization in Engineering Design

It is natural to exclude from consideration any design solution which can be improved without deterioration of any discipline and violation of the constraints; in other words, a solution which can be improved without any trade-off. This leads to the Pareto optimal set [21]. In practice, however, the decision maker (DM) would select the ultimate solution among the Pareto set on the basis of additional (often subjective) requirements. Considering the size and complexity of industrial multiobjective optimization problems such as aircraft conceptual design, it follows that the efficiency and effectiveness of the optimization methods used for trade-off analysis are of paramount importance.

In spite of the existence of many numerical methods for vector optimization, only a few are suitable for real-design industrial applications, especially for preliminary design, due to time-limit constraints.

The classical vector optimization methods reduce the multiobjective optimization problem to a scalar optimization of an aggregated objective function (AOF) which includes a combination of objective functions. The most popular AOF is represented by a linear combination of the objective functions [21]. Recent reviews of the methods employed in multiobjective programming can be found in [3, 8]. In contrast to the classical, preference-based, approaches, another class of evolutionary methods such as the genetic-based algorithms generate a set of Pareto solutions simultaneously (see, e.g., [3, 7]). This class of methods seems to be very promising for solving multiobjective problems. Unfortunately these do not guarantee either the generation of a well-distributed Pareto set or the representation of the entire Pareto frontier.

In real industrial design the DM is able to consider only a few possible solutions (Pareto points). In such a context, it is important to have an even distribution of Pareto points to obtain maximum information on the Pareto surface at minimum computational cost. A well-distributed Pareto set can also be a good foundation for visualizing the Pareto frontier. It can substantially simplify the work of the DM. A comprehensive analysis of different approaches for approximating and visualizing a Pareto surface is given in [11]. A recent review of the up-to-date methods to approximate the Pareto set is available in [22]. These questions as well as the analysis of the ways of the DM's search for a preferable solution are beyond the scope of the present work. The rest of this section outlines the state of the art in generating well distributed Pareto sets, which is the main focus of this paper.

Survey of Pareto quasi-even set generators

Das and Dennis [5] showed that an even spread of the weights in the AOF does not necessary result in an even distribution of the points in the Pareto set. Also, the spread of the points strongly depends on the relative scaling of the objectives. In [19], necessary and sufficient conditions for an AOF were obtained for capturing any Pareto point. In the authors' knowledge, there are only three methods which are able to provide an even distribution of a whole Pareto surface [17]:

a) The Normal-Boundary Intersection (NBI) Method was developed by Das and Dennis [4, 6]. The method might be nonrobust. Moreover, it generates non-Pareto and locally Pareto points that requires a filtering procedure [16].

b) The new Normal Constraint (NC) Method [16, 17] developed recently looks very promising. As the NBI method, it has a clear geometrical interpretation. Both methods are based on the well-known fact that a Pareto surface belongs to the boundary of the feasible space towards minimization of the objective functions [21]. So-called *anchor points* are obtained in the feasible objective space, first. An anchor point corresponds to the optimal value of one and only one objective function in the feasible space. Thus, *n* objective functions give *n* anchor points. Second, the *utopia plane* passing through the anchor points is considered. In both the NBI and NC methods, the Pareto surface is then obtained by the intersection of lines normal to the utopia plane and the boundary of the feasible space. The single optimization problem, used in the NC, is based only on inequality constraints. This modification makes the method flexible and stable. In both methods an even distribution of Pareto points is provided by the even distribution of the lines orthogonal to the utopia plane. Both methods may fail

to generate Pareto solutions over the entire Pareto frontier [17] in multidimensional case when the number of objectives exceeds two. The recent modification of the NC [17] eliminates this drawback and guarantees the complete representation of a Pareto frontier. Nevertheless, both methods may generate non-Pareto and locally Pareto solutions [16]; the NC apparently being the more effective of the two [17].

c) The Physical Programming (PP) Method was proposed by Messac in [14]. This method also generates Pareto points on both convex and non-convex Pareto frontiers as shown in [18]. The method does not use any weight coefficients and allows one to take into account the DM experience directly. In this sense, it appears to be the most interesting method for practical applications under the above stated conditions. In the PP, the designer assigns each objective to one of four categories (class-functions). The optimization is based on minimization of an aggregate preference function determined by the preference functions (class-functions) with the preferences set *a priori*. The notion of the generalized Pareto optimal solution is introduced in the PP-based method [20] on the basis of the PP class-functions. To provide a well-distributed Pareto set, the off-set strategy is introduced in the PP-based algorithm in [18]. The algorithm includes a few free parameters. Some evaluations of these parameters are given which nevertheless do not fully remove the uncertainties in their determination. To avoid the set-off strategy, an alternative approach, based on the generalized class-functions, is proposed in the current paper.

Aims and Structure of the Paper

The objective of this research is to develop an efficient method for generating a well representation of the entire Pareto frontier in an arbitrary case. In order to achieve this

aim we have undertaken to modify the PP-based method, which also has allowed us to combine the advantages of the PP, NBI and NC methods.

The rest of the paper is organised as follows. Section 2 contains the standard formulation of the nonlinear multiobjective optimization problem and the definition of Pareto optimum. A more detailed introduction to the Physical Programming method and the original PP approach to generate a well distribute Pareto set is given in sections 3 and 4 respectively, since we utilize some of the basic concepts. Our proposed modification is described in section 5. This includes the generalization of the class functions in order to shrink the search domain and make its location in space more optimal. The algorithm for obtaining a quasi-even distribution of the Pareto set is described and analysed. A generalization of the notion of an anchor point is also suggested. It is shown that the proposed modification of the PP-based method does not provide non-Pareto solutions while local Pareto solutions may be easily recognized and removed. The method is evaluated in section 6 by six test cases. The numerical solutions are obtained using the MATLAB optimization software. Finally, conclusions are drawn and future work outlined.

2. Multiobjective optimization problem. Pareto optimization

Mathematically, the multiobjective optimization problem can be formulated as a vector nonlinear optimization problem under constraints as follows. It is assumed that an optimization problem is described in terms of a design variable vector $\mathbf{x} = (x_1, x_2, ..., x_N)^T$ in the design space $\mathbf{X} \in R^N$. A function $f \in R^M$ evaluates the quality of a solution by assigning it to an objective vector $\mathbf{y} = (y_1, y_2, ..., y_M)^T$ ($y_i = f_i(\mathbf{x})$,

f_i: $R^N \rightarrow R^1$, $i = 1, 2, ..., M$) in the objective space $\mathbf{Y} \in R^M$. Thus, X is mapped by f onto Y: $X \mapsto Y$. A multiobjective optimization problem may be formulated in the following form:

$$\text{Minimize } [y(x)] \tag{1}$$

subject to K inequality constraints

$$g_i(x) \leq 0, \qquad i = 1, 2, ..., K \tag{2}$$

and P equality constraints

$$h_j(x) = 0, \qquad j = 1, 2, ..., P \tag{3}$$

The feasible design space X^* is defined as the set $\{x \mid g_j(x) \leq 0, j = 1, 2, ..., K;$ and $h_i(x) = 0, i = 1, 2, ..., P\}$. The feasible criterion (objective) space Y^* is defined as the set $\{Y(x) \mid \mathbf{x} \in \mathbf{X}^*\}$. The feasibility means no constraint is violated.

A design vector $\mathbf{a}$ ($\mathbf{a} \in \mathbf{X}^*$) is called a Pareto optimum iff it does not exist any $\mathbf{b} \in \mathbf{X}^*$ such that

$$\mathbf{y}(\mathbf{b}) \leq \mathbf{y}(\mathbf{a}) \text{ and exist } l \leq M: y_l(\mathbf{b}) < y_l(\mathbf{a}).$$

A design vector is called a local Pareto optimum if it is a Pareto optimum within its some neighbourhood.

Hereafter it is supposed that all vectors are considered in the appropriate Euclidean spaces. Thus, the trade-off analysis can be formulated as a vector nonlinear optimization problem under constraints. Generally speaking, the solution of such a problem is not unique since each Pareto point is a solution of the multiobjective optimization problem.

3. The Physical programming method

In the PP method, objective functions are treated as design metrics. Each design metric $F_i(x)$ is referred to one of possible (determined in advance) utility dimensionless functions $\overline{F}_i(F_i(\mathbf{x}))$ called *class functions*. The determination of the class-functions reflects the qualitative classifications of possible preferences. The preferences are split into two major groups: hard class-functions, *H* and soft ones, *S*. The former reflects the presence of constraints (2) and (3), while the latter reflects the preferences settled for objective functions which may include subjective preferences of the DM based on his/her experience and intuition.

Class functions

Four soft class-functions are considered: class *1S* (smaller is better) where an objective is to be minimized; the mirror function *2S* (larger is better) where an objective is to be maximized; class *3S* (value is better) where an objective is to be close to a particular preferred value; and class *4S* (range is better) which is similar to the *3S* function, but the preferable value lies in some range rather than corresponding to some value selected *a priori*. All soft class-functions are to be positive and dimensionless. The argument of a soft class-function is subdivided by the DM into different preference ranges: highly desirable, desirable, tolerant, undesirable, highly undesirable and unacceptable. Such subdivision is a part of the approximation of the class-functions and allows the DM to exploit her/his own experience. The values of the class-functions at the boundaries of the ranges are fixed. Therefore, scaling between different objective functions is automatically provided.

The qualitative behaviour of the *S* class-functions is given in Figure 1. The class-functions *1S* and *2S* tend to zero if the argument tends to minus or plus infinity, accordingly. Each of the class-functions *3S* and *4S* has one and only one minimum point. All class-functions have the same values at the boundary points of the preference ranges regardless of the type of a design metric. Only the locations of the ranges depend on a metric. The presence of the class-functions *3S* and *4S* does not violate the generality of formulation (2)-(4). The optimization problem can be reformulated as a minimization problem (2)-(4). The PP lexicon makes the formulation closer to the formulation of real design optimization problems. In fact, the whole approach is more reflective of real life [14]. In particular, classes *3S* and *4S* are very common in practical cases, when it is impossible to determine if a smaller or a larger design metric is better or worse.

It is easy to see from Figure 1 that if a range is more desirable, then the value of the class-function is smaller. Therefore the preference between the ranges is enforced. This means that the highly undesirable region is significantly worse than the undesirable region. In turn, the undesirable region is significantly worse than the tolerable region, and so on. This demand is reflected by the following conditions [14]:

$$\Delta_k \overline{F}_i > n_{sc} \Delta_{k-1} \overline{F}_i, \qquad\qquad 1 \le k \le 4, \qquad (4)$$

where $\Delta_k \overline{F}_i = \overline{F}_i(F_{i,k+1}) - \overline{F}_i(F_{i,k})$, $\Delta_0 \overline{F}_i = \overline{F}_i(F_{i,1})$; $F_{i,k}$ $(k = 1, 2, ..., 5)$ are the boundary points of the preference ranges for a metric F_i; I is a number of a soft-class function; n_{sc} is the number of soft design metrics.

Aggregate function. Single –objective optimization

Ultimately, the problem is reduced to the following optimization problem under constraints:

$$\min G(\mathbf{x}) = \frac{1}{n_{sc}} \log_{10} [\sum_{i=1}^{n_{sc}} \overline{F_i}(F_i)], \tag{5}$$

subject to:

$$
\begin{aligned}
F_i(\mathbf{x}) &\leq F_{i,5} && \text{(for 1}S\text{)} && (6)\\
F_i(\mathbf{x}) &\geq F_{i,5} && \text{(for 2}S\text{)}\\
F_{i,5}^{(L)} \leq F_i(\mathbf{x}) &\leq F_{i,5}^{(R)} && \text{(for 3}S\text{)}\\
F_{i,5}^{(L)} \leq F_i(\mathbf{x}) &\leq F_{i,5}^{(R)} && \text{(for 4}S\text{)}
\end{aligned}
$$

and constraints related with the hard classes

$$
\begin{aligned}
F_i(\mathbf{x}) &\leq F_{i,M} && \text{(for 1}H\text{)} && (7)\\
F_i(\mathbf{x}) &\geq F_{i,m} && \text{(for 2}H\text{)}\\
F_{i,m} \leq F_i(\mathbf{x}) &\leq F_{i,M} && \text{(for 3}H\text{)},
\end{aligned}
$$

where $F_{i,m}$ and $F_{i,M}$ are the lower and upper limits respectively.

The equality constraints (3) can be changed by double inequality constraints of *1H* and *2H* types:

$$
\begin{aligned}
h_j(\mathbf{x}) &\leq 0,\\
h_j(\mathbf{x}) &\geq 0
\end{aligned}
$$

Constraints on the design variables *x* are considered as ordinary constraints. The logarithm in (5) is only used to diminish the difference between the maximal and minimal values that may affect the convergence of an iteration algorithm. It is easy to see that the more "undesirable" the range is, the more it affects the value of the AOF. In this sense, there is some analogy with the method of penalty functions (see, e.g., [21]).

In turn, it is possible to show that the preferences play somewhat the same role as weights in the weighted-sum method [14]. In (5) and henceforth, we assume that in the formulation $\overline{F}_i(F_i)$ the appropriate category of the class-function is used. This means that if, e.g., an objective F_i is to be minimized then *1S* class-function is utilised.

As Messac reports in [14], a robust implementation of the PP necessitates that problem (5) - (7) is implemented in such a way to avoid the introduction of additional minima. To guarantee this, each class-function must be a strictly convex function [14], that is:

$$\frac{d\overline{F}_i}{dF_i} \subset C^1, \quad \frac{d^2\overline{F}_i}{dF_i^2} > 0 \tag{8}$$

Yet, possible class-functions satisfying condition (8) are not unique. It is possible to see that both high-order polynomial and cubic spline functions are not acceptable. Messac [14] derived the class-functions that satisfy all requirements mentioned above. In [24] we have suggested an alternative method for approximating the class-functions which also satisfy all the requirements, but it is based on an exponential representation. If *1S* class-functions are only considered, then the AOF G in (5) increases monotonically with respect to each objective F_i (i = 1, …, n_{sc}). Property (8) is sufficient [23] for any solution of the single-objective problem (5), (6), (7) to be a Pareto optimum solution for problem (1), (2), (3).

It is easy to see that the Hessian of $G(F_1,...,F_{n_{sc}})$ is positive semidefinite under conditions (8). Hence, G is a convex function. Thus, if the search domain X^* is convex, the optimization problem is reduced to the convex optimization without any local minima [2].

12

In contrast to the standard preference-based methods, the PP approach includes some uncertainness related to the determination of the preference intervals, which makes it consistent with real-life engineering design. In such a way the PP was successfully used for solving many MDO problems (see, e.g., [12]).

It should be noted that the standard definition of the Pareto optimal solution is not applicable to the *3S* and *4S* class-functions. To avoid this limitation the notion of a generalized Pareto optimal solution is introduced in [20]. The correspondence between the standard and generalized Pareto optimal solutions is provided in [24]. The implementation of the notion of the generalized Pareto optimal solution allows us to reduce the optimization problem formulated for the *3S* and *4S* class-functions to the standard statement of problem (1) – (3) [24].

4. Generation of an evenly distributed Pareto set – the original PP-based approach.

As stated in the introduction, in principle the PP-based method is capable of generating an evenly distributed Pareto set. The entire algorithm is given in [18]. The original approach is briefly described below, followed by the description of our proposed modification. The algorithm can be also generalized for the *2S – 4S* class-functions.

Payoff matrix. Anchor points

Let us define the payoff table *T* [21] as follows:

$$T = \begin{bmatrix} F_{1|\min} & F_{12} & ... & F_{1n_{sc}} \\ F_{21} & F_{2|\min} & ... & F_{2n_{sc}} \\ ... & ... & ... & ... \\ F_{n_{sc}1} & F_{n_{sc}2} & ... & F_{n_{sc}|\min} \end{bmatrix}, \tag{9}$$

In the payoff table T, an i-th row represents the coordinates of an anchor point $\boldsymbol{\mu}_i^*$ corresponding to the solution of single-optimization problem min F_i in the feasible criterion space $\mathbf{Y}^*$ for $M = n_{sc}$.

In the feasible space $\mathbf{Y}^*$ a hypercube H limiting the search domain is defined in the following manner. We first set the pseudo nadir point [21], $F_{i|\max} = \max_j F_{ij}$, which corresponds to the maximum i-th component among all anchor points. Then, the hypercube H is represented as follows:

$$H = [F_{1|\min} F_{1|\max}] \times [F_{2|\min} F_{2|\max}] \times ... \times [F_{n_{sc}|\min} F_{n_{sc}|\max}].$$

For the sake of simplicity, consider the 2D case where there are only two design metrics. We assume that each of the design metrics belongs to class *IS*, otherwise, as mentioned above, the problem can be reduced to the minimization problem via consideration of the generalized Pareto solution [20, 24]. Following [18], for each of the design metrics let us introduce the vector of pseudo-preferences P_i:

$$P_i \equiv (F_{i,1}, F_{i,2}, F_{i,3}, F_{i,4}, F_{i,5})^T = F_i^{(0)}(1,1,1,1,1)^T + \delta_i(0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1)^T, \tag{10}$$

where $\delta_i = (F_{i|\max} - F_{i|\min})/n_d$ and $F_i^{(0)}$ is a free parameter. The parameter n_d defines the box size.

Search domain

The regions $F_1 > F_{1,5}$ and $F_2 > F_{2,5}$ are considered as unacceptable. Thus, the box

$D = [F_{1,1} F_{1,5}] \times [F_{2,1} F_{2,5}]$ defined by the pseudo-preferences limits the search domain

from the right and upper sides leaving it in the other directions unlimited for a formal

search. Changing the free vector $\mathbf{F}^{(0)} = (F_1^{(0)}, F_2^{(0)}, ..., F_{n_{sc}}^{(0)})^T$, it is possible to shift the

box D in the hypercube H to seek Pareto solutions. The current location of the box D

determines a possible location of a Pareto point since the Pareto points outside D (more

precisely − higher or on the right of D) are excluded from the current consideration. To

a certain extent this method operates in a manner similar to the ε-constraint method

[21]. In PP approach space reduction is simultaneously performed for all objectives. For

example, by moving the box to the lower-right angle of the hypercube H (in 2D case),

we give preferences to low values of the second objective at the expense of high values

of the first objective.

The algorithm for shifting the box D over the space $\mathbf{Y}^*$ to seek the Pareto frontier

is given in [18]. To achieve this, the free vector $\mathbf{F}_i^{(0)}$ is specially chosen to move the box

along lines parallel to a diagonal of the hypercube H passing trough the lower-right

angle and the upper-left ones. In contrast to the search of a single Pareto solution, in

this, off-set, strategy the pseudo-preferences are only used to determine the AOF (5).

They are automatically obtained from formula (10) which guarantees that if it exists, the

appropriate Pareto solution would appear inside the sub-space: $F_i \leq F_{i,5}$ $(i = 1,...,n_{sc})$.

Thus, the size of the box D mainly affects the efficiency of solving the single-objective

optimization problem for the AOF rather than the solution. For example, a variation of

the parameter n_d changes the AOF, however, the solution has to be inside of the search

sub-space. If the parameter n_d is not appropriately chosen, it may lead to either the increase of the number of iterations in the numerical algorithm used for minimizing the AOF, or to divergence. A few free parameters are introduced to control the displacement of the box. Some evaluations of these parameters are given in [18] which nevertheless do not fully remove the uncertainties in their determination. One can note that, the approach described in [18] is only applicable to the minimization problem when only the class function *IS* is involved.

5. Generation of an evenly distributed Pareto set. Modified method based on a directed search domain.

We suggest another strategy to seek the Pareto frontier. Further, it is assumed that only a minimization problem is considered (all class functions are *IS*) and the problem is solved in the objective space **Y**. The generalization on the arbitrary case (with the other class-functions) is given in [24]. The core of the proposed method is the algorithm for generating a reduced and easily directed search domain. To realize it, we first assign a coordinate system to the search domain and then conduct an affine transform of the coordinate system in order to shrink the search domain. The displacement of the search domain in the objective space is related to the utopia plane.

Algorithm for displacement of the search domain. Modified anchor points

Similar to the NC method, let us consider the utopia plane created by anchor points $\boldsymbol{\mu}_i^*$. It is well known that any point **p** belonging to the interior of a convex polygon spanned by n_{sc} vertexes $\boldsymbol{\mu}_i^*$ can be represented as follows:

$$\mathbf{p} = \sum_{i=1}^{n_{sc}} \alpha_i \boldsymbol{\mu}_i^* \qquad (11)$$

where the parameters α_i must satisfy the following conditions:

$$0 \le \alpha_i \le 1, \qquad (12)$$

$$\sum_{j=1}^{n_{sc}} \alpha_j = 1.$$

Since our approach utilizes the notion of *anchor point,* an important general remark on its definition should be made. As it was mentioned above, the standard definition assumes that an anchor point $\boldsymbol{\mu}_i^*$ corresponds to the solution of the single-optimization problem *min F_i* in the feasible criterion space $\mathbf{Y}^*$ (see, e.g., [17]). This definition allows the anchor point corresponding to some objectives to be non-unique. Furthermore, it may not even belong to the boundary of $\mathbf{Y}^*$. Such an example is given below. We suggest the following specification which guarantees the uniqueness of the anchor point for each objective. If the solution of the problem $\boldsymbol{\mu}_i^* = \mathbf{F}(\mathbf{X}^{i*})$ $\{\mathbf{X}^{i*} : \mathbf{X}^{i*} = \arg\min_{\mathbf{Y}^*} F_i\}$ is not unique, then the point corresponding to the minimal values of the other design metrics ought to be chosen. This may lead to the problem of trade-off minimization for the remaining objectives. To avoid this, we perform a priority minimization procedure. First, F_i is to be minimized, then F_{i+1} and so on up to F_{i-1}. The following lexicographic-based prioritization is introduced in a circular order: $i+1$, $i+2$, ..., n_{sc}, 1, 2, ..., $i-1$. A *k-th* prioritization assumes that the *k-th* minimization must not violate all the previous *k-1* ones. It is easy to prove that all anchor points belong to the Pareto frontier because they are on the boundary of the feasible space Y^* and no objective can be improved without deterioration of any other objective.

The free vector $\mathbf{F}^{(0)} = (F_1^{(0)}, F_2^{(0)}, ..., F_{n_{sc}}^{(0)})^T$ in (10) is determined in the following way. Let us consider the box D. The box D is shifted in such a way that its vertex M corresponding to the maximal values of the design metrics $(M : M = (F_{1|\max}, F_{2|\max}, ..., F_{n_{sc}|\max}))$ lies in the utopia plane (see Figure 2). This means

$$\mathbf{F}^{(0)} = \sum_{i=1}^{n_{sc}} \alpha_i \boldsymbol{\mu}_i^* + \mathbf{g}^{(0)}, \tag{13}$$

where

$$\mathbf{F}^{(0)} = (F_1^{(0)}, F_2^{(0)}, ..., F_{n_{sc}}^{(0)})^T,$$
$$\mathbf{g}^{(0)} = s_n(\delta_1, \delta_2, ..., \delta_{n_{sc}})^T.$$

Here, we set $s_n = -1$. In some cases the search domain is required to be in the opposite side of the utopia plane. It can be easily reached by setting $s_n = 0$. Then, we obtain that the vertex $(F_{1|\min}, F_{2|\min}, ..., F_{n_{sc}|\min})$ belongs to the utopia plane and the search box D is located in the other semi-space limited by the utopia plane.

An even distribution of the coefficients α_i produces a quasi-even distribution of the Pareto set. In contrast to the NC and NBI methods, the approach described below allows us to generate the complete Pareto frontier considering only non-negative coefficients α_i from (12).

There are different ways for generating an even distribution of the coefficients α_i. One possible algorithm is given in [17] where the following induction procedure is suggested. First, an even distribution of the coefficient α_1 is considered. From the normalization condition, the sum of the remaining coefficients $\sum_{j=2}^{n_{sc}} \alpha_j$ equals to

$1 - \alpha_1$ for each selected value of α_1. Then, an even distribution of the coefficient α_2 is considered for each of these variants and so on until either the last coefficient $\alpha_{n_{sc}}$ is reached or the sum of the coefficients already determined equals 1. In the former case the last coefficient is determined uniquely by the normalization condition; in the latter case the remaining coefficients are set equal to zero.

The original algorithm assumes either offsetting the diagonal translation of the box D or magnifying it. This procedure may increase drastically the number of domains to be investigated since it is not fully determined and depends on the location of the Pareto set to be found. It is also shown below that the distribution of the Pareto set may be sensitive to the displacement of the box D along the utopia plane especially if the Pareto frontier is concave. To overcome these problems, a generalization of the class functions is suggested. It allows us to shrink the search domain substantially.

Generalized class functions. Affine transform

In order to shrink the search domain limited by box D, we propose the introduction of generalized class-functions as follows:

$$\overline{F}_i(\tilde{F}_i) \ (i = 1, \ ..., \ n_{sc}), \tag{14}$$

where $\tilde{F}_i$ is defined by an affine transform

$$\tilde{F}_i = \sum_1^{n_{sc}} F_j B_{ji}, \quad (i = 1, \ ..., \ n_{sc}). \tag{15}$$

Thus, each generalized class-function can depend on a linear combination of the objective functions rather than a single objective function. In the general case, affine transform (15) can be represented as consecutive combination of shear and scaling

operations. Yet in our case the primary aim of transform (15) is shearing the objective space to make the aggregate function to be localized around a chosen direction.

It is worth noting here that an affine transform retains the convexity property [2], therefore the AOF G in (5) remains convex. The affine transform (15) does not depend on the concrete representation of the class-functions as long as conditions (8) are satisfied.

In the objective space $\mathbf{Y}$, let us introduce a Cartesian coordinate system $\{\mathbf{e}_i\}$ $(i = 1, ..., n_{sc})$ with the origin at point M and axes aligned with the edges of box D. Thus, this coordinate system is assigned to box D. Assume that the axes of the Cartesian system are directed towards the increase of the objective functions. Then, affine transform (15) uniquely determines the assigned coordinate system having the basis vectors:

$$\mathbf{a}_i = \sum_1^{n_{sc}} A_{ij} \mathbf{e}_j, \quad (i = 1, ..., n_{sc}), \tag{16}$$
$$A^{-1} = B$$

where $\mathbf{e}_j$ ($j = 1, ..., n_{sc}$) are the basis vectors of the original coordinate system. It is easy to see that the assigned coordinate system, in turn, uniquely determines transform (15). Then, we are able to control the search domain choosing the appropriate assigned coordinate system.

Shrinking of search domain

The search domain can be changed as shown in Figure 3. In particular, it is possible to choose basis vectors $\mathbf{a}_i$ ($i = 1, ..., n_{sc}$) which form an angle γ_c to some selected direction $\mathbf{l}$. The 2D case is shown in Figure 3. Matrixes $\mathbf{A}$ and $\mathbf{B}$ are determined as follows:

$$A = \begin{pmatrix} \cos\gamma_- & \sin\gamma_- \\ \cos\gamma_+ & \sin\gamma_+ \end{pmatrix}, \quad B = \frac{1}{\sin 2\gamma_c} \begin{pmatrix} \sin\gamma_+ & -\sin\gamma_- \\ -\cos\gamma_+ & \cos\gamma_- \end{pmatrix}, \quad (17)$$

where $\gamma_+ = \gamma_n + \gamma_c$, $\gamma_- = \gamma_n - \gamma_c$, $\mathbf{l} = (\cos\gamma_n, \sin\gamma_n)^T$.

In the general case of $R^{n_{sc}}$, the following algorithm is proposed. Suppose that all basis vectors $\mathbf{a}_i$ of the new coordinate system form an angle γ_c with some unit vector $\mathbf{l}$. Thus, the vectors $\mathbf{a}_i$ must satisfy the following conditions:

$$(\mathbf{a}_i, \mathbf{l}) = \cos\gamma_c \qquad (i = 1, ..., n_{sc}), \qquad (18)$$

where $(.,.)$ means a scalar product. Thus, all vectors $\mathbf{a}_i$ belong to the lateral area of the hypercone with the apex at point M, angle γ_c and axis along vector $\mathbf{l}$ (Figure 4). In turn, it is important to guarantee a spread distribution of these vectors. At least, the basis created by these vectors must not vanish. It appears possible to obtain even a fully uniform distribution of the basis vectors of such a polyhedral cone in the Euclidean space of an arbitrary finite dimension.

First, suppose that $\mathbf{l}$ is directed in such a way that it has equal coordinates in the original Cartesian coordinate system $\{\mathbf{e}_i\}$ $(i = 1, ..., n_{sc})$:

$$\mathbf{l} = \mathbf{l}_0, \qquad (19)$$
$$\mathbf{l}_0 = (l_0, l_0, l_0, ..., l_0)^T$$

Since $\mathbf{l}$ is a unit vector, we obtain its coordinates:

$$l_0 \equiv \cos\gamma_0 = \frac{1}{\sqrt{n_{sc}}} \qquad (20)$$

Formula (20) immediately follows from the fact that $(\mathbf{l}, \mathbf{l}) = 1$. Each basis vector $\mathbf{a}_i$ can be determined in the plane created by the vectors $\mathbf{e}_i$ and $\mathbf{l}_0$ (see Figure 4). It is possible to show that

$$\mathbf{a}_i = \frac{\sin\gamma_c}{\sin\gamma_0}\mathbf{e}_i + \frac{\sin(\gamma_0 - \gamma_c)}{\sin\gamma_0}\mathbf{l}_0 \tag{21}$$

From (15), (20) and (21) we obtain

$$A = A_0 \equiv \frac{\sin\gamma_c}{\sin\gamma_0}I + \frac{\sin(\gamma_0 - \gamma_c)\cos\gamma_o}{\sin\gamma_0}E, \tag{22}$$

where all elements of the matrix E are unities: $\|E_{ij}\| = 1$, I is the unit matrix.

Arbitrary direction of search domain

In many cases the shrinking around the lines parallel to vector $\mathbf{l}_0$ is already sufficient. Nevertheless, it is important to obtain matrix A in the general case of an arbitrary unit vector $\mathbf{l}$. For this purpose, it is enough to perform a linear transform mapping the previous configuration of the basis vectors and the directing vector in such a way that vector $\mathbf{l}_0$ is mapped onto vector $\mathbf{l}$. This is achieved by multiplying both parts of equation (21) by an orthogonal matrix R:

$$R\mathbf{l}_0 = \mathbf{l} \tag{23}$$

Then, we obtain the basis of vectors $\{\mathbf{a}_i'\}$ ($i = 1, ..., n_{sc}$) evenly distributed on the lateral area of the hypercone having the axis parallel to vector $\mathbf{l}$:

$$\mathbf{a}_i' = \frac{\sin\gamma_c}{\sin\gamma_0}\mathbf{e}_i' + \frac{\sin(\gamma_0 - \gamma_c)}{\sin\gamma_0}\mathbf{l}, \tag{24}$$

where $\mathbf{e}_i' = R\mathbf{e}_i$ are the components of the Cartesian coordinate system in which vector $\mathbf{l}$ has equalled components. It is easy to see that the columns of transition matrix R are the coordinates of vectors $\mathbf{e}_i'$ in the basis $\{\mathbf{e}_j\}$. Since the transform is orthogonal, all angles are preserved. In particular, $(\mathbf{a}_i', \mathbf{l}) = \cos\gamma_0$. Now we can obtain matrix A in the general form:

$$A = A_0 R^T = \frac{\sin\gamma_c}{\sin\gamma_0} R^T + \frac{\sin(\gamma_0 - \gamma_c)}{\sin\gamma_0} E, \tag{25}$$

where $\| E_{ij} \| = \| l_j \|$.

If $\gamma_c = \gamma_0$, then obviously $\mathbf{a}_i' = \mathbf{e}_i'$ which means that the transform becomes orthogonal and is only reduced to a turn of the original Cartesian coordinate system. As a consequence in this case, the matrix A is orthogonal and $B = A^T$.

The general presentation requires the calculation of the orthogonal matrix R, the components of which must satisfy the following additional requirements:

$$\cos\gamma_0 \sum_{j=1}^{n_{sc}} R_{ij} = l_i \tag{26}$$

Matrix R is not unique. The simplest way to obtain it is to consider the rotation from vector $\mathbf{l}_0$ to vector $\mathbf{l}$ in some Cartesian coordinate system $\{\mathbf{b}_i\}$ $(i = 1,...,n_{sc})$ related to these vectors. Suppose that in this coordinate system vectors $\{\mathbf{b}_i\}$ $(i = 3,...,n_{sc})$ are orthogonal to both $\mathbf{l}_0$ and $\mathbf{l}$. Then, in basis $\{\mathbf{b}_i\}$ vector $\mathbf{l}_0$ is mapped onto $\mathbf{l}$ by an elementary rotation matrix T_R describing the rotation in the hyperplane created by the vectors $\mathbf{l}_0$ and $\mathbf{l}$. It is easy to obtain that

$$T_R = \begin{bmatrix} (\mathbf{l}_0, \mathbf{l}) & -\sqrt{1-(\mathbf{l}_0,\mathbf{l})} & 0 & ... & 0 \\ \sqrt{1-(\mathbf{l}_0,\mathbf{l})} & (\mathbf{l}_0,\mathbf{l}) & 0 & ... & 0 \\ 0 & 0 & 1 & ... & ... \\ ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & 1 \end{bmatrix} \tag{27}$$

Then, in the original basis the rotation matrix is represented as follows:

$$R = D_R T_R D_R^T \tag{28}$$

Here, matrix D_R is the orthogonal matrix of transition from the original basis to the basis

$$\{\mathbf{b}_i\}: \quad \mathbf{b}_1 = \mathbf{l}_0, \quad \mathbf{b}_2 = \frac{\mathbf{l} - (\mathbf{l}, \mathbf{l}_0)\mathbf{l}_0}{\sqrt{1 - (\mathbf{l}, \mathbf{l}_0)^2}}, \quad \dots$$. The remaining basis vectors $\mathbf{b}_3, \mathbf{b}_4, \dots, \mathbf{b}_{n_{sc}}$ can

be obtained as follows. Let us supplement vectors $\mathbf{b}_1$ and $\mathbf{b}_2$ to the full basis by those

$n_{sc} - 2$ vectors of the original basis $\{\mathbf{e}_i\}$ that have the minimal value of

$|(\mathbf{l}, \mathbf{e}_k)| \quad (k = 1, \dots, n_{sc})$. Then, the full basis $\{\mathbf{b}_i\}$ $(i = 1, \dots, n_{sc})$ is obtained by the Gram-

Schmidt orthogonolization procedure applied to the selected $n_{sc} - 2$ vectors of the

original basis. According to this procedure each subsequent basis vector has to be

orthogonal to all the previous ones. Thus, $D_{R\,|ij} = b_{ij}$, $(i, j = 1, \dots, n_{sc})$, where b_{ij} are

coordinates of a vector $\mathbf{b}_j$.

Obviously the vectors $\{\mathbf{a}_i\}$, as well as $\{\mathbf{a}'_i\}$, $(i = 1, \dots, n_{sc})$ create a basis in $\mathbf{Y}$ which

does not vanish. The basis vectors form a search cone similar to the 2D case shown in

Figure 3. In the 2D case, $n_{sc} = 2$, $\gamma_0 = \pi/4$ and we obtain formulas (17).

The boundaries of the preference ranges are mapped in according to (15):

$$\tilde{F}_{ik} = \sum_1^{n_{sc}} F_{jk} B_{ji}, \quad (i = 1, \dots, n_{sc}; \ k = 1, \dots, 5) \tag{29}$$

Transform (14), (15) allows us to shrink the search domain by conditions

$\tilde{F}_i \leq \tilde{F}_{i,5}$ $(i = 1, \dots, n_{sc})$ and focus on a much smaller area on the Pareto surface. It

makes the algorithm quite flexible and less sensitive to the displacement of box D. The

transform shrinks the search domain, much like a "light beam" which emits from point

M and highlights a spot on the boundary of feasible space $\mathbf{Y}^*$. It is natural to choose a

direction (vector -**l**) in alignment with the normal to the utopia hyperplane towards the

decrement of the objective functions. If no solution is found, the direction is reversed by setting $s_n = 0$ in (13).

It should be emphasised that the general representation of matrix A can play a substantial part in seeking a Pareto set nearby the boundary of the Pareto frontier. As mentioned above, if we consider an orthogonal projection of the Pareto set onto the utopia hyperplane, the images of some Pareto points may not belong to the interior of a convex polygon spanned by the n_{sc} vertexes $\boldsymbol{\mu}_i^*$ (11), (12). This fact was first noted in [17]. One of the possibilities to solve this problem, suggested in [19] for the NC method, is based on the use of negative coefficients α_i. However, this will cause another problem, this time with the lower evaluation of the coefficients. Inevitably some points on the utopia plane corresponding to negative coefficients may not be orthogonal images of any Pareto point. In [17], an additional optimization problem needs to be solved at each anchor point in order to remove these points. Another possibility, offered specifically by our approach is described below.

Rotation of the search domain

Let us consider the edge vectors of polygon (11), (12): $\mathbf{v}_i = \boldsymbol{\mu}_{i+1}^* - \boldsymbol{\mu}_i^*$ $(i = 1,...,n_{sc} - 1)$. From linear algebra (see, e.g., [8]) it follows that point $\mathbf{p_i}$ belongs to a *k-th* edge of the polygon iff $\alpha_m = 0$ ($m \neq k, k+1$). Assume that vector $\mathbf{l}$ is initially related to the normal of the utopia hyperplane. If point M belongs to an edge of the polygon, vector $\mathbf{l}$ is then rotated in the direction opposite to the polygon. In other words, $\mathbf{l}$ is changed in such a way that the orthogonal projection of the end of the vector, drawn from the edge, onto the utopia hyperplane must not fall in the interior of the polygon.

To formalize this procedure, in the utopia plane we introduce a unit vector which is the outer normal to the edge considered. The vector can be defined as the following linear combination of the edge vectors $\mathbf{v}_{i-1}$ and $\mathbf{v}_i$:

$$\mathbf{s}_i = \frac{\mathbf{v}_{i-1} + \beta_i \mathbf{v}_i}{|\mathbf{v}_{i-1} + \beta_i \mathbf{v}_i|}, \quad \beta_i = -\frac{(\mathbf{v}_{i-1}, \mathbf{v}_i)}{(\mathbf{v}_i, \mathbf{v}_i)}. \tag{30}$$

Then, the current vector $\mathbf{l}$ is determined via $\mathbf{s}_i$ and the normal $\mathbf{n}$ to the utopia hyperplane towards the origin as follows:

$$\mathbf{l} = -\cos\theta_r \mathbf{n} + \sin\theta_r \mathbf{s}_i, \tag{31}$$
$$0 < \theta_r < \pi/2.$$

The angle θ_r is a parameter. Changing θ_r from 0 to $\pi/2$, vector $\mathbf{l}$ is turned from the normal vector $-\mathbf{n}$ to the vector $\mathbf{s}_i$ (see Figures 5a and 5b).

General algorithm for generating the Pareto set

Thus, the algorithm can be formulated in the following way.

Step A: Utopia plane and search domain. The utopia plane is determined. The search box $\tilde{D}$ is determined by formulas (15), (20), (25), (27), (28).

Step B: Displacement of search domain in the utopia polygon. Point M, a vertex of the search box $\tilde{D}$, is moved along the utopia plane according to (11) - (13) while the altering of the coefficients α_i follows the induction procedure [5] which was briefly described above.

Step C: Switching search direction. If the point M in (13) belongs strictly to the interior of the polygon formed by the anchor points, then the vector $\mathbf{l}$ coincides with the normal $-\mathbf{n}$. If there appears to be no solution, then the search is switched to the opposite

direction **n** via the shift of the search box to the opposite side of the utopia hyperplane ($s_n = 0$).

Step D: Rotation strategy at utopia polygon edges. If point *M* appears to be on an edge of the polygon, then an additional rotation of the vector may be required in order to investigate the periphery of the Pareto surface. To obtain a quasi-even distribution of the Pareto set, the number of additional points N_r related to the rotation of vector **l** depends on the distance to the vertexes of the edge. For example, the rotation is not required at the anchor points. In general, it is reasonable to choose the maximal value of N_r at the centre of an edge. The following evaluation of N_r is suggested for a *k-th* edge:

$$N_r = \text{int}(4m\alpha_k\alpha_{k+1}) \quad (m \geq 1) \tag{32}$$

Thus, at the centre of the edge $N_r = m$. The choice of *m* depends on the level of evenness required in the problem studied. Finally, it is worth noting that the number N_r can be substantially optimized if the information on the current local distribution of the Pareto set is taken into account. For example, if a Pareto solution appears to be at an edge of the polygon, no additional rotation is required and $N_r = 0$.

The angle γ_c in (25) is the parameter which determines the search cone: $\tilde{F}_i \leq \tilde{F}_{i,5}$ ($i = 1,...,n_{sc}$). If $\gamma_c \to 0$, then the search domain degenerates into a line. Thus, the angle γ_c is to be determined by the DM for a concrete problem on the basis of the following two principles. On the one hand it is natural to chose γ_c as small as possible, on the other hand it should not affect the robustness of the algorithm. It is clear that these requirements are not too restrictive and can be easily satisfied.

Step E: Filtering. Some *a posteriori* filtering procedure described in the next section for nonconvex Pareto surfaces might be required.

The flowchart of the general algorithm is given in Fig. 7.

Filtering procedure

In some cases, such as the one shown in Figure 6, local Pareto points (e.g., a point **P**) may be obtained which are not global Pareto solutions. The PP-based algorithm is capable to exclude such points. Let us locate the box D in such a way that the point M is at some point **P** investigated as a candidate Pareto solution and set $A = I$ in (15), (22). If the point is a global Pareto solution (e.g., a point **P′**), another solution cannot be obtained in D. This statement fully complies with the statement of the contact theorem [25]. Hence, we have a criterion for verification if the solution is a global Pareto solution.

Scaling procedure

In order to avoid possible problems related to different scales and undesirable severe skewing of the search domain, the following preliminary scaling of the objective functions similar to [21] can be used:

$$F_i^{sc} = \frac{F_i - F_{i|\min}}{F_{i|\max} - F_{i|\min}} \tag{33}$$

Thus, in our approach we are able to set $n_d = 1$ in (10).

Inverting the transform matrix

As written above, in the suggested algorithm we use only explicit formulas related to shrinking and conducting the search domain. Iterations can only be required to find the inverse matrix A. Meanwhile, this process can be optimized since $A^{-1} = RA_0^{-1}$. This

immediately follows from (25) and the orthogonality of the rotation matrix R. Hence, it is only required to invert the matrix A_0 which does not change unless the angle γ_c is changed.

Limitation of the algorithm

In a practical application of the developed approach, one should avoid the use of excessively small values of the angle γ_c because it might lead to the degeneration of the search domain and affect the robustness. The application of the method is limited by the assumption that the number of the anchor points is equalled to the number of objective functions. If this assumption is violated then the algorithm requires a substantial modification because of the degeneration of the utopia polygon. It is to be noted that the same problem arises for the NBI, NC and original PP approaches.

Capability to generate an even entire Pareto set

Following the strategy described above we are able in principle to cover all areas on the Pareto frontier in the objective space **Y**. To prove this, it is enough to show that for any arbitrary Pareto point Q there exist a point M belonging to polygon (11), (12) and vector **l** pointed from M to Q. If the algorithm principally provides such vector **l**, then the search domain inevitably includes point Q and a Pareto point can be found in some neighborhood of Q.

By definition vector **l** is either orthogonal to the utopia plane or satisfies conditions (31). Let us suppose, first, that the orthogonal projection of point Q onto the utopia plane, denoted by a point Q', belongs to the interior of polygon (11), (12). Then, we locate point M at point Q' and chose the vector **l** to be normal to the utopia plane.

Thus, the search domain includes the point Q. Assume now that the point $Q^{'}$ appears to be outside of the polygon (11), (12). Let us consider then the orthogonal projection of $Q^{'}$ on the nearest side of the polygon in the utopia plane and denote it as $Q^{''}$. Then, point M has to coincide with $Q^{''}$ and the appropriate vector **l** is given by (31), where

$$\sin\theta_r = \frac{\left(\overrightarrow{MQ^{'}}, \overrightarrow{MQ}\right)}{\left|\overrightarrow{MQ^{'}}\right|\left|\overrightarrow{MQ}\right|}$$. Point $Q^{''}$ can always be found if point $Q^{'}$ is situated outside of

the polygon. To prove this, let us assume that the statement is not correct and point $Q^{'}$ is located nearby an i-th and $i+1$ sides of the polygon. The intersection line between the hyperplane $F_i = \mu_i^*$ including anchor point μ_i^* and utopia plane cannot include $Q^{'}$ otherwise inevitably there are Pareto points (some of the anchor points) for which $F_i < \mu_i^*$. Then, anchor point μ_i^* has to be an internal point of the orthogonal projection of the Pareto surface onto the utopia plane. It can be proven that it is impossible. It follows from the fact that hyperplanes $F_j = const$ $(j = 1,...,n_{sc})$ are tangential to the Pareto frontier at the anchor points and the normal vector to the utopia plane has only positive coordinates.

The solution of the single-objective problem (5), (6), (7), (14), (15) has to be in some vicinity $\omega(\gamma_c, Q)$: $\sigma(\omega) \to 0$ *if* $\gamma_c \to 0$ (where σ is a measure of ω) since the search domain is strictly limited by conditions $\tilde{F}_i \le \tilde{F}_{i,5}$ $(i = 1,...,n_{sc})$. In turn, as stated above, the solution of the single-objective problem for the AOF has to be a Pareto solution.

It is to be noted that if a Pareto point is captured in the objective space, its original image is captured in the decision space **X** [19]. Thus, for any Pareto point there is some neighborhood $\omega(\gamma_c)$ of this point where a Pareto solution is captured. In turn, an even distribution of parameters α_i in (12) and angle θ_r in (31) results in a quasi-even mapping of these onto the Pareto frontier in the objective space **Y** if angle γ_c is small enough.

We have to emphasize that our approach for obtaining the complete Pareto frontier differs from the one developed for the NC method [17] where the authors use an oversized utopia-plane section to investigate the entire Pareto surface. This can lead to utopia-plane points guiding the appropriate optimization problems for which no Pareto solution exists. In our approach we perform a sequence of rotation of the search domain $\tilde{D}$ only for those utopia-plane points which belong to the edges of the polygon. The rotation is performed until no new Pareto solution is found.

Efficiency of the algorithm

To conclude, the algorithm described above is able to generate an entire well distributed Pareto set. This is achieved by solving a number of single-objective optimization problems for the AOF. The algorithm is efficient because the number of the single-objective problems solved mostly equals the number of the Pareto points obtained. Since the AOF is a smooth exponential-type convex function, fast gradient-based numerical methods can be used for its minimization if the objective functions are smooth enough. In addition, it is to be noted that the method can naturally be realized on parallel processors because each Pareto search can be done independently from the others.

It has to be pointed out that for the purpose of generating the entire Pareto set, the *1S* class functions can be represented by simple exponential functions. Nevertheless, retaining the general PP-formalism makes the algorithm (computer code) more universal, which can be important in the search of a specific Pareto solution. The extension of the algorithm presented here to the generalized Pareto solutions including *2S-4S* class-functions is given in [24].

6. Test cases

The method described above is validated by six test cases which include both convex and concave Pareto surfaces. It is shown that the PP-based method may lead to a high sensitivity of the location of the Pareto points to the displacement of box D especially in the case of a concave Pareto frontier if no offsetting is performed.

Example 1:

First, we consider the following simple algebraic test case:

$$\min(x, y)^T \tag{34}$$

$$x^2 + y^2 \leq 1, \tag{35}$$

$$x > -1, \; y > -1$$

In this case the Pareto surface is convex and represented by the unit circle in the first quadrant. Utopia polygon (11), (12) is represented by the interval between the utopia points (-1, 0) and (0, -1). The solution of problem (34), (35) by the original approach is shown in Fig. 8a. Contour lines near the upper right corner of the search domain (the box D) correspond to higher values of the aggregate objective function (AOF), while the contour lines near the lower left corner correspond to minimum values of the AOF. The crosses on the utopia line in Fig. 8a and in all of the following figures mark further the positions of point M in the computations related to different positions of box D. An

even displacement of the point M (or box D) along the utopia line does not lead to a good distribution of the Pareto points since there are gaps nearby the anchor points.

Offsetting the diagonal translation of box D and magnifying it may improve the situation [18], but could make the algorithm more complicated and more dependable on the solution to be found. According to this strategy, the top of the search domain (point M) is shifted in the normal direction from the utopia plane towards the Pareto frontier. Eventually, this strategy leads to the reduction of Pareto point scattering. In this sense the displacement Δ_s plays the same role as the parameter γ_c. However, the angle γ_c can be chosen, for instance, as small as possible unless it affects the robustness of the algorithm, while the value Δ_s cannot be chosen arbitrary and depends on the location of the Pareto frontier to be found. The introduction of the generalized class functions (14), (15) allows us to avoid the off-set strategy.

In the case of shrinking the search domain via transform (15) we obtain an even representation of the Pareto frontier (Figure 8b) evenly displaying the point M along the utopia interval. Here and further, the angle γ_c is equalled to 10^0. As we see, the aggregate function is localized around the search direction (vector **l**) as the result of the shear of the objective space.

Example 2:

The second test case includes optimization problem (34) under the following constraints:

$$x^2 + y^2 \geq 1, \tag{36}$$

$$x \geq 0,\, y \geq 0$$

In problem (34), (36) the Pareto frontier appears to be concave and is denoted by the unit circle located in the third quadrant. The utopia plane (interval) is determined by utopia points (0, 1) and (1, 0). This test case also represents an example where the standard definition of an anchor point leads to non-uniqueness. In Example 2 any point with coordinates $x = 0,\, y \geq 1$ is a solution of the single-objective minimization

problem: $\min\limits_{x,y} x$. In our definition we have only two anchor points in total: (0,1) and

(1,0).

In the original approach the displacement along the utopia line does not provide

any solution, except the utopia points because box D is in the unfeasible space $\mathbf{Y}\backslash\mathbf{Y}^*$.

Therefore, the off-set strategy is required in this example. If the displacement of point M

is performed along the line parallel to the line A as shown in Figure 9a, a complete

Pareto frontier can be obtained by the original method. Yet, this Pareto set is not evenly

distributed. It is important to note here that only five different Pareto points are obtained

in spite of solving eleven local optimization problems. This result appears because some

locations of point M lead to the same Pareto solution. It is to be noted that we chose line

A utilizing our *a priori* knowledge of the Pareto solution. If we consider line B then we

obtain only 4 Pareto solutions marked by the circles. In combination, these two

computations altogether provide us an evenly distributed Pareto set. However, in order

to obtain 7 Pareto solutions we have to consider 13 single-objective optimization

problems if start from line A and 14 problems if we start from line B.

The performance of our algorithm is demonstrated in Figure 9b which shows the

complete solution. In this figure, the contour plots of the AOF are located inside each

search domain. If a solution is not found (as shown in Figure 9b), the direction of the

search is switched to the opposite side by setting $s_n = 0$ in (13).

It is natural to introduce the coefficient of evenness k_e characterizing how evenly a

Pareto set is distributed on the Pareto surface. For this purpose, let us introduce some

coordinate system $\{\mathbf{x}^i\}$ ($i = 1, ..., n_{sc}- 1$) on the Pareto surface in the objective space. In

the Riemann space $R^{n_{sc}-1}$, related to the Pareto surface, the Riemann metric is given by

$$d\mathbf{r}^2 = \sum_{i=1}^{n_{sc}-1} \sum_{j=1}^{n_{sc}-1} g_{ij} d\mathbf{x}^i d\mathbf{x}^j. \tag{37}$$

Then, the coefficient k_e is defined as follows:

$$k_e = \frac{\max_i \min_j r_{ij}}{\min_i \min_j r_{ij}} \ (i, j = 1, ..., N_p) \tag{38}$$

Here, N_p is the number of Pareto points, r_{ij} is the distance between an *i-th* and a *j-th* Pareto points in metric (37):

$$r_{ij} = |\mathbf{x}_i - \mathbf{x}_j|. \tag{39}$$

Coefficient k_e represents the ratio between the maximal possible distance from a Pareto point to its nearest, but different, Pareto point and the minimal one. Obviously, in the ultimate case of the completely even distributed set: $k_e = 1$.

In the first example considered above, the standard approach without the off-set strategy yields $k_e = 5.6$, while the modified method provides $k_e = 1.6$. In Example 2, the standard approach formally leads to infinite value of k_e due to the coincidence of some Pareto points. In our approach $k_e = 1.2$.

Example 3:

In this example, suggested in [15], the Pareto frontier consists of different convex and concave parts created by three ellipsoid segments centered at the origin. The problem is formulated as follows:

$$\min \ (x, y)^T \tag{40}$$

subject to

$$x^2 + (y/3)^2 \geq 1,$$

$$x^4 + y^4 \geq 16,$$

$$(x/3)^3 + y^3 \geq 1,$$

$$0 \leq x \leq 2.9,$$

$$0 \leq y \leq 2.9$$

The exact Pareto curve is shown by the dashed line in Figure 10. It is important to note that the Pareto frontier is not smooth and is located on both sides of the utopia line. As illustrated in Figure 10, the algorithm is capable of capturing the entire frontier by generating an evenly distributed Pareto set. The NC method provides the same Pareto set if we use the identical points (marked by the crosses) in the utopia plane.

Example 4:

This test case is a *3D* one and leads to an example where the orthogonal images of Pareto points onto the utopia surface are not necessary in the interior of the polygon defined by the anchor points.

Let us consider the following 3D minimization problem:

$$\min(x, y, z)^T \tag{41}$$

under constraints

$$x^2 + y^2 + z^2 \geq 1, \tag{42}$$

$$x \geq 0, \ y \geq 0, \ z \geq 0$$

The standard definition of the anchor points again leads to non-uniqueness. It is easy to see that in our formulation we have only three anchor points: $\boldsymbol{\mu}_1^* = (0,0,1)^T$,

$\mathbf{\mu}_2^* = (1,0,0)^T$ and $\mathbf{\mu}_3^* = (0,1,0)^T$. The Pareto surface is represented by the part of the

unit sphere corresponding to nonnegative coordinates. It appears that not the entire

orthogonal projection of the Pareto surface onto the utopia surface is in the triangle

created by the anchor points (see Figure 11a). For this reason, a method such as NBI, is

not able to catch the entire Pareto frontier if the coefficients α_i in (12) are not negative.

Algorithm (31), (32) is used to provide the complete representation of the Pareto

frontier. According to this algorithm, the search domain is rotated to capture the entire

Pareto set while point M is on a side of the triangle determined by the anchor points.

The utopia plane and the points, distributed in the polygon (triangle in this case)

according to algorithm (11), (12), are shown in Figure 11b. (As in the previous

examples, the crosses on the utopia plane correspond to different positions of the point

M.) The solution providing the complete Pareto frontier is given in Figure 11c.

The Pareto set generated by the modified NC method is given in Figure 11d and

Figure 11f. Because the orthogonal projection of the Pareto surface is beyond the utopia

triangle (Fig. 11b), the search set in the utopia plane must be extended. According to the

algorithm described in [16], the search domain is limited by the hypercube assigned

with the *nadir* point. The nadir point, by definition, corresponds to the worst value for

all objective functions. Evenly distributed set was considered in the part of the utopia

plane limited by the search domain which is represented by the cube in Fig. 11d and

Fig. 11f.

If the anchor points shown in Fig. 11b are considered, the NC fails to generate the

Pareto frontier. Among 66 points distributed in the utopia plane 30 provided the feasible

subspaces and they resulted in the solutions which appear at the boundary of feasible

space limited by the cube. These solutions are not Pareto optimal apart from the anchor

point: $(1,0,0)$. The feasible space in the original NC algorithm for the considered set of the anchor points is shown in Fig. 11e if the minimum of $F_3 \equiv z$ is sought. The vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ are the vectors used in the NC method [16]: $\mathbf{v}_1 = \boldsymbol{\mu}_3^* - \boldsymbol{\mu}_1^*$, $\mathbf{v}_3 = \boldsymbol{\mu}_3^* - \boldsymbol{\mu}_2^*$. In the modified NC method the feasible space, in addition, is limited the cube assigned with the nadir point, and the NC solution appears at its boundary as shown in Figure 11d.

In the formulation of the NC method [16] the standard definition of an anchor point is used. As mentioned above, this definition leads to non-uniqueness of the anchor point set. Its choice may strongly affect the efficiency of the method. To demonstrate this, the following anchor points were considered: $(0,1,1)$, $(1,0,1)$, $(1,1,0)$. In this case, 24 from 66 points, distributed in the utopia plane, were efficient and they lead to 22 Pareto solutions shown in Figure 11f. Thus, this set of the anchor points provides us the generation of the Pareto frontier.

In our approach, 66 points, distributed in the utopia triangle, result in 93 Pareto points. This happens because any point among those in the utopia triangle provides at least one Pareto point. In addition, the points at the edge of the triangle generate additional Pareto points due to the rotation strategy.

In the next 3D example, approximately half the Pareto surface (in the orthogonal projection onto the utopia plane) appears far beyond the utopia triangle which is created by the anchor points. In this case algorithm (30), (31) related with the rotation of the search domain plays a crucial role.

Example 5:

Let us consider optimization problem (41) under the following constraints:

$$x^2 + y^2 + z^2 \le 1, \tag{43}$$

$$x \ge -1,\ y \ge -1,\ z \ge -0.5$$

In this case the Pareto surface is represented by the part of the unit sphere corresponding to nonpositive coordinates and satisfying condition $z \ge -0.5$. The anchor points are: $(-1,0,0)$, $(0,-1,0)$, $(-\frac{\sqrt{3}}{2}, 0, -\frac{1}{2})$. The results shown in Figure 12a demonstrate that the entire Pareto frontier is represented regardless of its orthogonal image on the utopia plane nearby point (0, 0, -0.5), the latter being far beyond the utopia triangle.

In Figure 12b, the Pareto set obtained by the modified NC method is shown. In this example, in the utopia plane only 18 points, among 66 evenly distributed in the search domain, appeared efficient. They provided 16 Pareto solutions.

In the developed algorithm, 66 points were evenly distributed in the utopia triangle and all these lead to Pareto solutions. The total number of the obtained Pareto solutions was equalled to 139.

Example 6:

In this example the optimization of the four-bar truss problem [10, 17] is analysed. The truss scheme is shown in Figure 13. Suspended loads F with magnitude of 10 kN each, are applied at points 1 and 2. The size of each bar must not exceed 5 cm^2 and the stresses in the bars are limited to 10 kN/cm^2 for both tension and compression. The elasticity modulus equals $2 \ast 10^4$ kN/cm^2. The objective is to design a four-bar truss with minimal stress in Bars 1 and 4, and minimal volume of the structure. Thus, the design

metrics are as follows: f_1 and f_2 are the stresses in Bars 1 and 4, respectively; f_3 is the volume of the structure.

As shown in Figure 14, our approach provides even generation of the complete Pareto frontier. The vertexes of the utopia triangle have the following coordinates: (10, 6, 2.7*10³), (10, 10, 2.3*10³), (2, 6, 4.4*10³). It is important to note that for the same example [17] the NBI and original NC methods fail to capture the full range of the Pareto frontier. The modified NC method generates additional redundant Pareto points near the extreme boundaries of the Pareto frontier [17].

7. Conclusion

The Physical Programming based method has been substantially modified to generate an even distribution of the entire Pareto set. The new approach also shares conceptual similarity with the Normal-Boundary Intersection and the Normal Constraint methods while aiming to combine their respective strengths. The proposed method is based on the generalization of the class-functions which leads to the manageable orientation of the search domain in the objective space. It is also proven in this work that the proposed modification allows the method to generate an even distribution of the entire Pareto surface. The generation is performed for both convex and non-convex Pareto frontiers. The method does not generate non-Pareto solutions. A simple algorithm has been proposed to remove local Pareto solutions. The new proposed definition of the anchor points provides the uniqueness of the anchor point set compared to the standard definition.

The proposed approach has been verified by several test cases, including the generation of both convex and concave Pareto frontiers. Future work will include the application of our method to more complex problems supplied by industry.

Acknowledgements

References

[1] T.W. Athan and P.Y. Papalambros, A Note on Weighted Criteria Methods for Compromise Solutions in Multi-Objective Optimization. *Engineering Optimization*, **27** (1996) 155-176.

[2] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge University Press, 2004.

[3] Y. Collette and P. Siarry, *Multiobjective Optimization: Principles and Case Studies*, Springer: Berlin, Heidelberg, New York, 2003, 317 p.

[4] I. Das, An Improved Technique for Choosing Parameters for Pareto Surface Generation Using Normal-Boundary Intersection. WCSMO-3 Proceedings, Buffalo, NY, March, 1999.

[5] I. Das and J.E. Dennis, A Closer Look at Drawbacks of Minimizing Weighted Sums of Objectives for Pareto Set Generation in Multicriteria Optimization Problems. *Structural Optimization*, **14** (1997) 63-69.

[6] I. Das and J.E. Dennis, Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM Journal of Meter Design Optimization Problems*, In Proceeding of ASME Design Automation Conference, 1989, Montreal, Quebec, Canada, Sept. 17-20: 77-89.

[7] K. Deb, *Multi-objective Optimization Using Evolutionary Algorithms*, J. Wiley & Sons: Chichester, 2001, 515p.

[8] M. Ehrgott, *Multicriteria Optimization*, Springer: Berlin, Heidelberg, 2005, 323 p.

[9] J. Koski, Defectiveness of Weighting Method in Multicreterion Optimization of Structures. *Communications in Applied Numerical Methods*, **1** (1985) 333-337.

[10] J. Koski, Multicriteria Truss Optimization, *Multicriteria Truss Optimization in Engineering and in the Sciences*, Edited by W. Stadler, Plenum Press, New York, 1988: 263-307.

[11] A.V. Lotov, V.A. Bushenkov, G.K. Kamenev *Interactive Decision Maps: Approximation and Visualization of Pareto Frontier*, Kluwer: Applied Optimization, Vol. 89, 2004, 310p.

[12] M.P. Martinez, A. Messac, M. Rais-Rohani, Manufacturability-Based Optimization of Aircraft Structures Using Physical Programming, *AIAA J.*, **39** (3), (2001) 517-525.

[13] A. Messac, From the Dubious Construction of Objective Functions to the Application of Physical Programming. *AIAA Journal*, **38** (1), (2000) 155-163.

[14] A. Messac, Physical Programming Effective Optimization for Computational Design. *AIAA Journal*, **34** (1), (1996) 149-158.

[15] A. Messac and A. Ismail-Yahaya, Required Relationships Between Objective Function and Pareto Frontier Orders: Practical Implications, *AIAA Journal*, **39** (11), (2001) 2168-2174.

[16] A. Messac, A. Ismail-Yahaya, C.A. Mattson, The Normalized Normal Constraint Method for Generating the Pareto Frontier. *Structural and Multidisciplinary Optimization*, **25** (2), (2003) 86-98.

[17] A. Messac and C. Mattson, Normal Constraint Method with Guarantee of Even Representation of Complete Pareto Frontier. *AIAA Journal*, **42** (10), (2004) 2101-2111.

[18] A. Messac and C. Mattson, Generating Well-Distributed Sets of Pareto Points for Engineering Design Using Physical Programming. *Optimization and Engineering*, **3** (2002) 431-450.

[19] A. Messac, E. Melachrinoudis, C. Sukam, Aggregate Objective Functions and Pareto Frontiers: Required Relationships and Practical Implications. *Optimization and Engineering Journal*, **1** (2), (2000) 171-188.

[20] A. Messac, C.P. Sukam, E. Melachrinoudis, Mathematical and pragmatic perspectives of physical programming. *AIAA Journal*, **39** (5), (2001) 885-893.

[21] K.M. Miettinen, *Nonlinear Multiobjective Optimization*, Kluwer Academic: Boston, 1999; 298.

[22] S. Ruzika and M.M. Wiecek, Approximation Methods in Multiobjective Programming, *J. of Optimization Theory and Applications*, **126** (3), (2005) 473-501.

[23] W. Stadler, *Multicriteria Optimization in Engineering and in the Sciences*, Plenum Press, New York, 1988.

[24] M.D. Guenov, S.V. Utyuzhnikov, P. Fantini, Application of the modified physical programming method to generating the entire Pareto frontier in multiobjective optimization, EUROGEN, Munich, 2005. See, also, http://arxiv.org/abs/math.OC/0512055.

[25] T.L. Vincent and W.J. Grantham, *Optimality in Parametric Systems*, John Wiley and Sons, New York, 1981; 243.
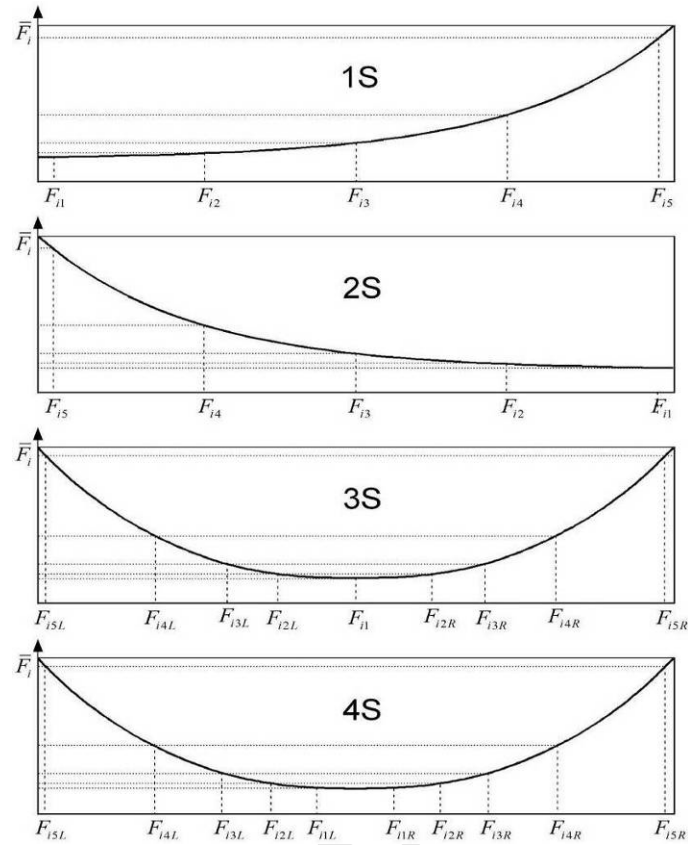
Fig. 1. The class-functions
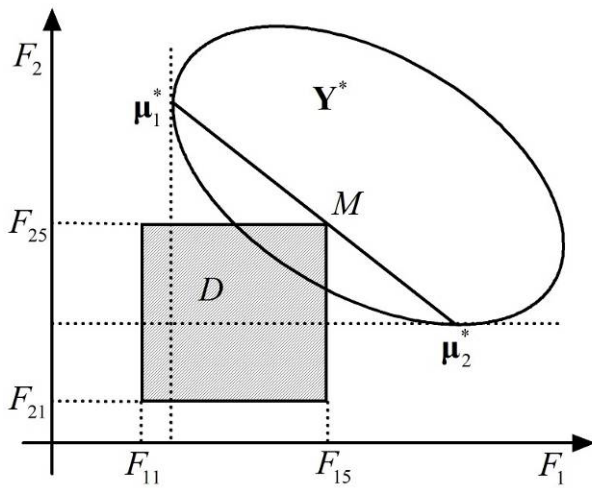
Fig. 2. Original search domain in the objective space
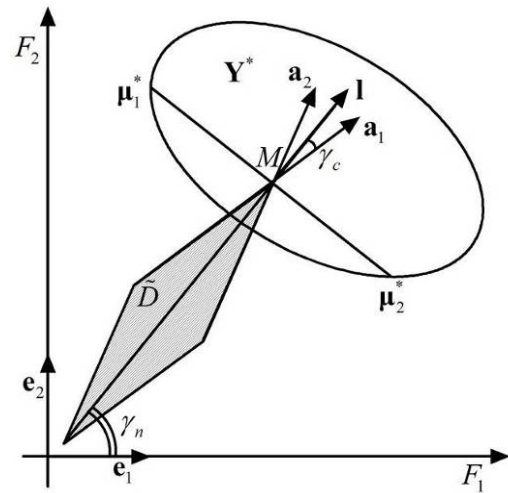
Fig. 3. New search domain after transform
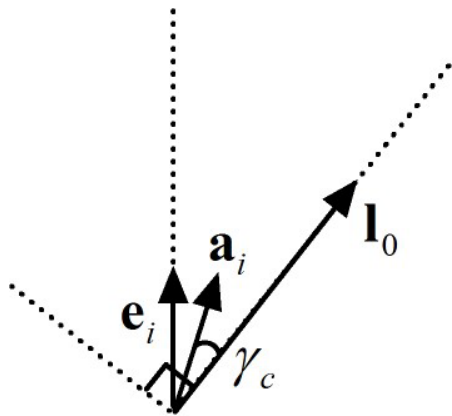
Fig. 4. Local basis vectors in an i-th hyperplane
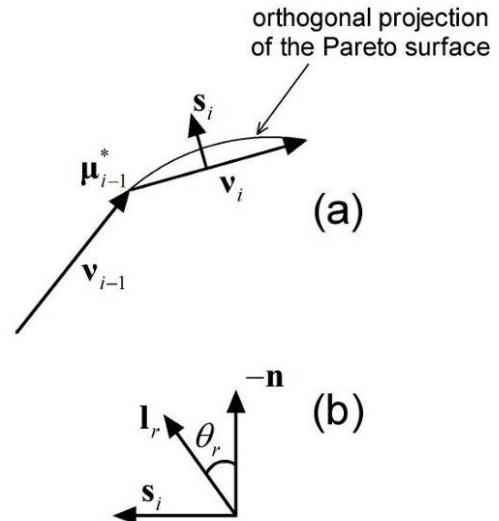
Fig. 5. Rotation of the search domain
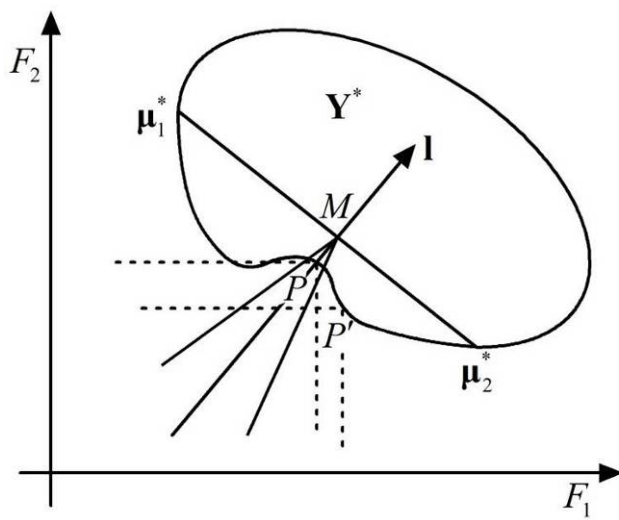
Fig. 6. Verification of global Pareto
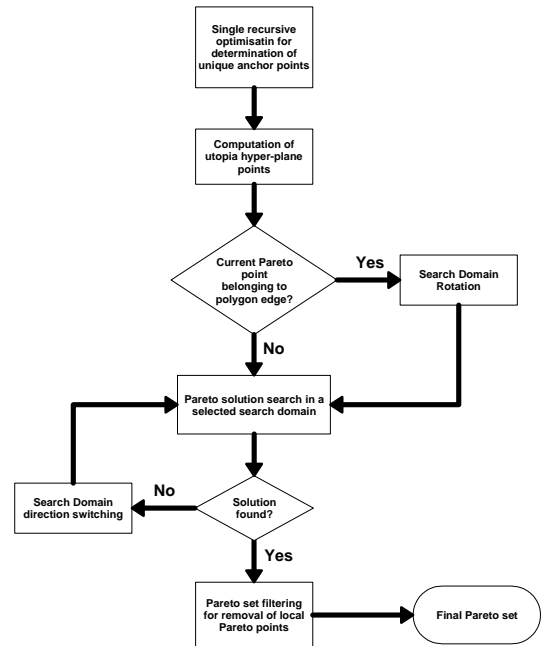
Fig. 7 Flowchart
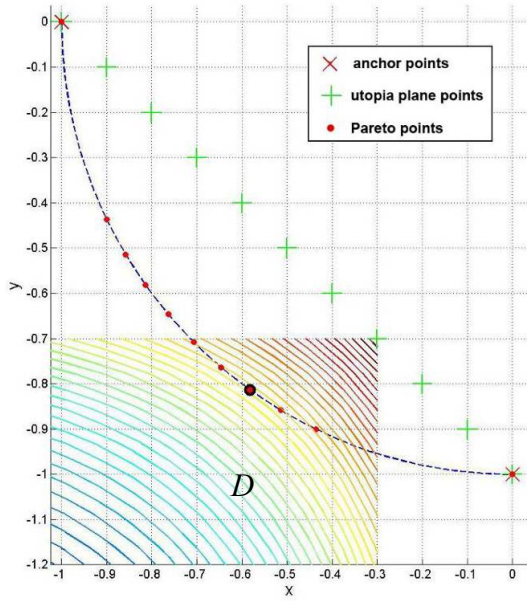
Fig. 8a. Standard search domain. Convex Pareto frontier
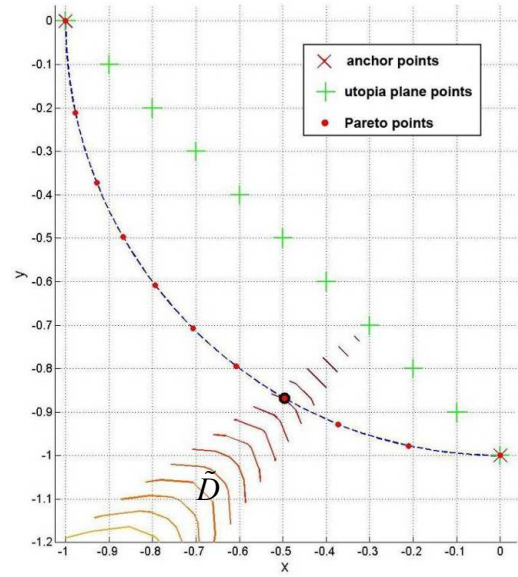
Fig. 8b. Transformed search domain. Convex Pareto frontier
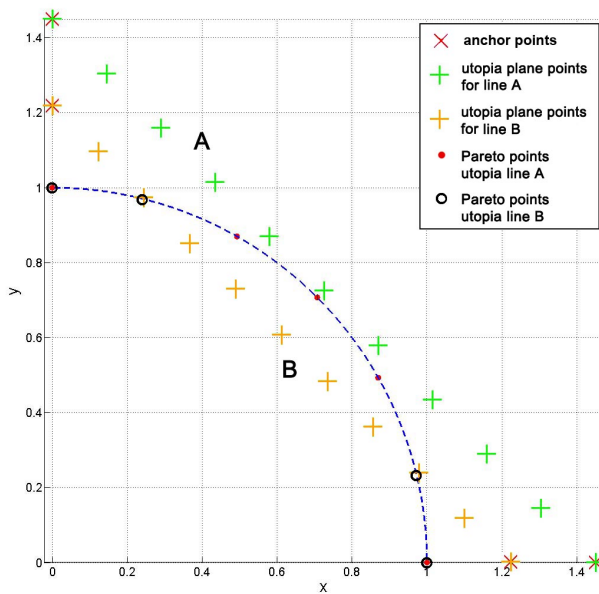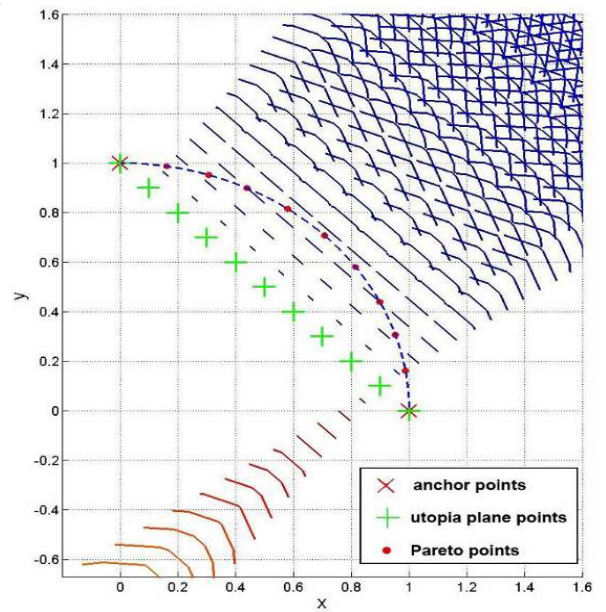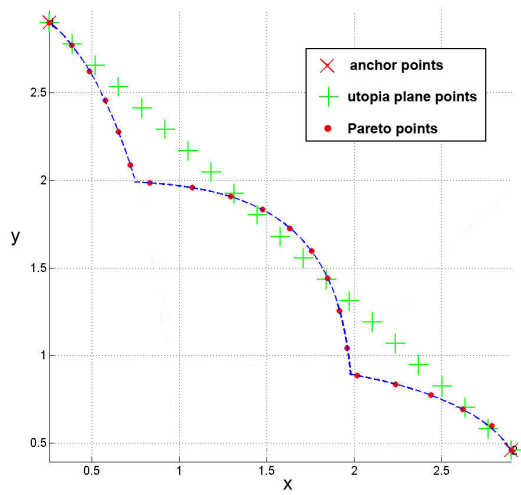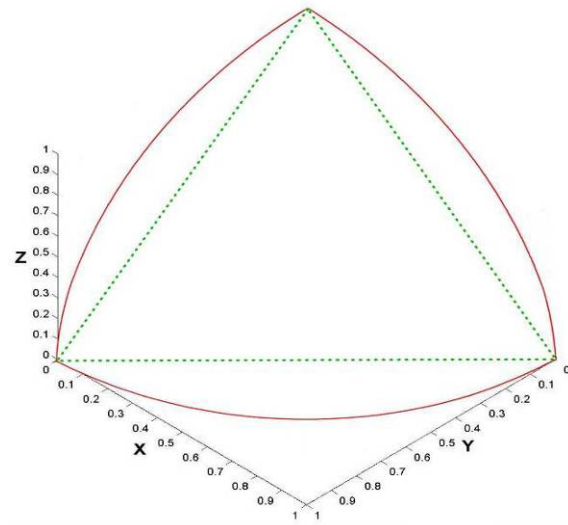

Fig. 9a. Standard search domain. Concave Pareto frontier

Fig. 9b. Transformed search domain. Concave Pareto frontier

Fig. 10. Hyperellipsoid frontier

Fig. 11a. 3D test case. Projection of the Pareto surface onto the utopia plane
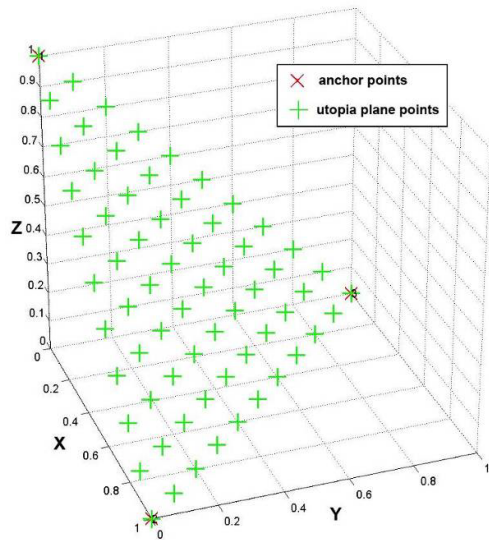
Fig. 11b. 3D test case. Utopia plane
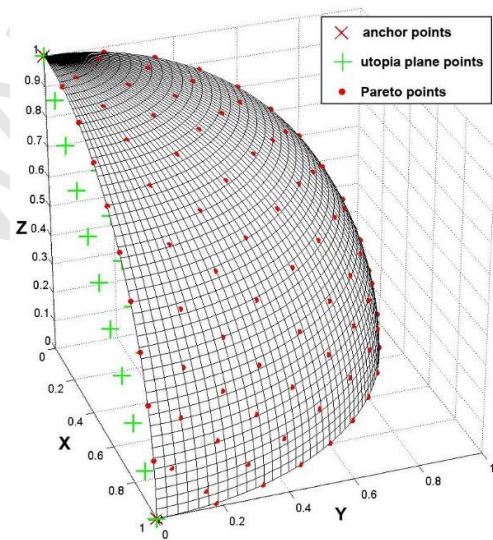
Fig. 11c. 3D test case. Pareto surface
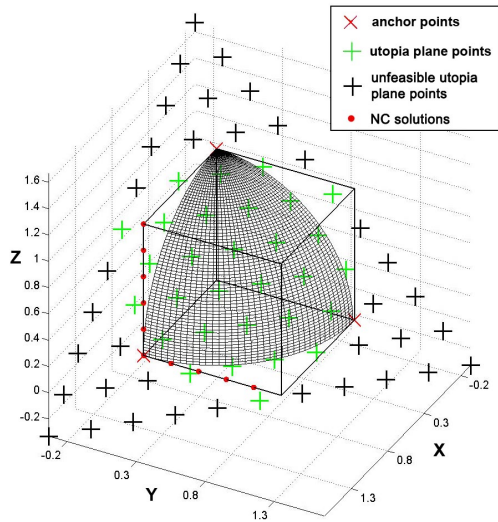
Fig. 11d. 3D test case. Pareto surface.
Modified NC method. Anchor points:
(0,0,1), (1,0,0), (0,1,0)
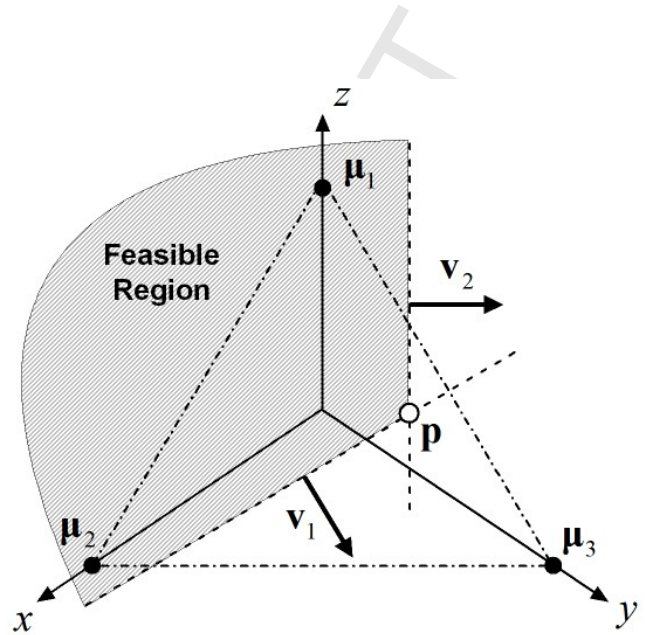
Fig. 11e. Feasible space in NC algorithm.
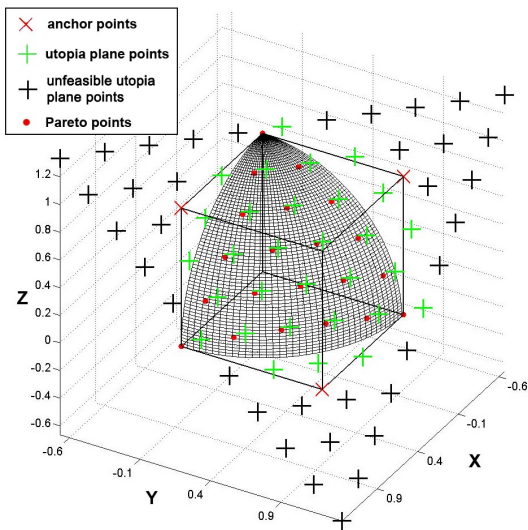
Fig. 11f. 3D test case. Pareto surface.
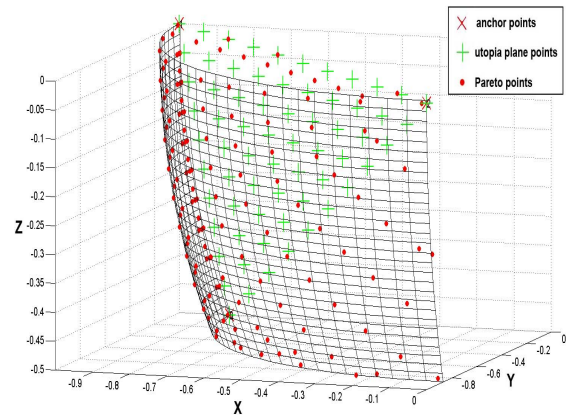Modified NC method. Anchor points:
(0,1,1), (1,0,1), (1,1,0)
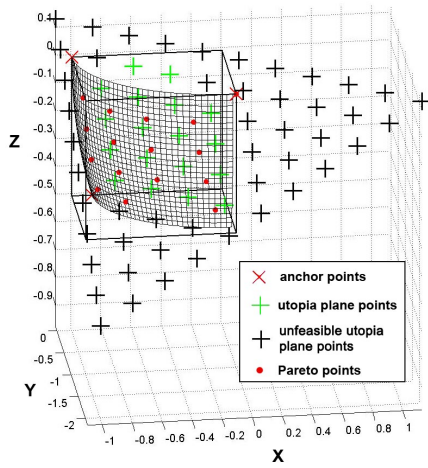

Fig. 12a. 3D test case. Pareto surface
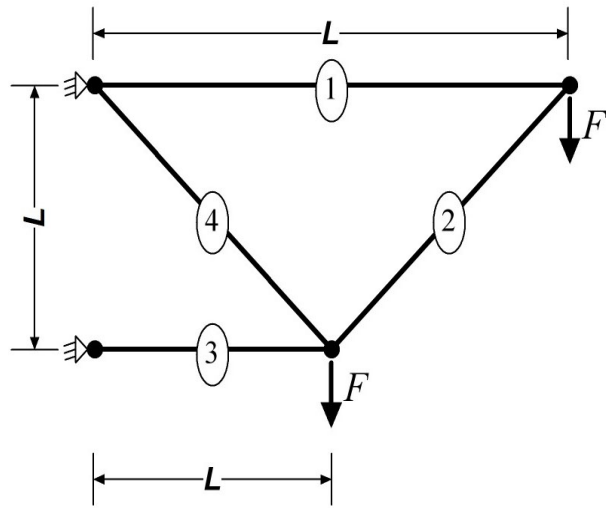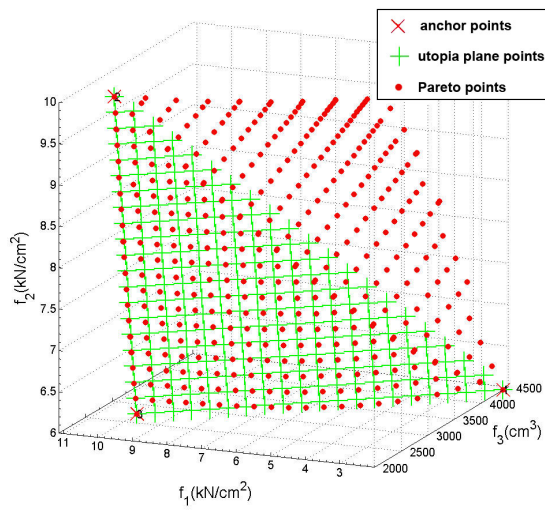
Fig. 12b. 3D test case. Pareto surface.

Modified NC method.

Fig. 13. 4-bar truss structure

Fig. 14. Four-bar truss problem