Some Methods for Nonlinear Multi-objective Optimization

Kaisa Miettinen[*]

University of Jyväskylä, Department of Mathematical Information Technology,
P.O. Box 35 (Agora), FIN-40351 Jyväskylä, Finland
miettine@mit.jyu.fi
WWW home page: http://www.mit.jyu.fi/miettine/engl.html

Abstract. A general overview of nonlinear multiobjective optimization methods is given. The basic features of several methods are introduced so that an appropriate method could be found for different purposes. The methods are classified according to the role of a decision maker in the solution process. The main emphasis is devoted to interactive methods where the decision maker progressively provides preference information so that the most satisfactory solution can be found.

1 Introduction

Multiple criteria decision making (MCDM) problems form an extensive field where the best possible compromise should be found by evaluating several conflicting objectives. There is a good reason to classify such problems on the basis of their different characteristics. Here we concentrate on problems involving continuous nonlinear functions with deterministic values. We present versatile methods for solving such problems.

The solution process usually requires the participation of a human decision maker (DM) who can give preference information related to conflicting goals. Here we assume that a single DM is involved.

Methods are divided into four classes according to the role of the DM. Either no DM takes part in the solution process or (s)he expresses preference relations before, after or during the process. The last-mentioned, interactive, methods form the most extensive class of methods.

Multiobjective optimization problems are usually solved by scalarization. *Scalarization* means that the problem is converted into one single or a family of single objective optimization problems. This new problem has a real-valued objective function that possibly depends on some parameters and it can be solved using single objective optimizers.

Further information about the methodology of deterministic multiobjective optimization can be found, e.g., in the monographs [6,12,14,20,27,30]. For a more detailed presentation of the methods treated here as well as other related topics we refer to [20] and references therein.

[*] This research was supported by the Academy of Finland, grant #65760.

E. Zitzler et al. (Eds.): EMO 2001, LNCS 1993, pp. 1–20, 2001.

2 Concepts and Background

A *multiobjective optimization problem* is of the form

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

involving k (≥ 2) conflicting *objective functions* $f_i : \mathbb{R}^n \to \mathbb{R}$ that we want to minimize simultaneously. The *decision (variable) vectors* $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$ belong to the (nonempty) *feasible region* $S \subset \mathbb{R}^n$. The feasible region is formed by *constraint functions* but we do not fix them here.

We denote the image of the feasible region by $Z \subset \mathbb{R}^k$ and call it a *feasible objective region*. The elements of Z are called *objective vectors* and they consist of *objective (function) values* $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))^T$. Note that if f_i is to be maximized, it is equivalent to minimize $-f_i$.

In what follows, a function is called *nondifferentiable* if it is locally Lipschitzian (and not necessarily continuously differentiable).

Definition 1. *When all the objective and the constraint functions are linear, the problem is called* linear *or an* MOLP *problem. If at least one of the functions is nonlinear, the problem is a* nonlinear multiobjective optimization problem. *Correspondingly, the problem is* nondifferentiable *is some of the functions is nondifferentiable and* convex *if all the objective functions and the feasible region are convex.*

Because of the contradiction and possible incommensurability of the objective functions, it is not possible to find a single solution that would optimize all the objectives simultaneously. In multiobjective optimization, vectors are regarded as optimal if their components cannot be improved without deterioration to at least one of the other components. This is usually called Pareto optimality.

Definition 2. *A decision vector $\boldsymbol{x}^* \in S$ is* Pareto optimal *if there does not exist another $\boldsymbol{x} \in S$ such that $f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{x}^*)$ for all $i = 1, \ldots, k$ and $f_j(\boldsymbol{x}) < f_j(\boldsymbol{x}^*)$ for at least one index j. An objective vector is Pareto optimal if the corresponding decision vector is Pareto optimal.*

There are usually a lot (infinite number) of Pareto optimal solutions and they form a set of Pareto optimal solutions or a *Pareto optimal set*. This set can be nonconvex and nonconnected.

Definition 2 introduces *global Pareto optimality*. Another important concept is local Pareto optimality defined in a small environment of the point considered. Naturally, any globally Pareto optimal solution is locally Pareto optimal. The converse is valid for convex problems. To be more specific, if the feasible region is convex and the objective functions are quasiconvex with at least one strictly quasiconvex function, then locally Pareto optimal solutions are also globally Pareto optimal.

Other related optimality concepts are weak and proper Pareto optimality. The properly Pareto optimal set is a subset of the Pareto optimal set which is a subset of the weakly Pareto optimal set.

A vector is *weakly Pareto optimal* if there does not exist any other feasible vector for which all the components are better. In other words, when compared to Definition 2, all the inequalities are strict. Weakly Pareto optimal solutions are often relevant from a technical point of view because they are sometimes easier to generate than Pareto optimal points.

Pareto optimal solutions can be divided into improperly and properly Pareto optimal ones according to whether unbounded trade-offs between objectives are allowed or not. Proper Pareto optimality can be defined in several ways (see, e.g., [20]). According to [9] a solution is properly Pareto optimal if there is at least one pair of objectives for which a finite decrement in one objective is possible only at the expense of some reasonable increment in the other objective.

Mathematically, all the Pareto optimal points are equally acceptable solutions of the multiobjective optimization problem. However, it is generally desirable to obtain one point as a solution. Selecting one out of the set of Pareto optimal solutions calls for a *decision maker (DM)*. (S)he is a person who has better insight into the problem and who can express preference relations between different solutions.

Finding a solution to (1) is called a *solution process*. It usually means the co-operation of the DM and an analyst. An *analyst* is a person or a computer program responsible for the mathematical side of the solution process. The analyst generates information for the DM to consider and the solution is selected according to the preferences of the DM.

By solving a multiobjective optimization problem we here mean finding a feasible decision vector such that it is Pareto optimal and satisfies the DM. Assuming such a solution exists, it is called a *final solution*.

The ranges of the Pareto optimal set provide valuable information for the solution process if the objective functions are bounded over the feasible region. The components $z_i^\star$ of the *ideal objective vector* $\boldsymbol{z}^\star \in \mathbb{R}^k$ are obtained by minimizing each of the objective functions individually subject to the constraints. The ideal objective vector is not feasible because of the conflict among the objectives. From the ideal objective vector we obtain the lower bounds of the Pareto optimal set. Note that in nonconvex problems we need a global optimizer for calculating the ideal objective vector.

The upper bounds of the Pareto optimal set, that is, the components of a *nadir objective vector* $\boldsymbol{z}^{\mathrm{nad}}$, are usually rather difficult to obtain. They can be estimated from a payoff table (see, e.g., [20]) but this is not a reliable way as can be seen, e.g., in [15,31].

For nonlinear problems, there is no constructive method for calculating the nadir objective vector. Nonetheless, the payoff table may be used as a rough estimate as long as its robustness is kept in mind. Because of the above-described difficulty of calculating the actual nadir objective vector, $\boldsymbol{z}^{\mathrm{nad}}$ is usually an approximation.

Sometimes we need a vector that is strictly better than every Pareto optimal solution. Such a vector is called a *utopian objective vector* $\boldsymbol{z}^{\star\star} \in \mathbb{R}^k$ and its components are formed by decreasing the components of $\boldsymbol{z}^\star$ by a positive scalar.

It is often assumed that the DM makes decisions on the basis of an underlying function. This function representing the preferences of the DM is called a *value function* $U : \mathbb{R}^k \to \mathbb{R}$ (see [14]). In many methods, the value (or utility) function is assumed to be known implicitly.

Value functions are important in the development of solution methods and as a theoretical background. Generally, the value function is assumed to be strongly decreasing. This means that the preference of the DM will increase if the value of an objective function decreases while all the other objective values remain unchanged (i.e., less is preferred to more). In this case, the maximum of U is Pareto optimal. Regardless of the existence of a value function, it is usually assumed that less is preferred to more by the DM.

Instead of as a maximum of the value function, a final solution can be understood as a satisficing one. *Satisficing decision making* means that the DM does not intend to maximize any value function but tries to achieve certain aspirations. A solution which satisfies all the aspirations of the DM is called a *satisficing solution*.

During solution processes, various kinds of information are solicited from the DM. *Aspiration levels* $\bar{z}_i$, $i = 1, \ldots, k$, are such desirable or acceptable levels in the objective function values that are of special interest and importance to the DM. The vector $\bar{\boldsymbol{z}} \in \mathbb{R}^k$ is called a *reference point*.

According to the definition of Pareto optimality, moving from one Pareto optimal solution to another necessitates trading off. This is one of the basic concepts in multiobjective optimization. A *trade-off* reflects the ratio of change in the values of the objective functions concerning the increment of one objective function that occurs when the value of some other objective function decreases. For details, see, e.g., [6,20].

It is said that two feasible solutions are situated on the same *indifference curve* if the DM finds them equally desirable. For any two Pareto optimal solutions on the same indifference curve there is a trade-off involving a certain increment in one objective function value that the DM can tolerate in exchange for a certain amount of decrement in some other objective function while the preferences of the two solutions remain the same. This is called the *marginal rate of substitution* $m_{ij}(\boldsymbol{x}^*)$ $(i, j = 1, \ldots, k,\ i \neq j)$.

To conclude this section, let us have a look at how the Pareto optimality of a feasible decision vector can be tested. This topic is investigated, e.g., in [1,20]. A decision vector $\boldsymbol{x}^* \in S$ is Pareto optimal if and only if the problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^k \varepsilon_i \\ \text{subject to} & f_i(\boldsymbol{x}) + \varepsilon_i = f_i(\boldsymbol{x}^*) \ \text{ for all } \ i = 1, \ldots, k, \\ & \varepsilon_i \geq 0 \ \text{ for all } \ i = 1, \ldots, k, \\ & \boldsymbol{x} \in S \end{array} \tag{2}$$

has an optimal objective function value of zero, where both $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{\varepsilon} \in \mathbb{R}^k_+$ are variables. On the other hand, if (2) has a finite nonzero optimal objective function value obtained at $\hat{\boldsymbol{x}}$, then $\hat{\boldsymbol{x}}$ is Pareto optimal. Note that the equalities in (2) can be replaced with inequalities.

3 Methods

Mathematically, the multiobjective optimization problem is considered to be solved when the Pareto optimal set is found. This is also known as *vector optimization*. However, this is not always enough. Instead, we want to obtain one final solution. This means that we must find a way to order the Pareto optimal solutions and here we need a DM and her/his preferences.

In what follows, we present several methods for finding a final solution. We cannot cover every existing method but we introduce several philosophies and ways of approaching the problem.

The methods can be classified in many ways. Here we apply the classification presented in [12] based on the participation of the DM in the solution process. The classes are *no-preference methods*, *a posteriori methods*, *a priori methods* and *interactive methods*. Note that no classification can be complete and overlapping and combinations of classes are possible.

In addition, we consider an alternative way of classification into ad hoc and non ad hoc methods. This division, suggested in [29], is based on the existence of an underlying value function. Even if one knew the DM's value function, one would not exactly know how to respond to the questions posed by an *ad hoc* algorithm. On the other hand, in *non ad hoc* methods the responses can be determined or at least confidently simulated with the help of a value function.

In no-preference methods, the opinions of the DM are not taken into consideration. Thus, the problem is solved using some relatively simple method and the solution is presented to the DM who may either accept or reject it. For details if this class see, e.g., [20]. Next we introduce examples of a posteriori, a priori and interactive methods.

4 A Posteriori Methods

A posteriori methods could also be called *methods for generating Pareto optimal solutions*. After the Pareto optimal set (or a part of it) has been generated, it is presented to the DM, who selects the most preferred solution. The inconveniences here are that the generation process is usually computationally expensive and sometimes in part, at least, difficult. On the other hand, it is hard for the DM to select from a large set of alternatives. An important question related to this is how to display the alternatives to the DM in an illustrative way.

If there are only two objective functions, the Pareto optimal set can be generated parametrically (see, e.g., [2,8]). The problem becomes more complicated with more objectives.

4.1 Weighting Method

In the weighting method (see, e.g. [8,35]), we solve the problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{k} w_i f_i(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{x} \in S \ , \end{array} \tag{3}$$

where $w_i \geq 0$ for all $i = 1, \ldots, k$ and $\sum_{i=1}^{k} w_i = 1$. The solution of (3) is weakly Pareto optimal and it is Pareto optimal if $w_i > 0$ for all $i = 1, \ldots, k$ or the solution is unique.

The weakness of the weighting method is that not all of the Pareto optimal solutions can be found unless the problem is convex. The conditions under which the whole Pareto optimal set can be generated by the weighting method with positive weights are presented in [5].

Systematic ways of perturbing the weights to obtain different Pareto optimal solutions are suggested, e.g., in [6] (pp. 234–236). In addition, an algorithm for generating different weights automatically for convex problems to produce an approximation of the Pareto optimal set is proposed in [4].

The method has several weaknesses. On the one hand, a small change in the weights may cause big changes in the objective vectors. On the other hand, dramatically different weights may produce nearly similar objective vectors. In addition, an evenly distributed set of weights does not necessarily produce an evenly distributed representation of the Pareto optimal set.

4.2 ε-Constraint Method

In the ε-constraint method, introduced in [11], one of the objective functions is optimized in the form

$$\begin{array}{lll} \text{minimize} & f_\ell(\boldsymbol{x}) & \\ \text{subject to} & f_j(\boldsymbol{x}) \leq \varepsilon_j & \text{for all} \ \ j = 1, \ldots, k, \ \ j \neq \ell, \\ & \boldsymbol{x} \in S \ , & \end{array} \tag{4}$$

where $\ell \in \{1, \ldots, k\}$ and ε_j are upper bounds for the objectives $j \neq \ell$.

The solution of (4) is weakly Pareto optimal. On the other hand, $\boldsymbol{x}^* \in S$ is Pareto optimal if and only if it solves (4) for every $\ell = 1, \ldots, k$, where $\varepsilon_j = f_j(\boldsymbol{x}^*)$ for $j = 1, \ldots, k$, $j \neq \ell$. In addition, the unique solution of (4) is Pareto optimal for any upper bounds. Thus, finding any Pareto optimal solution does not necessitate convexity.

In order to ensure Pareto optimality in this method, we have to either solve k different problems or obtain a unique solution. In general, uniqueness is not necessarily easy to verify. Systematic ways of perturbing the upper bounds to obtain different Pareto optimal solutions are suggested in [6] (pp. 283–295).

4.3 Method of Weighted Metrics

In the method of weighted metrics, the distance between some reference point and the feasible objective region is minimized. A common way is to use the ideal objective vector and L_p-metrics. We can produce different solutions by weighting the metrics. This method is also sometimes called *compromise programming*.

The solution obtained depends greatly on the value chosen for p. For $1 \leq p < \infty$ we have a problem

$$\begin{array}{ll} \text{minimize} & \left(\sum_{i=1}^{k} w_i \left| f_i(\boldsymbol{x}) - z_i^{\star} \right|^p \right)^{1/p} \\ \text{subject to} & \boldsymbol{x} \in S \ . \end{array} \tag{5}$$

The exponent $1/p$ can be dropped. For $p = \infty$ we have a *weighted Tchebycheff problem*

$$\begin{array}{ll} \text{minimize} & \max_{i=1,\dots,k} \left[w_i |f_i(\boldsymbol{x}) - z_i^{\star}| \right] \\ \text{subject to} & \boldsymbol{x} \in S \ . \end{array} \tag{6}$$

Notice that no absolute values are needed if we know the global ideal objective vector. The solution of (5) is Pareto optimal if either the solution is unique or all the weights are positive. Furthermore, the solution of (6) is weakly Pareto optimal for positive weights. Finally, (6) has at least one Pareto optimal solution.

Convexity of the problem is needed in order to guarantee that every Pareto optimal solution can be found by (5). On the other hand, any Pareto optimal solution can be found by (6) when $\boldsymbol{z}^{\star\star}$ is used as a reference point.

Weakly Pareto optimal solutions can be avoided in (6) by giving a slight slope to the contour of the metric (see [27]). The augmented problem to be solved is

$$\begin{array}{ll} \text{minimize} & \max_{i=1,\dots,k} \left[w_i |f_i(\boldsymbol{x}) - z_i^{\star\star}| \right] + \rho \sum_{i=1}^{k} |f_i(\boldsymbol{x}) - z_i^{\star\star}| \\ \text{subject to} & \boldsymbol{x} \in S \ , \end{array} \tag{7}$$

where ρ is a sufficiently small positive scalar. In this case, it may be impossible to find every Pareto optimal solution. Instead, (7) generates properly Pareto optimal solutions and any properly Pareto optimal solution can be found.

Let us mention that different connections between the weighting method, the ε-constraint method and the method of weighted metrics are presented in [17].

4.4 Achievement Scalarizing Function Approach

Scalarizing functions of a special type are called *achievement (scalarizing) functions*. They have been introduced by Wierzbicki, e.g., in [32,33]. These functions are of the form $s_{\bar{\boldsymbol{z}}} : Z \to \mathbb{R}$, where $\bar{\boldsymbol{z}} \in \mathbb{R}^k$ is an arbitrary reference point. Because we do not know Z explicitly, in practice we minimize the function $s_{\bar{\boldsymbol{z}}}(\boldsymbol{f}(\boldsymbol{x}))$ subject to $\boldsymbol{x} \in S$.

We can define so-called order-representing and order-approximating achievement functions. Then we have the following properties for any reference point: If the achievement function is order-representing, then its solution is weakly Pareto optimal and if the function is order-approximating, then its solution is Pareto optimal. On the other hand, any (weakly) Pareto optimal solution can be found if the achievement function is order-representing. Thus, weakly Pareto optimal or Pareto optimal solutions can be obtained by moving the reference point only.

There are many achievement functions satisfying the above-presented conditions. An example of order-representing functions is $s_{\bar{\boldsymbol{z}}}(\boldsymbol{z}) = \max_{i=1,\ldots,k}[\,w_i(z_i - \bar{z}_i)\,]$, where $\boldsymbol{w}$ is some fixed positive weighting vector. An example of order-approximating achievement functions is

$$s_{\bar{\boldsymbol{z}}}(\boldsymbol{z}) = \max_{i=1,\ldots,k}[\,w_i(z_i - \bar{z}_i)\,] + \rho \sum_{i=1}^{k} w_i(z_i - \bar{z}_i) \ , \tag{8}$$

where $\boldsymbol{w}$ is as above and $\rho > 0$.

5 A Priori Methods

In a priori methods, the DM must specify her/his preferences, hopes and opinions before the solution process. Unfortunately, the DM does not necessarily know beforehand what it is possible to attain in the problem and how realistic her/his expectations are.

5.1 Value Function Method

The value function approach was already mentioned earlier. It is an excellent method if the DM happens to know an explicit mathematical formulation for the value function and if that function represents wholly her/his preferences. Unfortunately, it may be difficult, if not impossible, to get that mathematical expression. On the other hand, it can be difficult to optimize because of its possible complicated nature.

Note that the DM's preferences must satisfy certain conditions so that a value function can be defined on them. The DM must, e.g., be able to specify consistent preferences.

5.2 Lexicographic Ordering

In lexicographic ordering, the DM must arrange the objective functions according to their absolute importance. This ordering means that a more important objective is infinitely more important than a less important objective. After ordering, the most important objective function is minimized subject to the original constraints. If this problem has a unique solution, it is the final one. Otherwise, the second most important objective function is minimized. Now, a new constraint is added to guarantee that the most important objective function preserves its optimal value. If this problem has a unique solution, it is the final one. Otherwise, the process goes on.

The solution of lexicographic ordering is Pareto optimal. The method is quite simple and people usually make decisions successively. However, the DM may have difficulties in specifying an absolute order of importance. Besides, it is very likely that the process stops before less important objective functions are taken into consideration.

Note that lexicographic ordering does not allow a small increment of an important objective function to be traded off with a great decrement of a less important objective. Yet, trading off might often be appealing to the DM.

5.3 Goal Programming

Goal programming is one of the first methods expressly created for multiobjective optimization. The DM is asked to specify aspiration levels $\bar{z}_i$ $(i = 1, \ldots, k)$ for the objective functions and deviations from these aspiration levels are minimized. An objective function jointly with an aspiration level forms a *goal*. For minimization problems, goals are of the form $f_i(\boldsymbol{x}) \leq \bar{z}_i$. Here the aspiration levels are assumed to be selected so that they are not achievable simultaneously. Next, the *overachievements* δ_i of the objective function values are minimized.

The method has several variants. In the *weighted* approach, see [7], the weighted sum of the deviational variables is minimized. This means that in addition to the aspiration levels, the DM must specify positive weights. Then we have a problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{k} w_i \delta_i \\ \text{subject to} & f_i(\boldsymbol{x}) - \delta_i \leq \bar{z}_i \ \text{ for all } \ i = 1, \ldots, k, \\ & \delta_i \geq 0 \ \text{ for all } \ i = 1, \ldots, k, \\ & \boldsymbol{x} \in S \ , \end{array} \tag{9}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ and δ_i $(i = 1, \ldots, k)$ are the variables and $\delta_i = \max\,[0,\, f_i(\boldsymbol{x}) - \bar{z}_i]$.

In the *lexicographic* approach, the DM must specify a lexicographic order for the goals in addition to the aspiration levels. After the lexicographic ordering, the problem with the deviational variables as objective functions is solved subject to the constraints of (9) as explained in Sect. 5.2.

A combination of the weighted and the lexicographic approaches is quite popular. In this case, several objective functions may belong to the same class of importance in the lexicographic order. In each priority class, a weighted sum of the deviational variables is minimized.

The solution of a weighted or a lexicographic goal programming problem is Pareto optimal if either the aspiration levels form a Pareto optimal reference point or all the deviational variables δ_i have positive values at the optimum. In other words, if the aspiration levels are all feasible, the solution is equal to the reference point that is not necessarily Pareto optimal.

Goal programming is a very widely used and popular solution method. Goal-setting is an understandable and easy way of making decisions. The specification of the weights or the lexicographic ordering may be more difficult. It may also be hard to specify weights because they have no direct physical meaning.

6 Interactive Methods

The extensive interest devoted to interactive methods can be explained by the fact that assuming the DM has enough time and capabilities for co-operation, interactive methods can be presumed to produce the most satisfactory results. Many of the weak points of the methods in the other method classes are overcome. Namely, only part of the Pareto optimal points has to be generated and evaluated, the DM is not overloaded with information, and the DM can specify and correct her/his preferences and selections as the solution process continues and (s)he gets to know the problem and its potentialities better. This also means that the DM does not have to know any global preference structure. In addition, the DM can be assumed to have more confidence in the final solution since (s)he is involved throughout the solution process.

In interactive methods, the DM works together with an analyst or an interactive computer program. One can say that the analyst tries to determine the preference structure of the DM in an interactive way. After every iteration, some information is given to the DM and (s)he is asked to answer some questions or provide some other type of information.

Interactive methods differ from each other by the form in which information is given to the DM, by the form in which information is provided by the DM, and how the problem is transformed into a single objective optimization problem. It is always important that the DM finds the method worthwhile and acceptable and is able to use it properly.

There are three main stopping criteria in interactive methods. Either the DM finds a desirable solution and is convinced that it is preferred to all the other Pareto optimal solutions (see [16]), some algorithmic stopping or convergence rule is fulfilled or the DM gets tired of the solution process.

The number of interactive methods is large. Here we briefly describe some of them. In all the methods, less is assumed to be preferred to more by the DM. For more details, see [20] and references therein.

6.1 Geoffrion-Dyer-Feinberg Method

The Geoffrion-Dyer-Feinberg (GDF) method, proposed in [10], is one of the most well-known interactive methods and it is based on the maximization of the underlying (implicitly known) value function. The objective functions are assumed to be continuously differentiable and the feasible region S must be compact and convex.

Marginal rates of substitution specified by the DM at the current point $\boldsymbol{x}^h$ are here used to approximate the direction of steepest ascent of the value function. We have $m_{ij}(\boldsymbol{x}^h) = m_i = \frac{\partial U(\boldsymbol{f}(\boldsymbol{x}^h))}{\partial f_j} \Big/ \frac{\partial U(\boldsymbol{f}(\boldsymbol{x}^h))}{\partial f_i}$. Then the approximation is optimized by the method of Frank and Wolfe by solving the problem

$$\begin{array}{ll} \text{minimize} & \left(\sum_{i=1}^k -m_i \nabla_x f_i(\boldsymbol{x}^h) \right)^T \boldsymbol{y} \\ \text{subject to} & \boldsymbol{y} \in S \end{array} \tag{10}$$

with $\boldsymbol{y} \in \mathbb{R}^n$ being the variable.

The basic phases of the GDF algorithm are the following.

1. Ask the DM to specify a reference function.
2. Ask the DM to specify marginal rates of substitution between the reference function and the other objectives at the current solution point.
3. Solve (10). Set the search direction as the difference between the old (i.e. current) and the new solution.
4. Determine with the help of the DM the appropriate step-size to be taken in the direction.
5. If the DM wants to continue, go to step (2). Otherwise, stop.

When determining the step-size, the DM is asked to select the most preferred objective vector obtained with different step-sizes taken in the search direction. Note that the alternatives are not necessarily Pareto optimal. It is obvious that the task of selection becomes more difficult for the DM as the number of objective functions increases.

The GDF method can be characterized to be a non ad hoc method. If one knows the value function, it is easy to specify the marginal rates of substitution and select the best alternative. In spite of the plausible theoretical foundation of the GDF method, it is not so convincing and powerful in practice. The most important difficulty for the DM is the determining of the $k-1$ marginal rates of substitution at each iteration. Even more difficult is to give consistent and correct marginal rates of substitution at every iteration.

6.2 Tchebycheff Method

The Tchebycheff method, presented in [27] (pp. 419–450) and refined in [28], is an interactive weighting vector space reduction method where value functions are not used. The method has been designed to be user-friendly for the DM and, thus, complicated information is not required. It is assumed that the objective functions are bounded (from below).

To start with, a utopian objective vector is established. Then the distance from the utopian objective vector to the feasible objective region, measured by the weighted Tchebycheff metric, is minimized. Different solutions are obtained with well dispersed positive weighting vectors in the metric, as introduced in Sect. 4.3.

At the first iteration, a sample of the whole Pareto optimal set is generated. The solution space is reduced by tightening the upper and the lower bounds for the weights. Then a concentrated group of weighting vectors centred about the selected one is formed. Thus, the idea is to develop a sequence of progressively smaller subsets of the Pareto optimal set until a final solution is located.

Every Pareto optimal solution of can be found by solving the weighted Tchebycheff problem with $\boldsymbol{z}^{\star\star}$ but some of the solutions may be weakly Pareto optimal. Here this weakness is overcome by formulating the distance minimization problem in a lexicographic form:

$$\begin{array}{ll} \text{lex minimize} & \max_{i=1,\dots,k} \left[w_i(f_i(\boldsymbol{x}) - z_i^{\star\star}) \right], \sum_{i=1}^{k} (f_i(\boldsymbol{x}) - z_i^{\star\star}) \\ \text{subject to} & \boldsymbol{x} \in S \ . \end{array} \tag{11}$$

The solution of (11) is Pareto optimal and any Pareto optimal solution can be found.

The number of the alternative objective vectors to be presented to the DM is denoted by P. It may be fixed or different at each iteration. We can now present the main steps of the Tchebycheff algorithm.

1. Specify values for the set size P and the number of iterations H. Construct the utopian objective vector. Set $h = 1$.
2. Form the current weighting vector space and generate $2P$ dispersed weighting vectors.
3. Solve (11) for each of the $2P$ weighting vectors.
4. Present the P most different of the resulting objective vectors to the DM and let her/him choose the most preferred among them.
5. If $h = H$, stop. Otherwise, gather information for reducing the weighting vector space, set $h = h + 1$ and go to step (2).

The predetermined number of iterations is not necessarily conclusive. The DM can stop iterating when (s)he obtains a satisfactory solution or continue the solution process longer if necessary.

All the DM has to do in the Tchebycheff method is to compare several Pareto optimal objective vectors and select the most preferred one. The ease of the comparison depends on the magnitude of P and on the number of objective functions. This can be characterized as a non ad hoc method. If the value function is known, it is easy to select the alternative maximizing the value function.

The flexibility of the method is reduced by the fact that the discarded parts of the weighting vector space cannot be restored if the DM changes her/his mind. Thus, some consistency is required. The weakness of the Tchebycheff method is that a great deal of calculation is needed at each iteration and many of the results are discarded. For large and complex problems, the Tchebycheff method is not a realistic choice. On the other hand, parallel computing can be utilized.

6.3 Reference Point Method

As its name suggests, the reference point method (see, e.g., [32]) is based on a reference point of aspiration levels. The reference point is used to derive achievement scalarizing functions as introduced in Sect. 4.4. No specific assumptions are set on the problem to be solved. The reference point idea has been utilized in several methods in different ways. Wierzbicki's reference point method (to be discussed here) was among the first of them.

Before the solution process starts, some information is given to the DM about the problem. If possible, the ideal objective vector and the (approximated) nadir objective vector are presented to illustrate the ranges of the Pareto optimal set. Another possibility is to minimize and maximize the objective functions individually in the feasible region (if it is bounded). An appropriate form for the achievement function must also be selected.

The basic steps of the reference point method are the following:

1. Present information about the problem to the DM.
2. Ask the DM to specify a reference point.
3. Minimize the achievement function. Present the solution to the DM.
4. Calculate a number of k other (weakly) Pareto optimal solutions by minimizing the achievement function with perturbed reference points.
5. Present the alternatives to the DM. If (s)he finds one of the $k+1$ solutions satisfactory, stop. Otherwise, go to step (2).

The reference point method can be characterized as an ad hoc method or a method having both non ad hoc and ad hoc features. Alternatives are easy to compare if the value function is known. On the other hand, a reference point cannot be directly defined with the help of the value function. However, it is possible to test whether a new reference point has a higher value function value than the earlier solutions.

The reference point method is quite easy for the DM to understand. The DM only has to specify appropriate aspiration levels and compare objective vectors. What has earlier been said about the comparison of alternatives is also valid here. The solutions are weakly or Pareto optimal depending on the achievement function employed.

The freedom of the DM has both positive and negative aspects. On the one hand, the DM can direct the solution process and is free to change her/his mind during the process. On the other hand, there is no clear strategy to produce the final solution since the method does not help the DM to find improved solutions. A software family called DIDAS (Dynamic Interactive Decision Analysis and Support) has been developed on the basis of the reference point ideas (see [34] for details).

6.4 GUESS Method

The GUESS method is a simple interactive method related to the reference point method. The method is also sometimes called a *naïve method* and it is presented in [3]. The ideal objective vector $\boldsymbol{z}^\star$ and the nadir objective vector $\boldsymbol{z}^{\text{nad}}$ are required to be available.

The method proceeds as follows. The DM specifies a reference point (or a guess) below the nadir objective vector and the minimum weighted deviation from the nadir objective vector is maximized. Then the DM specifies a new reference point and the iteration continues until the DM is satisfied with the solution produced.

The problem to be solved is

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

The solution of (12) is weakly Pareto optimal and any Pareto optimal solution can be found. The algorithm can be formulated as follows.

1. Calculate the ideal objective vector and the nadir objective vector and present them to the DM.
2. Let the DM specify upper or lower bounds to the objective functions if (s)he so desires. Update (12), if necessary.
3. Ask the DM to specify a reference point between the ideal and the nadir objective vectors.
4. Solve (12) and present the solution to the DM.
5. If the DM is satisfied, stop. Otherwise, go to step (2).

The only stopping rule is the satisfaction of the DM. No guidance is given to the DM in setting new aspiration levels. This is typical of many reference point-based methods. The GUESS method is an ad hoc method. The existence of a value function would not help in determining new reference points or upper or lower bounds for the objective functions.

The weakness of the GUESS method is its heavy reliance on the availability of the nadir objective vector. As mentioned earlier, the nadir objective vector is not easy to determine and it is usually only an approximation.

An interesting practical observation is mentioned in [3]. Namely, DMs are easily satisfied if there is a small difference between the reference point and the solution obtained. Somehow they feel a need to be satisfied when they have almost achieved what they wanted. In this case they may stop iterating 'too early.' The DM is naturally allowed to stop the solution process if the solution really is satisfactory. But, the coincidence of setting the reference point near an attainable solution may unnecessarily increase the DM's satisfaction.

6.5 Satisficing Trade-Off Method

The satisficing trade-off method (STOM) (see, e.g., [25]) is based on ideas similar to the two earlier methods with emphasis on finding a satisficing solution. The differentiating factor is the trade-off information utilized.

The functioning of STOM is the following. After a (weakly) Pareto optimal solution has been obtained by optimizing a scalarizing function, it is presented to the DM. On the basis of this information (s)he is asked to classify the objective functions into three classes. The classes are the unacceptable objective functions whose values (s)he wants to improve, the acceptable objective functions whose values (s)he agrees to relax (impair) and the acceptable objective functions whose values (s)he accepts as they are.

The objective and the constraint functions are assumed to be twice continuously differentiable. Under some additional special assumptions, trade-off information can be obtained from the KKT multipliers related to the scalarizing function. With this information, appropriate upper bounds can be determined for the functions to be relaxed. Thus, the DM only has to specify aspiration levels for functions to be improved. This is called *automatic trade-off*. Next, a modified scalarizing function is minimized and the DM is asked to classify the objective functions at the new solution.

Different scalarizing functions have been suggested for use in STOM. In the original formulation, the weighted Tchebycheff metric is used and the weights are set as $w_i = \frac{1}{\bar{z}_i - z_i^{\star\star}}$ for $i = 1, \ldots, k$, where $\bar{\boldsymbol{z}}$ is a reference point and $\boldsymbol{z}^{\star\star}$ is a utopian objective vector so that $\bar{\boldsymbol{z}} > \boldsymbol{z}^{\star\star}$. If weakly Pareto optimal solutions are to be avoided, the scalarizing function can be augmented as described in Sect. 4.3.

Even if a value function existed, it could not be directly used to determine the functions to be decreased and increased or the amounts of change. Thus the method is characterized as an ad hoc method. If automatic trade-off is not used, the method resembles the GUESS method.

6.6 Light Beam Search

Light beam search, described in [13], combines the reference point idea and tools of multiattribute decision analysis. The achievement function (8) is used with weights only in the maximum part. The reference point is here assumed to be an infeasible objective vector.

It is assumed that the objective and the constraint functions are continuously differentiable and the ideal and the nadir objective vectors are available. In addition, none of the objective functions is allowed to be more important than all the others together.

In the light beam search, the learning process of the DM is supported by providing additional information about the Pareto optimal set at each iteration. This means that other solutions in the neighbourhood of the current solution (based on the reference point) are displayed. However, an attempt is made to avoid frustration on the part of the DM caused, e.g., by indifference between the alternatives.

Concepts used in ELECTRE methods (see, e.g., [26]) are here employed. The idea is to establish *outranking relations* between alternatives. It is said that one alternative outranks the other if it is at least as good as the latter. In the light beam search, additional alternatives near the current solution are generated so that they outrank the current one. Incomparable or indifferent alternatives are not shown to the DM.

To be able to compare alternatives and to define outranking relations, we need several thresholds from the DM. The DM is asked to provide *indifference thresholds* for each objective function describing intervals where indifference prevails. Furthermore, the line between indifference and preference does not have to be sharp. The hesitation between indifference and preference can be expressed by *preference thresholds*. One more type of threshold, namely a *veto threshold* can be defined. It prevents a good performance in some objectives from compensating for poor values on some other objectives.

Let us now outline the light beam algorithm.

1. If the DM wants to or can specify the best and the worst values for each objective function, save them. Alternatively calculate $\boldsymbol{z}^{\star}$ and $\boldsymbol{z}^{\mathrm{nad}}$. Set $\boldsymbol{z}^{\star}$ as

a reference point. Ask the DM to specify indifference thresholds. If desired, (s)he can also specify preference and veto thresholds.
2. Minimize the achievement function.
3. Present the solution to the DM. Calculate k Pareto optimal characteristic neighbours and present them as well to the DM. If the DM wants to see alternatives between any two of the $k + 1$ alternatives displayed, set their difference as a search direction, take different steps in this direction and project them onto the Pareto optimal set before showing them to the DM. If desired, save the current solution.
4. If desired, let the DM revise the thresholds and go to step (3). Otherwise, if the DM wants to give another reference point, go to step (2). If, on the other hand, the DM wants to select one of the alternatives displayed or saved as a current solution, go to step (3). Finally, if one of the alternatives is satisfactory, stop.

Characteristic neighbours are new alternative objective vectors that outrank the current solution. See [13] for details. The option of saving desirable solutions increases the flexibility of the method. The DM can explore different directions and select the best among different trials.

The light beam search can be characterized as an ad hoc method. If a value function were available, it could not directly determine new reference points. It could, however, be used in comparing the set of alternatives. Yet, the thresholds are important in the method and they must come from the DM. Specifying different thresholds is a new aspect when compared to the methods presented earlier. This may be demanding for the DM. Anyway, it is positive that the thresholds are not assumed to be global but can be altered at any time.

The light beam search is a rather versatile solution method where the DM can specify reference points, compare a set of alternatives and affect the set of alternatives in different ways. Thresholds are used to try to make sure that the alternatives generated are not worse than the current solution. In addition, the alternatives must be different enough to be compared and comparable on the whole. This should decrease the burden on the DM.

6.7 NIMBUS Method

NIMBUS (Nondifferentiable Interactive Multiobjective BUndle-based optimization System), presented in [20,21,22], is an interactive multiobjective optimization method designed especially to be able to handle nondifferentiable functions efficiently. For this reason, it is capable of solving complicated real-world problems. It is assumed that the objective and the constraint functions are locally Lipschitzian (if a nondifferentiable solver is used) and the ideal objective vector is available.

NIMBUS is based on the classification of the objective functions where the DM can easily indicate what kind of improvements are desirable and what kind of impairments are tolerable. The DM examines at iteration h the values of the objective functions calculated at the current solution $\boldsymbol{x}^h$ and divides the objective functions into up to five classes; functions f_i whose values

- should be decreased ($i \in I^<$),
- should be decreased to a certain aspiration level $\bar{z}_i < f_i(\boldsymbol{x}^h)$ ($i \in I^\leq$),
- are satisfactory at the moment ($i \in I^=$),
- are allowed to increase to a certain upper bound $\varepsilon_i > f_i(\boldsymbol{x}^h)$ ($i \in I^>$) and
- are allowed to change freely ($i \in I^\diamond$),

where $I^< \cup I^\leq \neq \emptyset$ and $I^= \cup I^> \cup I^\diamond \neq \emptyset$.

The difference between the classes $I^<$ and $I^\leq$ is that the functions in $I^<$ are to be minimized as far as possible but the functions in $I^\leq$ only as far as the aspiration level. The classification is the core of NIMBUS. However, the DM can specify optional positive weights w_i summing up to one.

After the DM has classified the objective functions, a subproblem

$$\begin{array}{ll} \text{minimize} & \max_{\substack{i \in I^< \\ j \in I^\leq}} \left[w_i(f_i(\boldsymbol{x}) - z_i^\star), w_j \max\left[f_j(\boldsymbol{x}) - \bar{z}_j,\ 0\right] \right] \\ \text{subject to} & f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{x}^h) \ \text{ for all } \ i \in I^< \cup I^\leq \cup I^=, \\ & f_i(\boldsymbol{x}) \leq \varepsilon_i \ \text{ for all } \ i \in I^>, \\ & \boldsymbol{x} \in S \end{array} \tag{13}$$

is formed, where $z_i^\star$ ($i \in I^<$) are components of the ideal objective vector. The solution of (13) is weakly Pareto optimal if the set $I^<$ is nonempty. On the other hand, any Pareto optimal solution can be found with an appropriate classification.

If the DM does not like the solution of (13) for some reason, (s)he can explore other solutions between the old and this new one. Then we calculate a search direction as a difference of these two solutions and provide more solutions by taking steps of different sizes in this direction.

The NIMBUS algorithm is given below. Note that the DM must be ready to give up something in order to attain improvement for some other objective functions.

1. Ask the DM to classify the objective functions at the current point.
2. Solve the subproblem and present the solution to the DM. If (s)he wants to see different alternatives between the old and the new solution, go to step (3). If the DM prefers either of the two solutions and want to continue from it, go to step (1). Otherwise, go to step (4).
3. Ask the DM to specify the desired number of alternatives P and calculate alternative vectors. Present their Pareto optimal counterparts to the DM and let her/him choose the most preferred one among them. If the DM wants to continue, go to step (1).
4. Check Pareto optimality and stop.

In NIMBUS, the DM expresses iteratively her/his desires. Unlike some other methods based on classification, the success of the solution process does not depend entirely on how well the DM manages in specifying the classification and the appropriate parameter values. It is important that the classification is not irreversible. Thus, no irrevocable damage is caused in NIMBUS if the solution obtained is not what was expected. The DM is free to go back or explore intermediate points. (S)he can easily get to know the problem and its possibilities by specifying, e.g., loose upper bounds and examining intermediate solutions.

In NIMBUS, the DM can explore the (weakly) Pareto optimal set and change her/his mind if necessary. The DM can also extract undesirable solutions from further consideration.

The method is ad hoc in nature, since the existence of a value function would not directly advise the DM how to act to attain her/his desires. A value function could only be used to compare different alternatives.

The method has been implemented as a WWW-NIMBUS system on the Internet (see [23]). Via the Internet we can centralize the computing to one server computer (at the University of Jyväskylä) and the WWW is a way of distributing the graphical user interface to the computers of each individual user. Besides, the user always has the latest version of NIMBUS available.

The most important aspect of WWW-NIMBUS is that it is easily accessible and available to any Internet user (http://nimbus.mit.jyu.fi/). No special tools, compilers or software besides a WWW browser are needed. The user saves the trouble of installing any software and the system is independent of the computer and the operating system used.

The system contains both a nondifferentiable local solver (proximal bundle method) (see [19], pp. 112–143) and a global solver (genetic algorithms) for the subproblem. When the first version of WWW-NIMBUS was implemented in 1995 it was a pioneering interactive optimization system on the Internet.

7 Conclusions

As has been stressed, a large variety of methods exists for multiobjective optimization problems and none of them can be claimed to be superior to the others in every aspect. When selecting a solution method, the specific features of the problem to be solved must be taken into consideration. In addition, the opinions of the DM are important. One can say that selecting a multiobjective optimization method is a problem with multiple objectives itself.

The theoretical properties of the methods can rather easily be compared. A comparative table summarizing some of the features of interactive methods is given in [20]. However, in addition to theoretical properties, practical applicability also plays an important role in the selection of an appropriate method for the problem to be solved. The difficulty is that practical applicability is hard to determine without experience and experimentation.

The features of the problem and the capabilities of the DM have to be charted before a solution method can be chosen. Some methods may suit some problems and some DMs better than others. A decision tree is provided in [20] for easing the selection.

As far as the future is concerned, the obvious conclusion in the development of methods is the importance of continuing in the direction of user-friendliness. Methods must be even better able to correspond to the characteristics of the DM. If the aspirations of the DM change during the solution process, the algorithm must be able to cope with this situation.

Computational tests have confirmed the idea that DMs want to feel in control of the solution process, and consequently they must understand what is happening. However, sometimes the DM simply needs support, and this should be available as well. Thus, the aim is to have methods that support learning so that guidance is given whenever necessary. The DM can be supported by using visual illustrations and further development of such tools is essential. In addition to bar charts, value paths and petal diagrams of alternatives, we may use 3D slices of the feasible objective region (see [18]) and other tools. Specific methods for different areas of application that take into account the characteristics of the problems are also important.

References

1. Benson, H.P.: Existence of Efficient Solutions for Vector Maximization Problems. Journal of Optimization Theory and Applications **26** (1978) 569–580
2. Benson, H.P.: Vector Maximization with Two Objective Functions. Journal of Optimization Theory and Applications (1979) 253–257
3. Buchanan, J.T.: A Naïve Approach for Solving MCDM Problems: The GUESS Method. Journal of the Operational Research Society **48** (1997) 202–206
4. Caballero, R., Rey, L., Ruiz, F., González, M.: An Algorithmic Package for the Resolution and Analysis of Convex Multiple Objective Problems. In: Fandel, G., Gal, T. (eds.): Multiple Criteria Decision Making: Proceedings of the Twelfth International Conference, Hagen (Germany). Springer-Verlag (1997) 275–284
5. Censor, Y.: Pareto Optimality in Multiobjective Problems. Applied Mathematics and Optimization **4** (1977) 41–59
6. Chankong, V., Haimes, Y.Y.: Multiobjective Decision Making Theory and Methodology. Elsevier Science Publishing Co., Inc., (1983)
7. Charnes, A., Cooper, W.W.: Goal Programming and Multiple Objective Optimization; Part 1. European Journal of Operational Research **1** (1977) 39–54
8. Gass, S., Saaty, T.: The Computational Algorithm for the Parametric Objective Function. Naval Research Logistics Quarterly **2** (1955) 39–45
9. Geoffrion, A.M.: Proper Efficiency and the Theory of Vector Maximization. Journal of Mathematical Analysis and Applications **22** (1968) 618–630
10. Geoffrion, A.M., Dyer, J.S., Feinberg, A.: An Interactive Approach for Multi-Criterion Optimization, with an Application to the Operation of an Academic Department. Management Science **19** (1972) 357–368
11. Haimes, Y.Y., Lasdon, L.S., Wismer, D.A.: On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization. IEEE Transactions on Systems, Man, and Cybernetics **1** (1971) 296–297
12. Hwang, C.-L., Masud, A.S.M.: Multiple Objective Decision Making – Methods and Applications: A State-of-the-Art Survey. Springer-Verlag (1979)
13. Jaszkiewicz, A., Slowiński, R.: The 'Light Beam Search' Approach – An Overview of Methodology and Applications. European Journal of Operational Research **113** (1999) 300–314
14. Keeney, R.L., Raiffa, H.: Decisions with Multiple Objectives: Preferences and Value Tradeoffs. John Wiley & Sons, Inc. (1976)
15. Korhonen, P., Salo, S., Steuer, R.E.: A Heuristic for Estimating Nadir Criterion Values in Multiple Objective Linear Programming. Operations Research **45** (1997) 751–757
16. Korhonen, P., Wallenius, J.: Behavioural Issues in MCDM: Neglected Research Questions. Journal of Multi-Criteria Decision Analysis **5** (1996) 178–182

17. Li, D., Yang, J.-B., Biswal, M.P.: Quantitative Parametric Connections between Methods for Generating Noninferior Solutions in Multiobjective Optimization. European Journal of Operational Research **117** (1999) 84–99
18. Lotov, A.V., Bushenkov, V., Chernykh, O: Multi-Criteria DSS for River Water Quality Planning. Microcomputers in Civil Engineering **12** (1997) 57–67
19. Mäkelä, M.M., Neittaanmäki, P.: Nonsmooth Optimization: Analysis and Algorithms with Applications to Optimal Control. World Scientific Publishing Co. (1992)
20. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers (1999)
21. Miettinen, K., Mäkelä, M.M.: Interactive Bundle-Based Method for Nondifferentiable Multiobjective Optimization: NIMBUS. Optimization **34** (1995) 231–246
22. Miettinen, K., Mäkelä, M.M.: Comparative Evaluation of Some Interactive Reference Point-Based Methods for Multi-Objective Optimisation. Journal of the Operational Research Society **50** (1999) 949–959
23. Miettinen, K., Mäkelä, M.M.: Interactive Multiobjective Optimization System WWW-NIMBUS on the Internet. Computers & Operations Research **27** (2000) 709–723
24. Miettinen, K., Mäkelä, M.M, Männikkö, T.: Optimal Control of Continuous Casting by Nondifferentiable Multiobjective Optimization. Computational Optimization and Applications **11** (1998) 177–194
25. Nakayama, H.: Aspiration Level Approach to Interactive Multi-Objective Programming and Its Applications. In: Pardalos, P.M., Siskos, Y., Zopounidis, C. (eds.): Advances in Multicriteria Analysis. Kluwer Academic Publishers (1995) 147–174
26. Roy, B.: The Outranking Approach and the Foundations of ELECTRE Methods. In: Bana e Costa, C.A. (ed.): Readings in Multiple Criteria Decision Aid. Springer-Verlag (1990) 155–183
27. Steuer, R.E.: Multiple Criteria Optimization: Theory, Computation, and Applications. John Wiley & Sons, Inc. (1986)
28. Steuer, R.E.: The Tchebycheff Procedure of Interactive Multiple Objective Programming. In: Karpak, B., Zionts, S. (eds.): Multiple Criteria Decision Making and Risk Analysis Using Microcomputers. Springer-Verlag (1989) 235–249
29. Steuer, R.E., Gardiner, L.R.: On the Computational Testing of Procedures for Interactive Multiple Objective Linear Programming. In: Fandel, G., Gehring, H. (eds.): Operations Research. Springer-Verlag (1991) 121–131
30. Vincke, P.: Multicriteria Decision-Aid. John Wiley & Sons, Inc. (1992)
31. Weistroffer, H.R.: Careful Usage of Pessimistic Values is Needed in Multiple Objective Optimization. Operations Research Letters **4** (1985) 23–25
32. Wierzbicki, A.P.: A Mathematical Basis for Satisficing Decision Making. Mathematical Modelling **3** (1982) 391–405
33. Wierzbicki, A.P.: On the Completeness and Constructiveness of Parametric Characterizations to Vector Optimization Problems. OR Spektrum **8** (1986) 73–87
34. Wierzbicki, A.P., Granat, J.: Multi-Objective Modeling for Engineering Applications: DIDASN++ System. European Journal of Operational Research **113** (1999) 372–389
35. Zadeh, L.: Optimality and Non-Scalar-Valued Performance Criteria. IEEE Transactions on Automatic Control **8** (1963) 59–60