Comparison of multi-objective optimization methodologies for engineering applications

Comparison of multi-objective optimization methodologies for engineering applications

G. Chiandussi [a], M. Codegone [b,*], S. Ferrero [c], F.E. Varesio [a]

[a] *Politecnico di Torino, Department of Mechanical Engineering, Corso Duca degli Abruzzi 24, 10129 Torino, Italy*
[b] *Politecnico di Torino, Department of Mathematics, Corso Duca degli Abruzzi 24, 10129 Torino, Italy*
[c] *Nova Analysis s.n.c., Via Biella 72, 10098 Rivoli (TO), Italy*

ARTICLE INFO

Keywords:
Multi-objective optimization
Pareto front
Method effectiveness
Finite element method
Structural engineering

ABSTRACT

Computational models describing the behavior of complex physical systems are often used in the engineering design field to identify better or optimal solutions with respect to previously defined performance criteria. Multi-objective optimization problems arise and the set of optimal compromise solutions (Pareto front) has to be identified by an effective and complete search procedure in order to let the *decision maker*, the designer, to carry out the best choice. Four multi-objective optimization techniques are analyzed by describing their formulation, advantages and disadvantages. The effectiveness of the selected techniques for engineering design purposes is verified by comparing the results obtained by solving a few benchmarks and a real structural engineering problem concerning an engine bracket of a car.

1. Introduction

Computational models are commonly used in engineering design activities for the simulation of complex physical systems. They are often employed as virtual prototypes where a set of predefined system parameters are adjusted to improve or optimize the performance of the physical system as defined by one or more system performance objectives.

The optimization of a specific virtual prototype requires the implementation of the corresponding computational model, the evaluation of the performance objectives and the iterative adjustment of the system parameters in order to obtain an optimal solution. Multi-objective optimization problems arise in a natural fashion in the engineering field. It should be preferable to optimize the objective functions all at once but, in general, they are in competition with each other and the optimization process has to search for the best optimal compromise solution.

The primary goals in multi-objective optimization problem solution are:

- ✓ to preserve non dominated points in the objective space and associated solution points in the decision space;
- ✓ to keep making algorithmic progress toward the Pareto front in the objective function space;
- ✓ to maintain diversity of points on the Pareto front and of Pareto optimal solutions (decision space);
- ✓ to provide the *decision maker*, the designer, with a large enough but limited number of Pareto points for selection.

A preliminary review on single-objective optimization problems is required if the task of a multi-objective optimization problem and its attainment has to be fully understood. As a consequence the paper is organized as follows. At first, the basic terminology and nomenclature for use throughout the paper is introduced. Then, a classification of some mathematical

* Corresponding author. Tel.: +39 0110907507; fax: +39 0907599.
E-mail address: marco.codegone@polito.it (M. Codegone).

programming techniques that have been proposed to solve multi-objective optimization problems and the analysis of some of them will be presented in order to underline their advantages and disadvantages. The results obtained by solving several benchmark problems will be shown for comparison purposes. Finally, the results obtained by the study of a real world engineering design problem concerning an engine bracket will be presented.

1.1. The single-objective optimization problem

A single-objective optimization problem can be defined as:

Definition 1 (*General Single-Objective Optimization Problem*)**.** A general single-objective optimization problem is defined as the minimization (or maximization) of a scalar objective function $f(\boldsymbol{x})$ subject to inequality constraints $g_i(\boldsymbol{x}) \leq 0,\ i = \{1, \ldots, m\}$ and equality constraints $h_j(\boldsymbol{x}) = 0,\ j = \{1, \ldots, p\}$ where $\boldsymbol{x}$ is a n-dimensional decision variable vector $\boldsymbol{x} = (x_1, \ldots, x_n)$ from some universe Ω. Ω contains all possible $\boldsymbol{x}$ that can be used to satisfy an evaluation of $f(\boldsymbol{x})$ and its constraints. Of course, $\boldsymbol{x}$ can be a vector of continuous or discrete variables as well as f being continuous or discrete.

Observe that $g_i(\boldsymbol{x}) \leq 0$ and $h_j(\boldsymbol{x}) = 0$ represent constraints that must be fulfilled while optimizing (minimizing or maximizing) $f(\boldsymbol{x})$. Constraints can be explicit (i.e., given in algebraic form) or implicit, in which case the algorithm to compute $g_i(\boldsymbol{x})$ for any given vector $\boldsymbol{x}$ must be known. Note that p, the number of independent equality constraints, must be less than n, the number of decision variables, because if $p \geq n$ the problem is said to be over-constrained since there are no degrees of freedom left for optimizing (i.e., in other words, there would be more equations than unknowns). The number of degrees of freedom is given by $n - p$ if the equality constraints are independent and the inequality constraints do not reduce to equality constraints.

The method for finding the global optimum of any function (may not be unique) is referred to as Global Optimization. In general, the global minimum of a single-objective problem is presented in Definition 2 [1]:

Definition 2 (*Single-Objective Global Minimum Optimization*)**.** Given a function $f : \Omega \subseteq \mathbb{R}^n \to R$, $\Omega \neq \emptyset$, for $x \in \Omega$ the value $f^* \triangleq f(\boldsymbol{x}^*) > -\infty$ is called a global minimum if and only if:

$$\forall \boldsymbol{x} \in \Omega : f\left(\boldsymbol{x}^*\right) \leq f(\boldsymbol{x}) \tag{1.1}$$

where $\boldsymbol{x}^*$ is by definition the global minimum solution, f is the objective function and the set Ω is the feasible region of $\boldsymbol{x}$. The goal of determining the global minimum solution is called the global optimization problem for a single-objective problem.

1.2. The multi-objective optimization problem

Multi-objective problems are those problems where the goal is to optimize simultaneously k objective functions designated as: $f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})$ and forming a vector function $\boldsymbol{F}(\boldsymbol{x})$:

$$F(\boldsymbol{x}) = \begin{bmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \\ \vdots \\ f_k(\boldsymbol{x}) \end{bmatrix}. \tag{1.2}$$

Although single-objective optimization problems may have a unique optimal solution, multi-objective problems (as a rule) present a possibly uncountable *set* of solutions. Two n-space Euclidean spaces $\mathbb{R}^n$ are considered in multi-objective problems (Fig. 1.1):

- the n-dimensional space of the decision variables in which each coordinate axis corresponds to a component of vector $\boldsymbol{x}$;
- the k-dimensional space of the objective functions in which each coordinate axis corresponds to a component vector $f_k(\boldsymbol{x})$.

The evaluation function of a multi-objective problem, $\boldsymbol{F} : \Omega \to \Lambda$, maps the decision variables ($\boldsymbol{x} = x_1, \ldots, x_n$) to vectors ($\boldsymbol{y} = a_1, \ldots, a_k$). The set of solutions is found through the use of the Pareto Optimality Theory [2]. This mapping may or may not be onto some region of the objective function space depending on the functions and the constraints defining the multi-objective problem.

A *decision maker* has to choose one or more solutions by selecting one or more vectors. Note that the *decision maker* usually selects an acceptable solution belonging to the Pareto front. Identifying a set of Pareto optimal solutions is thus a key point for the *decision maker*'s selection of a compromise solution satisfying all the objectives as better as possible.

The multi-objective optimization problem (also called multi-criteria optimization, multi-performance or vector optimization problem) can then be defined as the problem of finding [3] "*a vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria which are usually in conflict with each other. Hence, the term 'optimize' means finding such a solution which would give the values of all the objective functions acceptable to the decision maker*".

A multi-objective global minimum (or maximum) problem is formally defined in Definition 3 [4]:

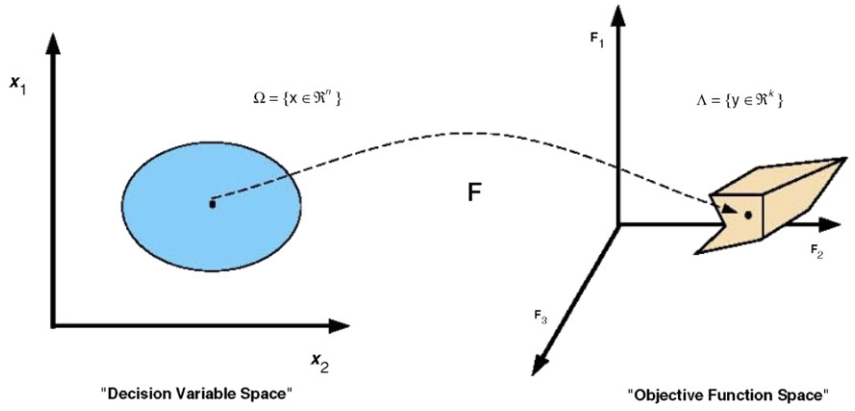

Fig. 1.1. Evaluation mapping of a multi-objective problem.

Definition 3 (*General Multi-Objective Optimization Problem*)**.** A general multi-objective optimization problem is defined as the minimization (or maximization) of the objective function set $F(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))$ subject to inequality constraints $g_i(\boldsymbol{x}) \leq 0$, $i = \{1, \ldots, m\}$, and equality constraints $h_j(\boldsymbol{x}) = 0$, $j = \{1, \ldots, p\}$. The solution of a multi-objective problem minimizes (or maximizes) the components of a vector $F(\boldsymbol{x})$ where $\boldsymbol{x}$ is a n-dimensional decision variable vector $\boldsymbol{x} = (x_1, \ldots, x_n)$ from some universe Ω. It is noted that $g_i(\boldsymbol{x}) \leq 0$ and $h_j(\boldsymbol{x}) = 0$ represent constraints that must be fulfilled while minimizing (or maximizing) $F(\boldsymbol{x})$ and Ω contains all possible $\boldsymbol{x}$ that can be used to satisfy an evaluation of $F(\boldsymbol{x})$.

Thus, a multi-objective problem consists of k objectives reflected in the k objective functions, $m + p$ constraints on the objective functions and n decision variables. The k objective functions may be linear or nonlinear and continuous or discrete in nature. Of course, the vector of decision variables x_i can also be continuous or discrete.

Definition 4 (*Ideal Vector*)**.** Let:

$$\boldsymbol{x}^{0(i)} = \left[x_1^{0(i)}, x_2^{0(i)}, \ldots, x_n^{0(i)}\right]^T \tag{1.3}$$

be a vector of variables which optimizes (either minimizes or maximizes) the ith objective function $f_i(x)$. In other words, the vector $x^{0(i)} \in \Omega$ is such that:

$$f_i\left(\boldsymbol{x}^{0(i)}\right) = \text{opt} f_i(\boldsymbol{x}). \tag{1.4}$$

Then, the vector:

$$\mathbf{f}^0 = \left[f_1^0, f_2^0, \ldots, f_k^0\right]^T \tag{1.5}$$

where f_i^0 denotes the optimum of the ith function, is ideal for an multi-objective problem and the point in $\mathbb{R}^n$ which determined this vector is the ideal solution and is consequently called the ideal vector. In other words, the ideal vector contains the optimum for each separately considered objective achieved at the same point in $\mathbb{R}^n$.

Definition 5 (*Convexity*)**.** A function $\phi(\boldsymbol{x})$ is called convex over the domain of $\mathbb{R}$ if for any two vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2 \in \mathbb{R}$:

$$\phi\left(\theta \boldsymbol{x}_1 + (1-\theta)\boldsymbol{x}_2\right) \leq \theta\phi\left(\boldsymbol{x}_1\right) + (1-\theta)\phi(\boldsymbol{x}_2) \tag{1.6}$$

where θ is a scalar in the range $0 \leq \theta \leq 1$. A convex function cannot have any value larger than the function values obtained by linear interpolation between $\phi\left(\boldsymbol{x}_1\right)$ and $\phi(\boldsymbol{x}_2)$. If the reverse inequality of the previous equation holds, the function is concave. Thus $\phi(\boldsymbol{x})$ is concave if $-\phi(\boldsymbol{x})$ is convex. Linear functions are convex and concave at the same time.

A set of points (or region) is defined as a convex set in n-dimensional space if, for all pairs of two points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ in the set, the straight-line segment joining them is also entirely in the set. Thus, every point $\boldsymbol{x}$, where:

$$\boldsymbol{x} = \theta \boldsymbol{x}_1 + (1-\theta)\,\boldsymbol{x}_2 \quad 0 \leq \theta \leq 1 \tag{1.7}$$

is also in the set. So, for example, the sets shown in Fig. 1.2 are convex, but the sets shown in Fig. 1.3 are not.

The concept of optimum with several objective functions changes because in multi-objective problems the aim is to find good compromise solutions (or trade-offs) rather than a single solution as in a global optimization problem. The concept of optimum most commonly adopted is that originally proposed by Francis Ysidro Edgeworth and later generalized by Vilfredo Pareto. The formal definition is provided next.

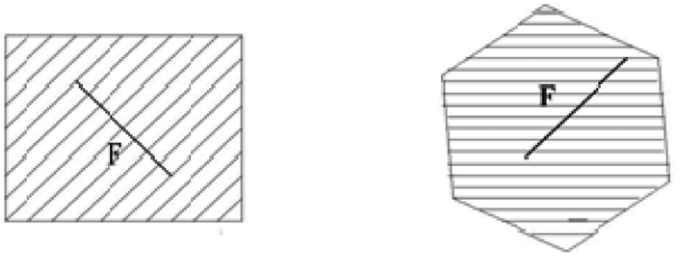

Fig. 1.2. Two examples of convex sets.

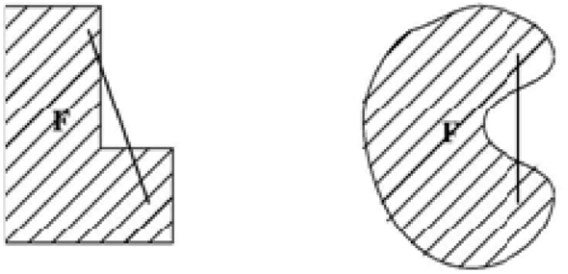

Fig. 1.3. Two examples of non-convex sets.

Definition 6 (*Pareto Optimality*)**.** A solution $\boldsymbol{x} \in \Omega$ is said to be Pareto optimal with respect to Ω if and only if there is no $x' \in \Omega$ for which $\boldsymbol{v} = F(\boldsymbol{x}') = (f_1(\boldsymbol{x}'), \ldots, f_k(\boldsymbol{x}'))$ dominates $\boldsymbol{u} = F(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))$. The phrase Pareto optimal is taken to mean with respect to the entire decision variable space unless otherwise specified. In words, this definition says that $\boldsymbol{x}^*$ is Pareto optimal if there exists no feasible vector $\boldsymbol{x}$ which would decrease some criterion without causing a simultaneous increase in at least one other criterion (assuming minimization).

Additionally, there are a few more definitions that are also adopted in multi-objective optimization:

Definition 7 (*Pareto Dominance*)**.** A vector $\boldsymbol{u} = (u_1, \ldots, u_k)$ is said to dominate another vector $\boldsymbol{v} = (v_1, \ldots, v_k)$ (denoted by $\boldsymbol{u} \preccurlyeq \boldsymbol{v}$) if and only if $\boldsymbol{u}$ is partially less than $\boldsymbol{v}$, i.e., $\forall i \in \{1, \ldots, k\},\ u_i \leq v_i \wedge \exists i \in \{1, \ldots, k\} : u_i < v_i$.

Definition 8 (*Pareto Optimal Set*)**.** For a given multi-objective problem, $F(\boldsymbol{x})$, the Pareto Optimal Set, $\mathcal{P}^*$, is defined as:

$$\mathcal{P}^* := \left\{ \boldsymbol{x} \in \Omega \mid \exists \boldsymbol{x}' \in \Omega F(\boldsymbol{x}') \preccurlyeq F(\boldsymbol{x}) \right\}. \tag{1.8}$$

Pareto optimal solutions are those solutions within the decision space whose corresponding variables cannot be all simultaneously improved. These solutions are also termed non-inferior, admissible or efficient solutions with the entire set represented by $\mathcal{P}^*$. Their corresponding vectors are termed non dominated. By selecting a vector from this vector set (the Pareto front set $\mathcal{P}F^*$) implicitly indicates acceptable Pareto optimal solutions, decision variables. These solutions may have no apparent relationship besides their membership in the Pareto optimal set. They form the set of all solutions whose associated vectors are non dominated. Pareto optimal solutions are classified as such based on their evaluated functional values.

Definition 9 (*Pareto Front*)**.** For a given multi-objective problem, $F(\boldsymbol{x})$, and Pareto optimal Set $\mathcal{P}^*$, the Pareto front $\mathcal{P}F^*$ is defined as:

$$\mathcal{P}F^* := \left\{ \boldsymbol{u} = F(\boldsymbol{x}) \,|\boldsymbol{x} \in \mathcal{P}^* \right\}. \tag{1.9}$$

When plotted in the objective space, non dominated vectors are collectively known as the Pareto front. Again, $\mathcal{P}^*$ is a subset of some solution set. Its evaluated objective vectors form $\mathcal{P}F^*$ of which each is non dominated with respect to all objective vectors produced by evaluating every possible solution in Ω. In general, it is not easy to find an analytical expression of the line or surface that contains these points and in most cases, it turns out to be impossible. The normal procedure to generate the Pareto front is to compute many points in Ω and their corresponding $f(\Omega)$. When there is a sufficient number of these, it is then possible to determine the non dominated points and to produce the Pareto front. A sample Pareto front is shown in Fig. 1.4.

Definition 10 (*Weak Pareto Optimality*)**.** A point $\boldsymbol{x}^* \in \Omega$ is a weak Pareto optimal if there is no $\boldsymbol{x} \in \Omega, \boldsymbol{x} \neq \boldsymbol{x}^*$ such that $f_i(\boldsymbol{x}) < f_i(\boldsymbol{x}^*)$ for $i = 1, \ldots, k$.

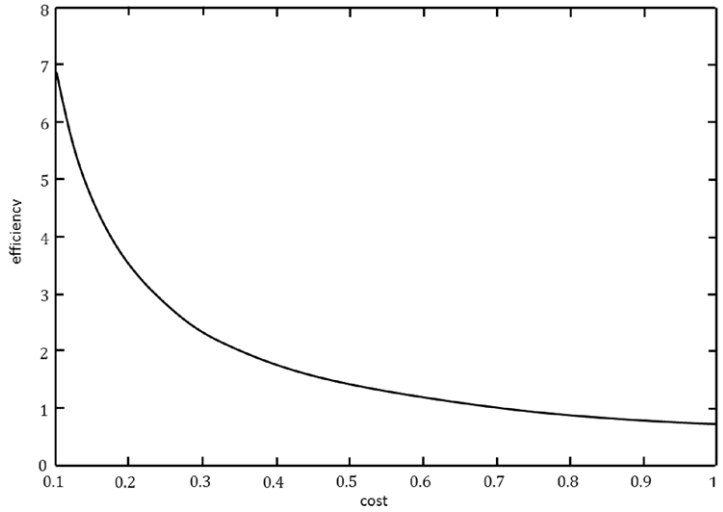

Fig. 1.4. Pareto front of a problem with two objective functions: cost and efficiency.

Definition 11 (*Strict Pareto Optimality*)**.** A point $\boldsymbol{x}^* \in \Omega$ is a strictly Pareto optimal if there is no $\boldsymbol{x} \in \Omega, \boldsymbol{x} \neq \boldsymbol{x}^*$ such that $f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{x}^*)$ for $i = 1, \ldots, k$.

Definition 12 (*Kuhn–Tucker Conditions for Non Inferiority*)**.** If a solution $\boldsymbol{x}$ to the general multi-objective problem is non inferior, then there exist $w_l = 0,\ l = 1, 2, \ldots, k$ (w_r is strictly positive for some $r = 1, 2, \ldots, k$), and $\lambda_i = 0, i = 1, 2, \ldots, m$, such that:

$$\boldsymbol{x} \in \Omega \tag{1.10}$$

and:

$$\sum_{l=1}^{k} w_l \nabla f_l(\boldsymbol{x}) - \sum_{i=1}^{m} \lambda_i \nabla g_i(\boldsymbol{x}) = 0. \tag{1.11}$$

These conditions are necessary for a non inferior solution and, when all of the $\mathbf{f}_l(\boldsymbol{x})$ are concave and Ω is a convex set, they are sufficient as well.

Pareto optimal solutions are those which, when evaluated, produce vectors whose performance f_i cannot be improved without adversely affecting another $f_j,\ i \neq j$. The Pareto front $\mathcal{P}F^*$ determined by evaluating $\mathcal{P}^*$ is fixed by the defined multi-objective problem and does not change. Thus, $\mathcal{P}^*$ represents the best solutions available and allows the definition of the global optimum of a multi-objective problem.

Definition 13 (*Multi-Objective Global Minimum*)**.** Given a function $\mathbf{f} : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}^k, \Omega \neq \emptyset, k > 2$, for $\boldsymbol{x} \in \Omega$ the set $\mathcal{P}F^* \triangleq \mathbf{f}\left(\boldsymbol{x}_i^*\right) > (-\infty, \ldots, -\infty)$ is called the global minimum if and only if:

$$\forall \boldsymbol{x} \in \Omega : \mathbf{f}\left(\boldsymbol{x}_i^*\right) \preccurlyeq \mathbf{f}(\boldsymbol{x}). \tag{1.12}$$

Then, $\boldsymbol{x}_i^*,\ i = 1, \ldots, n$ is the global minimum solution set (i.e., $\mathcal{P}^*$), $\mathbf{f}$ is the multiple objective function, and the set Ω is the feasible region. The problem of determining the global minimum solution set is called the multi-objective global optimization problem.

2. Multi-objective optimization techniques

There have been several attempts to classify the multi-objective optimization techniques currently in use. First of all, it is quite important to distinguish the two stages in which the solution of a multi-objective optimization problem can be divided: the optimization of the objective functions involved and the process of deciding what kind of trade-offs are appropriate from the *decision maker* perspective (the so-called multi-criteria decision making process). In this section, some of the many techniques available for these two stages are discussed by analyzing some of their advantages and disadvantages.

Cohon and Marks [5] proposed one of the most popular classification of techniques within the Operations Research community that focuses on the way in which each method handles the two problems of searching and making decisions:

1. *a priori* Preference Articulation: take decisions before searching (decide ⇒ search). This group of techniques includes those approaches that assume that either a certain desired achievable goals or a certain pre-ordering of the objectives can be performed by the decision maker prior to the search.
2. *a posteriori* Preference Articulation: search before making decisions (search ⇒ decide). These techniques do not require prior preference information from the *decision maker*. Some of the techniques included in this category are among the oldest multi-objective optimization approaches proposed.
3. Progressive Preference Articulation: integrate search and decision making (decide ⇔ search). These techniques normally operate in three stages:
   - ✓ find a non dominated solution,
   - ✓ get the reaction of the *decision maker* regarding this non dominated solution and modify the preferences of the objectives accordingly,
   - ✓ repeat the two previous steps until the *decision maker* is satisfied or no further improvement is possible.

A different kind of approach is represented by Evolutionary Algorithms that are based on Darwin's theory of survival of the fittest. They are found on the idea that as the population evolves in a genetic algorithm, solutions that are non-dominated are chosen to remain in the population.

One *a priori* Preference Articulation, two *a posteriori* Preference Articulations and one Evolutionary Algorithm have been analyzed and compared in terms of performance in the following sections.

2.1. Global criterion method

The global criterion method is an *a priori* Preference Articulation. Its aim is to minimize a function (global criterion) which is a measure of how close the *decision maker* can get to the ideal vector $\boldsymbol{f}^0$. The most common form of this function is [6]:

$$f(\boldsymbol{x}) = \sum_{i=1}^{k} \left(\frac{f_i^0 - f_i(\boldsymbol{x})}{f_i^0} \right)^p \tag{1.13}$$

where k is the number of objectives. For this formula Boychuk and Ovchinnikov [7] have suggested the exponent $p = 1$ whereas Salukvadze [8] has suggested $p = 2$, but other values of p can also be used. Obviously, the results differ greatly depending on the value of the exponent p chosen. Thus, the selection of the best p is an issue in this method and it could also be the case that any p could produce an unacceptable solution.

Another possible measure of closeness to the ideal solution is a family of L_p-metrics defined as follows:

$$L_p(f) = \left[\sum_{i=1}^{k} \left| f_i^0 - f_i(x) \right|^p \right]^{1/p}, \quad 1 \le p \le \infty. \tag{1.14}$$

In general, relative deviations of the form:

$$\frac{f_i^0 - f_i(x)}{f_i^0} \tag{1.15}$$

are preferred over absolute deviations because they have a substantive meaning in any context. The relevant L_p metrics are:

$$L_p(f) = \left[\sum_{i=1}^{k} \left| \frac{f_i^0 - f_i(\boldsymbol{x})}{f_i^0} \right|^p \right]^{1/p}, \quad 1 \le p \le \infty. \tag{1.16}$$

The value of p points out the type of distance. For $p = 1$, all deviations from f_i^* are taken into account in direct proportion to their magnitudes which corresponds to 'group utility'. For $2 \le p < \infty$, the larger deviations carry greater weight in L_p. For $p = \infty$, the largest deviation is the only one taken into consideration which leads to a purely 'individual utility' (min–max criterion) in which all weighted deviations are equal.

Koski [9] has suggested L_p-metrics with a normalized vector objective function of the form:

$$f_i(\boldsymbol{x}) = \frac{f_i(\boldsymbol{x}) - \min\limits_{x \in F} f_i(\boldsymbol{x})}{\max\limits_{x \in F} f_i(\boldsymbol{x}) - \min\limits_{x \in F} f_i(\boldsymbol{x})}. \tag{1.17}$$

In this case, the values of every normalized function are limited to the range [0, 1].

Using the global criterion method one non-inferior solution is obtained. If certain parameters w_i are used as weights for the criteria, a required set of non-inferior solutions can be found. Duckstein [10] calls this method compromise programming and his L_p-metric is:

$$L_p(\boldsymbol{x}) = \left[\sum_{i=1}^{\boldsymbol{k}} w_i^p \left[\frac{f_i(\boldsymbol{x}) - f_i^0}{f_{i\,\max} - f_i^0} \right]^p \right]^{1/p} \tag{1.18}$$

where w_i are the weights, $f_{i\,\max}$ is the worst value obtainable for criterion i, $f_i(\boldsymbol{x})$ is the result of implementing decision $\boldsymbol{x}$ with respect to the ith criterion. The displaced ideal technique which proceeds to define an ideal point, a solution point, another ideal point, etc. is an extension of compromise programming.

Another variation of this technique is the method suggested by Wierzbicki [11] in which the global function has a form that penalizes the deviations from the so-called reference objective. Any reasonable or desirable point in the space of objectives chosen by the decision maker can be considered as the reference objective. Let $\mathbf{f}^T = \left[f_1^r, f_2^r, \ldots, f_k^r\right]^T$ be a vector which defines this point. Then the function which is minimized has the form:

$$P\left(\boldsymbol{x}, \mathbf{f}^T\right) = -\sum_{i=1}^{k}(f_i(\boldsymbol{x}) - f_i^r)^2 + \varrho \sum_{i=1}^{k} \max(0, (f_i(\boldsymbol{x}) - f_i^r)^2) \tag{1.19}$$

where $\varrho > 0$ is a penalty coefficient which in this method can be chosen as constant. Minimizing (1.19) for the assumed point $\mathbf{f}^r$ a non-inferior solution which is close to this point can be obtained. If for different points $\mathbf{f}^r$ the procedure is carried out, some representation of non-inferior solutions can be found.

The main advantage of these methods is their simplicity and their effectiveness because they do not require a Pareto ranking procedure. However, their main disadvantage is the definition of the desired goals which requires some extra computational effort. An additional problem with these techniques is that they will yield a non dominated solution only if the goals are chosen in the feasible domain and such conditions may certainly limit their applicability. More information on these methods can be found in [6,12,13].

2.2. Linear combination of weights

Zadeh [14] was the first to show that the third of the Kuhn–Tucker conditions for non inferior solutions implies that non inferior solutions might be found by solving a scalar optimization problem in which the objective function is a weighted sum of the components of the original vector-valued function. That is, the solution to the following problem:

$$\min \sum_{i=1}^{k} \alpha_i f_i(\boldsymbol{x}) \tag{1.20}$$

subject to:

$$\boldsymbol{x} \in \Omega \tag{1.21}$$

where $\alpha_i \geq 0$ for all i and strictly positive for at least one objective, is usually non inferior. The non inferior set and the set of non inferior solutions can be generated by parametrically varying the weights α_i in the objective function. This was initially demonstrated by Gass and Saaty [15] for a two-objective problem.

The reduction of the problem to a single-objective function means to make all alternatives comparable with a preference framework that becomes a total order. Hence α_i values choice is very important to achieve the final decision and, for this reason, value choice is made by the *decision maker*. However the *decision maker*, in order to choose the coefficients, must have a clear perception of how this choice influence optimal points.

Let consider a particular solution $\overline{x}$ for which the value of the objective function is $\overline{f_i} = f_i(\overline{x})$. Let fix two criteria h and k and a value $\Delta_h > 0$ little enough. The *decision maker* is asked for which value $\Delta_k > 0$ there is no difference among $\overline{x}$ and an hypothetical alternative that gives values $f_i^* = \overline{f_i}$ for $i \neq h, k$ and $\overline{f_h} = \overline{f_h} - \Delta_h$ and $\overline{f_k} = \overline{f_k} + \Delta_k$. It is reasonable to think that such value $\Delta_h > 0$ exists. Indeed for $\Delta_h > 0$ the hypothetical alternative dominates $\overline{x}$ while increasing Δ_k the situation is expected to be reversed. Because of the indifference of the two alternatives, the following must hold:

$$\sum_i \alpha_i \overline{f_i} = \sum_i \alpha_i f_i^* \Rightarrow 0 = \alpha_k \Delta_k - \alpha_h \Delta_h \Rightarrow \frac{\alpha_h}{\alpha_k} = \frac{\Delta_k}{\Delta_h}. \tag{1.22}$$

The previous equation links the coefficients of the linear combination to the comparative evaluation among the two criteria. Varying h on all criteria, α_h can be described through α_k. Since coefficients are defined up to a positive constant (indeed multiply all for the same positive constant does not change the problem (1.20)), α_k can be set as $\alpha_k = 1$ and so $\alpha_h = \Delta_k/\Delta_h$. Obviously the bigger is the coefficient α_h, the more the objective h is taken into account in the decision choice. Moreover it has no meaning to consider a coefficient equal to zero since it would say non considering the corresponding objectives.

The procedure described implicitly assumed the linearity of the objective functions. In other words the values Δ_h can depend on levels $\overline{f_h}$ of the objective considered. If the h criterion has been satisfied yet, the *decision maker* could prefer to improve further while it could take an opposite behavior for a non-satisfying level. The interaction with the *decision maker* normally supposes of resolving several times the problem (1.20) attempting different values for α_i until a satisfying solution is found.

The positive aspect of this approach is that $\min\{F(x) : x \in X\}$ provides a Pareto optimal point. Indeed if y is a dominated solution by x, from dominance definition the following holds:

$$\alpha_i f_i(x) \leq \alpha_i f_i(y) \quad \text{with } i = 1, \ldots, m \quad \text{and} \quad \alpha_k f_k(x) < \alpha_k f_k(y) \tag{1.23}$$

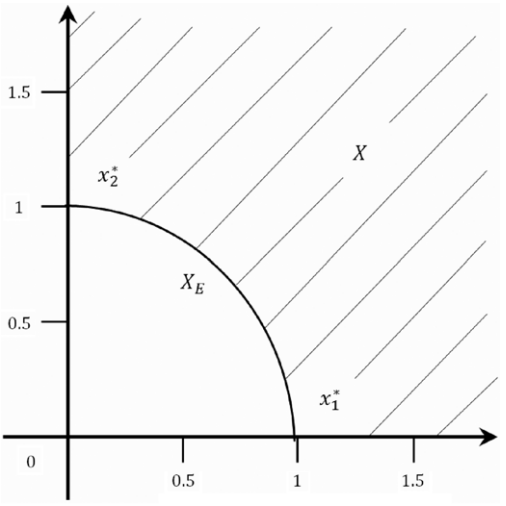

Fig. 2.1. The weighted sum method fails for non-convex problems.

leading to:

$$\sum_i \alpha_i f_i(x) < \sum_i \alpha_i f_i(y). \tag{1.24}$$

Thus no dominated solution can be optimum of (1.20).

On the contrary, a negative aspect is due to the fact that is usually not true that each Pareto optimal can be obtained through a suitable choice of α_i coefficients. The reason is the following: to solve (1.20) is equivalent to minimize the linear functional $\sum_i \alpha_i y_i$ for $y \in f(X)$. A minimum of a linear functional on a set Y belongs both to the set Y and to the border of the convex envelope of Y. Hence, those non dominated points that do not lie on the border of the convex envelope cannot be generated by (1.20). More exactly, since only positive α_i coefficients are admitted, solving (1.20) generates only solutions that lie on the border of the convex envelope $f(X) + \mathbb{R}^m_+$. In order to explain these concepts, let consider the following example: let $X = \left\{x \in \mathbb{R}^2 : x_1^2 + x_2^2 = 1\right\}$ and $f(x) = x$; in this case the efficient point set being $X_E = \left\{x \in \mathbb{R}^2 : x_1^2 + x_2^2 = 1\right\}$, yet $x_1^* = (1, 0)$ and $x_2^* = (0, 1)$ are the only feasible solutions that are optimal solutions of (1.20) for any $\alpha_i \geq 0$ (Fig. 2.1).

The main advantages of this method are its simplicity (in implementation and use) and its efficiency (computationally speaking). Its main disadvantage is the difficulty to determine the appropriate weight coefficients to be used when enough information about the problem is not available (this is an important concern, particularly in real-world applications). Also, a proper scaling of the objectives requires a considerable amount of extra knowledge about the problem. To obtain this information could be a very expensive process. A more serious drawback of this approach, as underlined before, is that it cannot generate certain portions of the Pareto front when its shape is concave, regardless of the weights combination used. Nevertheless, aggregating functions could be very useful to get a preliminary sketch of the Pareto front of a certain problem or to provide prior information to be exploited by another approach.

2.3. The ε-constraint method

Besides the weighted sum approach, the ε-constraint method is probably the best known technique to solve multi-criteria optimization problems. There is no aggregation of criteria, instead only one of the original objectives is minimized while the others are transformed to constraints. The idea was introduced by Haimes [16]. Through this approach among p objective function only one is kept as such, the other $p - 1$ are transformed in constraints fixing threshold values ε_k (with $k = 1, \ldots, p,\ k \neq j$) over them (if functions must be minimized). Therefore the problem:

$$\min_{x \in X} \left(f_1(x), \ldots, f_p(x)\right) \tag{1.25}$$

is substituted by the ε-constraint problem:

$$\min_{x \in X} f_j(x) \tag{1.26}$$

$$f_k(x) \leq \varepsilon_k \quad k = 1, \ldots, p,\ k \neq j. \tag{1.27}$$

Fig. 2.2 illustrates a bi-criterion problem where an upper bound constraint is put on $f_1(x)$. The optimal values of the (1.26) problem with $j = 2$ for two values of ε_1 are indicated. These show that the constraints $f_i(x) \leq \varepsilon_i$ might or might not be active at an optimal solution.

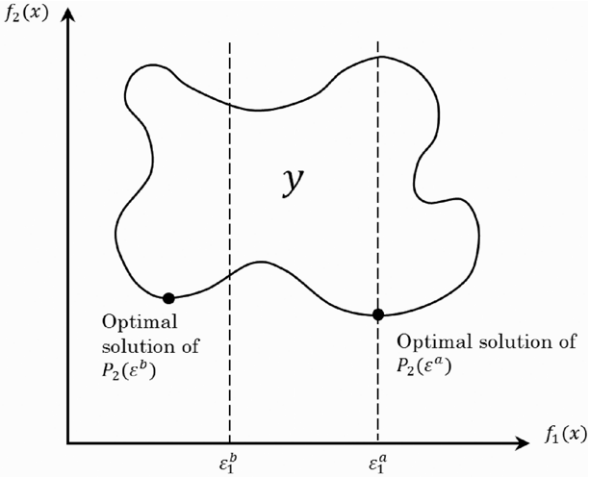

Fig. 2.2. Optimal solutions of ε-constraint problems (1.26).

The ε_i coefficients represent objective values that the *decision maker* wants however to be guaranteed. Even if the model (1.26)–(1.27) introduces a total order among alternatives, there is no mixing among objectives and each one keeps its identity. Even in this case, it is possible to consider these questions: given the $\varepsilon_i > 0$ values, it is possible to obtain a Pareto optimal solution? Given a Pareto optimal point x^*, do $\varepsilon_i > 0$ values exist for which x^* is an optimum for (1.25)? Which relationship exist among ε_i and Pareto optimum?

Unlike linear combination of weights, Eqs. (1.26)–(1.27) are able to generate all Pareto optimal points varying $\varepsilon_i > 0$ values. Indeed if x^* is a Pareto optimum, it is sufficient to choose ε such that:

$$\varepsilon_i = f_i\left(x^*\right) \quad i = 2, \ldots, p. \tag{1.28}$$

Thus if exists an admissible solution $\bar{x}$ such that $f_1(\bar{x}) < f_1(x^*)$ it would dominate x^* contrary to the Pareto optimality assumption of x^*. Hence no Pareto optimum is lost varying ε_i, independently from convexity or non-convexity of $f(X)$.

On the contrary, it is not true that, given $\varepsilon_i > 0$, a Pareto optimal solution can be obtained. In fact the solution of (1.26)–(1.27) is certainly non-dominated if it is the unique optimum. Unfortunately the property of the optimum uniqueness is not easy to check. It could happens that solving (1.26)–(1.27), an optimum x^* can be obtained, while another optimum $\bar{x}$ exists, unknown by the analyst, such that:

$$f_1\left(x^*\right) = f_1(\bar{x}), \qquad f_2(\bar{x}) < f_2\left(x^*\right) \leq \varepsilon_2. \tag{1.29}$$

Thus the point x^* is dominated. However in this case the problem can be overcome by solving again (1.26)–(1.27) with $\varepsilon_2 = f_2(x^*) - 1$ (if for example f_2 assumes only integer values). In this way it is possible to exclude from computation x^*, while if $\bar{x}$ is admissible, it would be generated solving (1.26) (unless it is a dominated solution yet). Comparing the two solutions x^* can be eliminated as it is dominated.

In order to get adequate ε_i values, single-objective optimizations are normally carried out for each objective function in turn by using mathematical programming techniques. For each objective function f_i $(i = 1, 2, \ldots, m)$ there is an optimal solution vector $\boldsymbol{x}_i^*$ for which $f_i\left(\boldsymbol{x}_i^*\right)$ is a minimum. Let be $f_i\left(\boldsymbol{x}_i^*\right)$ the lower bound on i.e.:

$$\varepsilon_i \geq f_i\left(\boldsymbol{x}_i^*\right) \quad i = 1, 2, \ldots, j-1, j+1, \ldots, k \tag{1.30}$$

and $f_i(\boldsymbol{x}_j^*)$ be the upper bound on ε_i, i.e.:

$$\varepsilon_i \leq f_i\left(\boldsymbol{x}_j^*\right) \quad i = 1, 2, \ldots, j-1, j+1, \ldots, k. \tag{1.31}$$

When the bounds ε_i are too low, there is no solution and at least one of these bounds must be relaxed. This approach is more effective than the previous as it is able to generate all Pareto optima. However there are some algorithmic problems that can be preferable linear combination.

There are relevant cases in which the set X has a particular structure such that allows efficient solving with only one objective. Objective aggregation keeps these features of algorithmic efficiency. Vice versa adding constraints, changing the structure of X, prevent application of know algorithms and often transform an easy problem into a difficult one. Hence what you gain in information amount you lost in computational efficiency.

The main disadvantage of this approach is its (potentially high) computational cost, also due to the preliminary individuation of ε_i values. Also, the encoding of the objective functions may be extremely difficult or even impossible for certain applications, particularly if there are too many objectives. Nevertheless, the relative simplicity of the technique (its

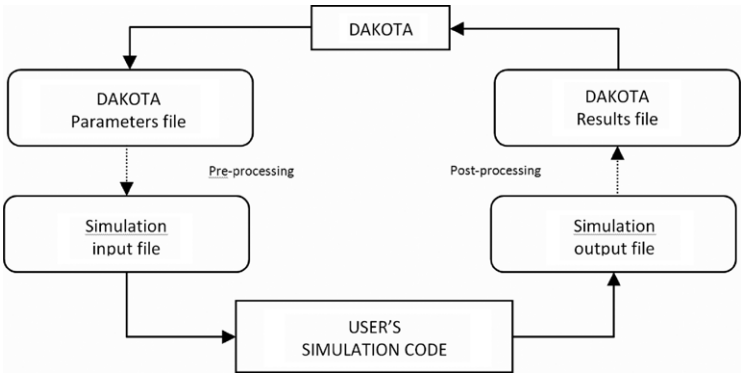

Fig. 3.1. A typical loosely-coupled, or "black-box", relationship between DAKOTA and the simulation code.

main advantage) has made it popular among some researchers (particularly in the engineering field). More information on this method can be found in [6,17].

2.4. MOGA

The JEGA library [18] contains two global optimization methods. The first is a Multi-Objective Genetic Algorithm (MOGA) which performs Pareto optimization. The second is a Single-Objective Genetic Algorithm (SOGA) which performs optimization on a single-objective function. Both methods support general constraints and a mixture of real and discrete variables.

Evolutionary algorithms are based on Darwin's theory of survival of the fittest. This kind of algorithm starts with a randomly selected population of design points in the parameter space, where the values of the design parameters form a 'genetic string' analogous to DNA in a biological system that uniquely represents each design point in the population. Then the method follows a sequence of generations, where the best design points in the population (i.e., those having low objective function values, in case of minimization) are considered to be the most 'fit' and are allowed to survive and reproduce. The algorithm simulates the evolutionary process by employing the mathematical analogs of processes such as natural selection, breeding, and mutation. Ultimately, the method identifies a design point (or a family of design points) that minimizes the objective function of the optimization problem.

Evolutionary algorithms seem particularly suitable to solve multi-objective optimization problems because they deal simultaneously with a set of possible solutions (the so-called population). This allows to find several members of the Pareto optimal set in a single 'run' of the algorithm, instead of having to perform a series of separate runs as in the case of the traditional mathematical programming techniques. The main disadvantage is the computational cost that is in general very high, this is due to the operational process of the method itself.

3. Benchmark description and analysis

The evaluation of the methods previously described has been carried out by solving five selected benchmarks representative of the different possible Pareto fronts (concave, convex, linear, discontinuous) and particular attention has been paid to their effectiveness in terms of number of evaluations required. In order to solve the benchmark optimization problems the program DAKOTA (Design Analysis Kit for Optimization and Terascale Applications, Sandia National Laboratories) has been used (Fig. 3.1). DAKOTA provides for a flexible, extensible interface between any simulation code and a variety of iterative methods and strategies and implements a small variety of techniques to solve multi-objective optimization problems as:

1. linear combination of weights method (see Section 2.2)
2. MOGA method (see Section 2.4)

These two methods show several disadvantages. So, in order to include more effective techniques in the performance comparison, two of them have been implemented within the DAKOTA program:

3. global criterion method (see Section 2.1)
4. ε-constraint method (see Section 2.3).

In the following section, the proposed benchmarks will be described and the results obtained by applying the proposed multi-objective techniques are shown.

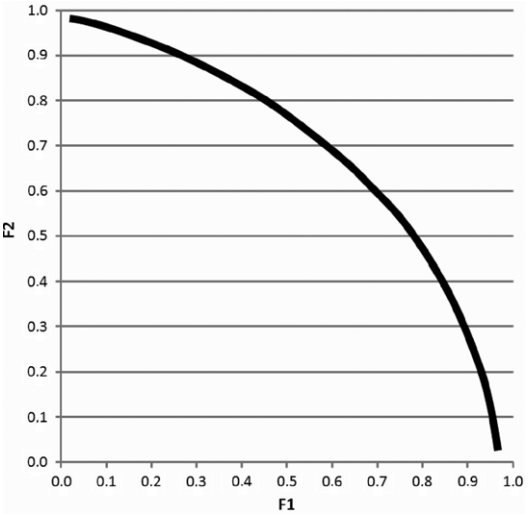

Fig. 3.2. Benchmark 1: reference Pareto front.

3.1. Benchmark 1

The first benchmark [19,20] is described by the minimization of two objective functions:

$$f_1(x) = 1 - \exp\left(-\sum_{i=1}^{3}\left(x_i - \frac{1}{\sqrt{3}}\right)^2\right) \tag{1.32}$$

$$f_2(x) = 1 - \exp\left(-\sum_{i=1}^{3}\left(x_i + \frac{1}{\sqrt{3}}\right)^2\right) \tag{1.33}$$

with side constraints on the decision variables $-4 \leq x_i \leq 4$. The minimum of the objective function $f_1(x)$ is located at $(x_1, \ldots, x_n) = \left(1/\sqrt{3}, \ldots, 1/\sqrt{3}\right)$ whereas the minimum of the objective function $f_2(x)$ is located at $(x_1, \ldots, x_n) = \left(-1/\sqrt{3}, \ldots, -1/\sqrt{3}\right)$. Due to the symmetry of the two functions, the Pareto optimal set corresponds to all points on the line defined by:

$$x_1 = x_2 = x_3 \cap -1/\sqrt{3} \leq x_1 \leq 1/\sqrt{3}. \tag{1.34}$$

It is a bound-constrained problem with two objective functions of three variables described by an exponential function powered to a quadratic function. Fig. 3.2 shows the solution of the problem, i.e. the Pareto front assumed as reference.

The problem has been solved through the methods described in the previous section and the results are shown in Fig. 3.3 (MOGA method), in Fig. 3.4 (linear combination of weights method), in Fig. 3.5 (global criterion method) and in Fig. 3.6 (ε-constraint method). The number of iterations required for the Pareto front identification is shown for each implemented method.

The Pareto front is concave as shown by Fig. 3.2. The optimal solutions obtained by implementing the MOGA method belong to the Pareto front analytically evaluated. The Pareto front can be fully described in detail with the drawback of a very large number of iterations. The method cannot be stopped before the conclusion of the process otherwise any optimal solution would be available for the *decision maker*.

The linear combination method is not able to find any point of the Pareto front with the exception of its extremities. Given that the Pareto front shape is usually unknown a priori, this means the *decision maker* is not able to figure out the other optimal solutions belonging to the Pareto front and is not able to correctly choose between the possible best compromise solutions available.

The global criterion and the ε-constraint methods allow to identify many optimal solutions belonging to the Pareto front with a limited number of iterations. The results obtained allow to identify the complete Pareto front and, as a consequence, the shape of the Pareto front allowing the *decision maker* to select between any possible best compromise solution by interpolation if required.

3.2. Benchmark 2

The second benchmark [20] is a multi-objective problem constructed using Deb's methodology [21]. The optimization problem is described by the minimization of two objective functions:

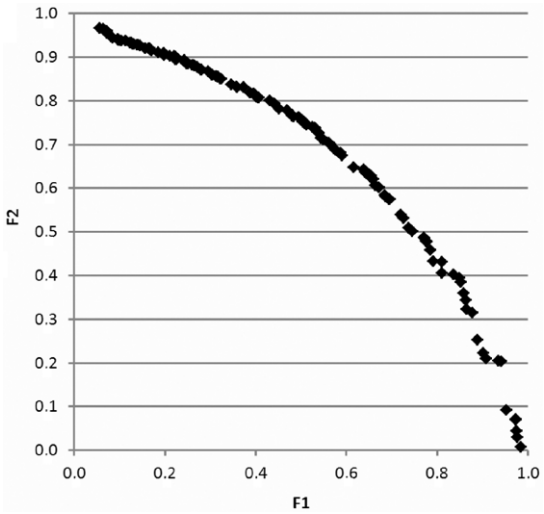

Fig. 3.3. Benchmark 1: MOGA method (iterations: 2500).

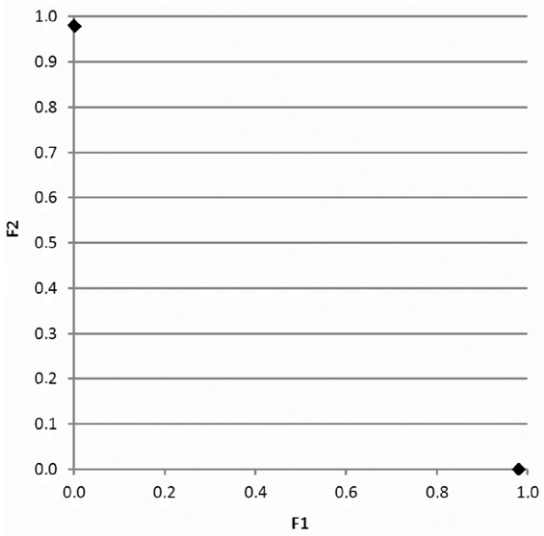

Fig. 3.4. Benchmark 1: Linear combination method (iterations: 258).

$$f_1(x) = x_1 \tag{1.35}$$

$$f_2(x) = (1 + 10x_2) \cdot \left[1 - \left(\frac{x_1}{1 + 10x_2}\right)^2 - \frac{x_1}{1 + 10x_2} \sin(8\pi x_1)\right] \tag{1.36}$$

with side constraints on the decision variables $0 \leq x_1 \leq 1$. It is a bound-constrained problem with two objective functions one is a plane and the other is a sinusoidal function with increasing amplitude. By allowing variables x_1 and x_2 to lie in the interval [0, 1], a two-objective optimization problem with a discontinuous Pareto-optimal front can be defined. In Fig. 3.7 is shown the solution of the problem, i.e. the Pareto front assumed as reference.

The problem has been solved through the methods described in the previous section and the results obtained are shown in Fig. 3.8 (MOGA method), in Fig. 3.9 (linear combination of weights method), in Fig. 3.10 (global criterion method) and in Fig. 3.11 (ε-constraint method). The number of iterations required for the Pareto front identification is shown for each implemented method.

The Pareto front is discontinuous as shown by Fig. 3.7. The comments on the results of the application of the proposed methods to the present benchmark problem are not so different with respect to those previously presented.

The optimal solutions obtained by implementing the MOGA method belong to the Pareto front analytically evaluated. The Pareto front can be fully described in detail with the drawback of a very large number of iterations. The method cannot be stopped before the conclusion of the process otherwise any optimal solution would be available for the *decision maker*.

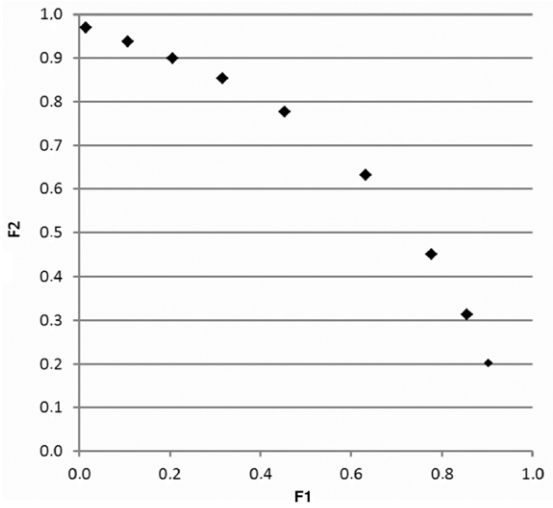

Fig. 3.5. Benchmark 1: Global criterion method (iterations: 286).

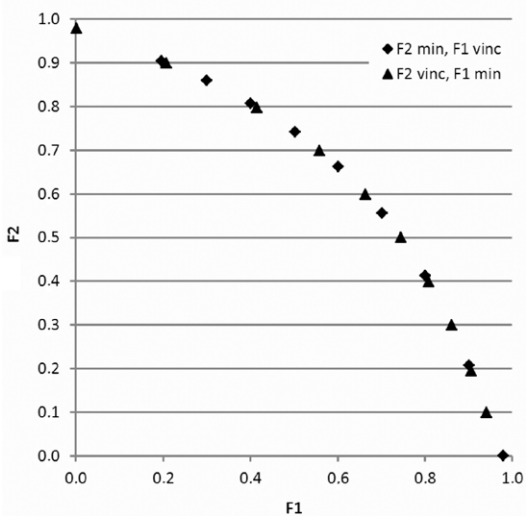

Fig. 3.6. Benchmark 1: ε-constraint method (iterations: 257).

The linear combination method is not able to find any point of the Pareto front with the exception of its extremities. Also for these points, the results are characterized by errors. Given that the Pareto front shape is usually unknown a priori, this means the *decision maker* is not able to figure out the other optimal solutions belonging to the Pareto front and is not able to correctly choose between the possible best compromise solutions available.

The global criterion and the ε-constraint methods allow to identify many optimal solutions belonging to the Pareto front with a limited number of iterations. Given the discontinuity of the Pareto front, its shape can be identified with more difficulties. In any case, the *decision maker* would be able to select between the possible best compromise solutions also by interpolation if required.

3.3. Benchmark 3

The third benchmark [22] is described by the minimization of the mass and the compliance of a four bar plane truss (Fig. 3.12) respectively described by the two objective functions:

$$f_1(\boldsymbol{x}) = L(2x_1 + \sqrt{2}x_2 + \sqrt{2}x_3 + x_4) \tag{1.37}$$

$$f_2(\boldsymbol{x}) = \frac{FL}{E}\left(\frac{2}{x_1} + \frac{2\sqrt{2}}{x_2} - \frac{2\sqrt{2}}{x_3} + \frac{2}{x_4}\right) \tag{1.38}$$

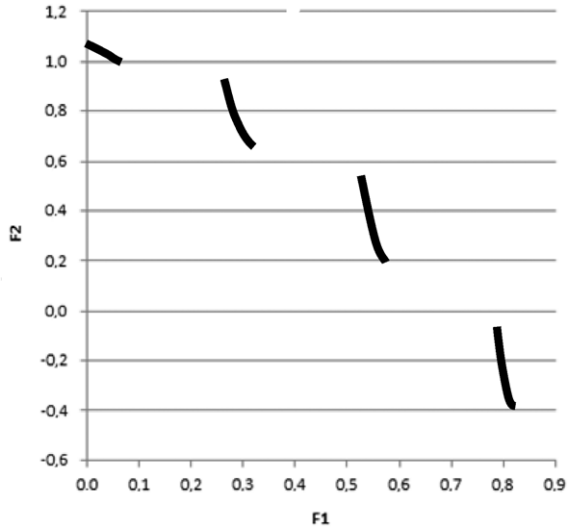

Fig. 3.7. Benchmark 2: reference Pareto front.

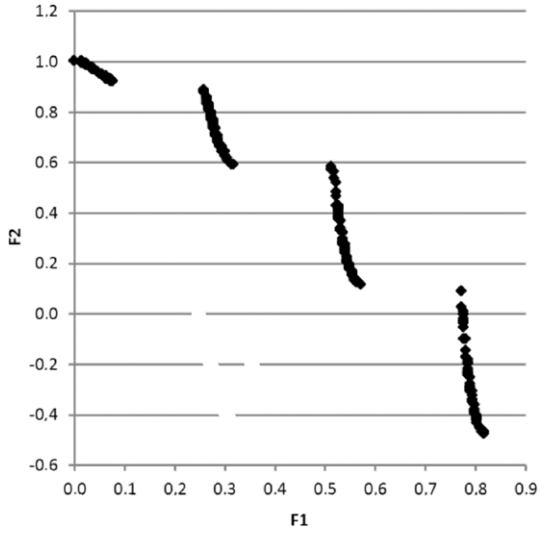

Fig. 3.8. Benchmark 2: MOGA method (iterations: 3000).

with side constraints on the decision variables:

$$(F/\sigma) \leq x_1 \leq 3\,(F/\sigma) \tag{1.39}$$

$$\sqrt{2}\,(F/\sigma) \leq x_2 \leq 3\,(F/\sigma) \tag{1.40}$$

$$\sqrt{2}\,(F/\sigma) \leq x_2 \leq 3\,(F/\sigma) \tag{1.41}$$

$$(F/\sigma) \leq x_4 \leq 3\,(F/\sigma) \tag{1.42}$$

where the design variables are the cross-sectional areas of the bars and $F = 10$ kN, $E = 2 \times 10^5$ kN/cm^2, $L = 200$ cm, $\sigma = 10$ kN/cm^2.

The problem has been solved through the methods described in the previous sections and the results obtained are shown in Fig. 3.14 (MOGA method), in Fig. 3.15 (linear combination of weights method), in Fig. 3.16 (global criterion method) and in Fig. 3.17 (ε-constraint method). The number of iterations required for the Pareto front identification is shown for each implemented method.

The Pareto front is convex as shown by Fig. 3.13. The optimal solutions obtained by implementing the MOGA method belong to the Pareto front analytically evaluated. The Pareto front can be fully described in detail with the drawback of a

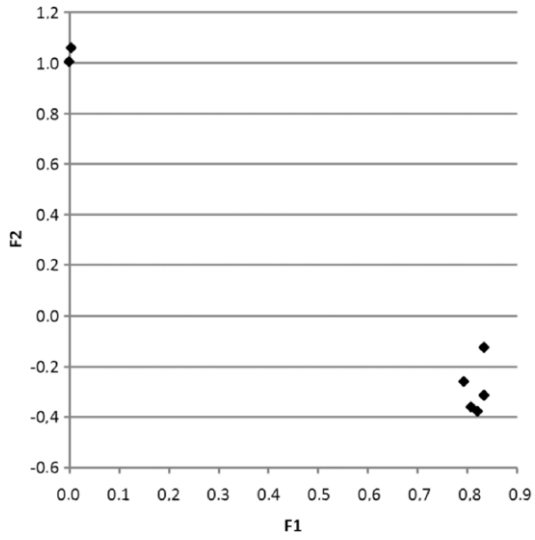

Fig. 3.9. Benchmark 2: Linear combination method (iterations: 875).

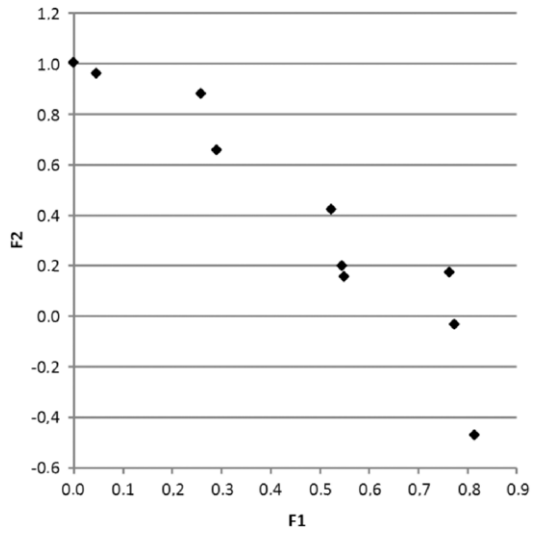

Fig. 3.10. Benchmark 2: Global criterion method (iterations: 427).

very large number of iterations. The method cannot be stopped before the conclusion of the process otherwise any optimal solution would be available for the *decision maker*.

The optimal solutions obtained by implementing the linear combination and the ε-constraint methods belong to the Pareto front and can be obtained with a limited number of iterations, particularly if the former is used. The results obtained allow to identify the complete Pareto front and, as a consequence, the shape of the Pareto front allowing the *decision maker* to select between any possible best compromise solution by interpolation if required.

The global criterion method leads to the identification of best compromise solutions that do not belong to the Pareto front. The results are characterized by errors that can lead to an incorrect evaluation of the best compromise by the *decision maker*.

3.4. Benchmark 4

The fourth benchmark [20,23] is described by the minimization of two objective functions:

$$f_1(x) = (x_1 - 2)^2 + (x_2 - 1)^2 + 2 \tag{1.43}$$

$$f_2(x) = 9x_1 - (x_2 - 1)^2 \tag{1.44}$$

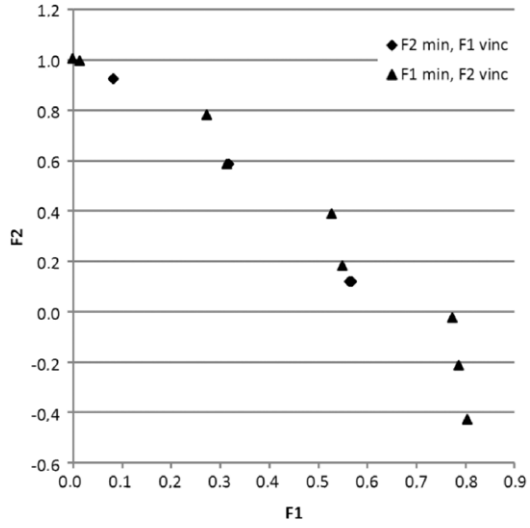

Fig. 3.11. Benchmark 2: ε-constraint method (iterations: 575).

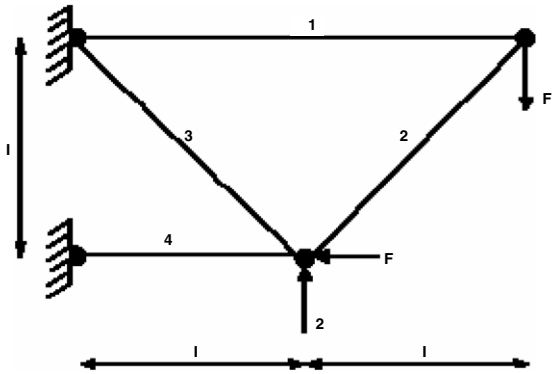

Fig. 3.12. A four bar plane truss.

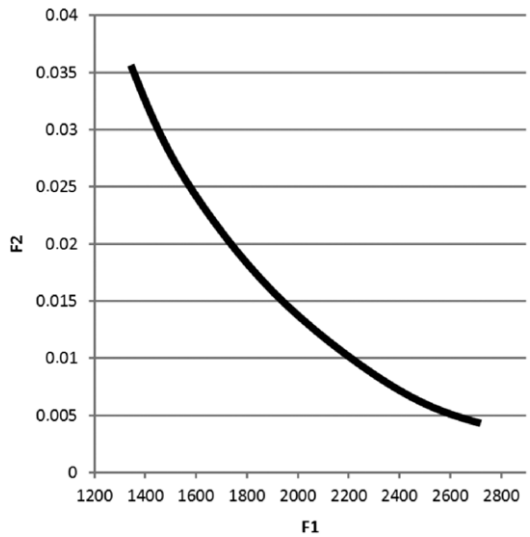

Fig. 3.13. Benchmark 3: reference Pareto front.

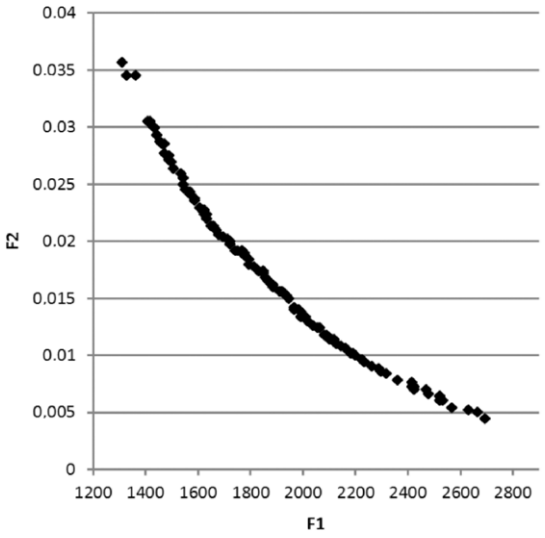

Fig. 3.14. Benchmark 3: MOGA method (iterations: 2500).

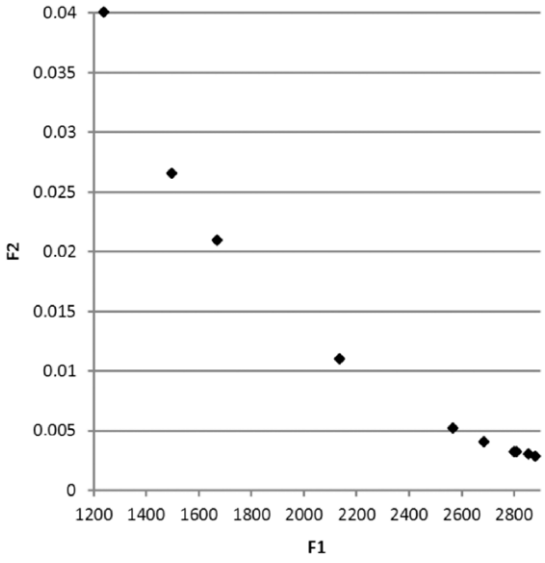

Fig. 3.15. Benchmark 3: Linear combination method (iterations: 261).

with side constraints on the decision variables:

$$x_1^2 + x_2^2 - 225 \leq 0 \tag{1.45}$$

$$x_1 - 3x_2 + 10 \leq 0 \tag{1.46}$$

$$-20 \leq x_1 \leq 20 \tag{1.47}$$

$$-20 \leq x_2 \leq 20. \tag{1.48}$$

It is a bound-constrained problem with two polynomial objective functions of two variables. The first objective function is a smooth unimodal function which has a minimum at point (2, 1). The second objective function decreases monotonically with decreasing x_1 or with increasing absolute value of x_2. The contours of the first function are concentric circles with the center at (2, 1). This function value increases with increasing diameter of the circle. The second function (parallel parabolas) constantly decreases along the line $x_2 = 1$ toward decreasing x_1. Careful observation reveals that the tangential points of circles and parabolas dominate all other points. This is because any such tangential point is better in the second objective than all other points belonging to the same circle. These tangential points are Pareto-optimal points. Fig. 3.18 shows the solution of the problem, i.e. the Pareto front assumed as reference.

The problem has been solved through the methods described in the previous section and the results obtained are shown in Fig. 3.19 (MOGA method), in Fig. 3.20 (linear combination of weights method), in Fig. 3.21 (global criterion method) and

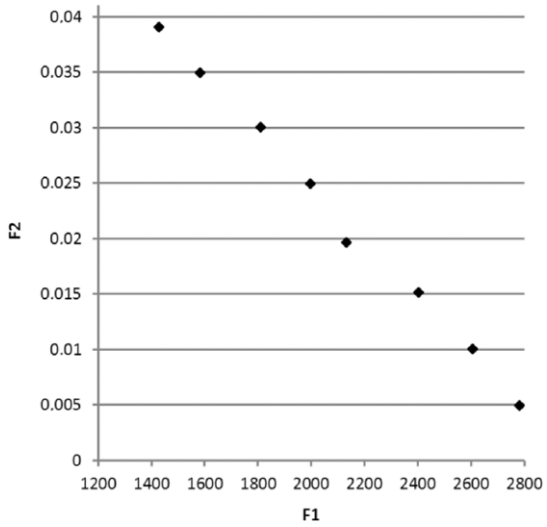

Fig. 3.16. Benchmark 3: Global criterion method (iterations: 450).

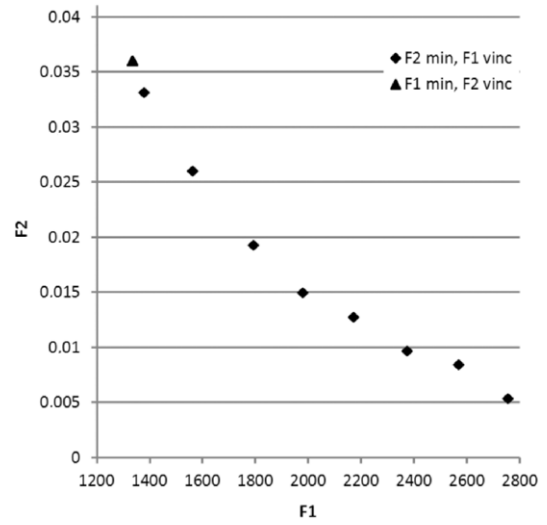

Fig. 3.17. Benchmark 3: ε-constraint method (iterations: 704).

in Fig. 3.22 (ε-constraint method). The number of iterations required for the Pareto front identification is shown for each implemented method.

The Pareto front is linear, to say, concave and convex at the same time, as shown by Fig. 3.18. The optimal solutions obtained by implementing the MOGA method belong to the Pareto front analytically evaluated. The Pareto front can be fully described in detail with the drawback of a very large number of iterations. The method cannot be stopped before the conclusion of the process otherwise any optimal solution would be available for the *decision maker*.

The linear combination method is not able to find any point of the Pareto front with the exception of its extremities. Given that the Pareto front shape is usually unknown a priori, this means the *decision maker* is not able to figure out the other optimal solutions belonging to the Pareto front and is not able to correctly choose between the possible best compromise solutions available.

The global criterion and the ε-constraint methods allow to identify many optimal solutions belonging to the Pareto front with a limited number of iterations. The results obtained allow to identify the complete Pareto front and, as a consequence, the shape of the Pareto front allowing the *decision maker* to select between any possible best compromise solution by interpolation if required.

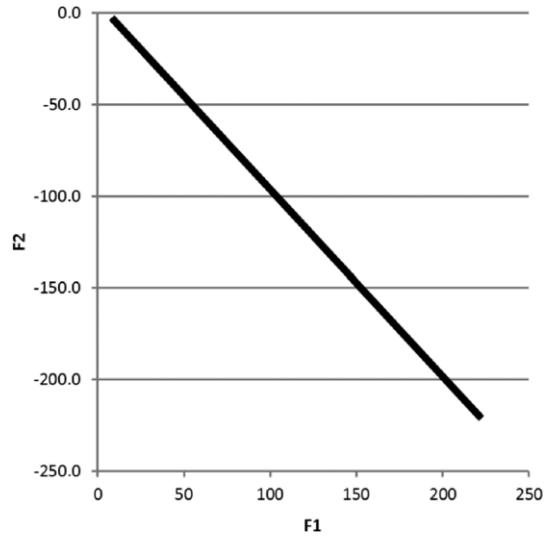

Fig. 3.18. Benchmark 4: reference Pareto front.

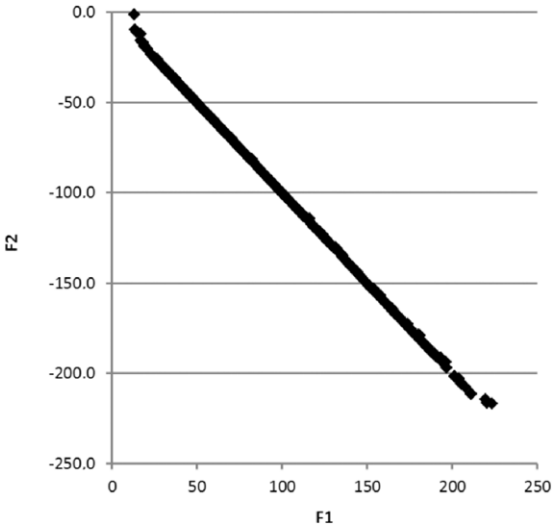

Fig. 3.19. Benchmark 4: MOGA method (iterations: 2000).

3.5. Benchmark 5

The fifth benchmark [24] is based on the analysis of the two-bar plane truss shown in Fig. 3.23. The area of the cross section of the members (A) and the position of the joints 1 and 2 (x) are treated as design variables. The truss is assumed to be symmetric about the y axis. The coordinates of joint 3 are held constant. The weight of the truss and the displacement of the joint 3 are considered as the objective functions f_1 and f_2. The stresses induced in the members are constrained to be smaller than the permissible stress σ_0. Thus the problem is formulated as the minimization of:

$$f_1(\boldsymbol{x}) = 2\rho h x_2 \sqrt{1 + x_1^2} \tag{1.49}$$

$$f_2(\boldsymbol{x}) = \frac{Ph\left(1 + x_1^2\right)^{1.5}\left(1 + x_1^4\right)^{0.5}}{2\sqrt{2}Ex_1^2 x_2} \tag{1.50}$$

subject to:

$$g_1(\boldsymbol{x}) = \frac{P(1 + x_1)(1 + x_1^2)^{0.5}}{2\sqrt{2}x_1 x_2} - \sigma_0 \leq 0 \tag{1.51}$$

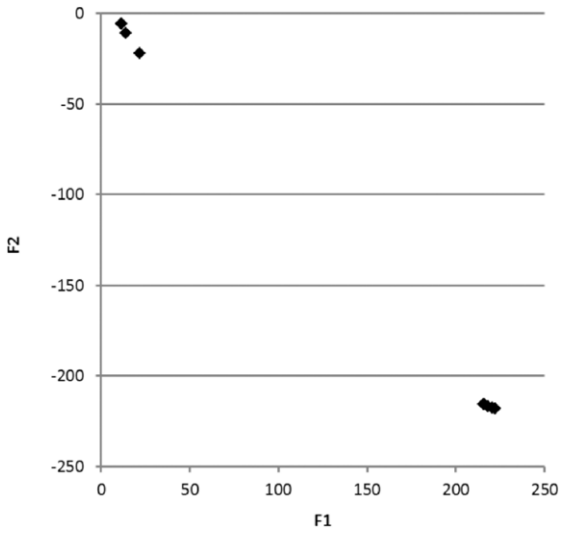

Fig. 3.20. Benchmark 4: Linear combination method (iterations: 480).

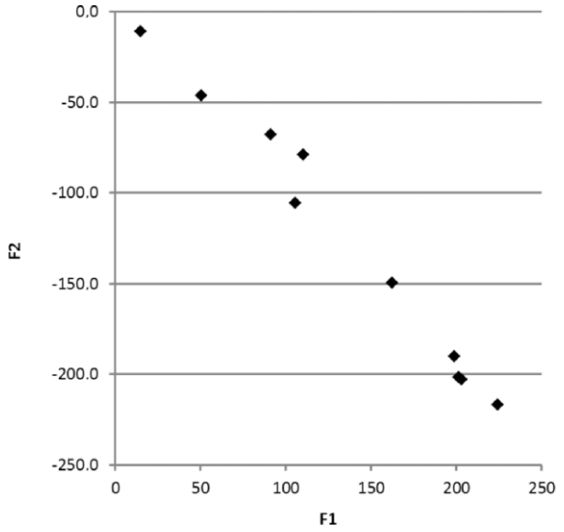

Fig. 3.21. Benchmark 4: Global criterion method (iterations: 288).

$$g_2(\boldsymbol{x}) = \frac{P(-x_1+1)(1+x_1^2)^{0.5}}{2\sqrt{2}x_1x_2} - \sigma_0 \leq 0 \tag{1.52}$$

where $f_1(\boldsymbol{x})$ is the structural weight of the truss, $f_2(\boldsymbol{x})$ is the displacement of joint 3 (Fig. 3.22) and $g_1(\boldsymbol{x})$ and $g_2(\boldsymbol{x})$ are the stress constraints with a limiting value σ_0. In the previous expressions, $x_1 = x/h$, $x_2 = A/A_{\min}$, $E =$ Young' modulus and $\rho =$ density of material. It is assumed that: $\rho = 0.283$ lb/in^3, $h = 100$ in, $P = 10^4$ lb, $E = 3 \times 10^7$ lb/in^2, $\sigma_0 = 2 \times 10^4$ lb/in^2, $A_{\min} = 1$ in^2, and the lower and upper bounds of the design variables are $0,1 \leq x_1 \leq 2,25$ and $0,5 \leq x_2 \leq 2,5$.

It is a bound problem of two variables and two polynomial objective functions. In Fig. 3.24 the solution of the problem is shown, i.e. the Pareto front assumed as reference. The problem has been solved through the methods described in the previous sections and the results obtained are shown in Fig. 3.25 (MOGA method), in Fig. 3.26 (linear combination of weights method), in Fig. 3.27 (global criterion method) and in Fig. 3.28 (ε-constraint method). The number of iterations required for the Pareto front identification is shown for each implemented method.

The Pareto front is convex as shown by Fig. 3.24. The optimal solutions obtained by implementing the MOGA method belong to the Pareto front analytically evaluated. The Pareto front can be fully described in detail with the drawback of a very large number of iterations. The method cannot be stopped before the conclusion of the process otherwise any optimal solution would be available for the *decision maker*.

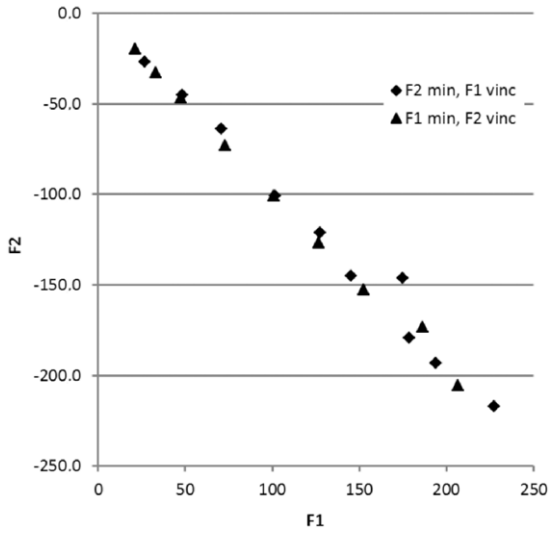

Fig. 3.22. Benchmark 4: ε-constraint method (iterations: 758).

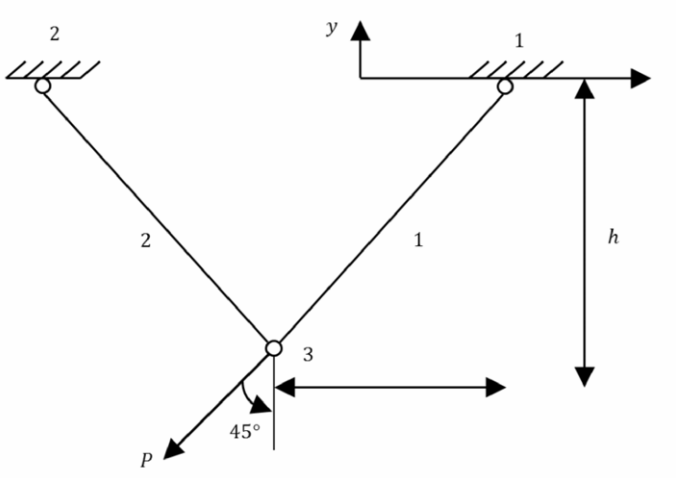

Fig. 3.23. A two-bar plane truss.

The optimal solutions obtained by implementing the linear combination and the ε-constraint methods belong to the Pareto front and can be obtained with a limited number of iterations, particularly if the former is used. The results allow to identify the complete Pareto front and, as a consequence, the shape of the Pareto front allowing the *decision maker* to select between any possible best compromise solution by interpolation if required.

The global criterion method leads to the identification of best compromise solutions that do not belong to the Pareto front. The results are characterized by errors that can lead to an incorrect evaluation of the best compromise by the *decision maker*.

3.6. Results analysis

The analysis of the results obtained by solving the proposed five benchmarks shows that the MOGA method leads always to a correct and accurate identification of the whole Pareto front. The results obtained by implementing the MOGA method do not depend on the functions analyzed, both objectives and constraints. The drawback of the MOGA method is the enormous number of iterations required and, consequently, the large computational time necessary to reach the Pareto front identification. The large computational time makes this method generally not acceptable in usual engineering problems where, e.g., the Finite Element Method is used and models with a large number of degrees of freedom are implemented.

The other three proposed methods allow for an effective reduction of the computational cost, but introduce several other disadvantages. The linear combination of weights method cannot generate certain portions of the Pareto front when its shape is concave because of its mathematical formulation (see Section 2.2). The global criterion method is particularly efficient with

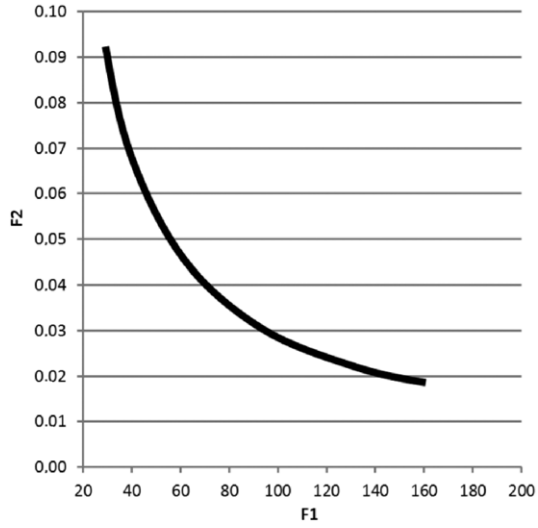

Fig. 3.24. Benchmark 5: reference Pareto front.

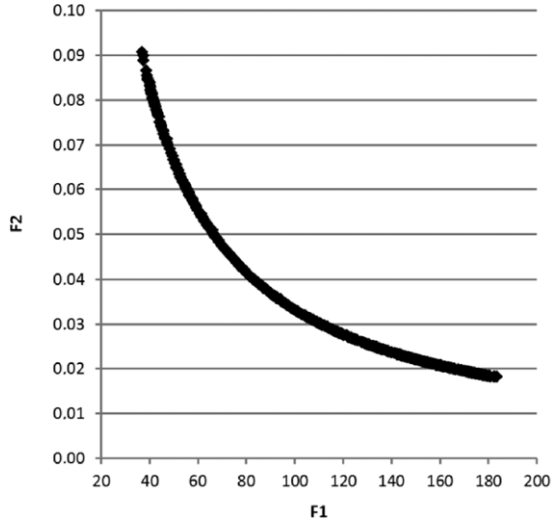

Fig. 3.25. Benchmark 5: MOGA method (iterations: 3000).

a non-convex or discontinuous Pareto front whereas it fails in the identification of concave Pareto fronts. The ε-constraint method is particularly efficient with convex and linear Pareto fronts and is always able to identify the best compromise solutions belonging to the Pareto front independently of its shape. In addition, the global criterion method is particularly suitable when the ideal value of the objective functions is known and can be set as target whereas the ε-constraint method is particularly convenient when the objective function bounds are known.

4. Engineering application

The proposed methods have been used to solve an engineering application problem concerning the engine bracket of a medium size car shown in Fig. 4.1. Engine brackets link the engine to the car body and have to support the static and dynamic loads due to the suspended engine mass. Each engine bracket is linked by spot welding to the car body structure and through threaded fasteners to the engine.

Fig. 4.2 shows the finite element model of the component obtained by using the pre-processor HyperMesh (Altair Engineering Inc., www.altair.com). Figs. 4.3 and 4.4 show the boundary conditions implemented in the multi-objective optimization problem proposed. The presence of the spot welds has been simulated by introducing full constraints (white, Fig. 4.3) and the mass of the engine has been applied as a concentrated load. The engine mass is uniformly distributed on the three threaded fasteners by rigid elements (light blue, Fig. 4.4). Moments acting on the engine bracket due to acceleration fields have been neglected and a constant velocity of the car has been taken into consideration.

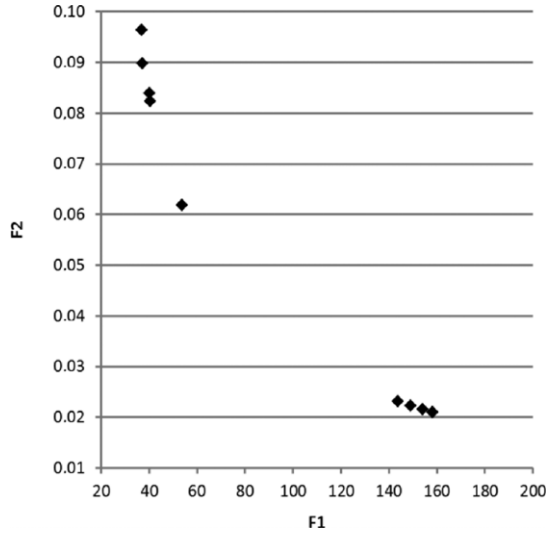

Fig. 3.26. Benchmark 5: Linear combination method (iterations: 273).

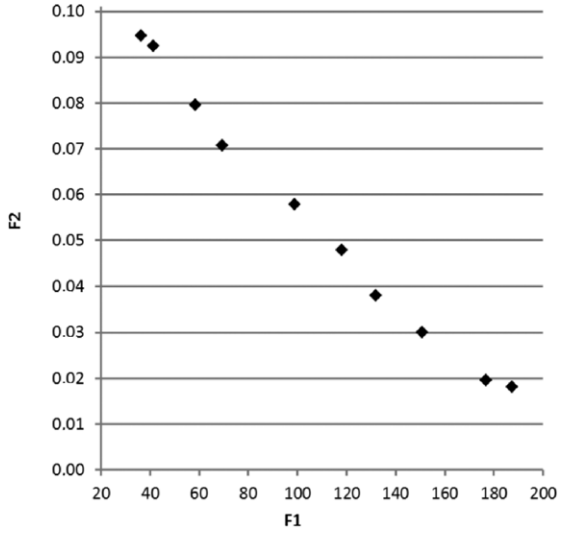

Fig. 3.27. Benchmark 5: Global criterion method (iterations: 405).

The properties of the material the bracket is made of (steel) are:

- material density: $\rho = 7850$ kg/m^3.
- Young's modulus: $E = 205\,000$ MPa.
- Poisson ratio: $\nu = 0.3$.
- yield stress: $\sigma_y = 460$ MPa.
- ultimate stress: $\sigma_u = 520$ MPa.

The engine bracket is manufactured by forming and bending a sheet metal plate of uniform thickness. As a consequence, the sheet metal thickness and its initial shape can be modified by taking into account the available space. Shape variables have been defined by a *morphing* process by preserving the partial symmetry of the component due to the use of the same mold for production cost reduction. Figs. 4.5–4.7 show three *shape* variables that allow to modify the curvature of the vertical walls of the component. Fig. 4.8 shows the *shape* variable that allows to modify the inclination of the vertical walls. Fig. 4.9 shows the *shape* variables that allows to create a circular edge at the internal radius of the central hole of the component. Finally Fig. 4.10 shows the *shape* variable that allows to modify the height of the vertical wall.

Design variable values have been bound by taking into account the working conditions of the component, the production technologies and the objectives of the optimization problem, i.e. mass minimization and frequency maximization. The thickness value ranges between 4 and 7 mm as required by the productive process. *Shapes* variables instead vary in a suitable range according to component geometry (Table 4.1):

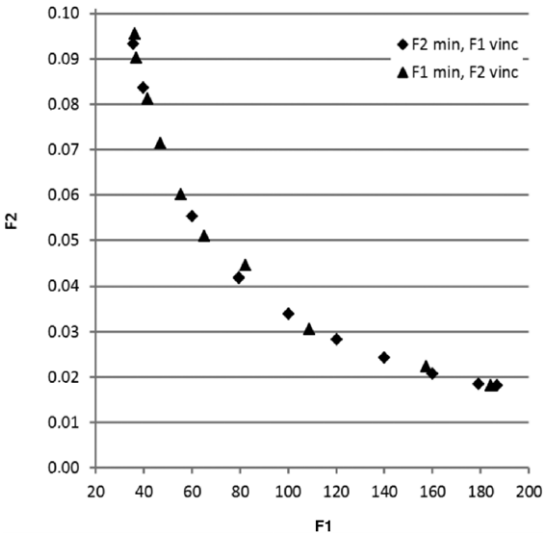

Fig. 3.28. Benchmark 5: ε-constraint method (iterations: 851).

Fig. 4.1. Engine bracket: CAD model.

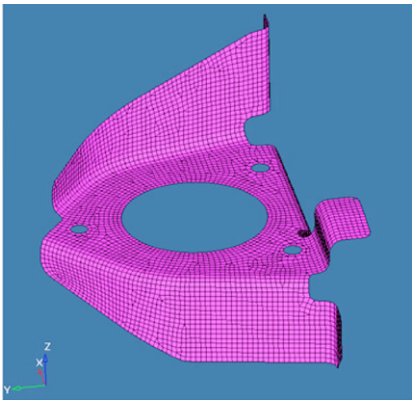

Fig. 4.2. Engine bracket: finite element model.

- shape variables 1–3 range between −0.5 and 1 in order to allow a correct variation of the curvature by avoiding irregular shapes or an ill conditioned mesh;
- shape variable 4 ranges among 0 and 1 in order to avoid the presence of undercuts (inclination angle larger than 90°);
- shape variables 5 and 6 range between −1 and 1 in order to allow the creation of an upward or downward edge and to increase or decrease the vertical height of the wall.

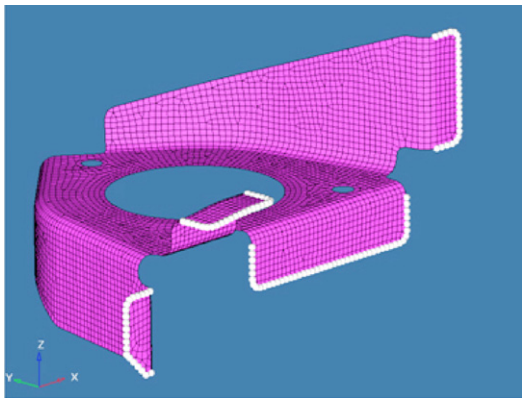

Fig. 4.3. Boundary conditions: constraints.

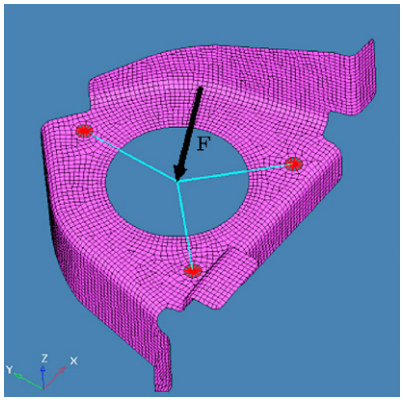

Fig. 4.4. Boundary conditions: applied load.

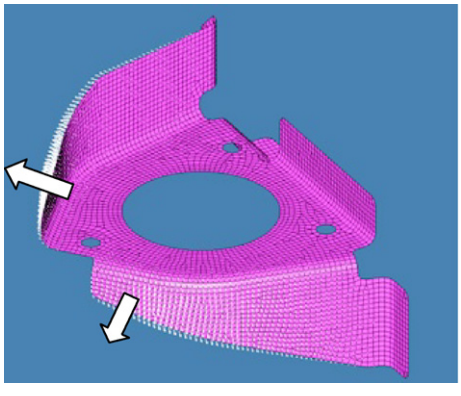

Fig. 4.5. Shape variable 1.

Table 4.1
Thickness and shape variables constraints.

Design variable	Initial value	Lower bound	Upper bound
THICK	5.0	4.0	7.0
SHAPE 01	0.0	−0.5	1.0
SHAPE 02	0.0	−0.5	1.0
SHAPE 03	0.0	−0.5	1.0
SHAPE 04	0.0	0.0	1.0
SHAPE 05	0.0	−1.0	1.0

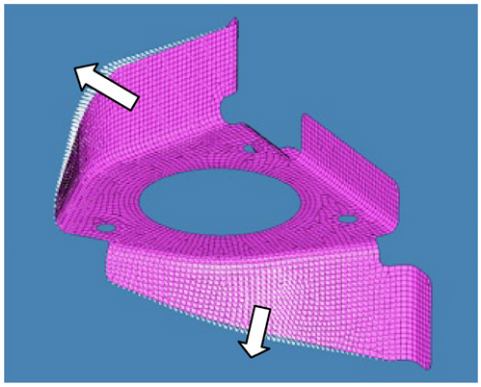

Fig. 4.6. Shape variable 2.

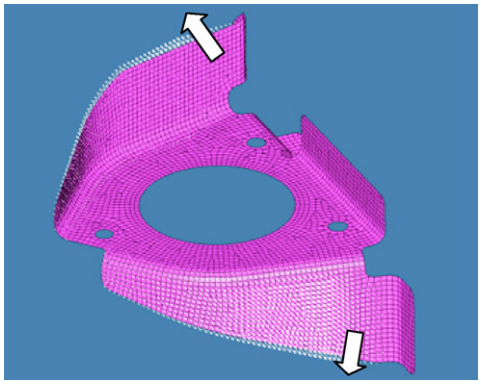

Fig. 4.7. Shape variable 3.

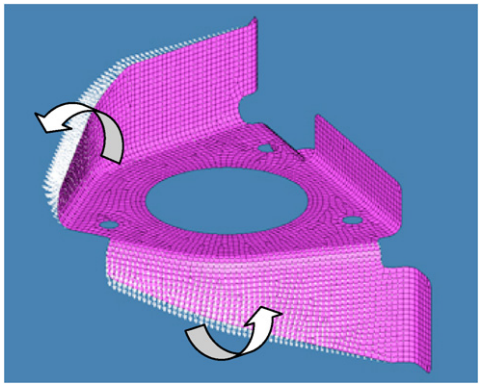

Fig. 4.8. Shape variable 4.

The structural responses have been obtained by using the solver Abaqus (Dassault Systèmes, www.simulia.com). They are:

- the maximum von Mises stress on the component;
- the maximum displacement in correspondence of the threaded fasteners;
- the first natural frequency of the component;
- the total mass of the component.

4.1. Single-objective problems

Two single-objective optimization problems have been defined first as the maximization of the first natural frequency and the minimization of the component mass by imposing a constraint on the component mass and the first natural

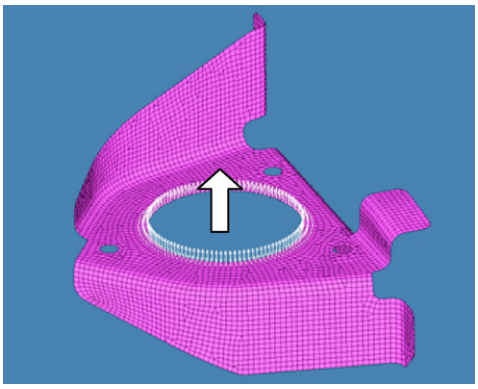

Fig. 4.9. Shape variable 5.

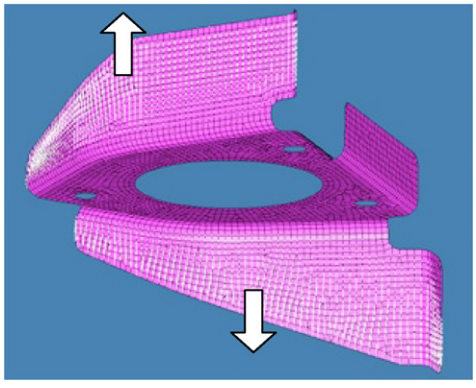

Fig. 4.10. Shape variable 6.

Table 4.2
Design variable, objective function and constraint values: first natural frequency maximization problem optimal solution.

Design variable	Optimal value	Objective function	Optimal value
THICK	6.89	FREQ 01	1051 Hz
SHAPE 01	−0.27		
SHAPE 02	−0.49	Design constraint	Value
SHAPE 03	−0.33	MAX STRESS	204 MPa
SHAPE 04	0.36	MAX DISPL	0.08 mm
SHAPE 05	0.67	MASS	1.89 kg
SHAPE 06	−0.88		

frequency, respectively. Then a multi-objective optimization problem looking for the first natural frequency maximization and the component mass minimization has been defined and solved.

4.1.1. First natural frequency maximization

The first single-objective optimization problem has been defined as the maximization of the first natural frequency of the component with constraints on the global mass, the maximum Von Mises stress and the maximum displacement of the connection points with the engine:

maximize:	I natural frequency (FREQ 01)
subject to:	Von Mises stress (MAX STRESS) $<$ 368 MPa
	displacement (MAX DISPL) $<$ 0.4 mm
	mass (MASS) $<$ 1.9 kg

The optimal solution is shown in Fig. 4.11 and the value of the design variables, of the objective function and of the constraints is summarized in Table 4.2.

The sheet metal thickness approaches the upper bound of 7 mm, shape variables 1–3 provide for a small modification of the side wall geometry while the fourth shape variable provides for a small inclination. The fifth shape variable controlling the height of a possible edge surrounding the central hole of the component shows a meaningful change due to its positive

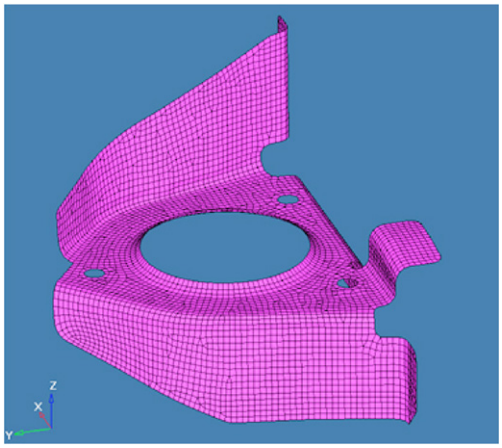

Fig. 4.11. Optimal solution for the first natural frequency maximization problem.

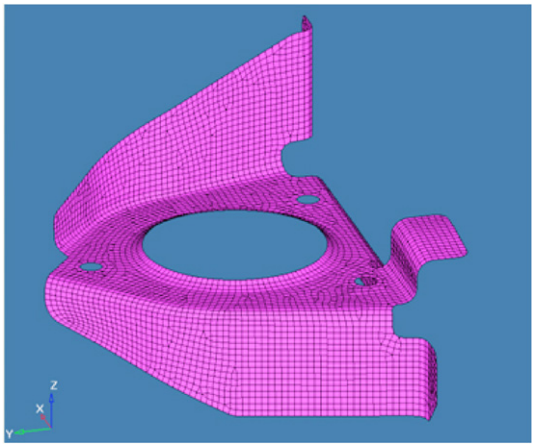

Fig. 4.12. Optimal solution for the mass minimization problem.

effect in increasing the natural frequencies. The sixth design variable controlling the height of the side walls increases leading to an increase of the component global mass. The first natural frequency maximization process has been controlled by the constraint on the maximum component mass that reaches the maximum allowable value.

4.1.2. Mass minimization

The second single-objective optimization problem has been defined as the minimization of the component mass with constraints on the first natural frequency, the maximum Von Mises stress and the maximum displacement of the connection points with the engine:

minimize:	mass (MASS)
subject to:	Von Mises stress (MAX STRESS) $<$ 368 MPa
	maximum displacement (MAX DISPL) $<$ 0.4 mm
	I natural frequency (FREQ 01) $>$ 800 Hz

The optimal solution is shown in Fig. 4.12 and the corresponding value of the design variables is summarized in Table 4.3.

The optimization process searches for a minimum mass by decreasing the sheet metal thickness that reaches its lower bound and by reducing the height of the side walls of the component. The curvature of the side walls has not been changed significantly whereas an edge surrounding the central hole has been created in order to fulfill the constraints concerning the maximum allowable displacement and the minimum first natural frequency constraints. The mass minimization process has been controlled by the constraints on the maximum displacement and the minimum first natural frequency of the component that reach their limit values.

Table 4.3
Design variable, objective function and constraint values: mass minimization problem optimal solution.

Design variable	Optimal value	Objective function	Optimal value
THICK	5.11	MASS	1.34 kg
SHAPE 01	−0.37		
SHAPE 02	−0.47	Design constraint	Value
SHAPE 03	0.03	MAX STRESS	333 MPa
SHAPE 04	0.11	MAX DISPL	0.18 mm
SHAPE 05	0.50	FREQ 01	823 Hz
SHAPE 06	0.74		

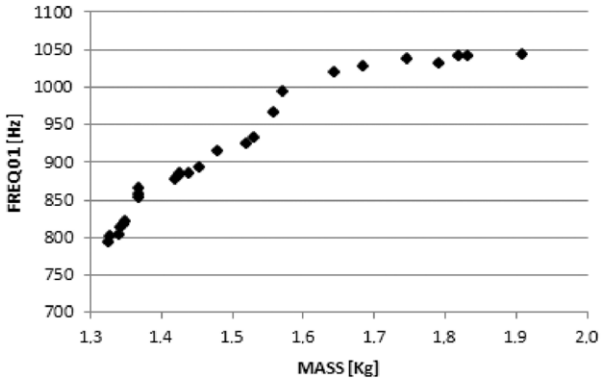

Fig. 4.13. MOGA method (iterations: 1000).

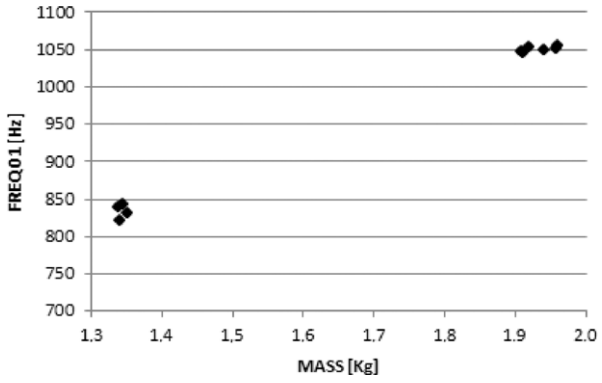

Fig. 4.14. Linear combination method (iterations: 612).

4.2. Multi-objective problem

The results of the single-objective optimization problems previously presented show that the maximization of the first natural frequency and the minimization of the mass of the engine bracket are competing objectives and that a multi-objective optimization problem is worth to be implemented in order to identify the complete set of optimal compromise solutions represented by the Pareto front.

The multi-objective optimization problem has been set up as:

minimize:	mass (MASS)
maximize:	I natural frequency (FREQ 01)
subject to:	Von Mises stress (MAX STRESS) $<$ 368 MPa
	displacement (MAX DISPL) $<$ 0.4 mm

It is a bound problem with two polynomial objective functions, two constraints on structural responses and 7 design variables. The problem has been solved through the methods described in previous sections and the results are shown in Fig. 4.13 (MOGA method), in Fig. 4.14 (linear combination of weights), in Fig. 4.15 (global criterion method) and in Fig. 4.16 (ε-constraint method).

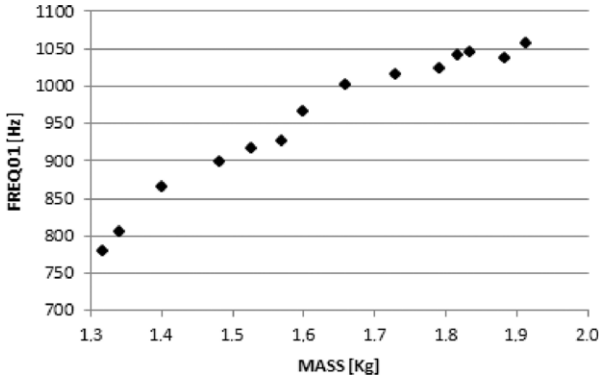

Fig. 4.15. Global criterion method (iterations: 556).

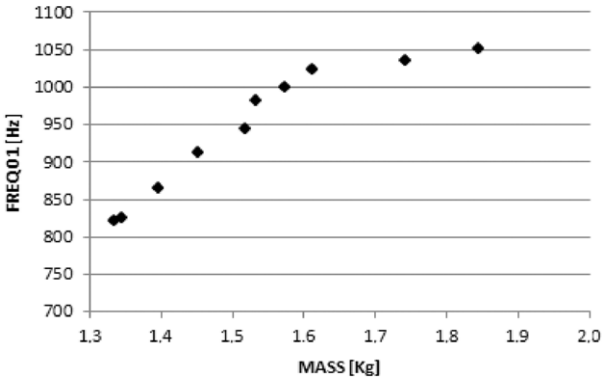

Fig. 4.16. ε-constraint method (iterations: 440).

The MOGA method describes a detailed and accurate Pareto front but its computational cost is inevitably large (limited to 1000 iterations). As already pointed out, the MOGA method cannot be stopped before the conclusion of the process otherwise any optimal solution would be available for the *decision maker*. The Pareto front as identified by using the MOGA method is non-convex and justifies the poor results obtained by using the linear combination method that is able to identify only its extremity points. The more convenient methods are the ε-constraint and the global criterion. The number of iterations required in order to identify the Pareto front with enough details is limited even if the global criterion method introduces some errors in its evaluation as can be seen comparing the results shown in Figs. 4.13–4.16. The ε-constraint and the global criterion methods are more useful in practice with respect to the MOGA method since the multi-objective optimization process is characterized by the solution of a sequence of different single-objective optimization problems and provides for a point of the front at the convergence as a result of the solution of each of them.

5. Conclusions

Computational models describing the behavior of complex physical systems are often used in the engineering design field to identify better or optimal solutions with respect to previously defined performance criteria. Multi-objective optimization problems arise and the set of optimal compromise solutions (Pareto front) has to be identified by an effective and complete search procedure in order to let the designer to carry out the best choice.

Four multi-objective optimization techniques have been analyzed by describing their formulation, advantages and disadvantages: the MOGA method, the linear combination of weights method, the global criterion method and the ε-constraint method. The evaluation of these methods has been carried out by solving five selected benchmarks representative of the different possible Pareto fronts (concave, convex, linear, discontinuous) and particular attention has been paid to their effectiveness in terms of number of evaluations required.

The MOGA method leads always to a correct and accurate identification of the whole Pareto front. The results obtained by implementing the MOGA method do not depend on the analyzed objective and constraint functions. The drawback of the MOGA method is the large number of iterations required and, consequently, the large computational effort required to identify the Pareto front. The large computational effort makes this method generally not acceptable in usual engineering problems where, e.g., the Finite Element Method is used and models with a large number of degrees of freedom are implemented.

The linear combination of weights and the global criterion methods allow for an effective reduction of the computational cost, but introduce several other advantages and disadvantages. The linear combination of weights method cannot generate certain portions of the Pareto front when its shape is concave because of its mathematical formulation. The global criterion method is particularly suitable when the ideal value of the objective functions is known and can be set as target, it is particularly efficient in the identification of non-convex or discontinuous Pareto fronts, but it fails in the identification of concave Pareto fronts.

The ε-constraint method allows for an effective reduction of the computational cost. It is particularly convenient when the objective function bounds are known and it is particularly efficient with convex and linear Pareto fronts. The ε-constraint method is always able to identify the best compromise solutions belonging to the Pareto front independently of its shape.

The Pareto front of common multi-objective engineering optimization problems is usually unknown a priori. If the evaluation of the objective and constraint functions is computationally expensive, it is necessary to implement optimization methods able to identify the shape of the Pareto front with a reduced number of evaluations. The ε-constraint method showed the best results and, as a consequence, has to be preferred in this kind of applications.

References

[1] T. Bäck, D.B. Fogel, Z. Michalewicz (Eds.), Evolutionary Algorithms in Theory and Practice, Institute of Physics Publications, Bristol, 2000.
[2] M. Ehrgott, Multicriteria Optimization, Springer, Berlin, 2000.
[3] A. Osyczka, Multicriteria optimization for engineering design, in: A. Osyczka (Ed.), Design Optimization, Academic Press, 1985, pp. 193–227.
[4] C.A. Coello Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Comput. Methods Appl. Mech. Engrg. 191 (2002) 1245–1287.
[5] J.L. Cohon, D.H. Marks, A review and evaluation of multiobjective programming techniques, Water Resour. Res. 11 (1975) 208–220.
[6] A. Osyczka, Multicriterion Optimization in Engineering with FORTRAN Programs, John Wiley & Sons, New York, 1984.
[7] L.M. Boychuk, V.O. Ovchinnikov, Principal methods of solution of multicriterial optimization problems, Sov. Automat. Control 6 (1973) 1–4.
[8] M.E. Salukvadze, On the existence of solution in problems of optimization, J. Optim. Theory Appl. 13 (1974) 203–217.
[9] K. Koski, Multicriterion optimization in structural design, in: A. Atrek (Ed.), New Directions in Optimum Structural Design, 1984.
[10] L. Duckstein, Multiobjective optimization in structural design: the model, in: A. Atrek (Ed.), New Directions in Optimum Structural Design, 1984.
[11] A.P. Wierzbicki, On the use of penalty functions in multiobjective optimization, in: Proceedings of the International Symposium on Operations Research, 1978.
[12] M. Zeleny, Compromise programming, in: M. Zeleny (Ed.), Multiple Criteria Decision Making, 1973.
[13] J.L. Cochrane, M. Zeleny (Eds.), Multiple Criteria Decision Making, McGraw-Hill Book Company, New York, 1982.
[14] L.A. Zadeh, Optimality and nonscalar-valued performance criteria, IEEE Trans. Automat. Control 8 (1963) 59–60.
[15] S. Gass, T.L. Saaty, The computational algorithm for the parametric objective function, Naval Res. Logist. Quart. 2 (1955) 39–45.
[16] L. Lasdon, D. Wismer, Y. Haimes, On a bicriterion formulation of the problems of integrated system identification and system optimization, IEEE Trans. Syst. Man Cybern. 1 (1971) 296–297.
[17] Z. Lounis, M.Z. Cohn, Multiobjective optimization of prestressed concrete structures, J. Struct. Eng. ASCE 119 (1993) 794–809.
[18] K. Lewis, J.E. Eddy, Effective generation of Pareto sets using genetic programming, in: Proceedings of ASME Design Engineering Technical Conference, 2001.
[19] P.J. Fleming, C.M. Fonseca, Multiobjective genetic algorithms made easy: selection, sharing, and mating restriction, in: Proceedings of the First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, 1995.
[20] C.A. Coello Coello, G.B. Lamont, D.A. Van Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems, Springer, New York, 2007.
[21] K. Deb, Multi-objective genetic algorithms: problem difficulties and construction of test problems, Evol. Comput. 7 (1999) 205–230.
[22] C.A. Coello Coello, A short tutorial on evolutionary multiobjective optimization, in: E. Zitzler, K. Deb, L. Thiele, C.A. Coello Coello, D. Corne (Eds.), First International Conference on Evolutionary Multi-Criterion Optimization, Springer-Verlag, 2001.
[23] N. Srinivas, K. Deb, Multiobjective optimization using nondominated sorting in genetic algortihms, Evol. Comput. 2 (1994) 221–248.
[24] S.S. Rao, Game theory approach for multiobjective structural optimization, Comput. Struct. 25 (1987) 119–127.