

**This is an electronic reprint of the original article.  
This reprint *may differ* from the original in pagination and typographic detail.**

**Author(s):** Miettinen, Kaisa; Hakanen, Jussi; Podkopaev, Dmitry

**Title:** Interactive Nonlinear Multiobjective Optimization Methods

**Year:** 2016

**Version:**

**Please cite the original version:**

Miettinen, K., Hakanen, J., & Podkopaev, D. (2016). Interactive Nonlinear Multiobjective Optimization Methods. In S. Greco, M. Ehrgott, & J. R. Figueira (Eds.), *Multiple Criteria Decision Analysis : State of the Art Surveys* (2nd ed., pp. 931-980). Springer Science+Business Media. International Series in Operations Research and Management Science, 233. [https://doi.org/10.1007/978-1-4939-3094-4\\_22](https://doi.org/10.1007/978-1-4939-3094-4_22)

All material supplied via JYX is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

## Chapter 1

# INTERACTIVE NONLINEAR MULTIOBJECTIVE OPTIMIZATION METHODS

Kaisa Miettinen, Jussi Hakanen

*University of Jyväskylä*

*Department of Mathematical Information Technology*

*P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland*

kaisa.miettinen@jyu.fi, <http://www.mit.jyu.fi/miettine/engl.html>

jussi.hakanen@jyu.fi

Dmitry Podkopaev

*University of Jyväskylä*

*Department of Biological and Environmental Science*

*P.O. Box 35 (YA), FI-40014 University of Jyväskylä, Finland*

dmitry.podkopaev@gmail.com

**Abstract** An overview of interactive methods for solving nonlinear multiobjective optimization problems is given. In interactive methods, the decision maker progressively provides preference information so that the most satisfactory Pareto optimal solution can be found for her or his. The basic features of several methods are introduced and some theoretical results are provided. In addition, references to modifications and applications as well as to other methods are indicated. As the role of the decision maker is very important in interactive methods, methods presented are classified according to the type of preference information that the decision maker is assumed to provide.

**Keywords:** Multiple criteria decision making, Multiple objectives, Nonlinear optimization, Interactive methods, Pareto optimality

## 1. Introduction

Nonlinear multiobjective optimization means multiple criteria decision making involving nonlinear functions of (continuous) decision variables. In these problems, the best possible compromise, that is, Pareto optimal solution, is to be found from an (infinite) number of alternatives represented by decision variables restricted by constraint functions. Thus, enumerating the solutions is impossible.

Solving multiobjective optimization problems usually requires the participation of a human decision maker who is supposed to have insight into the problem and who can express preference relations between alternative solutions or objective functions or some other type of preference information. Multiobjective optimization methods can be divided into four classes according to the role of the decision maker in the solution process [79, 136]. If the decision maker is not involved, we use methods where no articulation of preference information is used, in other words, *no-preference methods*. If the decision maker expresses preference information after the solution process, we speak about *a posteriori methods* whereas *a priori methods* require articulation of preference information before the solution process. The most extensive class is *interactive methods*, where the decision maker specifies preference information progressively during the solution process. Here we concentrate on this last-mentioned class and introduce several examples of interactive methods.

In the literature, interactive methods have proven useful for various reasons. They have been found efficient from both computational and cognitive points of view. Because the decision maker directs the solution process with one's preferences, only those Pareto optimal solutions that are interesting to her or him need to be calculated. This means savings in computational cost when compared to a situation where a big set of Pareto optimal solutions should be calculated. On the other hand, the amount of new information generated per iteration is limited and, in this way, the decision maker does not need to compare too many solutions at a time. An important advantage of interactive methods is learning. Once the decision maker has provided preferences, (s)he can see from the Pareto optimal solutions generated, how attainable or feasible the preferences were. In this way, the decision maker gains insight about the problem. (S)he learns about the interdependencies between the objective functions and also about one's own preferences. The decision maker can also change her or his mind after the learning, if so desired.

Many real-world phenomena behave in a nonlinear way. Besides, linear problems can always be solved using methods created for nonlinear

problems but not vice versa. For these reasons, we here devote ourselves to nonlinear problems. We assume that all the information involved is deterministic and that we have a single decision maker.

In this presentation, we concentrate on general-purpose interactive methods and, thus, methods tailored for some particular problem type are not included. In recent years, interactive approaches have been developed in the field of evolutionary multiobjective optimization (see, for example, [15]), but we do not consider them here. The literature survey of years since 2000 has been limited to journal articles (in English). We describe in more detail methods with published applications.

## 2. Concepts

Let us begin by introducing several concepts and definitions. We study *multiobjective optimization problems* of the form

$$\begin{array}{ll} \text{minimize} & \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})\} \\ \text{subject to} & \mathbf{x} \in S \end{array} \quad (1.1)$$

involving  $k$  ( $\geq 2$ ) *objective functions* or objectives  $f_i : S \rightarrow \mathbf{R}$  that we want to minimize simultaneously. The *decision (variable) vectors*  $\mathbf{x}$  belong to the (nonempty) *feasible region*  $S \subseteq \mathbf{R}^n$ . The feasible region is formed by *constraint functions* but we do not fix them here.

We denote the image of the feasible region by  $Z \subset \mathbf{R}^k$  and call it a *feasible objective region*. *Objective (function) values* form *objective vectors*  $\mathbf{z} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x}))^T$ . Note that if  $f_i$  is to be maximized, it is equivalent to minimize  $-f_i$ .

We call a multiobjective optimization problem *convex* if all the objective functions and the feasible region are convex. On the other hand, the problem is *nondifferentiable* if at least one of the objective or the constraint functions is nondifferentiable. (Here nondifferentiability means that the function is not necessarily continuously differentiable but that it is locally Lipschitz continuous.)

We assume that the objective functions are at least partly conflicting and possibly incommensurable. This means that it is not possible to find a single solution that would optimize all the objectives simultaneously. As the definition of optimality we employ Pareto optimality. An objective vector is Pareto optimal (or noninferior or efficient or nondominated) if none of its components can be improved without deterioration to at least one of the other components. More formally, we have the following definition.

**Definition 1** A decision vector  $\mathbf{x}^* \in S$  is (globally) Pareto optimal if there does not exist another decision vector  $\mathbf{x} \in S$  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  $j$ .

An objective vector  $\mathbf{z}^* \in Z$  is Pareto optimal if there does not exist another vector  $\mathbf{z} \in Z$  such that  $z_i \leq z_i^*$  for all  $i = 1, \dots, k$  and  $z_j < z_j^*$  for at least one index  $j$ ; or equivalently,  $\mathbf{z}^*$  is Pareto optimal if the decision vector corresponding to it is Pareto optimal.

Local Pareto optimality is defined in a small neighborhood of  $\mathbf{x}^* \in S$ . Naturally, any globally Pareto optimal solution is locally Pareto optimal. The converse is valid, for example, for convex multiobjective optimization problems; see [22, 136], among others.

For the sake of brevity, we usually speak about Pareto optimality in the sequel. In practice, however, we only have locally Pareto optimal solutions computationally available, unless some additional requirement, such as convexity, is fulfilled or unless we have global solvers available.

A *Pareto optimal set* consists of (an infinite number of) Pareto optimal solutions. In interactive methods, we usually move around the Pareto optimal set and forget the other solutions. However, one should remember that this limitation may have weaknesses. Namely, the real Pareto optimal set may remain unknown. This may be the case if an objective function is only an approximation of an unknown function or if not all the objective functions involved are explicitly expressed.

Moving from one Pareto optimal solution to another necessitates trading off. To be more specific, 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 (see, for example, [24, 136]).

For any two solutions equally preferable to the decision maker there is a trade-off involving a certain increment in the value of one objective function that the decision maker is willing to 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* (see, for example, [136] for further details and properties).

Usually, one of the objective functions is selected as a *reference function* when trade-offs and marginal rates of substitution are treated. The pairwise trade-offs and the marginal rates of substitution are generated with respect to it.

Sometimes Pareto optimal sets are not enough but we need wider or smaller sets: weakly and properly Pareto optimal sets, respectively. An objective vector is *weakly Pareto optimal* if there does not exist

any other objective vector for which all the components are smaller. Weakly Pareto optimal solutions are sometimes computationally easier to generate than Pareto optimal solutions. Thus, they have relevance from a technical point of view. On the other hand, a vector is *properly Pareto optimal* if unbounded trade-offs are not allowed. For a collection of different definitions of proper Pareto optimality, see, for example, [136].

Multiobjective optimization problems are usually solved by *scalarization* which means that the problem is converted into one or a family of single (scalar) objective optimization problems. This produces a new scalarized problem with a real-valued objective function, possibly depending on some parameters. The resulting new problem must be solved with a single objective optimization method which is appropriate to the characteristics of the problem in question (taking into account, for example, differentiability and convexity). When scalarization is done properly, it can be guaranteed that the solution obtained is Pareto optimal to the original multiobjective optimization problems. For further details see, for example, [136, 206].

Interactive methods differ from each other by the way the problem is transformed into a single objective optimization problem, by the form in which information is provided by the decision maker and by the form in which information is given to the decision maker at each iteration of the solution process.

One way of inquiring the decision maker's opinions is to ask for satisfactory or desirable objective function values. They are called *aspiration levels* and denoted by  $\bar{z}_i$ ,  $i = 1, \dots, k$ . They form a vector  $\bar{\mathbf{z}} \in \mathbf{R}^k$  to be called a *reference point*.

The ranges of the objective functions in the set of Pareto optimal solutions give valuable information to the decision maker about the possibilities and restrictions of the problem (assuming the objective functions are bounded over  $S$ ). The components of the *ideal objective vector*  $\mathbf{z}^* \in \mathbf{R}^k$  are the individual optima of the objective functions. This vector represents the lower bounds of the Pareto optimal set. (In nonconvex problems, we need a global solver for minimizing the  $k$  functions.) Note that we sometimes need a vector that is strictly better than the ideal objective vector. This vector is called a *utopian objective vector* and denoted by  $\mathbf{z}^{**}$ .

The upper bounds of the Pareto optimal set, that is, the components of a *nadir objective vector*  $\mathbf{z}^{\text{nad}}$ , are much more difficult to obtain. Actually, there is no constructive method for calculating the nadir objective vector for nonlinear problems. However, a rough estimate can be obtained by keeping in mind the solutions where each objective function

attains its lowest value and calculating the values of the other objectives. The highest value obtained for each objective can be selected as the estimated component of  $\mathbf{z}^{\text{nad}}$ . This approach was originally proposed in [10] and later named as a *pay-off table* method. Some approaches for estimating the nadir objective vector for nonlinear multiobjective optimization are summarized in [136]. Examples of latest approaches include [37, 38].

It is sometimes assumed that the decision maker makes decisions on the basis of an underlying *value function*  $U : \mathbf{R}^k \rightarrow \mathbf{R}$  representing her or his preferences among the objective vectors [92]. Even though value functions are seldom explicitly known, they have been important in the development of multiobjective optimization methods and as a theoretical background. Thus, the value function is sometimes presumed to be known implicitly.

The value function is usually assumed to be *strongly decreasing*. In other words, the preferences of the decision maker are assumed to increase if the value of one objective function decreases while all the other objective values remain unchanged. In brief, we can say that *less is preferred to more*. In that case, the maximal solution of  $U$  is assured to be Pareto optimal. Note that regardless of the existence of a value function, in what follows, we shall assume that lower objective function values are preferred to higher, that is, less is preferred to more by the decision maker.

An alternative to the idea of maximizing some value function is *satisficing decision making* [206]. In this approach, the decision maker tries to achieve certain aspirations. If the aspirations are achieved, the solution is called a *satisficing solution*.

### 3. Introduction to Interactive Methods

A large variety of methods has been developed for solving multiobjective optimization problems. We can say that none of them is generally superior to all the others. As mentioned earlier, we apply here the classification of the methods into four classes according to the participation of the decision maker in the solution process. This classification was originally suggested in [79] and it was followed later, for example, in [136].

While we discuss interactive methods, we divide them into ad hoc and non ad hoc methods (based on value functions) as suggested in [224]. Even if one knew the decision maker's value function, one would not exactly know how to respond to the questions posed by an *ad hoc* method. On the other hand, in *non ad hoc* methods, the responses

can be determined or at least confidently simulated based on a value function.

Before describing the methods, we mention several references for further information. This presentation is mainly based on [136]. Concepts and methods for multiobjective optimization are also treated in [17, 24, 43, 44, 79, 129, 196, 206, 219, 223, 228, 242, 246, 271].

Interactive multiobjective optimization methods, in particular, are collected in [153, 179, 207, 243, 255]. Furthermore, methods with applications to large-scale systems and industry are presented in [65, 216, 232].

We shall not discuss non-interactive methods here. However, we mention some of such methods by name and give references for further information. Examples of no-preference methods are the method of the global criterion [270, 273] and the multiobjective proximal bundle method [145]. From among a posteriori methods we mention the weighting method [56, 272], the  $\varepsilon$ -constraint method [64] and the hybrid method [32, 254] as well as the method of weighted metrics [273] and the achievement scalarizing function approach [257, 258, 259, 261]. Multiobjective evolutionary algorithms are also a posteriori in nature, see, for example, [15] and references therein. A priori methods include the value function method [92], the lexicographic ordering [52] and the goal programming [25, 26, 81, 194, 195].

In what follows, we concentrate on interactive methods. In interactive methods, a solution pattern is formed and repeated several times. After every iteration, some information is given to the decision maker and (s)he is asked to answer some questions or to provide some other type of information. In this way, only a part of the Pareto optimal solutions has to be generated and evaluated, and the decision maker can specify and correct her or his preferences and selections during the solution process when (s)he gets to know the problem better. Thus, the decision maker does not need to have any global preference structure. Further information about the topics treated here can be found in [136, 153].

An interactive method typically contains the following main steps (as outlined, for example, in [141]): (1) initialize (for example, calculate ideal and nadir objective vectors and show them to the decision maker), (2) generate a Pareto optimal starting point (some neutral compromise solution or solution given by the decision maker) and show it to the decision maker, (3) ask for preference information from the decision maker (for example, aspiration levels or number of new solutions to be generated, depending on the method in question), (4) generate new Pareto optimal solution(s) according to the preferences and show it/them and possibly some other information about the problem to the decision maker,



(5) if several solutions were generated, ask the decision maker to select the best solution so far, and (6) stop, if the decision maker wants to. Otherwise, go to step (3).

Three main stopping criteria can be identified in interactive methods. In the best situation, the decision maker finds a desirable solution and wants to stop. Alternatively, the decision maker gets tired and stops or some algorithmic stopping rule is fulfilled. In the last-mentioned case, one must check that the decision maker agrees to stop.

As a matter of fact, as stated in [153], solving a multiobjective optimization problem with an interactive method can be regarded as a constructive process where, while learning, the decision maker builds a conviction of what is possible (that is, what kind of solutions are available and attainable) and confronting this knowledge with her or his preferences that also evolve. Based on this understanding, in interactive methods we should pay attention to psychological convergence, rather than to mathematical convergence (like, for example, optimizing some value function).

Sometimes, two different phases can be identified in interactive solution processes: *learning phase* and *decision phase* [153]. In the learning phase, the decision maker learns about the problem and gains understanding of what kind of solutions are attainable whereas the most preferred solution is found in the decision phase in the region identified in the first phase. Naturally, the two phases can also be used iteratively.

In what follows, we present several interactive methods. The idea is to describe a collection of methods based on different approaches. In addition, plenty of references are included. Note that although all the calculations take place in the decision variable space, we mostly speak about the corresponding objective vectors and refer to both as solutions since the space is apparent from the context.

When presenting the methods we apply the classification given in [125, 201] according to the type of preference information that the methods utilize. This is an important aspect because a reliable and an understandable way of extracting preference information from the decision maker is essential for the success of applying interactive methods. The decision maker must feel being in control and must understand the questions posed. Otherwise, the answers cannot be relied on in the solution process. It is also important to pay attention to the cognitive load set on the decision maker, as discussed in [112]. Applying the method should not set too much cognitive load on the decision maker.

In the first class, the decision maker specifies aspiration levels (in other words, a reference point) representing desirable objective function values. In the second class, the decision maker provides a classification

indicating which of the objective function values should be improved, maintained at the current value or allowed to impair. One should note that providing aspiration levels and a classification are closely related as justified in [148]. From classification information one can derive a reference point but not vice versa. The third class is devoted to methods where the decision maker compares different solutions and chooses a solution among several ones. The fourth class involves marginal rates of substitution referring to the amount of decrement in the value of one objective function that compensates to the decision maker an infinitesimal increment in the value of another objective function while the values of other objective functions remain unaltered. In addition to the four classes given in [125, 201], we consider a fifth class devoted to navigation based methods where the decision maker moves around in the set of Pareto optimal solutions in real time and controls the direction of movement in different ways.

#### 4. Methods Using Aspiration Levels

What is common to the methods in this section is a reference point consisting of desirable aspiration levels. With a reference point, the decision maker can conveniently express one's desires without any cognitive mapping as (s)he gives objective function values and obtains objective function values generated by the method. Some of the methods in this section utilize other types of preference information as well but the reference point is an integral element of each method.

##### 4.1 Reference Point Method

The reference point method [256, 257, 259] is based on vectors formed of reasonable or desirable aspiration levels. These reference points are used to derive scalarizing functions having minimal values at weakly, properly or Pareto optimal solutions.

No specific assumptions are set in this method. The idea is to direct the search by changing the reference point  $\bar{z}^h$  in the spirit of satisficing decision making rather than optimizing any value function. It is important that reference points are intuitive and easy for the decision maker to specify and their consistency is not an essential requirement.

Note that specifying a reference point can be considered as a way of classifying the objective functions. If the aspiration level is lower than the current objective value, that objective function is currently unacceptable, and if the aspiration level is equal to or higher than the current objective value, that function is acceptable. The difference here is that the reference point can be infeasible in every component. Naturally,

trading off is unavoidable in moving from one Pareto optimal solution to another and it is impossible to get a solution where all objective values are better than in the previous Pareto optimal solution but different solutions can be obtained with different approaches.

Scalarizing functions used in the reference point method are so-called achievement (scalarizing) functions and the method relies on their properties. We can define so-called order-representing and order-approximating achievement functions.

An example of a scalarized problem with an order-representing achievement function is

$$\begin{aligned} & \text{minimize} && \max_{i=1,\dots,k} [w_i(f_i(\mathbf{x}) - \bar{z}_i^h)] \\ & \text{subject to} && \mathbf{x} \in S, \end{aligned} \quad (1.2)$$

where  $\mathbf{w}$  is some fixed weighting vector with positive components. An example of a scalarized problem with an order-approximating achievement function is

$$\begin{aligned} & \text{minimize} && \max_{i=1,\dots,k} [w_i(f_i(\mathbf{x}) - \bar{z}_i^h)] + \rho \sum_{i=1}^k w_i(f_i(\mathbf{x}) - \bar{z}_i^h) \\ & \text{subject to} && \mathbf{x} \in S, \end{aligned} \quad (1.3)$$

where  $\mathbf{w}$  is as above and  $\rho > 0$  sets bounds for trade-offs.

**Theorem 1** *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 properly Pareto optimal if the function is also strongly increasing. Any (weakly) Pareto optimal solution can be found if the achievement function is order-representing. Finally, any properly Pareto optimal solution can be found if the function is order-approximating.*

The reference point method is very simple. Before the solution process starts, some information is given to the decision maker about the problem. If possible, the ideal objective vector and the (approximated) nadir objective vector are presented. Another possibility is to minimize and maximize the objective functions individually in the feasible region (if it is bounded). Naturally, the maximized objective function values do not typically represent components of the nadir objective vector but they can give some information to the decision maker in any case.

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

- 1 Select the achievement function. Present information about the problem to the decision maker. Set  $h = 1$ .

- 2 Ask the decision maker to specify a reference point  $\bar{\mathbf{z}}^h \in \mathbf{R}^k$ .
- 3 Minimize the achievement function and obtain a (weakly, properly or) Pareto optimal solution  $\mathbf{z}^h$ . Present it to the decision maker.
- 4 Calculate a number of  $k$  other (weakly, properly or) Pareto optimal solutions with perturbed reference points  $\bar{\mathbf{z}}(i) = \bar{\mathbf{z}}^h + d^h \mathbf{e}^i$ , where  $d^h = \|\bar{\mathbf{z}}^h - \mathbf{z}^h\|$  and  $\mathbf{e}^i$  is the  $i$ th unit vector for  $i = 1, \dots, k$ .
- 5 Present the alternatives to the decision maker. If (s)he finds one of the  $k + 1$  solutions satisfactory, stop. Otherwise, ask the decision maker to specify a new reference point  $\bar{\mathbf{z}}^{h+1}$ . Set  $h = h + 1$  and go to step 3.

The idea in perturbing the reference point in step 4 is that the decision maker gets a better conception of the possible solutions around the current solution. If the reference point is far from the Pareto optimal set, the decision maker gets a wider description of the Pareto optimal set and if the reference point is near the Pareto optimal set, then a finer description of the Pareto optimal set is given.

In this method, the decision maker has to specify aspiration levels and compare objective vectors. The decision maker is free to change her or his mind during the solution process and can direct the solution process without being forced to understand complicated concepts and their meaning. On the other hand, the method does not necessarily help the decision maker to find more satisfactory solutions.

The reference point method is an ad hoc method because a reference point cannot directly be defined based on a value function. On the other hand, alternatives are easy to compare whenever a value function is known.

Let us mention that a software family called DIDAS (Dynamic Interactive Decision Analysis and Support) has been developed on the basis of the reference point ideas. It is described, for example, in [263].

Applications and modifications of the reference point method are provided in [12, 62, 155, 182, 183, 211, 213, 215, 225, 241, 244, 245, 260, 262].

## 4.2 GUESS Method

The GUESS method is also called a *naïve method* [19]. The method is related to the reference point method.

It is assumed that a global ideal objective vector  $\mathbf{z}^*$  and a global nadir objective vector  $\mathbf{z}^{\text{nad}}$  are available. The structure of the method is very simple: the decision maker specifies a reference point (or a guess)  $\bar{\mathbf{z}}^h$  and a Pareto optimal solution is generated which is somehow closest to the

reference point. Then the decision maker specifies a new reference point and so on.

The general idea is to maximize the minimum weighted deviation from the nadir objective vector. The scalarized problem to be solved is

$$\begin{aligned} & \text{maximize} && \min_{i=1,\dots,k} \left[ \frac{z_i^{\text{nad}} - f_i(\mathbf{x})}{z_i^{\text{nad}} - \bar{z}_i^h} \right] \\ & \text{subject to} && \mathbf{x} \in S. \end{aligned} \quad (1.4)$$

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

**Theorem 2** *The solution of (1.4) is weakly Pareto optimal and any Pareto optimal solution can be found.*

The GUESS algorithm has five basic steps.

- 1 Calculate  $\mathbf{z}^*$  and  $\mathbf{z}^{\text{nad}}$  and present them to the decision maker. Set  $h = 1$ .
- 2 Let the decision maker specify upper or lower bounds to the objective functions if (s)he so desires. Update the problem, if necessary.
- 3 Ask the decision maker to specify a reference point  $\bar{\mathbf{z}}^h$  between  $\mathbf{z}^*$  and  $\mathbf{z}^{\text{nad}}$ .
- 4 Solve (1.4) and present the solution to the decision maker.
- 5 If the decision maker is satisfied, stop. Otherwise, set  $h = h + 1$  and go to step 2.

In step 2, upper or lower bounds mean adding constraints to problem (1.4), but the ideal or the nadir objective vectors are not affected. The only stopping rule is the satisfaction of the decision maker. No guidance is given to the decision maker in setting new aspiration levels. This is typical of many reference point based methods.

The GUESS method is simple to use and no consistency of the preference information provided is assumed. The only information required from the decision maker is a reference point and possible upper and lower bounds, which are optional. Note that inappropriate lower bounds may lead into solutions that are not weakly Pareto optimal. Unfortunately, the GUESS method relies heavily on the availability of the nadir objective vector, which is usually only an estimation.

The GUESS method is an ad hoc method. The existence of a value function would not help in specifying reference points or bounds for the objective functions. The method has been compared to several other

interactive methods in [18, 21, 34] and it has performed surprisingly well. The reasons may be its simplicity and flexibility. One can say that decision makers seem to prefer solution methods where they can feel that they are in control.

### 4.3 Light Beam Search

The light beam search [82, 83] employs tools of multiattribute decision analysis (see, for example, [246]) together with reference point ideas. The basic setting is identical to the reference point method. The scalarized problem to be solved is

$$\begin{aligned} \text{minimize} \quad & \max_{i=1,\dots,k} [w_i(f_i(\mathbf{x}) - \bar{z}_i^h)] + \rho \sum_{i=1}^k (f_i(\mathbf{x}) - \bar{z}_i^h) \\ \text{subject to} \quad & \mathbf{x} \in S, \end{aligned} \quad (1.5)$$

where  $\mathbf{w}$  is a weighting vector,  $\bar{\mathbf{z}}^h$  is the current reference point and  $\rho > 0$ .

**Theorem 3** *The solution of (1.5) is properly Pareto optimal and any properly Pareto optimal solution can be found.*

The reference point is here assumed to be infeasible, that is, unattainable. It is also assumed that the objective and the constraint functions are continuously differentiable and that the objective functions are bounded over  $S$ . Furthermore, none of the objective functions is allowed to be more important than all the others together.

In the light beam search, the decision maker directs the search by specifying reference points. In addition, other solutions in the neighbourhood of the current solution are displayed. Thus, the idea is identical to that of the reference point method. The main difference is in the way the alternatives are generated. The motivation is to avoid comparing too similar alternatives or alternatives that are indifferent to the decision maker. To achieve this goal, concepts of ELECTRE methods (developed for handling with discrete problems in multiattribute decision analysis) are utilized (see, for example, [200]).

It is not always possible for the decision maker to distinguish between different alternatives. This means that there is an interval where indifference prevails. For this reason, the decision maker is asked to provide *indifference thresholds* for each objective function. The line between indifference and preference does not have to be sharp, either. The hesitation between indifference and preference can be expressed by *preference thresholds*. Finally, a *veto threshold* prevents a good performance

in some objectives from compensating for poor values on some other objectives.

In the light beam search, *outranking relations* are established between alternatives. An objective vector  $\mathbf{z}^1$  is said to outrank  $\mathbf{z}^2$  if  $\mathbf{z}^1$  is at least as good as  $\mathbf{z}^2$ . The idea is to generate  $k$  new alternative objective vectors such that they outrank the current solution. In particular, incomparable or indifferent alternatives are not shown to the decision maker. The alternatives to be shown are called *characteristic neighbours*. The neighbours are determined by projecting the gradient of one objective function at a time onto the linear approximation of those constraints that are active in the current solution.

We can now outline the light beam algorithm.

- 1 If the decision maker can specify the best and the worst values for each objective function, denote them by  $\mathbf{z}^*$  and  $\mathbf{z}^{\text{nad}}$ , respectively. Alternatively, calculate  $\mathbf{z}^*$  and  $\mathbf{z}^{\text{nad}}$ . Set  $h = 1$  and  $\bar{\mathbf{z}}^h = \mathbf{z}^*$ . Initialize the set of saved solutions as  $B = \emptyset$ . Ask the decision maker to specify an indifference threshold for each objective. If desired, (s)he can also specify preference and veto thresholds.
- 2 Calculate current Pareto optimal solution  $\mathbf{z}^h$  by solving (1.5).
- 3 Present  $\mathbf{z}^h$  to the decision maker. Calculate  $k$  Pareto optimal characteristic neighbours of  $\mathbf{z}^h$  and present them as well to the decision maker. If the decision maker 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 decision maker. If the decision maker wants to save  $\mathbf{z}^h$ , set  $B = B \cup \{\mathbf{z}^h\}$ .
- 4 If the decision maker wants to revise the thresholds, save them, set  $\mathbf{z}^h = \mathbf{z}^{h+1}$ ,  $h = h + 1$  and go then to step 3. If the decision maker wants to give another reference point, denote it by  $\bar{\mathbf{z}}^{h+1}$ , set  $h = h + 1$  and go to step 2. If the decision maker wants to select one of the alternatives or one solution in  $B$  as a current solution, set it as  $\mathbf{z}^{h+1}$ , set  $h = h + 1$  and go to step 3. If one of the alternatives is satisfactory, stop.

The option of saving desirable solutions in the set  $B$  increases the flexibility of the method. A similar option could be added to many other methods as well.

The name of the method comes from the idea of projecting a focused beam of light from the reference point onto the Pareto optimal set. The

lighted part of the Pareto optimal set changes if the location of the spotlight, that is, the reference point or the point of interest in the Pareto optimal set are changed.

In the light beam search, the decision maker specifies reference points, compares alternatives and affects the set of alternatives in different ways. Specifying different thresholds may be demanding for the decision maker. Note, however, that the thresholds are not constant but can be altered at any time. The developers of the method point out that it may be computationally rather demanding to find the exact characteristic neighbours in a general case. It is, however, noteworthy that the neighbours can be generated in parallel.

The light beam search is an ad hoc method because a value function could not directly determine new reference points. It could, however, be used in comparing alternatives. Remember that the thresholds are important here and they must come from the decision maker.

A modification of the method is described in [260].

#### 4.4 Other Methods Using Aspiration Levels

Many interactive methods of the class of methods using aspiration levels originate from the goal programming approach because the interpretation of a goal and a reference point are closely related. Examples of such methods include [130, 159, 181, 214, 233, 251]. Methods adopting a fuzzy approach to setting aspiration levels have been proposed in [75, 77, 156, 157, 205]. Some other aspiration level based interactive methods can be found in [14, 35, 61, 70, 96, 121, 180, 231, 234, 250, 252, 253].

### 5. Methods Using Classification

With a classification, the decision maker can express what kind of changes should be made to the current Pareto optimal solution to get a more desirable solution. Classification reminds the decision maker of the fact that it is not possible to improve all objective values of a Pareto optimal objective vector but impairment in some objective(s) must be allowed. The methods presented in this section utilize different numbers of classes. Some of the methods involve preference information other than classification but classification is the core element in all of them.

#### 5.1 Step Method

The step method (STEM) [10] is one of the first interactive methods developed for multiobjective optimization problems. Here we describe an extension for nonlinear problems according to [45] and [206], pp. 268–269.



STEM is based on the classification of the objective functions at the current iteration at  $\mathbf{z}^h = \mathbf{f}(\mathbf{x}^h)$ . It is assumed that the decision maker can indicate both functions that have acceptable values and those whose values are too high, that is, functions that are unacceptable. Then the decision maker is supposed to give up a little in the value(s) of some acceptable objective function(s)  $f_i$  (denoted by  $i \in I^>$ ) in order to improve the values of some unacceptable objective functions  $f_i$  (denoted by  $i \in I^<$ ) (here  $I^> \cup I^< = \{1, \dots, k\}$ ). To be more specific, the decision maker is asked to specify upper bounds  $\varepsilon_i^h > f_i(\mathbf{x}^h)$  for the functions in  $I^>$ .

The only requirement in the method is that the objective functions are bounded over  $S$  because distances are measured to the (global) ideal objective vector. The first scalarized problem to be solved is

$$\begin{aligned} & \text{minimize} && \max_{i=1, \dots, k} \left[ \frac{e_i}{\sum_{j=1}^k e_j} (f_i(\mathbf{x}) - z_i^*) \right] \\ & \text{subject to} && \mathbf{x} \in S, \end{aligned} \quad (1.6)$$

where  $e_i = \frac{1}{z_i^*} \frac{z_i^{\text{nad}} - z_i^*}{z_i^{\text{nad}}}$  as suggested in [45], or  $e_i = \frac{z_i^{\text{nad}} - z_i^*}{\max[|z_i^{\text{nad}}|, |z_i^*|]}$  as suggested in [243].

**Theorem 4** *The solution of (1.6) is weakly Pareto optimal. The problem has at least one Pareto optimal solution.*

After the decision maker has classified the objective functions, the feasible region is restricted according to the information of the decision maker. The weights of the relaxed objective functions are set equal to zero, that is  $e_i = 0$  for  $i \in I^>$ . Then a new distance minimization problem

$$\begin{aligned} & \text{minimize} && \max_{i=1, \dots, k} \left[ \frac{e_i}{\sum_{j=1}^k e_j} (f_i(\mathbf{x}) - z_i^*) \right] \\ & \text{subject to} && f_i(\mathbf{x}) \leq \varepsilon_i^h \text{ for all } i \in I^>, \\ & && f_i(\mathbf{x}) \leq f_i(\mathbf{x}^h) \text{ for all } i \in I^<, \\ & && \mathbf{x} \in S \end{aligned} \quad (1.7)$$

is solved.

The basic phases of the STEM algorithm are the following:

- 1 Calculate  $\mathbf{z}^*$  and  $\mathbf{z}^{\text{nad}}$  and the weighting coefficients  $e_i$  for  $i = 1, \dots, k$ . Set  $h = 1$ . Solve (1.6). Denote the solution by  $\mathbf{z}^h \in Z$ .
- 2 Ask the decision maker to classify the objective functions at  $\mathbf{z}^h$  into  $I^>$  and  $I^<$ . If the latter class is empty, stop. Otherwise, ask the decision maker to specify relaxed upper bounds  $\varepsilon_i^h$  for  $i \in I^>$ .

- 3 Solve (1.7) and denote the solution by  $\mathbf{z}^{h+1} \in Z$ . Set  $h = h + 1$  and go to step 2.

The solution process continues until the decision maker does not want to change any component of the current objective vector. If the decision maker is not satisfied with any of the components, then the procedure must also be stopped.

In STEM, the decision maker is moving from one weakly Pareto optimal solution to another. The idea of classification is quite simple for her or him. However, it may be difficult to estimate appropriate amounts of increment that would allow the desired amount of improvement in those functions whose values should be decreased.

STEM is an ad hoc method because the existence of a value function would not help in the classification process.

Applications and modifications of STEM are given in [7, 24, 36, 79, 85].

## 5.2 Satisficing Trade-Off Method

The satisficing trade-off method (STOM) [171, 175] utilizes classification and reference points. As its name suggests, STOM is based on satisficing decision making. The decision maker is asked to classify the objective functions at the current solution  $\mathbf{z}^h = \mathbf{f}(\mathbf{x}^h)$  into three classes: the unacceptable objective functions whose values should be improved ( $I^<$ ), the acceptable objective functions whose values may increase ( $I^>$ ) and the acceptable objective functions whose values are acceptable as they are (denoted by  $I^=$ ) (such that  $I^< \cup I^> \cup I^= = \{1, \dots, k\}$ ).

The decision maker only has to specify aspiration levels for the functions in  $I^<$ . The aspiration levels (that is, upper bounds) for the functions in  $I^>$  can be derived using so-called automatic trade-off. In addition, the aspiration levels for the functions in  $I^=$  are set equal to  $f_i(\mathbf{x}^h)$ . All the three kinds of aspiration levels form a reference point  $\bar{\mathbf{z}}^h$ .

Different scalarizing functions can be used in STOM. One alternative is to solve the scalarized problem

$$\begin{aligned} & \text{minimize} && \max_{i=1, \dots, k} \left[ \frac{f_i(\mathbf{x}) - z_i^{**}}{\bar{z}_i^h - z_i^{**}} \right] \\ & \text{subject to} && \mathbf{x} \in S, \end{aligned} \quad (1.8)$$

where the reference point must be strictly worse in each component than the utopian objective vector.

**Theorem 5** *The solution of (1.8) is weakly Pareto optimal and any Pareto optimal solution can be found.*

If weakly Pareto optimal solutions are to be avoided, the scalarized problem to be solved is

$$\begin{aligned} & \text{minimize} && \max_{i=1, \dots, k} \left[ \frac{f_i(\mathbf{x}) - z_i^{**}}{\bar{z}_i^h - z_i^{**}} \right] + \rho \sum_{i=1}^k \frac{f_i(\mathbf{x})}{\bar{z}_i^h - z_i^{**}} && (1.9) \\ & \text{subject to} && \mathbf{x} \in S, \end{aligned}$$

where  $\rho > 0$  is a sufficiently small scalar.

**Theorem 6** *The solution of (1.9) is properly Pareto optimal and any properly Pareto optimal solution can be found.*

Here the utopian objective vector must be known globally. However, if some objective function  $f_j$  is not bounded from below on  $S$ , then some small scalar value can be used as  $z_j^{**}$ .

Assuming all the functions involved are differentiable the scalarizing functions can be written in a differentiable form by introducing a scalar variable  $\alpha$  to be optimized and setting it as an upper bound for each function in the max-term. Under certain assumptions, trade-off rate information can be obtained from the Karush-Kuhn-Tucker multipliers connected to the solution of this formulation. In *automatic trade-off*, upper bounds for the functions in  $I^>$  are derived with the help of this trade-off information.

Let us now describe the STOM algorithm.

- 1 Select the scalarizing function. Calculate  $\mathbf{z}^{**}$ . Set  $h = 1$ .
- 2 Ask the decision maker to specify a reference point  $\bar{\mathbf{z}}^h \in \mathbf{R}^k$  such that  $\bar{z}_i^h > z_i^{**}$  for every  $i = 1, \dots, k$ .
- 3 Minimize the scalarizing function used. Denote the solution by  $\mathbf{z}^h$ . Present it to the decision maker.
- 4 Ask the decision maker to classify the objective functions. If  $I^< = \emptyset$ , stop. Otherwise, ask the decision maker to specify new aspiration levels  $\bar{z}_i^{h+1}$  for  $I \in I^<$ . Set  $\bar{z}_i^{h+1} = z_i^h$  for  $i \in I^=$ .
- 5 Use automatic trade-off to obtain new levels (upper bounds)  $\bar{z}_i^{h+1}$  for the functions in  $I^>$ . Set  $h = h + 1$  and go to step 3.

The decision maker can modify the levels calculated based on trade-off rate information if they are not agreeable. On the other hand, the decision maker can specify those upper bounds herself or himself, if so desired. If trade-off rate information is not available, for example, in a case when the functions are nondifferentiable, STOM is almost the same

as the GUESS method. The only difference is the scalarizing function used.

There is no need to repeat comments mentioned in connection with STEM and the GUESS method. In all of them, the role of the decision maker is easy to understand. STOM requires even less input from the decision maker if automatic trade-off is used.

As said before, in practice, classifying the objective functions into three classes and specifying the amounts of increment and decrement for their values is a subset of specifying a new reference point. A new reference point is implicitly formed.

STOM is an ad hoc method like all the other classification based methods. However, one must remember that the aim of the method is particularly in satisficing rather than optimizing some value function.

Modifications and applications of STOM are described in [95, 154, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 185, 249].

### 5.3 Reference Direction Method

In the classification based reference direction (RD) method [177, 178], a current objective vector  $\mathbf{z}^h$  is presented to the decision maker and (s)he is asked to specify a reference point  $\bar{\mathbf{z}}^h$  consisting of desired levels for the objective functions. However, as the idea is to move around the weakly Pareto optimal set, some objective functions must be allowed to increase in order to attain lower values for some other objectives.

As mentioned earlier, specifying a reference point is equivalent to an implicit classification indicating those objective functions whose values should be decreased till they reach some acceptable aspiration level, those whose values are satisfactory at the moment, and those whose values are allowed to increase to some upper bound. We denote again these three classes by  $I^<$ ,  $I^=$  and  $I^>$ , respectively. Furthermore, we denote the components of the reference point corresponding to  $I^>$  by  $\varepsilon_i^h$  (at iteration  $h$ ) because they represent upper bounds.

Here, steps are taken in the reference direction  $\bar{\mathbf{z}}^h - \mathbf{z}^h$  and the decision maker specifies a priori the number of steps to be taken, that is, the number of solutions to be generated. The idea is to move step by step as long as the decision maker wants to. In this way, extra computation is avoided when only those alternatives are calculated that the decision maker wants to see.

Alternatives are generated along the reference direction by solving the scalarized problem

$$\begin{aligned} & \text{minimize} && \max_{i \in I^<} \left[ \frac{f_i(\mathbf{x}) - z_i^h}{z_i^h - \bar{z}_i^h} \right] \\ & \text{subject to} && f_i(\mathbf{x}) \leq \varepsilon_i^h + \alpha(z_i^h - \varepsilon_i^h) \text{ for all } i \in I^>, \\ & && f_i(\mathbf{x}) \leq z_i^h \text{ for all } i \in I^=, \\ & && \mathbf{x} \in S, \end{aligned} \quad (1.10)$$

where  $0 \leq \alpha < 1$  is the step-size in the reference direction,  $\bar{z}_i^h < z_i^h$  for  $i \in I^<$  and  $\varepsilon_i^h > z_i^h$  for  $i \in I^>$ .

**Theorem 7** *The solution of (1.10) is weakly Pareto optimal for every  $0 \leq \alpha < 1$  and any Pareto optimal solution can be found.*

The steps of the RD algorithm are the following:

- 1 Find a starting solution  $\mathbf{z}^1$  and show it to the decision maker. Set  $h = 1$ .
- 2 If the decision maker does not want to decrease any component of  $\mathbf{z}^h$ , stop. Otherwise, ask the decision maker to specify  $\bar{\mathbf{z}}^h$ , where some of the components are lower and some higher or equal when compared to those of  $\mathbf{z}^h$ . If there are no higher values, set  $P = r = 1$  and go to step 3. Otherwise, ask the decision maker to specify the maximum number of alternatives  $P$  (s)he wants to see. Set  $r = 1$ .
- 3 Set  $\alpha = 1 - r/P$ . Solve (1.10) and get  $\mathbf{z}^h(r)$ . Set  $r = r + 1$ .
- 4 Show  $\mathbf{z}^h(r)$  to the decision maker. If (s)he is satisfied, stop. If  $r \leq P$  and the decision maker wants to see another solution, go to step 3. Otherwise, if  $r > P$  or the decision maker wants to change the reference point, set  $\mathbf{z}^{h+1} = \mathbf{z}^h(r)$ ,  $h = h + 1$  and go to step 2.

The RD method does not require artificial or complicated information from the decision maker; only reference points and the number of intermediate solutions are used. Some decision makers may appreciate the fact that they are not asked to compare several alternatives but only to decide whether another alternative is to be generated or not.

The decision maker must a priori determine the number of steps to be taken, and then intermediate solutions are calculated one by one as long as the decision maker wants to. This has both positive and negative sides. On one hand, it is computationally efficient since it may

be unnecessary to calculate all the intermediate solutions. On the other hand, the number of steps to be taken cannot be changed.

The RD method is an ad hoc method because a value function would not help in specifying reference points or the numbers of steps to be taken. It could not even help in selecting the most preferred alternative. Here one must decide for one solution at a time whether to calculate new alternative solutions or not. If the new alternative happens to be less preferred than its predecessor, one cannot return to the previous solution.

Applications and modifications of the RD method are described in [60, 146].

## 5.4 NIMBUS Method

The NIMBUS method was originally presented in [136, 145, 146] but here we describe the so-called synchronous version introduced in [149]. Originally, NIMBUS was particularly directed for nondifferentiable problems but nowadays it is a general interactive multiobjective optimization method for nonlinear problems.

NIMBUS offers flexible ways of performing interactive consideration of the problem and determining the preferences of the decision maker during the solution process. Classification is used as the means of interaction between the decision maker and the algorithm. In addition, the decision maker can ask for intermediate Pareto optimal solutions to be generated between any two Pareto optimal solutions.

In the classification, the decision maker can easily indicate what kind of improvements are desirable and what kind of impairments are tolerable. Opposed to the classification based methods introduced so far, NIMBUS has five classes available. The decision maker examines at every iteration  $h$  the current objective vector  $\mathbf{z}^h$  and divides the objective functions into up to five classes according to how the current solution should be changed to get a more desirable solution. The classes are functions  $f_i$  whose values

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

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

In addition to the classification, the decision maker is asked to specify the aspiration levels and the upper bounds if the second and the fourth class, respectively, are used. 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.

As mentioned, NIMBUS has more classes than STEM, STOM or the RD method. This means that the decision maker has more freedom and flexibility in specifying the desired changes in the objective values. Note that not all of the classes have to be used. The availability of the class  $I^{\diamond}$  means that some functions can be left unclassified for a while to be able to follow how their values change while the others are classified.

After the classification information has been obtained, a scalarized problem is solved and the Pareto optimal solution obtained reflects the desires of the decision maker as well as possible. In this way, the decision maker can learn about the attainability of her or his preferences. In the synchronous version of NIMBUS [149], the idea is to provide to the decision maker up to four slightly different Pareto optimal solutions based on the same preference information. The decision maker can decide how many solutions (s)he wants to see and compare. In this way, the decision maker can learn more about what kind of solutions are available in the area of the Pareto optimal set that (s)he is interested in.

After the classification, up to four scalarized problems are solved. The one that follows the classification information closest is

$$\begin{aligned}
& \text{minimize} && \max_{\substack{i \in I^< \\ j \in I^{\leq}}} \left[ \frac{f_i(\mathbf{x}) - z_i^*}{z_i^{\text{nad}} - z_i^{**}}, \frac{f_j(\mathbf{x}) - \hat{z}_j}{z_j^{\text{nad}} - z_j^{**}} \right] + \rho \sum_{i=1}^k \frac{f_i(\mathbf{x})}{z_i^{\text{nad}} - z_i^{**}} \\
& \text{subject to} && f_i(\mathbf{x}) \leq f_i(\mathbf{x}^h) \text{ for all } i \in I^< \cup I^{\leq} \cup I^=, \\
& && f_i(\mathbf{x}) \leq \varepsilon_i \text{ for all } i \in I^{\geq}, \\
& && \mathbf{x} \in S,
\end{aligned} \tag{1.11}$$

where a so-called augmentation coefficient  $\rho > 0$  is a relatively small scalar and  $z_i^*$  for  $i \in I^<$  are components of the ideal objective vector. The weighting coefficients  $1/(z_j^{\text{nad}} - z_j^{**})$  involving components of the nadir and the utopian objective vectors, respectively, have proven to facilitate capturing the preferences of the decision maker well. They also increase computational efficiency [150].

The other three problems are based on a reference point. As mentioned in Section 3, one can derive a reference point from classification information. If the decision maker has provided aspiration levels and upper bounds, they are directly used as components of the reference point. Similarly it is straightforward to use the current objective function value of the class  $I^=$ . In the class  $I^<$ , the component of the ideal

objective vector is used in the reference point and in the class  $I^\diamond$ , the component of the nadir objective vector is used. In this way, we can get a  $k$ -dimensional reference point and can solve reference point based scalarized problems. In the synchronous NIMBUS, the problems (1.4) of GUESS, (1.3) of the reference point method and (1.8) of the STOM method are used.

**Theorem 8** *The solution of (1.11) is Pareto optimal.*

The decision maker can also ask for intermediate solutions between any two Pareto optimal solutions  $\mathbf{x}^h$  and  $\hat{\mathbf{x}}^h$  to be generated. This means that we calculate a search direction  $\mathbf{d}^h = \hat{\mathbf{x}}^h - \mathbf{x}^h$  and provide more solutions by taking steps of different sizes in this direction. In other words, we generate  $P - 1$  new vectors  $\mathbf{f}(\mathbf{x}^h + t_j \mathbf{d}^h)$ ,  $j = 2, \dots, P - 1$ , where  $t_j = \frac{j-1}{P-1}$ . Their Pareto optimal counterparts (by setting each of the new vectors at a time as a reference point for (1.3)) are presented to the decision maker, who then selects the most satisfying solution among the alternatives.

The NIMBUS algorithm is given below. The solution process stops if the decision maker does not want to improve any objective function value or is not willing to impair any objective function value.

We denote the set of saved solutions by  $A$ . At the beginning, we set  $A = \emptyset$ . The starting point of the solution process can come from the decision maker or it can be some neutral compromise [261] between the objectives. The nadir and and utopian objective vectors must be calculated or estimated before starting the solution process.

The main steps of the synchronous NIMBUS algorithm are the following.

- 1 Generate a Pareto optimal starting point.
- 2 Ask the decision maker to classify the objective functions at the current Pareto optimal solution and to specify the aspiration levels and upper bounds if they are needed.
- 3 Ask the decision maker to select the maximum number of different solutions to be generated (between one and four) and solve as many problems (listed above).
- 4 Present the different new solutions obtained to the decision maker.
- 5 If the decision maker wants to save one or more of the new solutions to  $A$ , include it/them to  $A$ .



- 6 If the decision maker does not want to see intermediate solutions between any two solutions, go to step 8. Otherwise, ask the decision maker to select the two solutions from among the new solutions or the solutions in  $A$ . Ask the number of the intermediate solutions from the decision maker.
- 7 Generate the desired number of intermediate solutions and project them to the Pareto optimal set. Go to step 4.
- 8 Ask the decision maker to choose the most preferred one among the new and/or the intermediate solutions or the solutions in  $A$ . Denote it as the current Pareto optimal solution. If the decision maker wants to continue, go to step 2. Otherwise, stop.

In NIMBUS, the decision maker is free to explore the Pareto optimal set, to learn and also to change her or his mind if necessary. Furthermore, the selection of the most preferred alternative from a given set is possible, but not necessary. The decision maker can also extract undesirable solutions from further consideration. Unlike some other classification based methods, NIMBUS does not depend entirely on how well the decision maker manages in the classification. It is important that the classification is not irreversible. If the solution obtained is not satisfactory, the decision maker can go back to the previous solution or explore intermediate solutions. The method aims at being flexible and the decision maker can select to what extent (s)he exploits the versatile possibilities available. The method does not introduce too massive calculations, either.

Being a classification based method, NIMBUS is ad hoc in nature. A value function could only be used to compare different alternatives.

An implementation of NIMBUS is available on the Internet. This WWW-NIMBUS system is at the disposal of every academic Internet user at <http://nimbus.mit.jyu.fi/>. Positive sides of a WWW implementation are that the latest version of the system is always available and the user saves the trouble of installing the software. The operating system used or compilers available set no restrictions because all that is needed is a WWW browser. Furthermore, WWW provides a graphical user interface with possibilities for visualizing the classification phase, alternative solutions etc. The system contains both a nondifferentiable local solver and a global solver (genetic algorithm). For details, see [147, 149]. The first version of WWW-NIMBUS was implemented in 1995. Then, it was a pioneering interactive optimization system on the Internet.

There is also an implementation of NIMBUS in the Windows/Linux operating systems called IND-NIMBUS [1, 137]. It can be connected to

different simulation and modelling tools like Matlab and GAMS. Several local and global single objective optimization methods and their hybrids are available. It is also possible to utilize, for example, the optimization methods of GAMS. IND-NIMBUS has different tools for supporting graphical comparison of selected solutions and it also contains implementations of the Pareto Navigator method and the NAUTILUS method (see Subsections 8.2 and 6.2, respectively).

Applications and modifications of the NIMBUS method can be found in [47, 66, 67, 68, 71, 72, 114, 115, 138, 142, 145, 146, 148, 151, 152, 202, 209, 218].

### 5.5 Other Methods using Classification

Interactive physical programming is an interactive method developed for trade-off analysis and decision making in multidisciplinary optimization [236]. It is based on a physical programming approach to produce Pareto optimal solutions [133]. A second order approximation of the Pareto optimal set at the current Pareto optimal solution is produced and the decision maker is able to generate solutions in the approximation obeying her or his classification. However, this necessitates differentiability assumptions. A modification can be found in [76].

Some other classification based methods can be found in [8, 90, 135].

## 6. Methods where Solutions are Compared

In this section we present some methods where the decision maker is assumed to compare Pareto optimal solutions and select one of them. Thus, the decision maker is not assumed to provide much information but the cognitive load related to the comparison naturally depends on the number of alternatives to be considered.

### 6.1 Chebyshev Method

The Chebyshev method was originally called the Tchebycheff method. It was proposed in [219], pp. 419–450 and [222], refined in [220] and it is also known by the name *interactive weighted Tchebycheff procedure*. The idea in this weighting vector set reduction method is to develop a sequence of progressively smaller subsets of the Pareto optimal set until a most preferred solution is located.

This method does not have too many assumptions. All that is assumed is that the objective functions are bounded (from below) over  $S$ . To start with, a (global) utopian objective vector  $\mathbf{z}^{**}$  is established. Then the distance from the utopian objective vector to the feasible ob-

jective region is minimized by solving the scalarized problem

$$\begin{aligned} \text{lex minimize} \quad & \max_{i=1,\dots,k} [w_i^h(f_i(\mathbf{x}) - z_i^{**})], \sum_{i=1}^k (f_i(\mathbf{x}) - z_i^{**}) \\ \text{subject to} \quad & \mathbf{x} \in S. \end{aligned} \quad (1.12)$$

The notation above means that if the min-max problem does not have a unique solution, the sum term is minimized subject to the obtained solutions.

**Theorem 9** *The solution of (1.12) is Pareto optimal and any Pareto optimal solution can be found.*

In the Chebyshev method, different Pareto optimal solutions are generated by altering the weighting vector  $\mathbf{w}^h$ . At each iteration  $h$ , the weighting vector set  $W^h = \{\mathbf{w}^h \in \mathbf{R}^k \mid l_i^h < w_i^h < u_i^h, \sum_{i=1}^k w_i^h = 1\}$  is reduced to  $W^{h+1}$ , where  $W^{h+1} \subset W^h$ . At the first iteration, a sample of the whole Pareto optimal set is generated by solving (1.12) with well dispersed weighting vectors from  $W = W^1$  (with  $l_i^1 = 0$  and  $u_i^1 = 1$ ). The space  $W^h$  is reduced by tightening the upper and the lower bounds for the weights.

Let  $\mathbf{z}^h$  be the objective vector that the decision maker chooses from the sample at the iteration  $h$  and let  $\mathbf{w}^h$  be the corresponding weighting vector in the problem. Now a concentrated group of weighting vectors centred around  $\mathbf{w}^h$  is formed. In this way, a sample of Pareto optimal solutions centred about  $\mathbf{z}^h$  is obtained.

Before the solution process starts, the decision maker must set the number of alternative solutions  $P$  to be compared at each iteration and the number of iterations to be taken  $itn$ . We can now present the main features of the Chebyshev algorithm.

- 1 Set the set size  $P$  and a tentative number of iterations  $itn$ . Set  $l_i^1 = 0$  and  $u_i^1 = 1$  for all  $i = 1, \dots, k$ . Construct  $\mathbf{z}^{**}$ . Set  $h = 1$ .
- 2 Form the weighting vector set  $W^h$  and generate  $2P$  dispersed weighting vectors  $\mathbf{w}^h \in W^h$ .
- 3 Solve (1.12) for each of the  $2P$  weighting vectors.
- 4 Present the  $P$  most different of the resulting objective vectors to the decision maker and let her or him choose the most preferred among them.
- 5 If  $h = itn$ , stop.
- 6 Reduce  $W^h$  to get for  $W^{h+1}$ , set  $h = h + 1$  and go to step 2.

Problem (1.12) is solved more than  $P$  times so that solutions very close to each other do not have to be presented to the decision maker. On the other hand, the predetermined number of iterations is not necessarily conclusive. The decision maker can stop iterating when (s)he obtains a satisfactory solution or continue the solution process longer if necessary.

In this method, the decision maker is only asked to compare Pareto optimal objective vectors. The number of these alternatives and the number of objective functions affect the easiness of the comparison. The personal capabilities of the decision maker are also important. Note that some consistency is required from the decision maker because the discarded parts of the weighting vector space cannot be restored.

It must be mentioned that a great deal of calculation is needed in the method. That is why it may not be applicable for large and complex problems. However, parallel computing can be utilized when generating the alternatives.

The Chebyshev method is a non ad hoc method. It is easy to compare the alternative solutions with the help of a value function.

Applications and modifications of the Chebyshev method are given in [2, 87, 93, 126, 185, 192, 208, 221, 230, 265].

## 6.2 NAUTILUS Method

The NAUTILUS method, introduced in [140], has a different philosophy from many other interactive methods. It is based on the assumptions that past experiences affect the hopes of decision makers and that people do not react symmetrically to gains and losses. This is derived from the prospect theory of [86]. Typically, interactive multiobjective optimization methods move around the set of Pareto optimal solutions according to the preference of the decision maker and (s)he must trade-off, that is, give up in some objective functions in order to enable improvement in some others to get from one Pareto optimal solution to another. But according to the prospect theory, decision makers may have difficulties in allowing impairment, the decision maker may get anchored in the vicinity of the starting point and the solution process may even be prematurely terminated.

The NAUTILUS method is different from most interactive methods because it does not generate Pareto optimal solutions at every iteration. Instead, the solution process starts from the nadir objective vector representing bad values for all objective functions. In this way, the decision maker can attain improvement in each objective function without any trading-off and can simply indicate how much each of the objectives should be improved. It has also been observed that the decision maker

may be more satisfied with a given solution if the previous one was very undesirable and this lays the foundation of the NAUTILUS method.

The method utilizes the scalarized problem (1.3) of the reference point method but unlike other methods utilizing this problem where weights are kept unaltered during the whole solution process while their purpose is mainly to normalize different ranges of objectives, in NAUTILUS the weights have a different role as proposed in [124]. In NAUTILUS, the weights are varied to get different Pareto optimal solutions and some preference information is included in the weights. As mentioned earlier, the optimal solution of problem (1.3) is assured to be Pareto optimal for any reference point (see, for example, [136]).

As said, the NAUTILUS method starts from the nadir objective vector and at every iteration the decision maker gets a solution where all objective function values improve from the previous iteration. Thus, only the solution of the last iteration is Pareto optimal. To get started, the decision maker is asked to give the number of iterations (s)he plans to carry out, denoted by  $itn$ . This is an initial estimate and can be changed at any time.

As before, we denote by  $\mathbf{z}^h$  the objective vector corresponding to the iteration  $h$ . We set  $\mathbf{z}^0 = \mathbf{z}^{nad}$ . Therefore,  $\mathbf{z}^0$  (except in trivial problems) is not Pareto optimal. Furthermore, we denote by  $it^h$  the number of iterations left (including iteration  $h$ ). Thus,  $it^1 = itn$ . At each iteration, the range of reachable values that each objective function can have without impairment in any other objective function (in this and further iterations) will shrink. Lower and upper bounds on these reachable values will be calculated when possible. For iteration  $h$ , we denote by  $\mathbf{z}^{h,lo} = (z_1^{h,lo}, \dots, z_k^{h,lo})^T$  and  $\mathbf{z}^{h,up} = (z_1^{h,up}, \dots, z_k^{h,up})^T$  these lower and upper bounds, respectively. Initially,  $\mathbf{z}^{1,lo} = \mathbf{z}^*$  and  $\mathbf{z}^{1,up} = \mathbf{z}^{nad}$ . This information can be regarded as an actualization of the payoff table (see, for example, [136]) indicating new ideal and nadir values at each iteration, thus informing the decision maker of what values are achievable for each objective function.

For iteration  $h - 1$ , the objective vector  $\mathbf{z}^{h-1} = (z_1^{h-1}, \dots, z_k^{h-1})^T$  is shown to the decision maker, who has two possibilities to provide her or his preference information:

- 1 Ranking the objective functions according to the relative importance of improving current objective function values. Here the decision maker is not asked to give any global preference ranking of the objectives, but the local importance of improving each of the current objective function values. (S)he is asked to assign objective functions to classes in an increasing order of importance for improving the corresponding objective value  $z_i^{h-1}$ . With this

information the  $k$  objective functions can be allocated into index sets  $J_r$  which represent the importance levels  $r = 1, \dots, s$ , where  $1 \leq s \leq k$ . If  $r < t$ , then improving objective function values in  $J_r$  is less important than improving objective function values in  $J_t$ . Each objective function can belong to only one index set, but several objectives can be in the same index set  $J_r$ . We then set

$$w_i^h = \frac{1}{r(z_i^{\text{nad}} - z_i^{\star\star})} \quad \text{for all } i \in J_r, \quad r = 1, \dots, s. \quad (1.13)$$

2 Answering the question: *Assuming you have one hundred points available, how would you distribute them among the current objective values so that the more points you allocate, the more improvement on the corresponding current objective value is desired?* If the decision maker gives  $p_i^h$  points to the objective function  $f_i$ , we set  $\Delta q_i^h = p_i^h/100$  and

$$w_i^h = \frac{1}{\Delta q_i^h (z_i^{\text{nad}} - z_i^{\star\star})} \quad \text{for all } i = 1, \dots, k. \quad (1.14)$$

We set  $\bar{\mathbf{z}}^h = \mathbf{z}^{h-1}$ , and  $w_i = w_i^h$  ( $i = 1, \dots, k$ ), as defined in (1.13) or (1.14), depending on the way the decision maker specifies the preference information and solve the scalarized problem (1.3). Let us denote by  $\mathbf{x}^h$  the Pareto optimal decision vector obtained and set  $\mathbf{f}^h = \mathbf{f}(\mathbf{x}^h)$ . Then, at the next iteration we take a step from the current solution towards  $\mathbf{f}^h$  and show to the decision maker

$$\mathbf{z}^h = \frac{it^h - 1}{it^h} \mathbf{z}^{h-1} + \frac{1}{it^h} \mathbf{f}^h. \quad (1.15)$$

As mentioned, if  $h$  is the last iteration, then  $it^h = 1$  and  $\mathbf{z}^h = \mathbf{f}^h$  is the most preferred Pareto optimal objective vector and  $\mathbf{x}^h$  is the corresponding solution in the decision space. But if  $h$  is not the last iteration, then  $\mathbf{z}^h$  can even be an infeasible vector in the objective space. Nevertheless, it has the following properties:

**Theorem 10** *At any iteration  $h$ , components of  $\mathbf{z}^h$  are all better than the corresponding components of  $\mathbf{z}^{h-1}$ .*

It is important to point out that although  $\mathbf{z}^h$  is not a Pareto optimal objective vector of problem (1.1) (if  $h$  is not the last iteration), and it may even be infeasible for this problem, it is assured to either be in the

feasible objective set  $Z$  for problem (1.1) or there is some Pareto optimal objective vector where each objective function has a better value. On the other hand, each objective vector  $\mathbf{z}^h$  produced has better objective function values than the corresponding values in all previous iterations. In addition, at each iteration, a part of the Pareto optimal set is eliminated from consideration in the sense that it is not reachable unless a step backwards is taken.

Vectors  $\mathbf{z}^{h,lo}$  providing bounds for the objective values that can be attained at the next iteration can be calculated by solving  $k$  problems of the  $\varepsilon$ -constraint method so that each objective function is optimized in turn and the upper bounds for the other objective functions are taken from the corresponding components of  $\mathbf{z}^{h-1}$ .

Thus, the attainable values of  $\mathbf{z}^h$  are bound in the following way:

$$z_i^h \in [z_i^{h,lo}, z_i^{h-1}] \quad (i = 1, \dots, k).$$

By denoting  $\mathbf{z}^{h,up} = \mathbf{z}^{h-1}$ , we have

$$z_i^h \in [z_i^{h,lo}, z_i^{h,up}] \quad (i = 1, \dots, k). \quad (1.16)$$

Depending on the computational cost of solving the  $k$  problems of the  $\varepsilon$ -constraint method, it must be evaluated whether these bounds are worth to be calculated at each iteration. If this is regarded to be too time-consuming, calculating the bounds can be skipped.

In addition, a measure of the closeness of the current vector to the Pareto optimal set can be shown to the decision maker. This allows the decision maker to determine whether the approach rhythm to the Pareto optimal set is appropriate or whether it is too fast or too slow. The decision maker can affect this by adjusting the number of iterations still to be taken. Given the information available, the decision maker may take a step backwards if (s)he does not like the new solution generated or the bounds and/or change the number of remaining iterations. In the latter case, we assign a new value to  $it^h$ . In the former case, the decision maker can either:

- *continue with old preference information.* A new solution is obtained by considering a smaller step-size starting from the previous solution (for example, a half of the former step-size), or
- *provide new preference information.* Then a new iteration step is taken, starting from  $\mathbf{z}^{h-1}$ .

To get started, the ideal and the nadir objective vectors must be calculated or estimated. Then, an overview of the NAUTILUS algorithm can be summarized as follows.

- 1 Ask the decision maker to give the number of iterations,  $itn$ . Set  $h = 1$ ,  $\mathbf{z}^0 = \mathbf{f}^{1,up} = \mathbf{z}^{nad}$ ,  $\mathbf{f}^{1,lo} = \mathbf{z}^*$  and  $it^1 = itn$ .
- 2 Ask the decision maker to provide preference information in either of the two ways and calculate weights  $w_i^h$  ( $i = 1, \dots, k$ ).
- 3 Set the reference point and the weights and solve problem (1.3) to get  $\mathbf{x}^h$  and the corresponding  $\mathbf{f}^h$ .
- 4 Calculate  $\mathbf{z}^h$  according to (1.15).
- 5 Given  $\mathbf{z}^h$ , find  $\mathbf{f}^{h+1,lo}$  by solving  $k$   $\varepsilon$ -constraint problems. Furthermore, set  $\mathbf{f}^{h+1,up} = \mathbf{z}^h$ . Calculate the distance to the Pareto optimal set.
- 6 Show the current objective values  $z_i^h$  ( $i = 1, \dots, k$ ), together with the additional information  $[f_i^{h+1,lo}, f_i^{h+1,up}]$  ( $i = 1, \dots, k$ ) and the distance to the decision maker.
- 7 Set a new value for  $it^h$  if the decision maker wants to change the number of remaining iterations.
- 8 Ask the decision maker whether (s)he wants to take a step backwards. If so, go to step 10. Otherwise, continue.
- 9 If  $it^h = 1$ , stop with the last solution  $\mathbf{x}^h$  and  $\mathbf{f}^h$ . Otherwise, set  $it^{h+1} = it^h - 1$  and  $h = h + 1$ . If the decision maker wants to give new preference information, go to step 1. Alternatively, the decision maker can take a new step in the same direction (using the preference information of the previous iteration). Then, set  $\mathbf{f}^h = \mathbf{f}^{h-1}$ , and go to step 4.
- 10 Ask the decision maker whether (s)he would like to provide new preference information starting from  $\mathbf{z}^{h-1}$ . If so, go to step 2. Alternatively, the decision maker can take a shorter step with the same preference information given in step 2. Then, set  $\mathbf{z}^h = \frac{1}{2}\mathbf{z}^h + \frac{1}{2}\mathbf{z}^{h-1}$  and go to step 5.

The algorithm looks more complicated than it actually is. There are many steps to provide to the decision maker different options of how to continue the solution process. A good user interface plays an important role in making the options available intuitive.

The NAUTILUS method has been located in this class of methods because the decision maker must compare at each iteration the solution generated to the solution of the previous iteration and decide whether to proceed or to go backwards. Naturally, preference information indicating



how important it is to improve each of the objective functions from their current levels is also needed.

NAUTILUS is ad hoc in nature because all preference information needed cannot be obtained from a value function.

A modification of the NAUTILUS method is presented in [210].

### 6.3 Other Methods where Solutions are Compared

Methods where the decision maker is asked to compare different solutions have been developed rather recently. Such methods targeted at nonlinear problems can be found in [27, 88, 91, 102, 118, 127, 128].

## 7. Methods Using Marginal Rates of Substitution

In this section, we present methods that utilize preference information in the form of marginal rates of substitution or desirability of trade-off information provided. These methods are included here because they have played a role in the history of developing interactive methods. They aim at some sort of mathematical convergence in optimizing an estimated value function rather than psychological convergence. It is important that the decision maker understands well the concepts used in these methods to be able to apply them.

### 7.1 Interactive Surrogate Worth Trade-Off Method

The interactive surrogate worth trade-off (ISWT) method is introduced in [23] and [24], pp. 371–379. The ISWT method utilizes the scalarized  $\varepsilon$ -constraint problem where one of the objective functions is minimized subject to upper bounds on all the other objectives:

$$\begin{aligned} & \text{minimize} && f_\ell(\mathbf{x}) \\ & \text{subject to} && f_j(\mathbf{x}) \leq \varepsilon_j \quad \text{for all } j = 1, \dots, k, \quad j \neq \ell, \\ & && \mathbf{x} \in S, \end{aligned} \quad (1.17)$$

where  $\ell \in \{1, \dots, k\}$  and  $\varepsilon_j$  are upper bounds for the other objectives.

**Theorem 11** *The solution of (1.17) is weakly Pareto optimal. The decision vector  $\mathbf{x}^* \in S$  is Pareto optimal if and only if it solves (1.17) for every  $\ell = 1, \dots, k$ , where  $\varepsilon_j = f_j(\mathbf{x}^*)$  for  $j = 1, \dots, k, j \neq \ell$ . A unique solution is Pareto optimal for any upper bounds.*

The idea of the ISWT method is to maximize an approximation of an underlying value function. A search direction is determined based on the

opinions of the decision maker concerning trade-off rates at the current solution. The step-size to be taken in the search direction is determined by solving several  $\varepsilon$ -constraint problems and asking the decision maker to select the most satisfactory solution.

It is assumed that the underlying value function exists and is implicitly known to the decision maker. In addition, it must be continuously differentiable and strongly decreasing. Furthermore, the objective and the constraint functions must be twice continuously differentiable and the feasible region has to be compact. Finally, it is assumed that the Pareto optimality of the solutions of the  $\varepsilon$ -constraint problem is guaranteed and that trade-off rate information is available in the Karush-Kuhn-Tucker (KKT) multipliers related to the  $\varepsilon$ -constraint problem.

Changes in objective function values between a reference function  $f_\ell$  and all the other objectives are compared. For each  $i = 1, \dots, k, i \neq \ell$ , the decision maker must answer the following question: Let an objective vector  $\mathbf{z}^h$  be given. If the value of  $f_\ell$  is decreased by  $\lambda_i^h$  units, then the value of  $f_i$  is increased by one unit (or vice versa) and the other objective values remain unaltered. How desirable do you find this trade-off?

The response of the decision maker indicating the degree of preference is called a *surrogate worth* value. According to [23, 24] the response must be an integer between 10 and  $-10$  whereas it is suggested in [238] to use integers from 2 to  $-2$ .

The gradient of the underlying value function is then estimated with the help of the surrogate worth values. This gives a search direction with a steepest ascent for the value function. Several different steps are taken in the search direction and the decision maker must select the most satisfactory of them. In practice, the upper bounds of the  $\varepsilon$ -constraint problem are revised based on surrogate worth values with different step-sizes.

The main features of the ISWT algorithm can be presented with four steps.

- 1 Select  $f_\ell$  to be minimized and give upper bounds to the other objective functions. Set  $h = 1$ .
- 2 Solve (1.17) to get a solution  $\mathbf{z}^h$ . Trade-off rate information is obtained from the KKT multipliers.
- 3 Ask the decision maker for the surrogate worth values at  $\mathbf{z}^h$ .
- 4 If some stopping criterion is satisfied, stop. Otherwise, update the upper bounds with the help of the answers obtained in step 3 and solve several  $\varepsilon$ -constraint problems. Let the decision maker choose

the most preferred alternative  $\mathbf{z}^{h+1}$  and set  $h = h + 1$ . Go to step 3.

As far as stopping criteria are concerned, one can always stop when the decision maker wants to do so. A common stopping criterion is the situation where all the surrogate worth values equal zero. One more criterion is the case when the decision maker wants to proceed only in an infeasible direction.

In the ISWT method, the decision maker is asked to specify surrogate worth values and compare Pareto optimal alternatives. It may be difficult for the decision maker to provide consistent surrogate worth values throughout the solution process. In addition, if there is a large number of objective functions, the decision maker has to specify a lot of surrogate worth values at each iteration. On the other hand, the easiness of the comparison of alternatives depends on the number of objectives and on the personal abilities of the decision maker.

The ISWT method can be regarded as a non ad hoc method. The sign of the surrogate worth values can be judged by comparing trade-off rates with marginal rates of substitution (obtainable from the value function). Furthermore, when comparing alternatives, it is easy to select the one with the highest value function value.

Modification of the ISWT method are presented in [24, 28, 49, 63, 69].

## 7.2 Geoffrion-Dyer-Feinberg Method

In the Geoffrion-Dyer-Feinberg (GDF) method proposed in [57], the basic idea is related to that of the ISWT method. In both the methods, the underlying (implicitly known) value function is approximated and maximized. In the GDF method, the approximation is based on marginal rates of substitution.

It is assumed that an underlying value function exists, is implicitly known to the decision maker and is strongly decreasing with respect to the reference function  $f_\ell$ . In addition, the corresponding value function with decision variables as variables (i.e., arguments) must be continuously differentiable and concave on  $S$ . Furthermore, the objective functions have to be continuously differentiable and the feasible region  $S$  must be compact and convex.

Let  $\mathbf{x}^h$  be the current solution. We can obtain a local linear approximation for the gradient of the value function with the help of marginal rates of substitution  $m_i^h$  involving a reference function  $f_\ell$  and the other

functions  $f_i$ . Based on this information we solve the problem

$$\begin{aligned} & \text{maximize} && \left( \sum_{i=1}^k -m_i^h \nabla_x f_i(\mathbf{x}^h) \right)^T \mathbf{y} \\ & \text{subject to} && \mathbf{y} \in S, \end{aligned} \quad (1.18)$$

where  $\mathbf{y} \in \mathbf{R}^n$  is the variable. Let us denote the solution by  $\mathbf{y}^h$ . Then, the search direction is  $\mathbf{d}^h = \mathbf{y}^h - \mathbf{x}^h$ .

The following problem is to find a step-size. The decision maker can be offered objective vectors where steps of different sizes are taken in the search direction starting from the current solution. Unfortunately, these alternatives are not necessarily Pareto optimal.

Now we can present the GDF algorithm.

- 1 Ask the decision maker to select  $f_\ell$ . Set  $h = 1$ .
- 2 Ask the decision maker to specify marginal rates of substitution between  $f_\ell$  and the other objectives at the current solution  $\mathbf{z}^h$ .
- 3 Solve (1.18). Set the search direction  $\mathbf{d}^h$ . If  $\mathbf{d}^h = \mathbf{0}$ , stop.
- 4 Determine with the help of the decision maker the appropriate step-size  $t^h$  to be taken in  $\mathbf{d}^h$ . Denote the corresponding solution by  $\mathbf{z}^{h+1} = \mathbf{f}(\mathbf{x}^h + t^h \mathbf{d}^h)$ .
- 5 Set  $h = h + 1$ . If the decision maker wants to continue, go to step 2. Otherwise, stop.

In the GDF method, the decision maker has to specify marginal rates of substitution and select the most preferred solution from a set of alternatives. The theoretical foundation of the method is convincing but the practical side is not as promising. At each iteration the decision maker has to determine  $k - 1$  marginal rates of substitution in a consistent and correct way. On the other hand, it is obvious that in practice the task of selection becomes more difficult for the decision maker as the number of objective functions increases. Another drawback is that not all the solutions presented to the decision maker are necessarily Pareto optimal. They can naturally be projected onto the Pareto optimal set but this necessitates extra effort.

The GDF method is a non ad hoc method. The marginal rates of substitution and selections can be done with the help of value function information. Note that if the underlying value function is linear, the marginal rates of substitution are constant and only one iteration is needed.

Applications and modifications of the GDF method are described in [4, 40, 42, 51, 53, 73, 79, 84, 143, 144, 164, 186, 197, 203, 219, 264].

### 7.3 Other Methods Using Marginal Rates of Substitution

Although preference information about relative importance of different objectives in one form or another is utilized in many interactive methods, there are very few methods where the desirable marginal rates of substitute are the main preference information. Such methods are presented in [120, 131, 268].

## 8. Navigation Methods

By navigation we refer to methods where new Pareto optimal solution alternatives are generated in a real-time imitating fashion along directions that are derived from the information the decision maker has specified. In this way, the decision maker can learn about the interdependencies among the objective functions. The decision maker can either continue the movement along the current direction or change the direction, that is, one's preferences. Increased interest has been devoted to navigation based methods in the literature in recent years. In these methods, the user interface plays a very important role in enabling the navigation.

### 8.1 Reference Direction Approach

The reference direction approach [103, 108] is also known by the name *visual interactive approach*. It contains ideas from, for example, the GDF method and the reference point method. However, more information is provided to the decision maker.

In reference point based methods, a reference point is projected onto the Pareto optimal set by optimizing an achievement function. Here, instead, a so-called *reference direction* as a whole is projected onto the Pareto optimal set. It is a vector from the current solution  $\mathbf{z}^h$  to the reference point  $\bar{\mathbf{z}}^h$ . In practice, steps of different sizes are taken along the reference direction and projected. The idea is to plot the objective function values on a computer screen as value paths. The decision maker can move the cursor back and forth and see the corresponding numerical values at each solution.

Solutions along the reference direction are generated by solving the scalarized problem

$$\begin{aligned} & \text{minimize} && \max_{i \in I} \left[ \frac{f_i(\mathbf{x}) - \bar{z}_i^h}{w_i} \right] \\ & \text{subject to} && \bar{\mathbf{z}}^h = \mathbf{z}^h + t\mathbf{d}^{h+1}, \\ & && \mathbf{x} \in S, \end{aligned} \tag{1.19}$$

where  $I = \{i \mid w_i > 0\} \subset \{1, \dots, k\}$  and  $t$  has different discrete nonnegative values. The weighting vector can be, for example, the reference point specified by the decision maker.

**Theorem 12** *The solution of (1.19) is weakly Pareto optimal.*

The algorithm of the reference direction approach is as follows.

- 1 Find an arbitrary objective vector  $\mathbf{z}^1$ . Set  $h = 1$ .
- 2 Ask the decision maker to specify a reference point  $\bar{\mathbf{z}}^h \in \mathbf{R}^k$  and set  $\mathbf{d}^{h+1} = \bar{\mathbf{z}}^h - \mathbf{z}^h$ .
- 3 Find the set  $Z^{h+1}$  of weakly Pareto optimal solutions with different values of  $t$  in (1.19).
- 4 Ask the decision maker to select the most preferred solution  $\mathbf{z}^{h+1}$  in  $Z^{h+1}$ .
- 5 If  $\mathbf{z}^h \neq \mathbf{z}^{h+1}$ , set  $h = h + 1$  and go to step 2. Otherwise, check the optimality conditions. If the conditions are satisfied, stop. Otherwise, set  $h = h + 1$  and set  $\mathbf{d}^{h+1}$  to be a search direction identified by the optimality checking procedure. Go to step 3.

Checking the optimality conditions in step 5 is the most complicated part of the algorithm. Thus far, no specific assumptions have been set on the value function. However, we can check the optimality of  $\mathbf{z}^{h+1}$  if the cone containing all the feasible directions has a finite number of generators. We must then assume that an underlying value function exists and is pseudoconcave on  $Z$ . In addition,  $S$  must be convex and compact and the constraint functions must be differentiable.

The role of the decision maker is similar in the reference point method and in the reference direction approach: specifying reference points and selecting the most preferred alternative. But by providing similar reference point information, in the reference direction approach, the decision maker can explore a wider part of the weakly Pareto optimal set. This possibility brings the task of comparing the alternatives.

The performance of the method depends greatly on how well the decision maker manages to specify the reference directions that lead to more satisfactory solutions. The consistency of the decision maker's answers is not important and it is not checked in the algorithm.

The reference direction approach can be characterized as an ad hoc method like the other reference point based methods. The aim is to support the decision maker in getting to know the problem better.

A dynamic user interface to the reference direction approach and its adaptation to generalized goal programming is introduced in [110]. This

method for linear multiobjective optimization problems is called the *Pareto race*.

Applications and modifications of the reference direction approach are described in [11, 101, 103, 104, 105, 106, 107, 109].

## 8.2 Pareto Navigator Method

Pareto Navigator is an interactive method utilizing a polyhedral approximation of the Pareto optimal set for convex problems [48]. Pareto Navigator consists of two phases, namely an initialization phase, where the decision maker is not involved, and a navigation phase. In the initialization phase, a relatively small set of Pareto optimal objective vectors is assumed to be available to form a polyhedral approximation of the Pareto optimal set in the objective space. These objective vectors can be computed, for example, by using some a posteriori approach.

Pareto Navigator has been developed especially for the learning phase of interactive solution processes introduced in Section 3 and for computationally expensive problems where objective function and/or constraint function value evaluations may be time-consuming because the problem is, for example, simulation-based. In these problems, computing Pareto optimal solutions can take a lot of time. For this reason, besides the original (computationally expensive) problem, an approximation is used to enable fast computations so that the decision maker does not need to wait for new solutions being generated based on her or his preferences.

In Pareto Navigator, the decision maker is not involved in the part of the solution process where the set of objective vectors representing the Pareto optimal set is generated. Once the approximation has been created based on the objective vectors available, the original problem is not solved (in the navigation phase). When the navigation phase starts, the decision maker can navigate dynamically in the approximated Pareto optimal set in real time since approximated Pareto optimal solutions can be produced by solving linear programming problems that are computationally inexpensive.

Whenever the decision maker has found an interesting approximated Pareto optimal solution, the corresponding solution to the original problem can be generated by solving problem (1.3) with the approximated solution as a reference point. This can be seen as projecting the approximated solution to the Pareto optimal set of the original problem. However, this step may take time if the original problem is computationally expensive.

As mentioned, the multiobjective optimization problem is assumed to be convex, that is, the objective functions and the feasible region must be convex. The algorithm of Pareto Navigator is as follows.

- 1 Compute first a polyhedral approximation of the Pareto optimal set in the objective space based on a small set of Pareto optimal objective vectors. Use the extreme values present in this set to approximate the ideal and nadir objective vectors. Ask the decision maker to select a starting point for navigation (for example, one of the Pareto optimal objective vectors available).
- 2 Show the current objective vector to the decision maker and ask her or him whether a preferred solution has been found. If yes, go to step 6. Otherwise, continue.
- 3 Ask the decision maker whether (s)he would like to proceed to some other direction. If the decision maker does not want to change the direction, go to step 5.
- 4 Ask the decision maker to specify how the current objective vector should be improved by giving aspiration levels for the objectives. To aid her or him, show the ideal and the nadir objective vectors. Based on the resulting reference point  $\bar{\mathbf{z}}$  and the current objective vector  $\mathbf{z}^c$ , set a search direction.
- 5 Ask the decision maker to indicate a speed of movement, that is, a step-size  $\alpha > 0$  to the direction specified. Generate approximated Pareto optimal solutions in the direction specified by using a reference point based approach for each step in the direction starting from the current objective vector  $\mathbf{z}^c$ . Once an approximated solution is produced, it is instantly shown to the decision maker. New approximated solutions are produced to the direction specified until the decision maker stops the movement. Then go to step 2.
- 6 Once the decision maker has found a satisfactory solution, stop. Project the approximated Pareto optimal solution to the actual Pareto optimal set and show the resulting solution to the decision maker.

The search direction is based on decision maker's preferences and there are different ways of defining a direction where to move on the approximation. In Pareto Navigator, the direction is specified by  $\mathbf{d} = \bar{\mathbf{z}} - \mathbf{z}^c$ . The approximated Pareto optimal solutions are then computed



by solving scalarized problems of the form

$$\begin{aligned} & \text{minimize} && \max_{i=1,\dots,k} w_i (z_i - \bar{z}_i(\alpha)) \\ & \text{subject to} && \mathbf{A}\mathbf{z} \leq \mathbf{b}, \end{aligned} \tag{1.20}$$

where  $\bar{\mathbf{z}}(\alpha) = \mathbf{z}^c + \alpha\mathbf{d}$  is the reference point depending on the step parameter  $\alpha > 0$  (being varied) to the direction  $\mathbf{d}$  and  $w_i$ ,  $i = 1, \dots, k$ , are the scaling coefficients. The scaling coefficient can be set as one divided by the difference of the estimated nadir and ideal objective values. The linear constraints of problem (1.20) form a convex hull for a set of Pareto optimal solutions used to form the polyhedral approximation and, in practice, the reference point  $\bar{\mathbf{z}}(\alpha)$  is projected to the nondominated facets of the convex hull.

The objective function of problem (1.20) is nonlinear with respect to  $\mathbf{z}$  but can be linearized by adding a new real variable  $\xi \in \mathbf{R}$  replacing the max term. The resulting problem is then linear with respect to a new variable  $\mathbf{z}' = (\xi, \mathbf{z})^T$ . Due to linearity, approximated Pareto optimal solutions can be produced and shown to the decision maker in real time by moving the reference point along the direction  $\mathbf{d}$ . This is done by increasing the value of  $\alpha$ . At any point, the decision maker is able to find the closest actual Pareto optimal solution for any approximated Pareto optimal solution. However, as said, this can be time consuming.

Because the decision maker must specify desirable objective function values, this method is ad hoc by nature.

During the navigation, the approximated solutions are shown to the decision maker by presenting the approximated values as a continuous path (value path) for each objective function separately (bar charts can be used as well). Pareto navigator is implemented in the IND-NIMBUS system [1, 137] and the graphical user interface development is described in [237].

### 8.3 Pareto Navigation Method

The Pareto Navigation method developed in [160] assumes the convexity of all objective functions and a convex feasible region. Similar to the Pareto Navigator method, the idea is to enable a fast generation of new solutions in the navigation phase. Thus, the method starts with formulating a surrogate problem based on a set of pre-computed Pareto optimal decision vectors  $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ . The most preferred solution is sought among their convex hull

$$\mathcal{X} = \left\{ \sum_{j=1}^m v_j \mathbf{x}^{(j)} : \sum_{j=1}^m v_j = 1, v_j \geq 0 \text{ for all } j = 1, \dots, m \right\}.$$

This allows replacing the feasible region of the original problem with the set of convex combination coefficients  $v_1, \dots, v_m$  in the definition of  $\mathcal{X}$ .

The current state of the navigation process is represented by the current Pareto optimal solution  $\mathbf{x}^h$  and the vector of current upper bounds  $\mathbf{b} \in \mathbf{R}^k$  on objective function values. Using the surrogate problem with these bounds as additional constraints, the ideal objective vector is calculated and the nadir objective vector is estimated via a pay-off table. They define ranges of objective function values for Pareto optimal solutions. These ranges together with the current solution are displayed in a radar chart also known as a spider-web chart.

By moving sliders on the radar chart with the mouse, the decision maker can provide two types of preference information: upper bounds on objective values and a desired value (aspiration level) of any objective function. Changes made by the decision maker are immediately reflected in the current state of the navigation process and shown in the radar chart. Setting the upper bounds influences the objective function ranges as described above. Setting the value of any objective function  $f_{i^*}$  to a desired value  $\tau$  yields updating the current solution with the solution of the following problem

$$\begin{aligned} \text{minimize} \quad & \max_{\substack{i=1, \dots, k, \\ i \neq i^*}} y_i - f_i(x^h) \\ \text{subject to} \quad & \mathbf{y} = f \left( \sum_{j=1}^m v_j \mathbf{x}^{(j)} \right) + \mathbf{s}, \\ & y_i \leq b_i, \quad i = 1, \dots, k, \\ & y_{i^*} = \tau, \\ & \sum_{j=1}^m v_j = 1, \\ & \mathbf{v} \text{ and } \mathbf{s} \text{ are non-negative.} \end{aligned}$$

By using the two above-described mechanisms of expressing preferences the decision maker explores the set of Pareto optimal solutions of the surrogate problem until a most preferred or satisfactory solution is found. Because the decision maker must provide upper bounds and aspiration levels, the method is ad hoc by nature.

The method has been developed and implemented for intensity modulated radiation therapy treatment planning. Therefore, in addition to the radar chart, some application-specific information about the current solution (treatment plan) is displayed. Nevertheless, there are no obstacles of adapting the method elsewhere when the multiobjective optimization problem is convex and the convex hull of some finite set of pre-calculated Pareto optimal solutions may serve as a good enough approximation of the Pareto optimal set.

## 8.4 Other Navigation Methods

Other navigation based methods developed for nonlinear multiobjective optimization problems, implemented as software tools include [122, 123]. A collection of methods and software for solving linear multiobjective optimization problems [5, 6] can also be mentioned as they can partly be extended to nonlinear problems.

## 9. Other Interactive Methods

The number of interactive methods developed for multiobjective optimization is large. So far, we have given several examples of them. Let us next mention references to some more methods based on miscellaneous ideas: [9, 29, 30, 31, 39, 46, 50, 54, 55, 78, 89, 94, 97, 100, 116, 117, 119, 132, 158, 162, 163, 176, 187, 188, 189, 190, 198, 199, 204, 212, 216, 217, 223, 226, 227, 229, 235, 248, 266, 267, 269].

## 10. Comparing the Methods

None of the many multiobjective optimization methods can be claimed to be superior to the others in every aspect. One can say that selecting a multiobjective optimization method is a problem with multiple objectives itself. The properties of the problem and the capabilities and the desires of the decision maker have to be charted before a solution method can be chosen. Some methods may suit some problems and some decision makers better than some others.

A decision tree is provided in [136] for easing the method selection. The tree is based on theoretical facts concerning the assumptions on the problem to be solved and the preferences of the decision maker. Further aspects to be taken into account when evaluating and selecting methods are collected, for example, in [13, 58, 74, 80, 136, 228, 239, 240].

In addition to theoretical properties, practical applicability, in particular, plays an important role in the selection of an appropriate method. The difficulty is that practical applicability is hard to determine without experience.

Some comparisons of the methods have been reported in the literature. They have been carried out with respect to a variety of criteria and under varied circumstances. Instead of a human decision maker, one can sometimes employ value functions in the comparisons. Unfortunately, replacing the decision maker with a value function does not fully reflect the real usefulness of the methods. One of the problems is that value functions cannot really help in testing ad hoc methods.

Tests with human decision makers are described in [16, 18, 20, 21, 33, 34, 41, 111, 134, 184, 247] while tests with value functions are reported in [3, 59, 161, 191]. Finally, comparisons based on intuition are provided in [45, 98, 99, 113, 131, 135, 185, 193, 207, 243, 246].

## 11. Conclusions

We have outlined several interactive methods for solving nonlinear multi-objective optimization problems and indicated references to many more. One of the challenges in this area is spreading the word about the existing methods to those who solve real-world problems. Another challenge is to develop methods that support the decision maker even better. User-friendliness cannot be overestimated because interactive methods must be able to correspond to the characteristics of the decision maker. Specific methods for different areas of application that take into account the characteristics of the problems are also important.

An alternative to creating new methods is to use different methods in different phases of the solution process. This hybridization means that the positive features of various methods can be exploited to their best advantage in appropriate phases. In this way, it may also be possible to overcome some of the weaknesses of the methods like proposed. Ways to enable changing the type of preference information specified, that is, the method used during the solution process are presented in [125, 201].

The decision maker can be supported by using visual illustrations and further development of such tools is essential. For instance, one may visualize (parts of) the Pareto optimal set and, for example, use 3D slices of the feasible objective region (see [122, 123], among others) and other tools. On the other hand, one can illustrate sets of alternatives by means of bar charts, value paths, spider-web charts and petal diagrams etc. For more details see, for example, [136] and references therein as well as [139] for a more detailed survey.

## References

- [1] IND-NIMBUS website. <http://ind-nimbus.it.jyu.fi/>.
- [2] P. J. Agrell, B. J. Lence, and A. Stam. An interactive multicriteria decision model for multipurpose reservoir management: The Shellmouth reservoir. *Journal of Multi-Criteria Decision Analysis*, 7:61–86, 1998.
- [3] Y. Aksoy, T. W. Butler, and E. D. Minor III. Comparative studies in interactive multiple objective mathematical programming. *European Journal of Operational Research*, 89:408–422, 1996.

- [4] J. E. Al-alvani, B. F. Hobbs, and B. Malakooti. An interactive integrated multiobjective optimization approach for quasiconcave/quasiconvex utility functions. In A. Goicoechea, L. Duckstein, and S. Zionts, editors, *Multiple Criteria Decision Making: Proceedings of the Ninth International Conference: Theory and Applications in Business, Industry, and Government*, pages 45–60, New York, 1992. Springer-Verlag.
- [5] C. H. Antunes, M. J. Alves, A. L. Silva, and J. N. Clímaco. An integrated MOLP method base package – a guided tour of TOM-MIX. *Computers & Operations Research*, 19:609–625, 1992.
- [6] C. H. Antunes, M. P. Melo, and J. N. Clímaco. On the integration of an interactive MOLP procedure base and expert system technique. *European Journal of Operational Research*, 61:135–144, 1992.
- [7] J.-P. Aubin and B. Näslund. An exterior branching algorithm. Working Paper 72-42, European Institute for Advanced Studies in Management, Brussels, 1972.
- [8] N. Baba, H. Takeda, and T. Miyake. Interactive multi-objective programming technique using random optimization method. *International Journal of Systems Science*, 19:151–159, 1988.
- [9] J. F. Bard. A multiobjective methodology for selecting subsystem automation options. *Management Science*, 32:1628–1641, 1986.
- [10] R. Benayoun, J. de Montgolfier, J. Tergny, and O. Laritchev. Linear programming with multiple objective functions: Step method (STEM). *Mathematical Programming*, 1:366–375, 1971.
- [11] H. P. Benson and Y. Aksoy. Using efficient feasible directions in interactive multiple objective linear programming. *Operations Research Letters*, 10:203–209, 1991.
- [12] E. Bischoff. Two empirical tests with approaches to multiple-criteria decision making. In M. Grauer, M. Thompson, and A. P. Wierzbicki, editors, *Plural Rationality and Interactive Decision Processes*, pages 344–347, Berlin, Heidelberg, 1985. Springer-Verlag.
- [13] E. Bischoff. Multi-objective decision analysis – the right objectives? In G. Fandel, M. Grauer, A. Kurzhanski, and A. P. Wierzbicki, editors, *Large-Scale Modelling and Interactive Decision Analysis*, pages 155–160. Springer-Verlag, 1986.
- [14] P. Bogetoft, Å. Hallefjord, and M. Kok. On the convergence of reference point methods in multiobjective programming. *European Journal of Operational Research*, 34:56–68, 1988.

- [15] J. Branke, K. Deb, K. Miettinen, and R. Slowinski, editors. *Multi-objective Optimization: Interactive and Evolutionary Approaches*. Springer-Verlag, Berlin, Heidelberg, 2008.
- [16] K. Brockhoff. Experimental test of MCDM algorithms in a modular approach. *European Journal of Operational Research*, 22:159–166, 1985.
- [17] J. T. Buchanan. Multiple objective mathematical programming: A review. *New Zealand Operational Research*, 14:1–27, 1986.
- [18] J. T. Buchanan. An experimental evaluation of interactive MCDM methods and the decision making process. *Journal of the Operational Research Society*, 45:1050–1059, 1994.
- [19] J. T. Buchanan. A naïve approach for solving MCDM problems: The GUESS method. *Journal of the Operational Research Society*, 48:202–206, 1997.
- [20] J. T. Buchanan and J. L. Corner. The effects of anchoring in interactive MCDM solution methods. *Computers & Operations Research*, 24:907–918, 1997.
- [21] J. T. Buchanan and H. G. Daellenbach. A comparative evaluation of interactive solution methods for multiple objective decision models. *European Journal of Operational Research*, 29:353–359, 1987.
- [22] Y. Censor. Pareto optimality in multiobjective problems. *Applied Mathematics and Optimization*, 4:41–59, 1977.
- [23] V. Chankong and Y. Y. Haimes. The interactive surrogate worth trade-off (ISWT) method for multiobjective decision-making. In S. Zionts, editor, *Multiple Criteria Problem Solving*, pages 42–67, Berlin, Heidelberg, 1978. Springer-Verlag.
- [24] V. Chankong and Y. Y. Haimes. *Multiobjective Decision Making Theory and Methodology*. Elsevier, New York, 1983.
- [25] A. Charnes and W. W. Cooper. *Management Models and Industrial Applications of Linear Programming*, volume 1. John Wiley & Sons, New York, 1961.
- [26] A. Charnes and W. W. Cooper. Goal programming and multiple objective optimization; part 1. *European Journal of Operational Research*, 1:39–54, 1977.
- [27] J. Chen and S. Lin. An interactive neural network-based approach for solving multiple criteria decision-making problems. *Decision Support Systems*, 36:137–146, 2003.
- [28] T. Chen and B.-I. Wang. An interactive method for multiobjective decision making. In A. Straszak, editor, *Large Scale Systems*:

*Theory and Applications 1983*, pages 277–282. Pergamon Press, 1984.

- [29] J. N. Clímaco and C. H. Antunes. Flexible method bases and man-machine interfaces as key features in interactive MOLP approaches. In P. Korhonen, A. Lewandowski, and J. Wallenius, editors, *Multiple Criteria Decision Support*, pages 207–216. Springer-Verlag, 1991.
- [30] J. N. Clímaco and C. H. Antunes. Man-machine interfacing in MCDA. In G. H. Tzeng, H. F. Wand, U. P. Wen, and P. L. Yu, editors, *Multiple Criteria Decision Making – Proceedings of the Tenth International Conference: Expand and Enrich the Domains of Thinking and Application*, pages 239–253, New York, 1994. Springer-Verlag.
- [31] J. N. Clímaco, C. H. Antunes, and M. J. Alves. From TRIMAP to SOMMIX – building effective interactive MOLP computational tools. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making: Proceedings of the Twelfth International Conference*, pages 285–296, Berlin, Heidelberg, 1997. Springer-Verlag.
- [32] H. W. Corley. A new scalar equivalence for Pareto optimization. *IEEE Transactions on Automatic Control*, 25:829–830, 1980.
- [33] J. L. Corner and J. T. Buchanan. Experimental consideration of preference in decision making under certainty. *Journal of Multi-Criteria Decision Analysis*, 4:107–121, 1995.
- [34] J. L. Corner and J. T. Buchanan. Capturing decision maker preference: Experimental comparison of decision analysis and MCDM techniques. *European Journal of Operational Research*, 98:85–97, 1997.
- [35] J. P. Costa and J. N. Clímaco. A multiple reference point parallel approach in MCDM. In G. H. Tzeng, H. F. Wand, U. P. Wen, and P. L. Yu, editors, *Multiple Criteria Decision Making – Proceedings of the Tenth International Conference: Expand and Enrich the Domains of Thinking and Application*, pages 255–263, New York, 1994. Springer-Verlag.
- [36] Y. Crama. Analysis of STEM-like solutions to multi-objective programming problems. In S. French, R. Hartley, L. C. Thomas, and D. J. White, editors, *Multi-Objective Decision Making*, pages 208–213. Academic Press, 1983.
- [37] K. Deb and K. Miettinen. Nadir point estimation using evolutionary approaches: Better accuracy and computational speed through focused search. In M. Ehrgott, B. Naujoks, T. J. Stewart, and

- J. Wallenius, editors, *Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems, Proceedings*, pages 339–354, Berlin, Heidelberg, 2010. Springer-Verlag.
- [38] K. Deb, K. Miettinen, and Chaudhuri S. Towards an estimation of nadir objective vector using a hybrid of evolutionary and local search approaches. *IEEE Transactions on Evolutionary Computation*, 14:821–841, 2010.
- [39] A. Diaz. Interactive solution to multiobjective optimization problems. *International Journal for Numerical Methods in Engineering*, 24:1865–1877, 1987.
- [40] J. S. Dyer. Interactive goal programming. *Management Science*, 19:62–70, 1972.
- [41] J. S. Dyer. An empirical investigation of a man-machine interactive approach to the solution of the multiple criteria problem. In J. L. Cochrane and M. Zeleny, editors, *Multiple Criteria Decision Making*, pages 202–216, Columbia, South Carolina, 1973. University of South Carolina Press.
- [42] J. S. Dyer. A time-sharing computer program for the solution of the multiple criteria problem. *Management Science*, 19:1379–1383, 1973.
- [43] J. S. Dyer and R. K. Sarin. Multicriteria decision making. In A. G. Holzman, editor, *Mathematical Programming for Operations Researchers and Computer Scientists*, pages 123–148. Marcel Dekker, 1981.
- [44] M. Ehrgott. *Multicriteria Optimization*. Springer-Verlag, Berlin, Heidelberg, 2000.
- [45] H. A. Eschenauer, A. Osyczka, and E. Schäfer. Interactive multi-criteria optimization in design process. In H. Eschenauer, J. Koski, and A. Osyczka, editors, *Multicriteria Design Optimization Procedures and Applications*, pages 71–114, Berlin, Heidelberg, 1990. Springer-Verlag.
- [46] H. A. Eschenauer, E. Schäfer, and H. Bernau. Application of interactive vector optimization methods with regard to problems in structural mechanics. In H. A. Eschenauer and G. Thierauf, editors, *Discretization Methods and Structural Optimization – Procedures and Applications*, pages 95–101, Berlin, Heidelberg, 1989. Springer-Verlag.
- [47] P. Eskelinen and K. Miettinen. Trade-off analysis approach for interactive nonlinear multiobjective optimization. *OR Spectrum*, 34:803–816, 2012.



- [48] P. Eskelinen, K. Miettinen, K. Klamroth, and J. Hakanen. Pareto Navigator for interactive nonlinear multiobjective optimization. *OR Spectrum*, 32:211–227, 2010.
- [49] J. Feng. Dual worth trade-off method and its application for solving multiple criteria decision making problems. *Journal of Systems Engineering and Electronics*, 17:554–558, 2006.
- [50] P. A. V. Ferreira and J. C. Geromel. An interactive projection method for multicriteria optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics*, 20:596–605, 1990.
- [51] P. A. V. Ferreira and M. E. S. Machado. Solving multiple-objective problems in the objective space. *Journal of Optimization Theory and Applications*, 89:659–680, 1996.
- [52] P. C. Fishburn. Lexicographic orders, utilities and decision rules: A survey. *Management Science*, 20:1442–1471, 1974.
- [53] T. L. Friesz. Multiobjective optimization in transportation: The case of equilibrium network design. In N. N. Morse, editor, *Organizations: Multiple Agents with Multiple Criteria*, pages 116–127, Berlin, Heidelberg, 1981. Springer-Verlag.
- [54] L. R. Gardiner and R. E. Steuer. Unified interactive multiple objective programming. *European Journal of Operational Research*, 74:391–406, 1994.
- [55] L. R. Gardiner and R. E. Steuer. Unified interactive multiple objective programming: An open architecture for accommodating new procedures. *Journal of the Operational Research Society*, 45:1456–1466, 1994.
- [56] S. Gass and T. Saaty. The computational algorithm for the parametric objective function. *Naval Research Logistics Quarterly*, 2:39–45, 1955.
- [57] A. M. Geoffrion, J. S. Dyer, and A. Feinberg. An interactive approach for multi-criterion optimization, with an application to the operation of an academic department. *Management Science*, 19:357–368, 1972.
- [58] M. Gershon and L. Duckstein. An algorithm for choosing of a multiobjective technique. In P. Hansen, editor, *Essays and Surveys on Multiple Criteria Decision Making*, pages 53–62, Berlin, Heidelberg, 1983. Springer-Verlag.
- [59] M. Gibson, J. J. Bernardo, C. Chung, and R. Badinelli. A comparison of interactive multiple-objective decision making procedures. *Computers & Operations Research*, 14:97–105, 1987.

- [60] V. G. Gouljashki, L. M. Kirilov, S. C. Narula, and V. S. Vassilev. A reference direction interactive algorithm of the multiple objective nonlinear integer programming. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making: Proceedings of the Twelfth International Conference*, pages 308–317, Berlin, Heidelberg, 1997. Springer-Verlag.
- [61] J. Granat and M. Makowski. Interactive specification and analysis of aspiration-based preferences. *European Journal of Operational Research*, 122:469–485, 2000.
- [62] M. Grauer, A. Lewandowski, and A. Wierzbicki. DIDASS – theory, implementation and experiences. In M. Grauer and A. P. Wierzbicki, editors, *Interactive Decision Analysis*, pages 22–30. Springer-Verlag, 1984.
- [63] Y. Y. Haimes. The surrogate worth trade-off (SWT) method and its extensions. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making Theory and Application*, pages 85–108, Berlin, Heidelberg, 1980. Springer-Verlag.
- [64] Y. Y. Haimes, L. S. Lasdon, and D. A. Wismer. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, 1:296–297, 1971.
- [65] Y. Y. Haimes, K. Tarvainen, T. Shima, and J. Thadathil. *Hierarchical Multiobjective Analysis of Large-Scale Systems*. Hemisphere Publishing Corporation, New York, 1990.
- [66] J. Hakanen, Y. Kawajiri, K. Miettinen, and L. T. Biegler. Interactive multi-objective optimization for simulated moving bed processes. *Control and Cybernetics*, 36:282–320, 2007.
- [67] J. Hakanen, K. Miettinen, and K. Sahlstedt. Wastewater treatment: New insight provided by interactive multiobjective optimization. *Decision Support Systems*, 51:328–337, 2011.
- [68] J. Hakanen, K. Sahlstedt, and K. Miettinen. Wastewater treatment plant design and operation under multiple conflicting objective functions. *Environmental Modelling & Software*, 46:240–249, 2013.
- [69] W. A. Hall and Y. Y. Haimes. The surrogate worth trade-off method with multiple decision-makers. In M. Zeleny, editor, *Multiple Criteria Decision Making Kyoto 1975*, pages 207–233, Berlin, Heidelberg, 1976. Springer-Verlag.
- [70] Å. Hallefjord and K. Jörnsten. An entropy target-point approach to multiobjective programming. *International Journal of Systems Science*, 17:639–653, 1986.

- [71] J. P. Hämäläinen, K. Miettinen, P. Tarvainen, and J. Toivanen. Interactive solution approach to a multiobjective optimization problem in a paper machine headbox design. *Journal of Optimization Theory and Applications*, 116:265–281, 2003.
- [72] E. Heikkola, K. Miettinen, and P. Nieminen. Multiobjective optimization of an ultrasonic transducer using NIMBUS. *Ultrasonics*, 44:368–380, 2006.
- [73] T. Hemming. Some modifications of a large step gradient method for interactive multicriterion optimization. In N. N. Morse, editor, *Organizations: Multiple Agents with Multiple Criteria*, pages 128–139, Berlin, Heidelberg, 1981. Springer-Verlag.
- [74] B. F. Hobbs. What can we learn from experiments in multiobjective decision analysis? *IEEE Transactions on Systems, Man, and Cybernetics*, 16:384–394, 1986.
- [75] C. Hu, Y. Shen, and S. Li. An interactive satisficing method based on alternative tolerance for fuzzy multiple objective optimization. *Applied Mathematical Modelling*, 33:1886–1893, 2009.
- [76] H. Huang and Z. Tian. Application of neural network to interactive physical programming. In J. Wang, X. Liao, and Z. Yi, editors, *Advances in Neural Networks*, pages 725–730. Springer-Verlag, Berlin, Heidelberg, 2005.
- [77] H.-Z. Huang, Y.-K. Gu, and X. Du. An interactive fuzzy multi-objective optimization method for engineering design. *Engineering Applications of Artificial Intelligence*, 19:451–460, 2006.
- [78] M. L. Hussein and F. S. A. El-Ghaffar. An interactive approach for vector optimization problems. *European Journal of Operational Research*, 89:185–192, 1996.
- [79] C.-L. Hwang and A. S. M. Masud. *Multiple Objective Decision Making – Methods and Applications: A State-of-the-Art Survey*. Springer-Verlag, Berlin, Heidelberg, 1979.
- [80] C.-L. Hwang and K. Yoon. *Multiple Attribute Decision Making Methods and Applications: A State-of-the-Art Survey*. Springer-Verlag, Berlin, Heidelberg, 1981.
- [81] J. P. Ignizio. *Goal Programming and Extensions*. Lexington Books. D.C. Heath and Company, 1976.
- [82] A. Jaszkievicz and R. Slowiński. The light beam search – outranking based interactive procedure for multiple-objective mathematical programming. In P. M. Pardalos, Y. Siskos, and C. Zopounidis, editors, *Advances in Multicriteria Analysis*, pages 129–146, Dordrecht, 1995. Kluwer Academic Publishers.

- [83] A. Jaskiewicz and R. Słowiński. The ‘light beam search’ approach – an overview of methodology and applications. *European Journal of Operational Research*, 113:300–314, 1999.
- [84] P. Jedrzejowicz and L. Rosicka. Multicriterial reliability optimization problem. *Foundations of Control Engineering*, 8:165–173, 1983.
- [85] I.-J. Jeong and K.-J. Kim. D-STEM: A modified Step method with desirability function concept. *Computers & Operations Research*, 32:3175–3190, 2005.
- [86] D. Kahneman and A. Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47:263–291, 1979.
- [87] I. Kaliszewski. A modified weighted Tchebycheff metric for multiple objective programming. *Computers & Operations Research*, 14:315–323, 1987.
- [88] I. Kaliszewski. Using trade-off information in decision-making algorithms. *Computers & Operations Research*, 27:161–182, 2000.
- [89] I. Kaliszewski and W. Michalowski. Searching for psychologically stable solutions of multiple criteria decision problems. *European Journal of Operational Research*, 118:549–562, 1999.
- [90] I. Kaliszewski, W. Michalowski, and G. Kersten. A hybrid interactive technique for the MCDM problems. In M. H. Karwan, J. Spronk, and J. Wallenius, editors, *Essays in Decision Making: A Volume in Honour of Stanley Zionts*, pages 48–59. Springer-Verlag, Berlin, Heidelberg, 1997.
- [91] I. Kaliszewski and S. Zionts. Generalization of the Zionts-Wallenius algorithm. *Control and Cybernetics*, 33:477–500, 2004.
- [92] R. L. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. John Wiley & Sons, 1976.
- [93] J. Kim and S.-K. Kim. A CHIM-based interactive Tchebycheff procedure for multiple objective decision making. *Computers & Operations Research*, 33:1557–1574, 2006.
- [94] S. H. Kim and T. Gal. A new interactive algorithm for multi-objective linear programming using maximally changeable dominance cone. *European Journal of Operational Research*, 64:126–137, 1993.
- [95] S. Kitayama and K. Yamazaki. Compromise point incorporating trade-off ratio in multi-objective optimization. *Applied Soft Computing*, 12:1959–1964, 2012.

- [96] K. Klamroth and K. Miettinen. Integrating approximation and interactive decision making in multicriteria optimization. *Operations Research*, 56:222–234, 2008.
- [97] L. Y. Klepper. An interactive method for optimization of irradiation plans in radiation therapy of malignant tumors. *Biomedical Engineering*, 40:291–297, 2006.
- [98] M. Kok. Scalarization and the interface with decision makers in interactive multi objective linear programming. In P. Serafini, editor, *Mathematics of Multi Objective Optimization*, pages 433–438, Wien, New York, 1985. Springer-Verlag.
- [99] M. Kok. The interface with decision makers and some experimental results in interactive multiple objective programming method. *European Journal of Operational Research*, 26:96–107, 1986.
- [100] M. Kok and F. A. Lootsma. Pairwise-comparison methods in multiple objective programming, with applications in a long-term energy-planning model. *European Journal of Operational Research*, 22:44–55, 1985.
- [101] M. Köksalan and R. D. Plante. Interactive multicriteria optimization for multiple-response product and process design. *Manufacturing & Service Operations Management*, 5:334–347, 2003.
- [102] M. M. Köksalan and H. Moskowitz. Solving the multiobjective decision making problem using a distance function. In G. H. Tzeng, H. F. Wand, U. P. Wen, and P. L. Yu, editors, *Multiple Criteria Decision Making – Proceedings of the Tenth International Conference: Expand and Enrich the Domains of Thinking and Application*, pages 101–107, New York, 1994. Springer-Verlag.
- [103] P. Korhonen. Reference direction approach to multiple objective linear programming: Historical overview. In M. H. Karwan, J. Spronk, and J. Wallenius, editors, *Essays in Decision Making: A Volume in Honour of Stanley Zionts*, pages 74–92. Springer-Verlag, Berlin, Heidelberg, 1997.
- [104] P. Korhonen and M. Halme. Using lexicographic parametric programming for searching a nondominated set in multiple objective linear programming. *Journal of Multi-Criteria Decision Analysis*, 5:291–300, 1996.
- [105] P. Korhonen and J. Laakso. A visual interactive method for solving the multiple-criteria problem. In M. Grauer and A. P. Wierzbicki, editors, *Interactive Decision Analysis*, pages 146–153. Springer-Verlag, 1984.

- [106] P. Korhonen and J. Laakso. On developing a visual interactive multiple criteria method – an outline. In Y. Y. Haimes and V. Chankong, editors, *Decision Making with Multiple Objectives*, pages 272–281, Berlin, Heidelberg, 1985. Springer-Verlag.
- [107] P. Korhonen and J. Laakso. Solving generalized goal programming problems using a visual interactive approach. *European Journal of Operational Research*, 26:355–363, 1986.
- [108] P. Korhonen and J. Laakso. A visual interactive method for solving the multiple criteria problem. *European Journal of Operational Research*, 24:277–287, 1986.
- [109] P. Korhonen and S. C. Narula. An evolutionary approach to support decision making with linear decision models. *Journal of Multi-Criteria Decision Analysis*, 2:111–119, 1993.
- [110] P. Korhonen and J. Wallenius. A Pareto race. *Naval Research Logistics*, 35:615–623, 1988.
- [111] P. Korhonen and J. Wallenius. Observations regarding choice behaviour in interactive multiple criteria decision-making environments: An experimental investigation. In A. Lewandowski and I. Stanchev, editors, *Methodology and Software for Interactive Decision Support*, pages 163–170. Springer-Verlag, 1989.
- [112] O. Larichev. Cognitive validity in design of decision-aiding techniques. *Journal of Multi-Criteria Decision Analysis*, 1:127–138, 1992.
- [113] O. I. Larichev, O. A. Polyakov, and A. O. Nikiforov. Multicriterion linear programming problems. *Journal of Economic Psychology*, 8:389–407, 1987.
- [114] T. Laukkanen, T.-M. Tveit, V. Ojalehto, K. Miettinen, and C.-J. Fogelholm. An interactive multi-objective approach to heat exchanger network synthesis. *Computers and Chemical Engineering*, 34:943–952, 2010.
- [115] T. Laukkanen, T.-M. Tveit, V. Ojalehto, K. Miettinen, and C.-J. Fogelholm. Bilevel heat exchanger network synthesis with an interactive multi-objective optimization method. *Applied Thermal Engineering*, 48:301–316, 2012.
- [116] R. Lazimy. Interactive relaxation method for a broad class of integer and continuous nonlinear multiple criteria problems. *Journal of Mathematical Analysis and Applications*, 116:553–573, 1986.
- [117] R. Lazimy. Solving multiple criteria problems by interactive decomposition. *Mathematical Programming*, 35:334–361, 1986.

- [118] R. Lazimy. Interactive polyhedral outer approximation (IPOA) strategy for general multiobjective optimization problems. *Annals of Operations Research*, 210:73–99, 2012.
- [119] E. R. Lieberman. Soviet multi-objective mathematical programming methods: An overview. *Management Science*, 37:1147–1165, 1991.
- [120] G. V. Loganathan and H. D. Sherali. A convergent interactive cutting-plane algorithm for multiobjective optimization. *Operation Research*, 35:365–377, 1987.
- [121] A. V. Lotov. Computer-based support for planning and negotiation on environmental rehabilitation of water resource systems. In D. P. Loucks, editor, *Restoration of Degraded Rivers: Challenges, Issues and Experiences*, pages 417–445. Kluwer Academic Publishers, 1998.
- [122] A. V. Lotov, V. A. Bushenkov, and O. L. Chernykh. Multicriteria DSS for river water-quality planning. *Microcomputers in Civil Engineering*, 12:57–67, 1997.
- [123] A. V. Lotov, V. A. Bushenkov, and G. K. Kamenev. *Interactive Decision Maps: Approximation and Visualization of Pareto Frontier*. Kluwer Academic Publishers, Norwell, 2004.
- [124] M. Luque, K. Miettinen, P. Eskelinen, and F. Ruiz. Incorporating preference information in interactive reference point methods for multiobjective optimization. *Omega*, 37:450–462, 2009.
- [125] M. Luque, F. Ruiz, and K. Miettinen. Global formulation for interactive multiobjective optimization. *OR Spectrum*, 33:27–48, 2011.
- [126] M. Luque, F. Ruiz, and R. E. Steuer. Modified interactive Chebyshev algorithm (MICA) for convex multiobjective programming. *European Journal of Operational Research*, 204:557–564, 2010.
- [127] P. Mackin, A. Roy, and J. Wallenius. An interactive weight space reduction procedure for nonlinear multiple objective mathematical programming. *Mathematical Programming*, 127:425–444, 2011.
- [128] B. Malakooti and V. Raman. An interactive multi-objective artificial neural network approach for machine setup optimization. *Journal of Intelligent Manufacturing*, 11:41–50, 2000.
- [129] R.T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26:369–395, 2004.

- [130] A. S. M. Masud and C. L. Hwang. Interactive sequential goal programming. *Journal of the Operational Research Society*, 32:391–400, 1981.
- [131] A. S. M. Masud and X. Zheng. An algorithm for multiple-objective non-linear programming. *Journal of the Operational Research Society*, 40:895–906, 1989.
- [132] Z. Meng, R. Shen, and M. Jiang. An objective penalty functions algorithm for multiobjective optimization problem. *American Journal of Operations Research*, 1:229–235, 2011.
- [133] A. Messac. Physical programming: Effective optimization for computational design. *AIAA Journal*, 34:149–158, 1996.
- [134] W. Michalowski. Evaluation of a multiple criteria interactive programming approach: An experiment. *INFOR: Information Systems & Operational Research*, 25:165–173, 1987.
- [135] W. Michalowski and T. Szapiro. A bi-reference procedure for interactive multiple criteria programming. *Operations Research*, 40:247–258, 1992.
- [136] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, 1999.
- [137] K. Miettinen. IND-NIMBUS for demanding interactive multiobjective optimization. In T. Trzaskalik, editor, *Multiple Criteria Decision Making '05*, pages 137–150, Katowice, 2006. The Karol Adamiecki University of Economics in Katowice.
- [138] K. Miettinen. Using interactive multiobjective optimization in continuous casting of steel. *Materials and Manufacturing Processes*, 22:585–593, 2007.
- [139] K. Miettinen. Survey of methods to visualize alternatives in multiple criteria decision making problems. *OR Spectrum*, 36:3–37, 2014.
- [140] K. Miettinen, P. Eskelinen, F. Ruiz, and M. Luque. NAUTILUS method: An interactive technique in multiobjective optimization based on the nadir point. *European Journal of Operational Research*, 206:426–434, 2010.
- [141] K. Miettinen and J. Hakanen. Why use interactive multi-objective optimization in chemical process design? In G. P. Rangaiah, editor, *Multi-Objective Optimization Techniques and Applications in Chemical Engineering*, pages 148–183. World Scientific Publishers, Singapore, 2009.
- [142] K. Miettinen, A. V. Lotov, G. K. Kamenev, and V. E. Berezkin. Integration of two multiobjective optimization methods for non-



- linear problems. *Optimization Methods and Software*, 18:63–80, 2003.
- [143] K. Miettinen and M. M. Mäkelä. An interactive method for non-smooth multiobjective optimization with an application to optimal control. *Optimization Methods and Software*, 2:31–44, 1993.
- [144] K. Miettinen and M. M. Mäkelä. A nondifferentiable multiple criteria optimization method applied to continuous casting process. In A. Fasano and M. Primicerio, editors, *Proceedings of the Seventh European Conference on Mathematics in Industry*, pages 255–262, Stuttgart, 1994. B. G. Teubner.
- [145] K. Miettinen and M. M. Mäkelä. Interactive bundle-based method for nondifferentiable multiobjective optimization: NIMBUS. *Optimization*, 34:231–246, 1995.
- [146] K. Miettinen and M. M. Mäkelä. Comparative evaluation of some interactive reference point-based methods for multi-objective optimisation. *Journal of the Operational Research Society*, 50:949–959, 1999.
- [147] K. Miettinen and M. M. Mäkelä. Interactive multiobjective optimization system WWW-NIMBUS on the Internet. *Computers & Operations Research*, 27:709–723, 2000.
- [148] K. Miettinen and M. M. Mäkelä. On scalarizing functions in multiobjective optimization. *OR Spectrum*, 24:193–213, 2002.
- [149] K. Miettinen and M. M. Mäkelä. Synchronous approach in interactive multiobjective optimization. *European Journal of Operational Research*, 170:909–922, 2006.
- [150] K. Miettinen, M. M. Mäkelä, and K. Kaario. Experiments with classification-based scalarizing functions in interactive multiobjective optimization. *European Journal of Operational Research*, 175:931–947, 2006.
- [151] K. Miettinen, M. M. Mäkelä, and T. Männikkö. Optimal control of continuous casting by nondifferentiable multiobjective optimization. *Computational Optimization and Applications*, 11:177–194, 1998.
- [152] K. Miettinen, J. Mustajoki, and T. J. Stewart. Interactive multiobjective optimization with NIMBUS for decision making under uncertainty. *OR Spectrum*, 36:39–56, 2014.
- [153] K. Miettinen, F. Ruiz, and A. P. Wierzbicki. Introduction to multiobjective optimization: Interactive approaches. In J. Branke, K. Deb, K. Miettinen, and R. Slowinski, editors, *Multiobjective Op-*

- timization: Interactive and Evolutionary Approaches*, pages 27–57. Springer-Verlag, Berlin, Heidelberg, 2008.
- [154] K. Mitani and H. Nakayama. A multiobjective diet planning support system using the satisficing trade-off method. *Journal of Multi-Criteria Decision Analysis*, 6:131–139, 1997.
- [155] U. Mocci and L. Primicerio. Ring network design: An MCDM approach. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making: Proceedings of the Twelfth International Conference*, pages 491–500, Berlin, Heidelberg, 1997. Springer-Verlag.
- [156] C. Mohan and H. T. Nguyen. Reference direction interactive method for solving multiobjective fuzzy programming problems. *European Journal of Operational Research*, 107:599–613, 1998.
- [157] C. Mohan and H. T. Nguyen. An interactive satisficing method for solving multiobjective mixed fuzzy-stochastic programming problems. *Fuzzy Sets and Systems*, 117:61–79, 2001.
- [158] M. A. Moldavskiy. Singling out a set of undominated solutions in continuous vector optimization problems. *Soviet Automatic Control*, 14:47–53, 1981.
- [159] D. E. Monarchi, C. C. Kisiel, and L. Duckstein. Interactive multiobjective programming in water resources: A case study. *Water Resources Research*, 9:837–850, 1973.
- [160] M. Monz, K. H. Kufer, T. R. Bortfeld, and C. Thieke. Pareto navigation – algorithmic foundation of interactive multi-criteria IMRT planning. *Physics in Medicine and Biology*, 53:985–998, 2008.
- [161] J. Mote, D. L. Olson, and M. A. Venkataramanan. A comparative multiobjective programming study. *Mathematical and Computer Modelling*, 10:719–729, 1988.
- [162] A. M’silti and P. Tolla. An interactive multiobjective nonlinear programming procedure. *European Journal of Operational Research*, 64:115–125, 1993.
- [163] H. Mukai. Algorithms for multicriterion optimization. *IEEE Transactions on Automatic Control*, 25:177–186, 1980.
- [164] K. Musselman and J. Talavage. A tradeoff cut approach to multiple objective optimization. *Operations Research*, 28:1424–1435, 1980.
- [165] H. Nakayama. On the components in interactive multiobjective programming methods. In M. Grauer, M. Thompson, and A. P. Wierzbicki, editors, *Plural Rationality and Interactive Decision Processes*, pages 234–247, Berlin, Heidelberg, 1985. Springer-Verlag.

- [166] H. Nakayama. Sensitivity and trade-off analysis in multiobjective programming. In A. Lewandowski and I. Stanchev, editors, *Methodology and Software for Interactive Decision Support*, pages 86–93. Springer-Verlag, 1989.
- [167] H. Nakayama. Satisficing trade-off method for problems with multiple linear fractional objectives and its applications. In A. Lewandowski and V. Volkovich, editors, *Multiobjective Problems of Mathematical Programming*, pages 42–50. Springer-Verlag, 1991.
- [168] H. Nakayama. Trade-off analysis based upon parametric optimization. In P. Korhonen, A. Lewandowski, and J. Wallenius, editors, *Multiple Criteria Decision Support*, pages 42–52. Springer-Verlag, 1991.
- [169] H. Nakayama. Trade-off analysis using parametric optimization techniques. *European Journal of Operational Research*, 60:87–98, 1992.
- [170] H. Nakayama. Engineering applications of multi-objective programming: Recent results. In G. H. Tzeng, H. F. Wand, U. P. Wen, and P. L. Yu, editors, *Multiple Criteria Decision Making – Proceedings of the Tenth International Conference: Expand and Enrich the Domains of Thinking and Application*, pages 369–378, New York, 1994. Springer-Verlag.
- [171] H. Nakayama. Aspiration level approach to interactive multi-objective programming and its applications. In P. M. Pardalos, Y. Siskos, and C. Zopounidis, editors, *Advances in Multicriteria Analysis*, pages 147–174, Dordrecht, 1995. Kluwer Academic Publishers.
- [172] H. Nakayama and K. Furukawa. Satisficing trade-off method with an application to multiobjective structural design. *Large Scale Systems*, 8:47–57, 1985.
- [173] H. Nakayama, K. Kaneshige, S. Takemoto, and Y. Watada. An application of a multi-objective programming technique to construction accuracy control of cable-stayed bridges. *European Journal of Operational Research*, 87:731–738, 1995.
- [174] H. Nakayama, J. Nomura, K. Sawada, and R. Nakajima. An application of satisficing trade-off method to a blending problem of industrial materials. In G. Fandel, M. Grauer, A. Kurzhanski, and A. P. Wierzbicki, editors, *Large-Scale Modelling and Interactive Decision Analysis*, pages 303–313. Springer-Verlag, 1986.

- [175] H. Nakayama and Y. Sawaragi. Satisficing trade-off method for multiobjective programming. In M. Grauer and A. P. Wierzbicki, editors, *Interactive Decision Analysis*, pages 113–122. Springer-Verlag, 1984.
- [176] H. Nakayama, T. Tanino, and Y. Sawaragi. An interactive optimization method in multicriteria decisionmaking. *IEEE Transactions on Systems, Man, and Cybernetics*, 10:163–169, 1980.
- [177] S. C. Narula, L. Kirilov, and V. Vassilev. An interactive algorithm for solving multiple objective nonlinear programming problems. In G. H. Tzeng, H. F. Wand, U. P. Wen, and P. L. Yu, editors, *Multiple Criteria Decision Making – Proceedings of the Tenth International Conference: Expand and Enrich the Domains of Thinking and Application*, pages 119–127, New York, 1994. Springer-Verlag.
- [178] S. C. Narula, L. Kirilov, and V. Vassilev. Reference direction approach for solving multiple objective nonlinear programming problems. *IEEE Transactions on Systems, Man, and Cybernetics*, 24:804–806, 1994.
- [179] S. C. Narula and H. R. Weistroffer. Algorithms for multi-objective nonlinear programming problems: An overview. In A. G. Lockett and G. Islei, editors, *Improving Decision Making in Organisations*, pages 434–443, Berlin, Heidelberg, 1989. Springer-Verlag.
- [180] S. C. Narula and H. R. Weistroffer. A flexible method for nonlinear multicriteria decisionmaking problems. *IEEE Transactions on Systems, Man, and Cybernetics*, 19:883–887, 1989.
- [181] P. Nijkamp and J. Spronk. Interactive multiple goal programming: An evaluation and some results. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making Theory and Applications*, pages 278–293, Berlin, Heidelberg, 1980. Springer-Verlag.
- [182] W. Ogryczak. Preemptive reference point method. In J. Clímaco, editor, *Multicriteria Analysis*, pages 156–167, Berlin, Heidelberg, 1997. Springer-Verlag.
- [183] M. Olbrisch. The interactive reference point approach as a solution concept for econometric decision models. In M. J. Beckmann, K.-W. Gaede, K. Ritter, and H. Schneeweiss, editors, *X. Symposium on Operations Research, Part 1, Sections 1-5*, pages 611–619, Knigstein, 1986. Verlag Anton Hain Meisenheim GmbH.
- [184] D. L. Olson. Review of empirical studies in multiobjective mathematical programming: Subject reflection of nonlinear utility and learning. *Decision Sciences*, 23:1–20, 1992.

- [185] D. L. Olson. Tchebycheff norms in multi-objective linear programming. *Mathematical and Computer Modelling*, 17:113–124, 1993.
- [186] K. R. Oppenheimer. A proxy approach to multi-attribute decision making. *Management Science*, 24:675–689, 1978.
- [187] A. Osyczka and S. Kundu. A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm. *Structural Optimization*, 10:94–99, 1995.
- [188] R. Ramesh, M. H. Karwan, and S. Zionts. Theory of convex cones in multicriteria decision making. *Annals of Operations Research*, 16:131–147, 1988.
- [189] R. Ramesh, M. H. Karwan, and S. Zionts. Interactive multicriteria linear programming: An extension of the method of Zionts and Wallenius. *Naval Research Logistics*, 36:321–335, 1989.
- [190] R. Ramesh, M. H. Karwan, and S. Zionts. Preference structure representation using convex cones in multicriteria integer programming. *Management Science*, 35:1092–1105, 1989.
- [191] G. R. Reeves and J. J. Gonzalez. A comparison of two interactive MCDM procedures. *European Journal of Operational Research*, 41:203–209, 1989.
- [192] G. R. Reeves and K. R. MacLeod. Robustness of the interactive weighted Tchebycheff procedure to inaccurate preference information. *Journal of Multi-Criteria Decision Analysis*, 8:128–132, 1999.
- [193] P. Rietveld. *Multiple Objective Decision Methods and Regional Planning*. North-Holland Publishing Company, 1980.
- [194] C. Romero. *Handbook of Critical Issues in Goal Programming*. Pergamon Press, 1991.
- [195] C. Romero. Extended lexicographic goal programming: A unifying approach. *Omega*, 29:63–71, 2001.
- [196] R. E. Rosenthal. Principles of multiobjective optimization. *Decision Sciences*, 16:133–152, 1985.
- [197] E. E. Rosinger. Interactive algorithm for multiobjective optimization. *Journal of Optimization Theory and Applications*, 35:339–365, 1981. Errata Corrige in *Journal of Optimization Theory and Applications*, 38:147–148, 1982.
- [198] A. Roy and P. Mackin. Multicriteria optimization (linear and nonlinear) using proxy value functions. In P. Korhonen, A. Lewandowski, and J. Wallenius, editors, *Multiple Criteria Decision Support*, pages 128–134. Springer-Verlag, 1991.

- [199] A. Roy and J. Wallenius. Nonlinear multiobjective optimization: An algorithm and some theory. *Mathematical Programming*, 55:235–249, 1992.
- [200] B. Roy. The outranking approach and the foundations of ELECTRE methods. In C. A. Bana e Costa, editor, *Readings in Multiple Criteria Decision Aid*, pages 155–183, Berlin, Heidelberg, 1990. Springer-Verlag.
- [201] F. Ruiz, M. Luque, and K. Miettinen. Improving the computational efficiency of a global formulation (GLIDE) for interactive multiobjective optimization. *Annals of Operations Research*, 197:47–70, 2012.
- [202] H. Ruotsalainen, K. Miettinen, J.-E. Palmgren, and T. Lahtinen. Interactive multiobjective optimization for anatomy based three-dimensional HDR brachytherapy. *Physics in Medicine and Biology*, 55:4703–4719, 2010.
- [203] S. Sadagopan and A. Ravindran. Interactive algorithms for multiple criteria nonlinear programming problems. *European Journal of Operational Research*, 25:247–257, 1986.
- [204] M. Sakawa. Interactive multiobjective decision making by the sequential proxy optimization technique: SPOT. *European Journal of Operational Research*, 9:386–396, 1982.
- [205] M. Sakawa and K. Yauchi. An interactive fuzzy satisficing method for multiobjective nonconvex programming problems through floating point genetic algorithms. *European Journal of Operational Research*, 117:113–124, 1999.
- [206] Y. Sawaragi, H. Nakayama, and T. Tanino. *Theory of Multiobjective Optimization*. Academic Press, Orlando, Florida, 1985.
- [207] W. S. Shin and A. Ravindran. Interactive multiple objective optimization: Survey I – continuous case. *Computers & Operations Research*, 18:97–114, 1991.
- [208] J. Silverman, R. E. Steuer, and A. W. Whisman. A multi-period, multiple criteria optimization system for manpower planning. *European Journal of Operational Research*, 34:160–170, 1988.
- [209] K. Sindhya, V. Ojalehto, J. Savolainen, H. Niemistö, J. Hakanen, and K. Miettinen. Coupling dynamic simulation and interactive multiobjective optimization for complex problems: An APROS-NIMBUS case study. *Expert Systems with Applications*, 41:2546–2558, 2014.
- [210] K. Sindhya, A. B. Ruiz, and K. Miettinen. A preference based interactive evolutionary algorithm for multiobjective optimization:

- PIE. In R.H.C. Takahashi, K. Deb, E.F. Wanner, and S. Greco, editors, *Evolutionary Multi-Criterion Optimization: 6th International Conference, Proceedings*, pages 212–225, Berlin, Heidelberg, 2011. Springer-Verlag.
- [211] A. M. J. Skulimowski. *Decision Support Systems Based on Reference Sets*. Wydawnictwa AGH, Kraków, 1996.
- [212] R. Slowinski. Interactive multiobjective optimization based on ordinal regression. In A. Lewandowski and V. Volkovich, editors, *Multiobjective Problems of Mathematical Programming*, pages 93–100. Springer-Verlag, 1991.
- [213] A. Song and W.-M. Cheng. A method for multihuman and multicriteria decision making. In A. Sydow, S. G. Tzafestas, and R. Vichnevetsky, editors, *Systems Analysis and Simulation 1988 I: Theory and Foundations*, pages 213–216, Berlin, 1988. Akademie-Verlag.
- [214] J. Spronk. Interactive multifactorial planning: State of the art. In C. A. Bana e Costa, editor, *Readings in Multiple Criteria Decision Aid*, pages 512–534, Berlin, Heidelberg, 1990. Springer-Verlag.
- [215] A. Stam, M. Kuula, and H. Cesar. Transboundary air pollution in Europe: An interactive multicriteria tradeoff analysis. *European Journal of Operational Research*, 56:263–277, 1992.
- [216] R. B. Statnikov. *Multicriteria Design: Optimization and Identification*. Kluwer Academic Publishers, Dordrecht, 1999.
- [217] R. B. Statnikov and J. Matusov. Use of  $p_r$ -nets for the approximation of the Edgeworth-Pareto set in multicriteria optimization. *Journal of Optimization Theory and Applications*, 91:543–560, 1996.
- [218] I. Steponavice, S. Ruuska, and K. Miettinen. A solution process for simulation-based multiobjective design optimization with an application in paper industry. *Computer-Aided Design*, 47:45–58, 2014.
- [219] R. E. Steuer. *Multiple Criteria Optimization: Theory, Computation, and Applications*. John Wiley & Sons, 1986.
- [220] R. E. Steuer. The Tchebycheff procedure of interactive multiple objective programming. In B. Karpak and S. Zionts, editors, *Multiple Criteria Decision Making and Risk Analysis Using Microcomputers*, pages 235–249, Berlin, Heidelberg, 1989. Springer-Verlag.
- [221] R. E. Steuer. Implementing the Tchebycheff method in a spreadsheet. In M. H. Karwan, J. Spronk, and J. Wallenius, editors, *Es-*

- says in Decision Making: A Volume in Honour of Stanley Zionts*, pages 93–103. Springer-Verlag, Berlin, Heidelberg, 1997.
- [222] R. E. Steuer and E.-U. Choo. An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26:326–344, 1983.
- [223] R. E. Steuer and L. R. Gardiner. Interactive multiple objective programming: Concepts, current status, and future directions. In C. A. Bana e Costa, editor, *Readings in Multiple Criteria Decision Aid*, pages 413–444, Berlin, Heidelberg, 1990. Springer-Verlag.
- [224] R. E. Steuer and L. R. Gardiner. On the computational testing of procedures for interactive multiple objective linear programming. In G. Fandel and H. Gehring, editors, *Operations Research*, pages 121–131, Berlin, Heidelberg, 1991. Springer-Verlag.
- [225] R. E. Steuer, J. Silverman, and A. W. Whisman. A combined Tchebycheff/aspiration criterion vector interactive multiobjective programming procedure. *Management Science*, 39:1255–1260, 1993.
- [226] R. E. Steuer and M. Sun. The parameter space investigation method of multiple objective nonlinear programming: A computational investigation. *Operations Research*, 43:641–648, 1995.
- [227] R. E. Steuer and A. W. Whisman. Toward the consolidation of interactive multiple objective programming procedures. In G. Fandel, M. Grauer, A. Kurzhanski, and A. P. Wierzbicki, editors, *Large-Scale Modelling and Interactive Decision Analysis*, pages 232–241. Springer-Verlag, 1986.
- [228] T. J. Stewart. A critical survey on the status of multiple criteria decision making theory and practice. *Omega*, 20:569–586, 1992.
- [229] M. Sun, A. Stam, and R. E. Steuer. Solving multiple objective programming problems using feed-forward artificial neural networks: The interactive FFANN procedure. *Management Science*, 42:835–849, 1996.
- [230] M. Sun, A. Stam, and R. E. Steuer. Interactive multiple objective programming using Tchebycheff programs and artificial neural networks. *Computers & Operations Research*, 27:601–620, 2000.
- [231] T. Sunaga, M. A. Mazed, and E. Kondo. A penalty function formulation for interactive multiobjective programming problems. In M. Iri and K. Yajima, editors, *System Modelling and Optimization*, pages 221–230. Springer-Verlag, 1988.
- [232] M. T. Tabucanon. *Multiple Criteria Decision Making in Industry*. Elsevier, Amsterdam, 1988.



- [233] M. Tamiz and D. F. Jones. A general interactive goal programming algorithm. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making: Proceedings of the Twelfth International Conference*, pages 433–444, Berlin, Heidelberg, 1997. Springer-Verlag.
- [234] C. G. Tapia and B. A. Murtagh. The use of preference criteria in interactive multiobjective mathematical programming. *Asia-Pacific Journal of Operational Research*, 6:131–147, 1989.
- [235] C. G. Tapia and B. A. Murtagh. A Markovian process in interactive multiobjective decision-making. *European Journal of Operational Research*, 57:421–428, 1992.
- [236] R. V. Tappeta, J. E. Renaud, A. Messac, and G. J. Sundararaj. Interactive physical programming: Tradeoff analysis and decision making in multidisciplinary optimization. *AIAA Journal*, 38:917–926, 2000.
- [237] S. Tarkkanen, K. Miettinen, J. Hakanen, and H. Isomäki. Incremental user-interface development for interactive multiobjective optimization. *Expert Systems with Applications*, 40:3220–3232, 2013.
- [238] K. Tarvainen. On the implementation of the interactive surrogate worth trade-off (ISWT) method. In M. Grauer and A. P. Wierzbicki, editors, *Interactive Decision Analysis*, pages 154–161, Berlin, Heidelberg, 1984. Springer-Verlag.
- [239] A. Teclé and L. Duckstein. A procedure for selecting MCDM techniques for forest resources management. In A. Goicoechea, L. Duckstein, and S. Zionts, editors, *Multiple Criteria Decision Making: Proceedings of the Ninth International Conference: Theory and Applications in Business, Industry, and Government*, pages 19–32, New York, 1992. Springer-Verlag.
- [240] J. Teghem Jr., C. Delhaye, and P. L. Kunsch. An interactive decision support system (IDSS) for multicriteria decision aid. *Mathematical and Computer Modelling*, 12:1311–1320, 1989.
- [241] A. Udink ten Cate. On the determination of the optimal temperature for the growth of an early cucumber crop in a greenhouse. In M. Grauer, M. Thompson, and A. P. Wierzbicki, editors, *Plural Rationality and Interactive Decision Processes*, pages 311–318, Berlin, Heidelberg, 1985. Springer-Verlag.
- [242] D. Vanderpooten. Multiobjective programming: Basic concepts and approaches. In R. Slowinski and J. Teghem, editors, *Stochastic versus Fuzzy Approaches to Multiobjective Mathematical Pro-*

- gramming under Uncertainty*, pages 7–22, Dordrecht, 1990. Kluwer Academic Publishers.
- [243] D. Vanderpooten and P. Vincke. Description and analysis of some representative interactive multicriteria procedures. *Mathematical and Computer Modelling*, 12:1221–1238, 1989.
- [244] R. Vetschera. Feedback-oriented group decision support in a reference point framework. In A. Lewandowski and V. Volkovich, editors, *Multiobjective Problems of Mathematical Programming*, pages 309–314. Springer-Verlag, 1991.
- [245] R. Vetschera. A note on scalarizing functions under changing sets of criteria. *European Journal of Operational Research*, 52:113–118, 1991.
- [246] P. Vincke. *Multicriteria Decision-Aid*. John Wiley & Sons, Chichester, 1992.
- [247] J. Wallenius. Comparative evaluation of some interactive approaches to multicriterion optimization. *Management Science*, 21:1387–1396, 1975.
- [248] S. Wang. Algorithms for multiobjective and nonsmooth optimization. In P. Kleinschmidt, F. J. Radermacher, W. Schweitzer, and H. Wildermann, editors, *Methods of Operations Research 58*, pages 131–142, Frankfurt am Main, 1989. Athenum Verlag.
- [249] S. Wang. An interactive method for multicriteria decision making. In K. H. Phua, C. M. Wang, W. Y. Yeong, T. Y. Leong, H. T. Loh, K. C. Tan, and F. S. Chou, editors, *Optimization: Techniques and Applications*, volume 1, pages 307–316. Proceedings of the International Conference (ICOTA), World Scientific Publishing Co., 1992.
- [250] H. R. Weistroffer. Multiple criteria decision making with interactive over-achievement programming. *Operations Research Letters*, 1:241–245, 1982.
- [251] H. R. Weistroffer. An interactive goal-programming method for non-linear multiple-criteria decision-making problems. *Computers & Operations Research*, 10:311–320, 1983.
- [252] H. R. Weistroffer. A combined over- and under-achievement programming approach to multiple objective decision-making. *Large Scale Systems*, 7:47–58, 1984.
- [253] H. R. Weistroffer. A flexible model for multi-objective optimization. In J. Jahn and W. Krabs, editors, *Recent Advances and Historical Developments of Vector Optimization*, pages 311–316, Berlin, Heidelberg, 1987. Springer-Verlag.

- [254] R. E. Wendell and D. N. Lee. Efficiency in multiple objective optimization problems. *Mathematical Programming*, 12:406–414, 1977.
- [255] D. J. White. A selection of multi-objective interactive programming methods. In S. French, R. Hartley, L. C. Thomas, and D. J. White, editors, *Multi-Objective Decision Making*, pages 99–126. Academic Press, 1983.
- [256] A. P. Wierzbicki. The use of reference objectives in multiobjective optimization. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making Theory and Applications*, pages 468–486, Berlin, Heidelberg, 1980. Springer-Verlag.
- [257] A. P. Wierzbicki. A mathematical basis for satisficing decision making. *Mathematical Modelling*, 3:391–405, 1982.
- [258] A. P. Wierzbicki. A methodological approach to comparing parametric characterizations of efficient solutions. In G. Fandel, M. Grauer, A. Kurzhanski, and A. P. Wierzbicki, editors, *Large-Scale Modelling and Interactive Decision Analysis*, pages 27–45. Springer-Verlag, 1986.
- [259] A. P. Wierzbicki. On the completeness and constructiveness of parametric characterizations to vector optimization problems. *OR Spectrum*, 8:73–87, 1986.
- [260] A. P. Wierzbicki. Convergence of interactive procedures of multiobjective optimization and decision support. In M. H. Karwan, J. Spronk, and J. Wallenius, editors, *Essays in Decision Making: A Volume in Honour of Stanley Zionts*, pages 19–47. Springer-Verlag, Berlin, Heidelberg, 1997.
- [261] A. P. Wierzbicki. Reference point approaches. In T. Gal, T. J. Stewart, and T. Hanne, editors, *Multicriteria Decision Making: Advances in MCDM Models, Algorithms, Theory, and Applications*, pages 9–1–9–39. Kluwer Academic Publishers, Boston, 1999.
- [262] A. P. Wierzbicki and J. Granat. Multi-objective modeling for engineering application in decision support. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making: Proceedings of the Twelfth International Conference*, pages 529–540, Berlin, Heidelberg, 1997. Springer-Verlag.
- [263] A. P. Wierzbicki and J. Granat. Multi-objective modeling for engineering applications: DIDASN++ system. *European Journal of Operational Research*, 113:372–389, 1999.

- [264] H.-M. Winkels and M. Meika. An integration of efficiency projections into the Geoffrion approach for multiobjective linear programming. *European Journal of Operational Research*, 16:113–127, 1984.
- [265] E. Wood, N. P. Greis, and R. E. Steuer. Linear and nonlinear applications of the Tchebycheff metric to the multi criteria water allocation problem. In S. Rinaldi, editor, *Environmental Systems Analysis and Management*, pages 363–376. North-Holland Publishing Company, 1982.
- [266] J.-B. Yang. Gradient projection and local region search for multiobjective optimization. *European Journal of Operational Research*, 112:432–459, 1999.
- [267] J.-B. Yang, C. Chen, and Z.-J. Zhang. The interactive step tradeoff method (ISTM) for multiobjective optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, 20:688–695, 1990.
- [268] J.-B. Yang and D. Li. Normal vector identification and interactive tradeoff analysis using minimax formulation in multiobjective optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, 32:305–319, 2002.
- [269] J.-B. Yang and P. Sen. Preference modelling by estimating local utility functions for multiobjective optimization. *European Journal of Operational Research*, 95:115–138, 1996.
- [270] P. L. Yu. A class of solutions for group decision problems. *Management Science*, 19:936–946, 1973.
- [271] P. L. Yu. *Multiple-Criteria Decision Making Concepts, Techniques, and Extensions*. Plenum Press, New York, 1985.
- [272] L. Zadeh. Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*, 8:59–60, 1963.
- [273] M. Zeleny. Compromise programming. In J. L. Cochrane and M. Zeleny, editors, *Multiple Criteria Decision Making*, pages 262–301, Columbia, South Carolina, 1973. University of South Carolina Press.