OR Spectrum (2002) 24: 193–213

OR Spectrum

On scalarizing functions in multiobjective optimization*

Kaisa Miettinen and Marko M. Mäkelä

Department of Mathematical Information Technology, University of Jyväskylä, P.O. Box 35 (Agora), 40351 Jyväskylä, Finland (e-mail: {miettine,makela}@mit.jyu.fi)

Received: June 11, 2001 / Accepted: October 31, 2001

Abstract. Scalarizing functions play an essential role in solving multiobjective optimization problems. Many different scalarizing functions have been suggested in the literature based on different approaches. Here we concentrate on classification and reference point-based functions. We present a collection of functions that have been used in interactive methods as well as some modifications. We compare their theoretical properties and numerical behaviour. In particular, we are interested in the relation between the information provided and the results obtained. Our aim is to select some of them to be used in our WWW-NIMBUS optimization system.

Key words: Multiple criteria decision making (MCDM) – Nonlinear optimization – Scalarizing function – Classification – Reference point

1 Introduction

Many real-life optimization problems involve multiple conflicting criteria. In these problems, it is not possible to mathematically define a single optimal solution but we have a set of compromises, that is, a set of so-called Pareto optimal solutions. Finding a solution to a multiobjective optimization problem usually necessitates additional preference information from a human decision maker.

A large variety of methods have been suggested for solving multiobjective optimization problems (see, e.g. [10, 17, 34, 36]). Multiobjective optimization problems are usually solved by utilizing scalarization. Via scalarization, the problem is transformed into a single objective optimization problem involving possibly some

* The research was supported by the Academy of Finland, grants #65760, #71822 and #8583. The authors thank Mr. J. Huikari for his significant efforts in providing numerical results and Mr. H. Maaranen for helping with Figure 5.

parameters or additional constraints. In most scalarizing functions, preference information of the decision maker is taken into consideration. After the scalarization phase, the widely developed theory and methods of single objective optimization are available.

Multiobjective optimization methods utilize different scalarizing functions in different ways. The input requested from the decision maker may consist of trade-off information, marginal rates of substitution or desirable objective function values, etc. Furthermore, the scalarization may be performed once or repeatedly as a part of an iterative process. When methods are introduced in the literature, the optimality of the results produced is usually proved. On the other hand, it is not so common to justify why some specific form of scalarization is used.

Comparisons of multiobjective optimization methods have been realized in different ways. Some of them use human decision makers (see, e.g., [5,28,41]) whereas some of them replace the decision maker with a value (or utility) function (see, e.g. [1,23,32]). In addition, some methods have been compared based on intuition (see, e.g. [12,15–17,19,35]). However, comparisons of the underlying scalarizing functions are rare. Here, [3] makes an exception. There, two scalarizing functions are analyzed and their ability to generate different solutions is studied.

In what follows, we concentrate on classification (see, e.g., [2]) and reference point-based (see, e.g., [42]) scalarizing functions. A reference point consists of aspiration levels, that is, objective function values that are desirable or acceptable for the decision maker. In classification, the decision maker specifies what kind of changes (s)he would like to see in the objective function values when compared to the current solution. Thus, these two ways of specifying preference information are closely related. The difference between them is the fact that in classification, some objective function values must be allowed to be impaired (thus, we move around the Pareto optimal set). On the other hand, a reference point can be specified irrespective of the current solution. In other words, the reference point can be infeasible.

Note that some reference point-based scalarizing functions have been developed so that they utilize both aspiration and so-called reservation levels. The reservation levels represent objective function values that are still tolerable for the decision maker. In this presentation, we do not consider scalarizing function involving reservation levels. For more information about them, we refer, for example, to [44,45] and references therein.

In this paper, we discuss several widely-known scalarizing functions and their modifications together with some variations introduced more recently. The main objective is to compare the scalarizing functions both theoretically and numerically. In addition to collecting the characteristics of the functions, we demonstrate their performance with a set of test problems. The basic idea is to study what kind of solutions different functions generate based on very similar preference data. In other words, we examine how the choice of the scalarizing function affects the solution obtained. We wish to emphasize that we do not compare the methods but only the scalarization part. This means that we do not study the goodness of the methods where the scalarizing functions have been used.

The rest of the paper is organized as follows. The basic concepts are introduced in Section 2. Section 3 is devoted to the scalarizing functions and their theoretical properties and behaviour. The performances of the scalarizing functions in twenty numerical tests are presented in Section 4. Finally, the paper is concluded in Section 5.

2 Concepts and notations

We handle *multiobjective optimization problems* of the form

$$\begin{array}{ll} \text{minimize} & \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})\} \\ \text{subject to} & \mathbf{x} \in S \end{array} \tag{1}$$

involving k (≥ 2) conflicting lower semicontinuous *objective functions* $f_i : \mathbf{R}^n \to \mathbf{R}$ that we want to minimize simultaneously. The *decision (variable) vectors* $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ belong to the nonempty compact *feasible region* $S \subset \mathbf{R}^n$.

Objective vectors consist of *objective (function) values* $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x}))^T$ and the image of the feasible region is denoted by $Z = \mathbf{f}(S) \subset \mathbf{R}^k$.

In multiobjective optimization, vectors are regarded as optimal if their components cannot be improved without deterioration to at least one of the other components.

Definition 1 *A decision vector* $\mathbf{x}' \in S$ *is* Pareto optimal *if there does not exist another* $\mathbf{x} \in S$ *such that* $f_i(\mathbf{x}) \leq f_i(\mathbf{x}')$ *for all* $i = 1, \dots, k$ *and* $f_j(\mathbf{x}) < f_j(\mathbf{x}')$ *for at least one index* j*. On the other hand,* $\mathbf{x}'$ *is* weakly Pareto optimal *if there does not exist another* $\mathbf{x} \in S$ *such that* $f_i(\mathbf{x}) < f_i(\mathbf{x}')$ *for all* $i = 1, \dots, k$*. An objective vector is (weakly) Pareto optimal if the corresponding decision vector is (weakly) Pareto optimal.*

Under the assumptions mentioned in the problem formulation, we know that Pareto optimal solutions exist (see [34], Corollary 3.2.1). The set of Pareto optimal solutions is a subset of weakly Pareto optimal solutions. Pareto optimality corresponds to the intuitive idea of a compromise. However, we deal with both the definitions because weakly Pareto optimal solutions are sometimes computationally more convenient to produce than Pareto optimal solutions.

Pareto optimal solutions can be divided into properly and improperly Pareto ones. In properly Pareto optimal solutions, infinite or large enough trade-offs are not allowed between the solutions. Proper Pareto optimality can be defined in many different ways (see [17]). Here we employ the definitions of Geoffrion [7] and Wierzbicki [42]. The latter may be called ε-proper Pareto optimality. (Note that in that case trade-offs are limited between ε and $1/\varepsilon$.)

Definition 2 *A decision vector* $\mathbf{x}' \in S$ *is* properly Pareto optimal (in the sense of Geoffrion) *if it is Pareto optimal and if there is some real number* $M > 0$ *such that for each* f_i *and each* $\mathbf{x} \in S$ *satisfying* $f_i(\mathbf{x}) < f_i(\mathbf{x}')$*, there exists at least one* f_j *such that* $f_j(\mathbf{x}') < f_j(\mathbf{x})$ *and*

$$\frac{f_i(\mathbf{x}') - f_i(\mathbf{x})}{f_j(\mathbf{x}) - f_j(\mathbf{x}')} \leq M.$$

A decision vector $\mathbf{x}' \in S$ *and the corresponding objective vector* $\mathbf{z}' \in Z$ *are* ε-properly Pareto optimal (in the sense of Wierzbicki) *if*

$$(\mathbf{z}' - \mathbf{R}^k_\varepsilon \setminus \{\mathbf{0}\}) \cap Z = \emptyset,$$

where $\mathbf{R}^k_\varepsilon = \{\mathbf{z} \in \mathbf{R}^k \mid \max_{i=1,\dots,k} z_i + \varepsilon \sum_{i=1}^k z_i \geq 0\}$ *and* $\varepsilon > 0$ *is a predetermined scalar.*

Note that Definitions 1 and 2 introduce global optimality. Computationally, it may, however, be difficult to generate globally optimal solutions if the problem is not convex (that is, if some the objective functions or the feasible region are not convex). In that case, the solutions obtained may be only locally optimal unless a global solver is available.

The ranges of the Pareto optimal solutions provide valuable information if the objective functions are bounded over the feasible region. The components $z_i^\star$ of the *ideal objective vector* $\mathbf{z}^\star \in \mathbf{R}^k$ are obtained by minimizing each of the objective functions individually subject to the feasible region. Lower bounds of the Pareto optimal set are, thus, available in $\mathbf{z}^\star$. Sometimes, a vector strictly better than $\mathbf{z}^\star$ is required. This vector is called a *utopian objective vector* and denoted by $\mathbf{z}^{\star\star}$. In practice, the components of the utopian objective vector are calculated by subtracting some small positive scalar from the components of the ideal objective vector.

The upper bounds of the Pareto optimal set, that is, the components of a *nadir objective vector* $\mathbf{z}^{\mathrm{nad}}$, are usually difficult to obtain. Unfortunately, there exists no constructive way to obtain the exact nadir objective vector for nonlinear problems. They can, however, be estimated from a payoff table (see, e.g., [17]).

Weakly, properly or Pareto optimal solutions are usually generated by using real-valued *scalarizing functions* $s : \mathbf{R}^n \to \mathbf{R}$ possibly involving additional parameters. In other words, the problem with multiple objectives is transformed into one or a series of single objective optimization problems

$$\begin{array}{ll} \text{minimize} & s(\mathbf{f}(\mathbf{x})) \\ \text{subject to} & \mathbf{x} \in S. \end{array} \tag{2}$$

The scalarizing function s includes some kind of preference information from the decision maker who is supposed to have deeper insight into the problem and who is responsible for finding the most satisfactory compromise among the conflicting goals.

3 Scalarizing functions

In this section, we introduce several classification and reference point-based scalarizing functions from the literature as well as their modifications and more recent variations. In classification, the objective function values calculated at the current solution point $\mathbf{x}^c \in S$ are presented to the decision maker. (S)he can then express what kind of changes would be desirable to her/him by classifying the objective functions into different classes. The most commonly used classes are those for functions whose values are desired to be improved (i.e., decreased) from the current level, for functions whose values are acceptable at the moment and for functions whose values may be impaired (i.e., increased).

We denote the class of functions whose values should be improved by $I^<$. If the decision maker is able to provide additional information about the desired amount of improvement in the objective function value, such functions are placed in the class $I^\leq$ and the corresponding desired aspiration levels for functions f_i in $I^\leq$ are denoted by $\bar{z}_i$. If the current value of some objective function is acceptable, it is placed in the class $I^=$. Furthermore, the class for functions whose values are allowed to be increased is denoted by $I^>$. However, if the decision maker wants to specify upper bounds for such functions, the class is denoted by $I^\geq$ and the bounds by ε_i. In some connections, the decision maker does not have to classify all the objective functions. In this case, the rest of them are placed in $I^\diamond$. In this last-mentioned class, the objective function values can change freely and those functions are ignored in the scalarizing function.

In reference point-based methods, the decision maker specifies a reference point $\bar{\mathbf{z}}$ consisting of desirable aspiration levels $\bar{z}_i$ for each objective function f_i. This reference point does not have to depend on the current iteration point. It only indicates what kind of objective function values the decision maker prefers.

Note that usually the ideal objective vector (and possibly an approximation of the nadir objective vector) are calculated and presented to the decision maker before any classification or reference point information is requested. This gives the decision maker some perspective about the possibilities and limitations of the problem. Thus, we can assume that the decision maker does not specify hopes beyond the ideal values, in other words, $\bar{z}_i \geq z_i^\star$.

In the following, we introduce scalarizing functions used in different methods. However, we do not introduce the methods themselves. We illustrate the behaviour of some of the scalarizing functions with figures involving two objective functions. One should note that these simple illustrations cannot fully express the versatility of the functions.

3.1 STEM

The following scalarizing function is used in the Step method (STEM) [2]. The decision maker is supposed to classify the objective functions at the current solution point to two classes: $I^<$ and $I^\geq$. In addition, the decision maker must specify the upper bounds ε_i for f_i in $I^\geq$. The global ideal and nadir objective vectors must also be available.

The problem to be solved will be called *stem* and it is of the form

$$\begin{array}{ll}\text{minimize} & \max\limits_{i=1,\dots,k} \left[\dfrac{e_i}{\sum_{j=1}^k e_j} (f_i(\mathbf{x}) - z_i^\star) \right] \\ \text{subject to} & f_i(\mathbf{x}) \leq \varepsilon_i \quad \text{for all} \quad i \in I^\geq, \\ & f_i(\mathbf{x}) \leq f_i(\mathbf{x}^c) \quad \text{for all} \quad i \in I^<, \\ & \mathbf{x} \in S, \end{array} \tag{3}$$

where $e_i = 0$ for $i \in I^{\geq}$ and elsewhere $e_i = (z_i^{\mathrm{nad}} - z_i^{\star}) / \max\left[|z_i^{\mathrm{nad}}|, |z_i^{\star}|\right]$, as suggested in [38]. The solution obtained by (3) is guaranteed to be weakly Pareto optimal (see, e.g. [17]).

3.2 STOM

The following scalarizing functions are used in the satisficing trade-off method (STOM) [24,25]. The decision maker is assumed to specify a reference point $\bar{\mathbf{z}}$. Here the utopian objective vector must be known globally. However, if some objective function f_i is not bounded from below in S, then some small scalar value can be used as $z_i^{\star\star}$.

Different scalarizing functions can be used in STOM. One alternative is

$$s(\mathbf{f}(\mathbf{x})) = \max_{i=1,\dots,k}\left[\frac{f_i(\mathbf{x}) - z_i^{\star\star}}{\bar{z}_i - z_i^{\star\star}}\right], \tag{4}$$

where the reference point must be strictly worse than the utopian objective vector. We denote this function by *stom*.

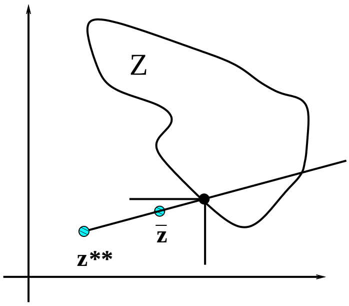

Fig. 1. Behaviour of stom

Figure 1 illustrates how the solution of stom is generated based on the utopian objective vector and the reference point. These two points are connected with a line and the negative orthant cone is shifted along the line until it touches Z. The solution is the intersection point.

It can be shown that the solution of (4) is weakly Pareto optimal and any Pareto optimal solution can be found by changing the reference point (see, e.g., [17]).

If weakly Pareto optimal solutions are to be avoided, the function to be used is *stom-aug*

$$s(\mathbf{f}(\mathbf{x})) = \max_{i=1,\dots,k}\left[\frac{f_i(\mathbf{x}) - z_i^{\star\star}}{\bar{z}_i - z_i^{\star\star}}\right] + \rho\sum_{i=1}^{k}\frac{f_i(\mathbf{x})}{\bar{z}_i - z_i^{\star\star}}, \tag{5}$$

where $\rho > 0$ is some sufficiently small scalar. With this function, every solution is properly Pareto optimal and any properly Pareto optimal solution can be found (see, e.g., [17]). The behaviour of stom-aug corresponds to that of stom with the exception in the shape of the cone. The augmentation coefficient defines the slope so that the cone becomes slightly blunt.

3.3 Achievement scalarizing functions

Achievement (scalarizing) functions have been introduced by Wierzbicki in [42–44], among others. They are used, for example, in the reference point method (see the references above).

The idea is that the decision maker specifies reasonable or desirable aspiration levels forming a reference point. So-called order-representing and order-approximating achievement functions can be defined. An example of an order-representing achievement function is *ach*

$$s(\mathbf{f}(\mathbf{x})) = \max_{i=1,\dots,k} [\, w_i(f_i(\mathbf{x}) - \bar{z}_i) \,], \tag{6}$$

where $\mathbf{w} = (w_1, \dots, w_k)^T$ is some fixed positive weighting vector. Here we set $w_i = 1/|z_i^\star|$ for each i. If $|z_i^\star| \le \delta$ for some small $\delta > 0$, then we set $w_i = 1$. Notice that if we set $w_i = 1/(\bar{z}_i - z_i^{\star\star})$, the resulting scalarizing function is equal to stom.

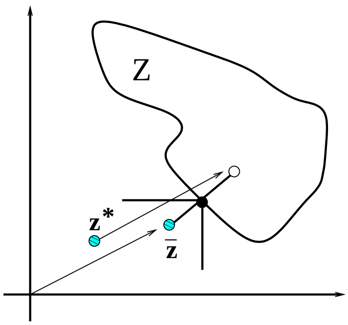

Fig. 2. Behaviour of ach

Figure 2 illustrates how the solution of ach is generated based on the ideal objective vector and the reference point. First, the origin of the coordinate system is transformed into the reference point. Then, (the vector consisting of the absolute values of) the ideal objective vector is located in the new coordinate system. The new point is connected to the reference point with a line and the negative orthant cone is shifted along the line until the cone touches Z. The solution is the intersection point.

An example of an order-approximating achievement function is *ach-aug*

$$s(\mathbf{f}(\mathbf{x})) = \max_{i=1,\dots,k} [\, w_i(f_i(\mathbf{x}) - \bar{z}_i) \,] + \rho \sum_{i=1}^{k} w_i(f_i(\mathbf{x}) - \bar{z}_i), \tag{7}$$

where $\mathbf{w}$ is as above and $\rho > 0$. (In [13], it is suggested to set $\rho = \varepsilon/k$ whereas [45] sets $\rho = \varepsilon$.)

The behaviour of ach-aug corresponds to that of stom-aug as far as the augmentation coefficient is concerned. We can prove that if the achievement function is order-representing, then its solution is weakly Pareto optimal. If the function is order-approximating, then its solution is Pareto optimal and the solution is ε-properly Pareto optimal if the function is also strongly increasing (see [17]). Any (weakly) Pareto optimal solution can be found if the achievement function is order-representing. Finally, any ε-properly Pareto optimal solution can be found if the function is order-approximating.

Other examples of achievement functions can be found, for example, in [8,13, 44,45] and references therein. More general formulations include so-called component achievement functions for each objective function. A way how the decision maker can adjust these component functions is suggested in [8]. Here we, however, consider only the simplest cases ach and ach-aug and leave the others as topics of further research.

3.4 GUESS

The GUESS method is also known as a *naïve method* [4,5]. The scalarizing function used requires global information about the nadir objective vector $\mathbf{z}^{\mathrm{nad}}$. The decision maker is assumed to specify a reference point (or a guess) $\bar{\mathbf{z}}$. The general idea is to maximize the minimum weighted deviation from the nadir objective vector.

The function to be minimized is *guess*

$$s(\mathbf{f}(\mathbf{x})) = \max_{i=1,\dots,k} \left[\frac{f_i(\mathbf{x}) - z_i^{\mathrm{nad}}}{z_i^{\mathrm{nad}} - \bar{z}_i} \right]. \tag{8}$$

Notice that the aspiration levels have to be strictly lower than the nadir objective vector.

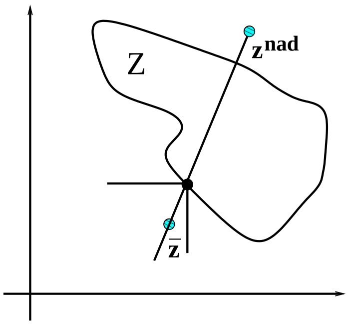

Fig. 3. Behaviour of guess

Figure 3 illustrates how the solution of guess is generated based on the nadir objective vector and the reference point. These two points are connected with a line and the negative orthant cone is shifted along the line until the cone touches Z. The solution is the intersection point. It can be shown that the solution of (8) is weakly Pareto optimal and any Pareto optimal solution can be found (see, e.g., [17]).

3.5 NIMBUS

The NIMBUS method (Nondifferentiable Interactive Multiobjective BUndle-based optimization System) is presented in [17–19]. In NIMBUS, a (global) ideal objective vector is assumed to be known. The decision maker is asked to classify the objective functions into up to five classes $I^{<}$, $I^{\leq}$, $I^{=}$, $I^{\geq}$ and $I^{\diamond}$. In addition to the classification, the decision maker is asked to specify the corresponding aspiration levels and upper bounds.

The decision maker can tune the order of importance inside the classes $I^{<}$ and $I^{\leq}$ with optional positive weighting coefficients w_i summing up to one. If the decision maker does not want to specify any weighting coefficients, they are set equal to one.

After the classification, a problem *nimbus-a*

$$
\begin{array}{ll}
\text{minimize} & \max\limits_{\substack{i\in I^{<}\\ j\in I^{\leq}}} \left[w_i(f_i(\mathbf{x}) - z_i^{\star}), w_j \max\left[f_j(\mathbf{x}) - \bar{z}_j,\ 0\right]\right] \\
\text{subject to} & f_i(\mathbf{x}) \leq f_i(\mathbf{x}^c) \ \text{ for all } \ i \in I^{<} \cup I^{\leq} \cup I^{=}, \\
 & f_i(\mathbf{x}) \leq \varepsilon_i \ \text{ for all } \ i \in I^{\geq}, \\
 & \mathbf{x} \in S
\end{array}
\tag{9}
$$

is solved.

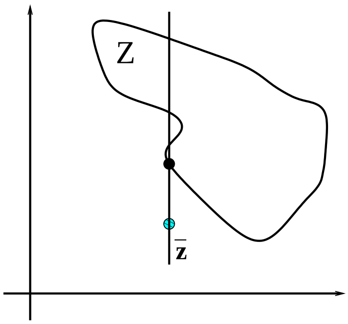

Fig. 4. Behaviour of nimbus-a

Figure 4 illustrates how the solution of nimbus-a is generated based on the aspiration level and the upper bound. The upper bound restricts the area where the point closest to the reference point is to be found. It is shown in [17] that the solution of (9) is weakly Pareto optimal if the set $I^{<}$ is nonempty and any Pareto optimal solution can be found.

Problem (9) can be formulated in an alternative form *nimbus-b*

$$
\begin{array}{ll}
\text{minimize} & \max\limits_{i\in I^{<}\cup I^{\leq}} \left[w_i \max\left[f_i(\mathbf{x}), \bar{z}_i\right]\right] \\
\text{subject to} & f_i(\mathbf{x}) \leq f_i(\mathbf{x}^c) \ \text{ for all } \ i \in I^{<} \cup I^{\leq} \cup I^{=}, \\
 & f_i(\mathbf{x}) \leq \varepsilon_i \ \text{ for all } \ i \in I^{\geq}, \\
 & \mathbf{x} \in S,
\end{array}
\tag{10}
$$

where $\bar{z}_i = z_i^\star$ for $i \in I^<$. The optimality results correspond to those of (9).

It is possible to modify (9) so that no weighting coefficients are asked from the decision maker. Instead, information about the best values of each objective function is taken into account. We call this variant by the name *nimbus-c*

$$\begin{array}{ll} \text{minimize} & \displaystyle\max_{\substack{i\in I^< \\ j\in I^\leq}} \left[\frac{1}{|z_i^\star|}(f_i(\mathbf{x}) - z_i^\star), \frac{1}{|z_j^\star|} \max\left[f_j(\mathbf{x}) - \bar{z}_j,\ 0\right] \right] \\ \text{subject to} & f_i(\mathbf{x}) \leq f_i(\mathbf{x}^c) \ \text{ for all } \ i \in I^< \cup I^\leq \cup I^=, \\ & f_i(\mathbf{x}) \leq \varepsilon_i \ \text{ for all } \ i \in I^\geq, \\ & \mathbf{x} \in S. \end{array} \tag{11}$$

Note that if $|z_i^\star| \leq \delta$ with some small $\delta > 0$, we replace $z_i^\star$ by 1 in the denominator.

3.6 Lexicographic formula

The augmented scalarizing functions stom-aug and ach-aug guarantee that weakly Pareto optimal solutions are not generated. Their common drawback is how to select the augmentation coefficient ρ. An alternative way of avoiding weakly Pareto optimal solutions is to use a lexicographic approach in the spirit of Tchebycheff method of [37] (see also [17]).

The following *stom-lex* problem in two phases is a lexicographic variant of (5). The first problem to be solved is

$$\begin{array}{ll} \text{minimize} & \displaystyle\max_{i=1,\dots,k} \left[\frac{f_i(\mathbf{x}) - z_i^{\star\star}}{\bar{z}_i - z_i^{\star\star}} \right] \\ \text{subject to} & \mathbf{x} \in S. \end{array} \tag{12}$$

Let us denote the optimal objective function value of (12) by y^*. The final solution is obtained by solving the problem

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{k} \frac{f_i(\mathbf{x})}{\bar{z}_i - z_i^{\star\star}} \\ \text{subject to} & \displaystyle\max_{i=1,\dots,k} \left[\frac{f_i(\mathbf{x}) - z_i^{\star\star}}{\bar{z}_i - z_i^{\star\star}} \right] \leq y^*, \\ & \mathbf{x} \in S. \end{array} \tag{13}$$

The lexicographic variant of (7) is to be called *ach-lex*. First we solve the problem

$$\begin{array}{ll} \text{minimize} & \displaystyle\max_{i=1,\dots,k} \left[w_i(f_i(\mathbf{x}) - \bar{z}_i) \right] \\ \text{subject to} & \mathbf{x} \in S, \end{array} \tag{14}$$

where $w_i = 1/|z_i^\star|$ for each i. If $|z_i^\star| \leq \delta$, then we set $w_i = 1$. Let us denote the optimal objective function value of (14) by y^*. Next we solve the problem

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{k} w_i(f_i(\mathbf{x}) - \bar{z}_i) \\ \text{subject to} & \displaystyle\max_{i=1,\dots,k} \left[w_i(f_i(\mathbf{x}) - \bar{z}_i) \right] \leq y^*, \\ & \mathbf{x} \in S. \end{array} \tag{15}$$

The solutions of (13) and (15) can be shown to be Pareto optimal. Furthermore, any Pareto optimal solution can be found. The proofs follow the ideas given in [17].

3.7 Four variants

Recently, several different scalarizing functions have been proposed in connection with integer-valued multiobjective linear optimization (see [39]). They can, however, also be used in more general problems. None of them requires additional information in the form of ideal or nadir objective vectors. The first variant *var-a* involves the classes $I^{\leq}$ (with related aspiration levels), $I^{=}$ and $I^{>}$ and the problem is of the form

$$\begin{aligned} \text{minimize} \quad & \max\left[\max_{i\in I^{\leq}}\left[\frac{f_i(\mathbf{x})-\bar{z}_i}{|f_i(\mathbf{x}^c)|}\right], \max_{j\in I^{>}}\left[\frac{f_j(\mathbf{x})-f_j(\mathbf{x}^c)}{|f_j(\mathbf{x}^c)|}\right]\right] \\ \text{subject to} \quad & f_i(\mathbf{x}) \leq f_i(\mathbf{x}^c) \ \text{ for all } \ i\in I^{\leq}\cup I^{=}, \\ & \mathbf{x}\in S. \end{aligned} \tag{16}$$

If the denominators are smaller than some positive δ, we replace them by 1. It is easy to show that the solutions obtained are weakly Pareto optimal and any Pareto optimal solution can be found.

The second variant *var-b* avoids weakly Pareto optimal solutions with the help of an augmentation term. Thus, we have

$$\begin{aligned} \text{minimize} \quad & \max\left[\max_{i\in I^{\leq}}\left[\frac{f_i(\mathbf{x})-\bar{z}_i}{|f_i(\mathbf{x}^c)|}\right], \max_{j\in I^{>}}\left[\frac{f_j(\mathbf{x})-f_j(\mathbf{x}^c)}{|f_j(\mathbf{x}^c)|}\right]\right] + \\ & \rho\left[\sum_{i\in I^{\leq}}(f_i(\mathbf{x})-\bar{z}_i) + \sum_{j\in I^{>}}\big(f_j(\mathbf{x})-f_j(\mathbf{x}^c)\big)\right] \\ \text{subject to} \quad & f_i(\mathbf{x}) \leq f_i(\mathbf{x}^c) \ \text{ for all } \ i\in I^{\leq}\cup I^{=}, \\ & \mathbf{x}\in S. \end{aligned} \tag{17}$$

Corresponding to the earlier results involving functions with augmentation terms, we can say that the solutions of (17) are properly Pareto optimal and any properly Pareto optimal solution can be found.

The third variant *var-c* uses the classes $I^{<}$, $I^{=}$ and $I^{>}$. The idea is not to require any aspiration level or upper bound information from the decision maker. We have the problem

$$\begin{aligned} \text{minimize} \quad & \max_{i\in I^{<}}\left[\frac{f_i(\mathbf{x})-f_i(\mathbf{x}^c)}{|f_i(\mathbf{x}^c)|}\right] + \max_{j\in I^{>}}\left[\frac{f_j(\mathbf{x})-f_j(\mathbf{x}^c)}{|f_j(\mathbf{x}^c)|}\right] \\ \text{subject to} \quad & f_i(\mathbf{x}) \leq f_i(\mathbf{x}^c) \ \text{ for all } \ i\in I^{<}\cup I^{=}, \\ & \mathbf{x}\in S. \end{aligned} \tag{18}$$

The treatment with denominators corresponds to that in (16). It can be shown that the solution of (18) is weakly Pareto optimal.

The fourth variant *var-d* uses the classes $I^<$, $I^{\leq}$, $I^=$ and $I^>$. The decision maker is supposed to give aspiration levels to the functions in $I^{\leq}$. The problem is of the form

$$\begin{aligned} \text{minimize} \quad & \max\left[\max_{i\in I^{\leq}}\left[\frac{f_i(\mathbf{x})-\bar{z}_i}{|f_i(\mathbf{x}^c)|}\right], \max_{j\in I^>}\left[\frac{f_j(\mathbf{x})-f_j(\mathbf{x}^c)}{|f_j(\mathbf{x}^c)|}\right]\right] + \\ & \max_{i\in I^<}\left[\frac{f_i(\mathbf{x})-f_i(\mathbf{x}^c)}{|f_i(\mathbf{x}^c)|}\right] \\ \text{subject to} \quad & f_i(\mathbf{x}) \leq f_i(\mathbf{x}^c) \ \text{ for all } \ i \in I^< \cup I^{\leq} \cup I^=, \\ & \mathbf{x} \in S. \end{aligned} \tag{19}$$

The treatment with denominators corresponds to that in (16). It can be shown that the solution of (19) is weakly Pareto optimal.

3.8 Properties of the scalarizing functions

All the scalarizing functions introduced are nondifferentiable because of the max-term. Note that they can be expressed in a differentiable form with the help of an additional variable (see, e.g. [17,36]). In this way, it is possible to solve them with a differentiable optimizer assuming the objective and the possible constraint functions involved are differentiable.

We collect some basic properties of the scalarizing functions introduced in Table 1. The column I indicates classification-based functions while the column $\bar{\mathbf{z}}$ represents reference point-based functions. The next three columns show whether the function utilizes the ideal, the utopian or the nadir objective vector, respectively. The next column ρ indicates whether the function involves an augmentation term. The column con. differentiates between the functions that use additional constraints or involve the original constraints only. The last three columns express the type of the optimality of the solution: weak Pareto optimality, Pareto optimality and proper Pareto optimality, respectively.

Let us next illustrate the functioning of the scalarizing functions introduced with a simple two-dimensional example

$$\begin{aligned} \text{minimize} \quad & \{-x_1^2 + 3x_2^2 + 2, \\ & 2(x_1-1)^2 + (x_2-1.5)^2 + 1\} \\ \text{subject to} \quad & -1 \leq x_1, x_2 \leq 2. \end{aligned} \tag{20}$$

We use a starting point $\mathbf{x} = (1.7394, 0.4225)$. At that point we have $f_1(\mathbf{x}) = -0.48999361$ and $f_2(\mathbf{x}) = 3.25443097$. This vector is denoted by $\mathbf{z}^c$ in Figure 5. Now we have the ideal objective vector $\mathbf{z}^\star = \mathbf{z}^{\text{icv}} = (-2, 1)$ and the nadir objective vector $\mathbf{z}^{\text{nad}} = (7.75, 5.251)$.

Our aim is to illustrate the variety of the different solutions that the scalarizing functions produce for problem (20). We select the reference point as $(1.0, 1.5)$. It corresponds to the classification $I^{\leq} = \{2\}$ with $z_2 = 1.5$ and $I^{\geq} = \{1\}$ with $\varepsilon_1 = 1.0$. With these selections, the preference information used in different scalarizing function is the same and, thus, the results obtained are comparable.

Table 1. Basic properties of scalarizing functions

Function	I	$\bar{\mathbf{z}}$	$\mathbf{z}^{\star}$	$\mathbf{z}^{\star\star}$	$\mathbf{z}^{\mathrm{nad}}$	ρ	con.	weak P	P	prop P
stem	X		X		X		X	X		
stom		X		X				X		
stom-aug		X		X		X				X
ach		X	X					X		
ach-aug		X	X			X				X
guess		X			X			X		
nimbus-a	X		X				X	X		
nimbus-b	X		X				X	X		
nimbus-c	X		X				X	X		
stom-lex		X		X			X		X	
ach-lex		X	X				X		X	
var-a	X						X	X		
var-b	X					X	X			X
var-c	X						X	X		
var-d	X						X	X		

Table 2. Solutions obtained with different functions

Function	$\mathbf{f}(\mathbf{x})$
stem	(0.996905, 2.194451)
stom	(2.118540, 1.686734)
stom-aug	(2.118540, 1.686734)
ach	(1.702306, 1.851908)
ach-aug	(1.699851, 1.851897)
guess	(1.659221, 1.866234)
nimbus-a	(0.999877, 2.192758)
nimbus-b	(0.999877, 2.192758)
nimbus-c	(0.999877, 2.192758)
stom-lex	(2.118540, 1.686734)
ach-lex	(1.694637, 1.851091)
var-a	(-0.255207, 3.059398)
var-b	(-0.255207, 3.059398)
var-c	(-0.489994, 3.254431)
var-d	(-0.255207, 3.059398)

In Table 2, we collect the solutions obtained using different scalarizing functions. The differences and the similarities of the solutions obtained can easily be seen in Figure 5. The theoretical behaviour of the scalarizing functions described earlier can be verified in the figure. Note that even though some functions happened to produce similar solutions, this kind of a phenomenon cannot be generalized. How-

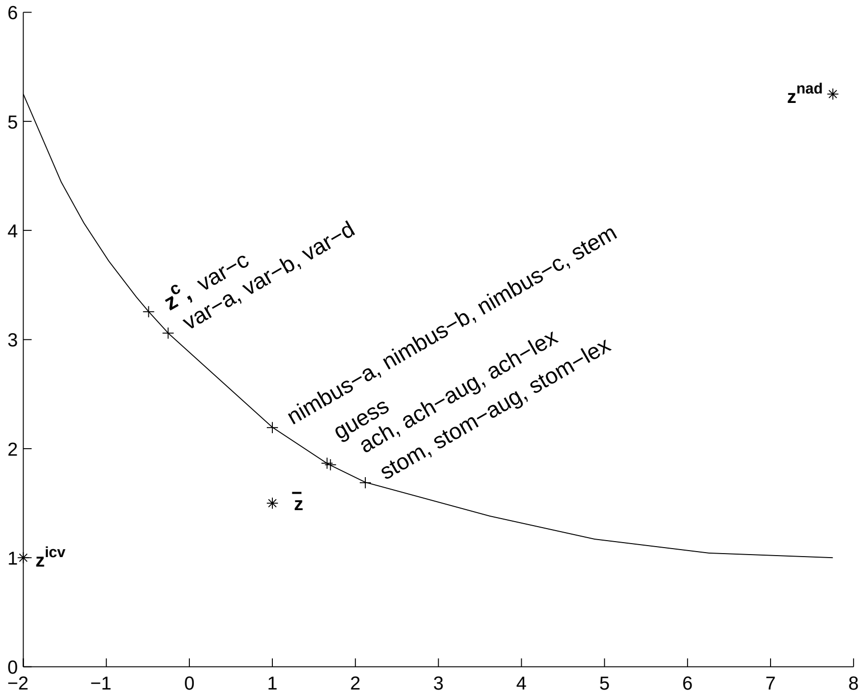

Fig. 5. Illustration of solutions obtained

ever, the figure gives a general overview of the location of the solutions obtained with respect to the reference point.

4 Results and analysis

Next, we compare the performances of the fifteen scalarizing functions introduced with the help of a set of test problems. The aim is to compare both the satisfaction of the decision maker and the computational complexity.

Twenty test problems mainly from the literature were selected and used in the numerical experiments. A collection of some of their properties is given in Table 3. In addition to the reference in the literature, the number of objective functions k, variables n and constraint functions m, we indicate whether the problem is linear, nonlinear but differentiable or nondifferentiable. Some of the problems were slightly modified. This is indicated by the word modif. in the table. Let us mention that slightly more than a half of the problems were convex. More details of the test problems can be found in [21].

The scalarizing functions were numerically compared on an HP9000/ J280, 180 MHz computer. The single objective optimizer used was the proximal bundle method PB (see [14]), where the computational accuracy was 10^{-12}. In the calculations, we used $z_i^{\star\star} = z_i^\star - 0.001$ for $i = 1, \ldots, k$. Furthermore, we set $\delta = 10^{-2}$ and $\rho = 10^{-3}$.

Table 3. Problems in brief

Number	Reference	k	n	m	Type
1	[10] p. 214	3	4	3	linear
2	[11], modif.	3	3	4	linear
3	[18]	7	2	0	nondiff.
4	[18], modif.	4	2	0	nondiff.
5	[18, 27], modif.	5	2	0	nondiff.
6	[10] p. 111	3	6	6	linear
7	[30] pp. 93–98	5	5	2	nonlin.
8	[21]	4	3	0	nonlin.
9	[33]	3	3	1	nonlin.
10	[31]	2	6	6	nonlin.
11	[6]	3	2	1	nonlin.
12	[21]	5	2	0	nondiff.
13	[29]	2	2	1	nonlin.
14	[40]	2	30	0	nonlin.
15	[29]	2	2	3	nonlin.
16	[10] pp. 136–138, modif.	3	3	4	nonlin.
17	[19], modif.	6	3	1	nondiff.
18	[26], modif.	3	2	3	nonlin.
19	[9]	2	2	1	nonlin.
20	[22]	5	325	0	nondiff.

The test problems were solved with each of the scalarizing functions (for one iteration). Their performances were evaluated based on three criteria with the scale 1–10 (with 10 being the best). First, it was evaluated, how the input format of the scalarizing function corresponded to the desires of the decision maker and information available. In other words, this criterion measured whether the classes available or the reference point ideology could reflect and capture the preferences of the decision maker. This means that the rating is the same for all reference point-based functions and those classification-based functions utilizing the same classes. A rating was given in relation to each test problem and the mean values and the standard deviations of these ratings are presented in the column input/desires in Table 4.

The column solutions/input in Table 4 represents ratings for the question how well the solution obtained matched up to the input specified. In other words, this rating measures how well the solution reflects the preference information specified. Thus, the rating is independent of the original desires and information available and it is restricted by the input format used in each scalarizing function.

Finally, the most important thing was to rate how well the solution obtained could meet the hopes and the desires of the decision maker. In other words, the third rating measures the satisfaction of the decision maker with the final solution and the tools of transmitting the preference information are ignored. The mean values and

Table 4. Numerical results

Function	Input/desires		Solutions/input		Solutions/desires		f.eval.
	mean	st.dev.	mean	st.dev.	mean	st.dev.	mean
stem	6.90	1.17	6.40	2.16	6.00	2.03	61.15
stom	7.80	0.62	5.25	2.67	5.45	2.76	58.35
stom-aug	7.80	0.62	5.35	2.62	5.55	2.68	65.35
ach	7.80	0.62	6.70	2.64	7.05	2.74	57.55
ach-aug	7.80	0.62	**6.85**	2.21	**7.20**	2.21	66.40
guess	7.80	0.62	6.50	2.74	6.60	2.50	75.20
nimbus-a	**8.05**	0.22	6.45	2.28	6.00	2.15	52.30
nimbus-b	**8.05**	0.22	5.85	1.98	5.50	1.88	46.00
nimbus-c	**8.05**	0.22	6.80	2.24	6.40	2.14	52.90
stom-lex	7.80	0.62	5.65	2.39	5.85	2.52	109.90
ach-lex	7.80	0.62	6.75	2.61	**7.20**	2.75	105.20
var-a	6.80	0.89	6.40	2.46	5.60	2.23	47.50
var-b	6.80	0.89	6.15	2.68	5.50	2.35	64.75
var-c	4.20	1.11	3.85	3.22	2.95	2.37	**39.75**
var-d	6.90	0.91	6.00	2.68	5.10	2.40	48.30

the standard deviations of these ratings are given in the column solutions/desires in Table 4.

The objective function evaluations were also counted. Their mean values (i.e., f.eval.) are also given in Table 4. Note that the possible calculation of the ideal objective vector is not included in the figures because this information is usually calculated independently of the scalarizing function used.

In Table 4, we emphasize the best mean values for each column with bold face. We present information about standard deviations so that the reader could have some kind of an impression about the diversity of the ratings of each scalarizing function in different test problems.

Because of the setting selected, the methods themselves were not compared, different decision makers were not involved and only one iteration was taken. This means that the scores of the different functions connected to each individual test problem are comparable.

4.1 General observations

One should note that it is impossible to draw any absolute conclusions of the tests and Table 4 because of the limited number of test problems and decision makers. However, some trendsetting inferences can be formulated. In general, avoiding weakly Pareto optimal results necessitates more computational effort. In other words, the augmented and the lexicographic variants of functions required more computation but produced more satisfactory results. Surprisingly, the computational burden with other scalarizing functions involving additional constraints did not increase. On the other hand, classification-based functions needed less calculation than reference point-based functions.

As far as the decision maker-dependent evaluation criteria (that is, input/desires, solutions/input and solutions/desires) are concerned, the methods utilizing information about the ranges in the Pareto optimal set performed relatively well, on the average. However, one should note that generating the ideal objective vector requires additional calculation. Especially, if the problem is nonconvex, a global optimizer is needed. Otherwise, insufficient local information may weaken the performance of the scalarizing function. In spite of the additional computational burden, the generation of the ideal objective vector is usually worthwhile in any practical problem even if the scalarizing function does not explicitly require this information.

The achievement scalarizing functions ach-aug, ach-lex and ach worked quite well, although ach-lex required a lot of computation. They had good scores in with respect to the criteria solutions/input and solutions/desires, which can be regarded to represent important criteria. The performance of these three functions is not so bad with respect to input/desires, either. The same comments are also valid for guess. On the other hand, guess required quite a lot of computational effort, on the average. One more thing decreasing the applicability of the method is its high dependence on the availability of the global nadir objective vector.

The differences in the performances of the members of the nimbus family are rather evident. Nimbus-c and nimbus-a produced more satisfactory solutions but required more computation than nimbus-b. In general, we can say that nimbus-c was the best of the three. As far as stem is concerned, it performed relatively well in all the four criteria.

The differences between the members of the stom family and the four variants are not too clear. However, the stom family can be ordered as stom-lex, stom-aug and stom and the four variants as var-a, var-b, var-d and var-c. As far as computational efficiency is concerned, stom-lex required a lot of function evaluations. This can be explained by the lexicographic nature of the function. The opposite in function evaluations is var-c. The price of the low computational demand is a very poor performance in all the three decision maker-dependent criteria. In most cases, var-c was able to move only a little from the starting point.

If we want to draw some general suggestive conclusions from Table 4, we can rank the ’best’ scalarizing functions according to the decision maker-dependent criteria in the order: ach-aug, ach-lex, nimbus-c, ach, guess, nimbus-a and stem. As scalarizing functions with moderate performance, we can list var-a, stom-lex, nimbus-b, stom-aug, var-b, stom and var-d. Clearly, the weakest of all was var-c. As far as computational efficiency is concerned, we can distinguish four categories. The most efficient ones were var-c, nimbus-b, var-a, var-d, nimbus-a and nimbus-c. The next category with moderate efficiency consists of ach, stom, stem, var-b, stom-aug and ach-aug. Guess was rather inefficient computationally but not as bad as ach-lex and stom-lex.

To conclude, for example, nimbus-c and ach can be regarded as some kind of compromises between computational cost and functionality of the scalarizing function when decision maker's preferences are concerned. However, they share the weakness of producing weakly Pareto optimal solutions.

4.2 Relations to WWW-NIMBUS

One of the goals of the experiments was to find scalarizing functions that are able to produce solutions that follow the preference information of the decision maker as closely as possible. The aim was to select some scalarizing functions that satisfy the above-mentioned requirement and still generate somewhat different solutions. These scalarizing functions are then to be included as a part of the interactive WWW-NIMBUS optimization system [20] operating on the Internet. The idea is to provide different but satisfactory solutions to the decision maker based on the same classification information. Then, the decision maker can select the solution that (s)he wants to continue with.

Based on the tests carried out, nimbus-a, that was earlier used in the WWW-NIMBUS system, was decided to be replaced by nimbus-c. In addition, ach-aug, guess, stem and stom-lex were selected as candidates to be included in the system. The functions within ach and stom families performed in similar ways and, thus, only one representative was selected from each of them.

In the previous step, such scalarizing functions were selected that produced somewhat satisfactory solutions. The next step was to drop those functions that produced too similar solutions. For this purpose, the solutions of the test problems with each of the five functions were gone through. Possible unsatisfactory solutions were ignored and the L_2-distance between the solutions in the objective space were calculated. Because the aim was to include new functions into the WWW-NIMBUS system, nimbus-c was selected as a point of reference when calculating the distances.

In our test problems, stem and nimbus-c produced mainly similar results. That is why it is not reasonable to include stem in WWW-NIMBUS. On the other hand, the other functions ach-aug, guess and stom-lex managed to generate satisfactory but clearly different solutions when compared to nimbus-c in roughly a half of the problems. Note that for some problems stom-lex was unable to move from the starting point. To conclude, ach-aug, guess and stom-lex can be considered to be included in WWW-NIMBUS. They seem to follow the preference information of the decision maker in a satisfactory but still different way. That is why it is best to leave the final selection between them up to the decision maker.

5 Conclusions

We have presented fifteen scalarizing functions used in various methods in multiobjective optimization. We have collected their theoretical properties and behaviour and compared them in twenty numerical test problems (mainly from the literature). We have taken only one iteration with each scalarizing function so that the results obtained could be compared. Thus, we have examined scalarizing functions, not methods. The aim has been to study the differences of the solutions obtained with the same or very similar preference information. On the other hand, analysis concerning the effects of changing preference information can be found in [3].

In the numerical tests, the functions have been compared according to four criteria. The first three criteria measured the functionality of the scalarizing function with respect to the decision maker's preferences whereas the last criterion described computational complexity. To be more precise, the evaluation criteria used were: how well the input format in question corresponded to the desires of the decision maker, how well the solution obtained matched up to the input specified, how well the solution obtained could meet the hopes of the decision maker and how many objective function evaluations were needed.

If the problem is to select a scalarizing function to be used in an interactive method, the choice depends on which of these factors are considered to be most important. A realistic option is to use more than one scalarizing function in order to generate different solutions, as was decided to do with WWW-NIMBUS. In this case, the decision maker can make the final choice.

One of the goals was to augment the WWW-NIMBUS system with additional scalarizing functions. Thus, the preference information once expressed by the decision maker could be used in order to generate several different solutions still satisfying the preferences. Based on our studies, the scalarizing function in the WWW-NIMBUS system was decided to be replaced by another form. In addition, three other scalarizing functions were selected to be included in the system. In this way, the system can provide a more versatile view of the behaviour of the problem considered.

References

1. Aksoy Y, Butler TW, Minor III ED (1996) Comparative studies in interactive multiple objective mathematical programming. European Journal of Operational Research 89: 408–422
2. Benayoun R, de Montgolfier J, Tergny J, Laritchev O (1971) Linear programming with multiple objective functions: Step method (STEM). Mathematical Programming 1: 366–375
3. Buchanan J, Gardiner L (2001) A comparison of two reference point methods in multiple objective mathematical programming. Research Report Series 2001-03, The University of Waikato
4. Buchanan JT (1997) A naïve approach for solving MCDM problems: The GUESS method. Journal of the Operational Research Society 48: 202–206
5. Buchanan JT, Daellenbach HG (1987) A comparative evaluation of interactive solution methods for multiple objective decision models. European Journal of Operational Research 29: 353–359
6. Caballero R, Rey L, Ruiz F (1996) Determination of satisfying and efficient solutions in convex multi-objective programming. Optimization 37: 125–137
7. Geoffrion AM (1968) Proper efficiency and the theory of vector maximization. Journal of Mathematical Analysis and Applications 22: 618–630
8. Granat J, Makowski M (2000) Interactive specification and analysis of aspiration-based preferences. European Journal of Operational Research 122: 469–485
9. Gulati TR, Islam MA (1990) Efficiency and proper efficiency in nonlinear vector maximum problems. European Journal of Operational Research 44: 373–382

10. Hwang C-L, Masud ASM (1979) Multiple objective decision making – methods and applications: A state-of-the-art survey. Springer, Berlin Heidelberg New York
11. Korhonen P, Laakso J (1986) Solving generalized goal programming problems using a visual interactive approach. European Journal of Operational Research 26: 355–363
12. Larichev OI, Polyakov OA, Nikiforov AO (1987) Multicriterion linear programming problems. Journal of Economic Psychology 8: 389–407
13. Lewandowski A, Kreglewski T, Rogowski T, Wierzbicki AP (1989) Decision support systems of DIDAS family (dynamic interactive decision analysis & support). In: Lewandowski A, Wierzbicki AP (eds) Aspiration based decision support systems: Theory, software and applications, pp 21–47. Springer, Berlin Heidelberg New York
14. Mäkelä MM, Neittaanmäki P (1992) Nonsmooth optimization: Analysis and algorithms with applications to optimal control. World Scientific Publishing, Singapore
15. Masud ASM, Zheng X (1989) An algorithm for multiple-objective non-linear programming. Journal of the Operational Research Society 40: 895–906
16. Michalowski W, Szapiro T (1992) A bi-reference procedure for interactive multiple criteria programming. Operations Research 40: 247–258
17. Miettinen K (1999) Nonlinear multiobjective optimization. Kluwer, Boston
18. Miettinen K, Mäkelä MM (1995) Interactive bundle-based method for nondifferentiable multiobjective optimization: NIMBUS. Optimization 34: 231–246
19. Miettinen K, Mäkelä MM (1999) Comparative evaluation of some interactive reference point-based methods for multi-objective optimisation. Journal of the Operational Research Society 50: 949–959
20. Miettinen K, Mäkelä MM (2000) Interactive multiobjective optimization system WWW-NIMBUS on the Internet. Computers & Operations Research 27: 709–723
21. Miettinen K, Mäkelä MM (2001) Classification and reference point based scalarizing functions in multiobjective optimization. Reports of the Department of Mathematical Information Technology, Series B. Scientific Computing B8/2001, University of Jyväskylä
22. Miettinen K, Mäkelä MM, Männikkö T (1998) Optimal control of continuous casting by nondifferentiable multiobjective optimization. Computational Optimization and Applications 11: 177–194
23. Mote J, Olson DL, Venkataramanan MA (1988) A comparative multiobjective programming study. Mathematical and Computer Modelling 10: 719–729
24. Nakayama H (1995) Aspiration level approach to interactive multi-objective programming and its applications. In: Pardalos PM, Siskos Y, Zopounidis C (eds) Advances in multicriteria analysis, pp 147–174. Kluwer, Dordrecht
25. Nakayama H, Sawaragi Y (1984) Satisficing trade-off method for multiobjective programming. In: Grauer M, Wierzbicki AP (eds) Interactive decision analysis, pp 113–122. Springer, Berlin Heidelberg New York
26. Narula SC, Kirilov L, Vassilev V (1994) An interactive algorithm for solving multiple objective nonlinear programming problems. In: Tzeng GH, Wand HF, Wen UP, Yu PL (eds) Multiple criteria decision making – Proceedings of the Tenth International Conference: Expand and Enrich the Domains of Thinking and Application, pp 119–127. Springer, Berlin Heidelberg New York
27. Narula SC, Weistroffer HR (1989) A flexible method for nonlinear multicriteria decisionmaking problems. IEEE Transactions on Systems, Man, and Cybernetics 19: 883–887
28. Olson DL (1992) Review of empirical studies in multiobjective mathematical programming: Subject reflection of nonlinear utility and learning. Decision Sciences 23: 1–20

29. Osyczka A (1984) Multicriterion Optimization in engineering with FORTRAN Programs. Ellis Horwood, Chichester
30. Osyczka A (1992) Computer aided multicriterion optimization system (CAMOS): software package in FORTRAN. International Software Publishers, Krakow
31. Osyczka A, Kundu S (1995) A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm. Structural Optimization 10: 94–99
32. Reeves GR, Gonzalez JJ (1989) A comparison of two interactive MCDM procedures. European Journal of Operational Research 41: 203–209
33. Sakawa M, Mori N (1984) Interactive multiobjective decisionmaking for nonconvex problems based on the penalty scalarizing functions. European Journal of Operational Research 17: 320–330
34. Sawaragi Y, Nakayama H, Tanino T (1985) Theory of multiobjective optimization. Academic Press, Orlando, FL
35. Shin WS, Ravindran A (1991) Interactive multiple objective optimization: Survey I – continuous case. Computers & Operations Research 18: 97–114
36. Steuer RE (1986) Multiple criteria optimization: Theory, computation, and applications. Wiley, New York
37. Steuer RE (1989) The Tchebycheff procedure of interactive multiple objective programming. In: Karpak B, Zionts S (eds) Multiple criteria decision making and risk analysis using microcomputers, pp 235–249. Springer, Berlin Heidelberg New York
38. Vanderpooten D, Vincke P (1989) Description and analysis of some representative interactive multicriteria procedures. Mathematical and Computer Modelling 12: 1221–1238
39. Vassileva M (2000) Scalarizing problems of multiobjective linear integer programming. Problems of Engineering Cybernetics and Robotics 50: 54–64
40. Vicini A, Quagliarella D (2001) A multiobjective mathematical test problem, 2001. Available on the Internet at http://ingenet.ulpgc.es/functional/ databases/ceani/
41. Wallenius J (1975) Comparative evaluation of some interactive approaches to multicriterion optimization. Management Science 21: 1387–1396
42. Wierzbicki AP (1980) The use of reference objectives in multiobjective optimization. In: Fandel G, Gal T (eds) Multiple criteria decision making theory and applications, pp 468–486. Springer, Berlin Heidelberg New York
43. Wierzbicki AP (1982) A mathematical basis for satisficing decision making. Mathematical Modelling 3: 391–405
44. Wierzbicki AP (1986) On the completeness and constructiveness of parametric characterizations to vector optimization problems. OR Spektrum 8: 73–87
45. Wierzbicki AP (1999) Reference point approaches. In: Gal T, Stewart TJ, Hanne T (eds) Multicriteria decision making: Advances in MCDM models, algorithms, theory, and applications, ch 9, pp 1–39. Kluwer, Boston