

**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

**Title:** Why Use Interactive Multi-Objective Optimization in Chemical Process Design?

**Year:** 2009

**Version:**

**Please cite the original version:**

Miettinen, K., & Hakanen, J. (2009). Why Use Interactive Multi-Objective Optimization in Chemical Process Design?. In G. P. Rangaiah (Ed.), *Multi-Objective Optimization: Techniques and Applications in Chemical Engineering* (pp. 153-188). World Scientific.

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 6

### Why Use Interactive Multi-Objective Optimization in Chemical Process Design?

Kaisa Miettinen and Jussi Hakanen  
Department of Mathematical Information Technology  
P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland  
kaisa.miettinen@jyu.fi, jussi.hakanen@jyu.fi

#### Abstract

Problems in chemical engineering, like most real-world optimization problems, typically, have several conflicting performance criteria or objectives and they often are computationally demanding, which sets special requirements on the optimization methods used. In this paper, we point out some shortcomings of some widely used basic methods of multi-objective optimization. As an alternative, we suggest using interactive approaches where the role of a decision maker or a designer is emphasized. Interactive multi-objective optimization has been shown to suit well for chemical process design problems because it takes the preferences of the decision maker into account in an iterative manner that enables a focused search for the best Pareto optimal solution, that is, the best compromise between the conflicting objectives. For this reason, only those solutions that are of interest to the decision maker need to be generated making this kind of an approach computationally efficient. Besides, the decision maker does not have to compare many solutions at a time which makes interactive approaches more usable from the cognitive point of view. Furthermore, during the interactive solution process the decision maker can learn about the interrelationships among the objectives. In addition to describing the general philosophy of interactive approaches, we discuss the possibilities of interactive multi-objective optimization in chemical process design and give some examples of interactive methods to illustrate the ideas. Finally, we demonstrate the usefulness of interactive approaches in chemical process design by summarizing some reported studies related to, for example, paper making and sugar industries. Let us emphasize that the approaches described are appropriate for problems with more than two objective functions.

**Keywords:** Multiple criteria decision making (MCDM), interactive methods, scalarization, chemical engineering, Pareto optimality

## 6.1. Introduction

Problems involving multiple conflicting criteria or objectives are generally known as multiple criteria decision making (MCDM) problems. In such problems, instead of a well-defined single optimal solution, there are many compromise solutions, so-called Pareto optimal solutions, that are mathematically incomparable. In the MCDM literature, solving a multi-objective optimization problem is usually understood as helping a human decision maker (DM) in considering the multiple objectives simultaneously and in finding a Pareto optimal solution that pleases him/her the most. In other words, the solution process needs some involvement of the DM and the final solution is determined by his/her preferences.

Examples of surveys of methods available for multi-objective optimization are Chankong and Haimes (1983); Hwang and Masud (1979); Marler and Arora (2004); Miettinen (1999); Sawaragi *et al.* (1985); Steuer (1986); Vincke (1992). The methods can be classified in different ways. In Hwang and Masud (1979); Miettinen (1999) they are divided into four classes according to the role of the DM in the solution process. If there is no DM and his/her preference information available, we can use so-called *no-preference methods* which find some neutral compromise solution without any additional preference information. In *a priori methods*, the DM first gives preference information and then the method looks for a Pareto optimal solution satisfying the hopes as well as possible. This is a straightforward approach but the difficulty is that the DM may have too optimistic or pessimistic hopes and then the solution generated may be far from them and, thus, disappointing.

In *a posteriori methods*, a representative set of Pareto optimal solutions is generated and then the DM must select the most preferred one. In this way, the DM gets an overview of the problem but it may be difficult for the DM to analyze a large amount of information. A natural visualization on a plane is possible only for problems involving two objectives. Furthermore, generating the set of Pareto optimal solutions may be computationally expensive. Evolutionary multi-objective optimization (EMO) algorithms belong to this class but it may happen that the solutions generated are not really Pareto optimal but only nondominated in the current population. The fourth class is that of *interactive methods*. Many interactive methods exist but they should become more widely known among people solving real applications. In interactive approaches, a solution pattern is formed and then repeated and the DM can specify and adjust one's

preference information between each iteration.

In this chapter, we introduce scalarization based approaches and, in particular, interactive methods as alternatives to EMO approaches. Our aim here is to widen the awareness of the readers of the existence of interactive methods and the advantages and usefulness of using them. Sometimes multi-objective and bi-objective optimization are regarded as synonyms but this kind of thinking is very limiting. Our approaches are capable of handling genuine multi-objective optimization with more than two objectives. Besides discussing drawbacks of some widely used methods (like the weighting method) that sometimes are regarded as the only nonevolutionary multi-objective optimization methods available, we introduce some interactive methods and the NIMBUS method (Miettinen, 1999; Miettinen and Mäkelä, 2006), in particular. In addition, we summarize encouraging experiences of solving some chemical engineering problems with the interactive NIMBUS method. The motivation here is that it is important to get to know that a variety of methods and approaches exists. In this way, people solving different problems are able to use the most appropriate approaches. In this respect, scalarization based and interactive methods complement evolutionary approaches. More information about bringing the MCDM and EMO fields closer is available in Branke *et al.* (2008).

This paper is organized as follows. In Section 6.2, main concepts and the idea of scalarization based methods are introduced. In addition, some basic multi-objective optimization methods and their shortcomings are presented and comparative aspects between scalarization based and evolutionary approaches are discussed. Section 6.3 concentrates on interactive multi-objective optimization and some methods. Advantages of using interactive approaches in chemical process design are discussed in Section 6.4 and some applications related to sugar and papermaking industries are summarized in Section 6.5. Finally, concluding remarks are given in Section 6.6.

## 6.2. Concepts, Basic Methods and Some Shortcomings

### 6.2.1. Concepts

Let us consider *multi-objective optimization problems* of the form

$$\begin{aligned} & \text{minimize} && \{f_1(\mathbf{x}), \dots, f_k(\mathbf{x})\} \\ & \text{subject to} && \mathbf{x} \in S, \end{aligned} \tag{6.1}$$

where we have  $k$  ( $\geq 2$ ) conflicting *objective functions*  $f_i : \mathbf{R}^n \rightarrow \mathbf{R}$  that we want to minimize simultaneously. In addition, we have *decision (variable)*

or design vectors  $\mathbf{x} = (x_1, \dots, x_n)^T$  belonging to the nonempty feasible region  $S \subset \mathbf{R}^n$  defined by equality, inequality and/or box constraints. In multi-objective optimization, we typically are interested in objective vectors consisting of objective (function) values  $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))^T$  and the image of the feasible region is called a feasible objective region  $Z = \mathbf{f}(S)$ . (Note that if some function  $f_i$  should be maximized, it is equivalent to minimize  $-f_i$ . Thus, without losing any generality, we consider problems of the form (6.1).)

For multi-objective optimization, theoretical background has been laid, e.g., in Edgeworth (1881); Koopmans (1951); Kuhn and Tucker (1951); Pareto (1896, 1906). Typically, there is no unique optimal solution but a set of mathematically incomparable solutions can be identified. An objective vector can be regarded as optimal if none of its components (i.e., objective values) can be improved without deterioration to at least one of the other objectives. To be more specific, a decision vector  $\mathbf{x}' \in S$  and the corresponding objective vector  $\mathbf{f}(\mathbf{x}')$  are called *Pareto optimal* if there does not exist another  $\mathbf{x} \in S$  such that  $f_i(\mathbf{x}) \leq f_i(\mathbf{x}')$  for all  $i = 1, \dots, k$  and  $f_j(\mathbf{x}) < f_j(\mathbf{x}')$  for at least one index  $j$ . In the MCDM literature, widely used synonyms of Pareto optimal solutions are nondominated, efficient, noninferior or Edgeworth-Pareto optimal solutions.

As mentioned in the introduction, we here assume that a DM is able to participate in the solution process. (S)he is expected to know the problem domain and be able to specify preference information related to the objectives and/or different solutions. We assume that less is preferred to more in each objective for him/her. (In other words, all the objective functions are to be minimized.) If the problem is correctly formulated, the final solution of a rational DM is always Pareto optimal. Thus, we can restrict our consideration to Pareto optimal solutions. For this reason, it is important that the multi-objective optimization method used is able to find any Pareto optimal solution and produce only Pareto optimal solutions. However, weakly Pareto optimal solutions are sometimes used because they may be easier to generate than Pareto optimal ones. A decision vector  $\mathbf{x}' \in S$  (and the corresponding objective vector) is *weakly Pareto optimal* if there does not exist another  $\mathbf{x} \in S$  such that  $f_i(\mathbf{x}) < f_i(\mathbf{x}')$  for all  $i = 1, \dots, k$ . Note that Pareto optimality implies weak Pareto optimality but not vice versa.

The DM may find information about the ranges of feasible Pareto optimal objective vectors useful. Lower bounds form a so-called *ideal objective vector*  $\mathbf{z}^* \in \mathbf{R}^k$ . Its components  $z_i^*$  are obtained by minimizing each ob-

jective function individually subject to the feasible region. Sometimes, for computational reasons, we also need a strictly better *utopian objective vector*  $\mathbf{z}^{**}$  defined as  $z_i^{**} = z_i^* - \varepsilon$  for  $i = 1, \dots, k$ , where  $\varepsilon$  is some small positive scalar.

The upper bounds of the Pareto optimal set, that is, the components of a *nadir objective vector*  $\mathbf{z}^{\text{nad}}$ , are in practice difficult to obtain. It can be estimated using a payoff table but the estimate may be unreliable (see, e.g., Miettinen (1999) and references therein).

Finding a final solution to problem (6.1) is called a *solution process*. It usually involves the DM and an analyst. An *analyst* can be a human being or a computer program. The analyst's role is to support the DM and generate information for the DM. Let us emphasize that the DM is not assumed to know multi-objective optimization theory or methods but (s)he is supposed to be an expert in the problem domain, that is, understand the application considered and have insight into the problem. Based on that, (s)he is supposed to be able to specify preference information related to the objectives considered and different solutions. The DM can be, e.g., a *designer*. The task of a multi-objective optimization method is to help the DM in finding the most preferred solution as the final one. The *most preferred solution* is a Pareto optimal solution which is satisfactory for the DM.

Multi-objective optimization problems can be solved by *scalarizing* the problem, in other words, by forming a problem (or several problems) involving a single objective function (and possibly some additional constraints). Because the *scalarized problem* has a real-valued objective function (possibly depending on some parameters originating, e.g., from preference information), it can be solved using appropriate (local, global, mixed-integer etc.) single objective optimizers and, thus, we can utilize the theoretical background and large amount of methods developed for single objective optimization. The real-valued objective function can be called a *scalarizing function*. Such *scalarizing approaches* should be favored that generate Pareto optimal solutions and can find any Pareto optimal solution (as discussed earlier). Depending on whether a local or a global optimizer is used we get either locally or globally Pareto optimal solutions (for nonconvex problems). Because locally Pareto optimal solutions are irrational for DMs, it is important to use appropriate optimizers.

As said in the definition, to move from one Pareto optimal solution to another Pareto optimal solution means trading off. More formally, a *trade-off* is the ratio of change in objective function values involving the

increment of one objective function that occurs when the value of some other objective function decreases. For details, see, e.g., Chankong and Haimes (1983); Miettinen (1999).

The DM can specify preference information in many ways and the task is to find a format that the DM finds most natural and intuitive. One possibility is that the DM specifies *aspiration levels*  $\bar{z}_i$  ( $i = 1, \dots, k$ ) that are desirable or acceptable objective function values. The vector  $\bar{\mathbf{z}} \in \mathbf{R}^k$  consisting of aspiration levels is called a *reference point*.

### 6.2.2. Some Basic Methods

When discussing methods, it is in order to begin with two widely used ones, the weighting method and the  $\varepsilon$ -constraint method. They can be called basic methods. In many applications one can actually see that they are used without necessarily recognizing them as multi-objective optimization methods or explicitly even saying that the problem considered is a multi-objective optimization one. This means that when formulating and solving the problem, the difference between modeling and optimization phases is not always clear. One can say that these basic methods represent ideas that first come to one's mind when one wants to consider several objective functions simultaneously. However, these methods have some shortcomings that are not necessarily widely known and, for that reason, we want to point them out. In this section, we also briefly discuss some characteristics of EMO approaches when compared to scalarizing approaches. Proofs of theorems related to optimality as well as further details about the methods can be found in Miettinen (1999).

#### 6.2.2.1. Weighting Method

The scalarized problem to be solved in the *weighting method* (Gass and Saaty, 1955; Zadeh, 1963) is

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^k w_i f_i(\mathbf{x}) \\ & \text{subject to} && \mathbf{x} \in S, \end{aligned} \tag{6.2}$$

where the weights are nonnegative, that is,  $w_i \geq 0$  for all  $i = 1, \dots, k$  and, typically,  $\sum_{i=1}^k w_i = 1$ . The solution of (6.2) is weakly Pareto optimal and Pareto optimal if  $w_i > 0$  for all  $i = 1, \dots, k$  or the solution is unique.

The weighting method can be used (as an *a posteriori* method) so that different weights are set to generate different Pareto optimal solutions and

then the DM must select the most satisfactory one. Alternatively, the DM can be asked to specify the weights reflecting his/her preferences (as an *a priori* method).

It is important to point out that if the problem is nonconvex, the weighting method does not work as it is expected to. However, surprisingly few people applying it seem to realize this. To be more specific, any Pareto optimal solution can be found by altering the weights only if the problem is convex. Thus, it may happen that some Pareto optimal solutions of nonconvex problems remain undiscovered no matter how the weights are varied. This is a serious drawback because it is not always easy to check the convexity in real applications, e.g., involving black box objective functions. If the method is used for generating a representation of the Pareto optimal set, the DM gets a completely misleading impression because only some parts of the Pareto optimal set are covered. Furthermore, as shown by Das and Dennis (1997), an evenly distributed set of weights does not necessarily produce an evenly distributed representation of the Pareto optimal set, even if the problem is convex.

If the method is used as an *a priori* method, the DM is expected to represent his/her preferences in the form of weights. However, in general, the role of the weights may be greatly misleading and it is not at all clear what the concept 'relative importance of an objective' means (Podinovski, 1994; Roy and Mousseau, 1996). Besides, giving weights implies eliciting global preferences which may be hard if not impossible. Furthermore, the DM may get an unsatisfactory solution if some of the objective functions correlate with each other (Steuer, 1986). This is also demonstrated in Tanner (1991) with an example originally formulated by P. Korhonen. The problem (involving three candidates and five objectives) is about choosing a spouse. There, the weights representing the preferences of the DM result with a spouse who is the worst in the objective that was given the biggest weight, that is, the highest importance.

Overall, we can say that it is not necessarily easy for the DM (or the analyst) to control a solution process with weights because weights behave in an indirect way. It makes no sense to end up in a situation where one tries to guess such weights that would produce a desirable solution. Because the DM can not be properly supported in this, (s)he is likely to get frustrated. Instead, it is then better to use real interactive methods (see Section 6.3) where more intuitive preference information can be used.

### 6.2.2.2. $\varepsilon$ -Constraint Method

If one of the objective functions is selected to be optimized and the others are converted into constraints, we get the scalarization of the  $\varepsilon$ -constraint method (Haimes *et al.*, 1971; Chankong and Haimes, 1983):

$$\begin{aligned} & \text{minimize} && f_\ell(\mathbf{x}) \\ & \text{subject to} && f_j(\mathbf{x}) \leq \varepsilon_j \text{ for all } j = 1, \dots, k, j \neq \ell, \\ & && \mathbf{x} \in S, \end{aligned} \quad (6.3)$$

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

The solution of problem (6.3) is always weakly Pareto optimal and Pareto optimal if it is unique. On the other hand,  $\mathbf{x}^* \in S$  is Pareto optimal if and only if it solves (6.3) for every  $\ell = 1, \dots, k$ , where  $\varepsilon_j = f_j(\mathbf{x}^*)$  for  $j = 1, \dots, k$ ,  $j \neq \ell$ . Thus, ensuring Pareto optimality means either solving  $k$  problems or obtaining a unique solution (which is not necessarily easy to verify in practice). What is positive when compared to the weighting method is that the  $\varepsilon$ -constraint method can find any Pareto optimal solution even for nonconvex problems.

In practice, it is not always easy to set the upper bounds so that problem (6.3) has feasible solutions. It may also be difficult to select which of the objective functions should be the one to be optimized. These choices may affect the solutions obtained. For using the  $\varepsilon$ -constraint method as an *a posteriori* method, systematic ways of perturbing the upper bounds are suggested in Chankong and Haimes (1983). On the other hand, when used in an *a priori* way, the drawback is that if there is a promising solution really close to the bound specified but on the infeasible side, it will never be found because single objective optimizers must obey the constraints specified. However, the DM may want to study solutions corresponding to different bounds. If this is the case, it is again recommended to use interactive methods.

### 6.2.2.3. Evolutionary Multi-Objective Optimization

Since most of this book is devoted to evolutionary methods for multi-objective optimization, we here only wish to discuss some differences between EMO approaches and scalarization based approaches. As mentioned before, EMO approaches are *a posteriori* type of methods and they try to generate an approximation of the Pareto optimal set. In bi-objective optimization problems, it is easy to plot the objective vectors produced on a plane and ask the DM to select the most preferred one. While looking at the

visualization, the DM sees an overview of the trade-offs in the problem and most likely can choose the final solution. However, if the problem has more than two objectives, there is no natural way of visualization for objective vectors but one has to settle for projections or use other additional tools which are not necessarily very intuitive. This means that it is problematic to represent the many different solutions for the DM to compare. Another question is how to support him/her in selecting one of many solutions in a reasoned way. Furthermore, generating a good representation of a Pareto optimal set in a higher-dimensional feasible objective region necessitates high population sizes which implies high computational costs. If function evaluations are costly, the calculation takes a lot of effort. Besides, there may be many areas in the Pareto optimal set that the DM is not interested in. In such cases, we waste computational resources in finding solutions that are not needed at all.

On the other hand, it is not sensible, e.g., to restrict consideration to two objectives only, for the purpose of intuitive visualization. It is better to consider the problem as a whole and use as many objectives as needed instead of artificial simplifications. Furthermore, as mentioned earlier, EMO approaches do not necessarily guarantee that they generate Pareto optimal solutions. Because of the above-mentioned aspects, EMO approaches may not always be the best methods for solving multi-objective optimization problems and that is why we introduce scalarization based and interactive methods, in particular, to be considered as alternative approaches. When using them, the DM can concentrate on interesting solutions only and computational effort is not wasted. Furthermore, the DM can decide how many solutions (s)he wants to compare at a time.

The strength of evolutionary approaches is their wide applicability to, e.g., nondifferentiable and nonconvex problems. We wish to emphasize that this positive feature can be combined with scalarization based approaches by using evolutionary algorithms (i.e., not EMO but single objective optimizers) for solving the scalarized problem.

### **6.3. Interactive Multi-objective Optimization**

As said in the introduction, in interactive multi-objective optimization methods, a solution pattern is formed and repeated and the DM specifies preference information progressively during the solution process. In other words, the solution process is iterative and the phases of preference elicitation and solution generation alternate. In brief, the main steps of a

general interactive method are the following: (1) initialization (e.g., calculating ideal and nadir values and showing them to the DM), (2) generate a Pareto optimal starting point (some neutral compromise solution or solution given by the DM), (3) ask for preference information from the DM (e.g., aspiration levels or number of new solutions to be generated), (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 DM. If several solutions were generated, ask the DM to select the best solution so far, and (6) stop, if the DM wants to. Otherwise, go to step (3).

The most important stopping criterion is the satisfaction of the DM in some solution. (Some interactive methods use also algorithmic stopping criteria but we do not go into such details here.) In each iteration, some information about the problem or solutions available is given to the DM and then (s)he is supposed to answer some questions or to give some other kind of information. New solutions are generated based on the information specified. In this way, the DM directs the solution process towards such Pareto optimal solutions that (s)he is interested in and only such solutions are generated.

The advantage of interactive methods is that the DM can specify and correct his/her preferences and selections during the solution process. Because of the iterative nature, the DM does not need to have any global preference structure and (s)he can *learn* during the solution process. This is a very important strength of interactive methods. Actually, finding the final solution is not always the only task but it is also noteworthy that the DM gets to know the problem, its possibilities and limitations.

We can say that interactive methods overcome weaknesses of *a priori* and *a posteriori* methods: the DM does not need a global preference structure and only interesting Pareto optimal solutions need to be considered. The latter means both savings in computational cost, which in many computationally complicated real problems is a significant advantage, and avoids setting cognitive overload on the DM, which the comparison of many solutions typically implies.

Many interactive methods exist and none of them is superior to all the others but some methods may suit different DMs and problems better than the others. Methods differ from each other by both the style of interaction and technical realization: e.g., what kind of information is given to the DM, the form of preference information specified by the DM and what kind of a scalarizing function is used or, more generally, which inner process is used to generate Pareto optimal solutions (Miettinen, 1999). It is always

important that the DM finds the method worthwhile and is able to use it properly, in other words, the DM must find preferences easy and intuitive to provide in the style selected. In many cases, we can identify two phases in the solution process: a *learning phase* when the DM wants to learn about the problem and what kind of solutions are feasible and a *decision phase* when the most preferred solution is found in the region identified in the first phase. If so desired, the two phases can be used iteratively, as well.

Descriptions of interactive methods are given, e.g., in Buchanan (1986); Chankong and Haimes (1983); Hwang and Masud (1979); Miettinen (1999); Sawaragi *et al.* (1985); Steuer (1986); Stewart (1992); Vanderpooten and Vincke (1989) and methods with applications to large-scale systems and industry are presented in Haimes *et al.* (1990); Statnikov (1999); Tabucanon (1988). Special attention to describing methods for nonlinear problems is paid in Miettinen (1999). Here we only describe a few interactive methods. We concentrate on methods where the DM specifies preferences in the form of reference points or classification.

### 6.3.1. Reference Point Approaches

In reference point based methods, the DM first specifies a reference point  $\bar{\mathbf{z}} \in \mathbf{R}^k$  consisting of desirable aspiration levels for each objective and then this reference point is projected onto the Pareto optimal set. That is, a Pareto optimal solution closest to the reference point is found. The distance can be measured in different ways. Specifying a reference points is an intuitive way for the DM to direct the search of the most preferred solution. It is straightforward to compare the point specified and the solution obtained without artificial concepts. Examples of methods of this type are the reference point method and the 'light beam search'.

The reference point method is based on using a so-called *achievement (scalarizing) function* (Wierzbicki, 1982). The achievement function measures the distance between the reference point and Pareto optimal solutions and produces a new Pareto optimal solution closest to the reference point. The beauty here is that Pareto optimal solutions are generated no matter how the reference point is specified, that is, they can be attainable or not. We have an example of an achievement function in the problem

$$\begin{aligned} & \text{minimize} && \max_{i=1, \dots, k} [w_i(f_i(\mathbf{x}) - \bar{z}_i)] + \rho \sum_{i=1}^k w_i f_i(\mathbf{x}) && (6.4) \\ & \text{subject to} && \mathbf{x} \in S, \end{aligned}$$

where  $w_i$  ( $i = 1, \dots, k$ ) are fixed scaling coefficients, e.g.,  $w_i = 1/(z_i^{\text{nad}} - z_i^{**})$  and  $\rho > 0$  is a relatively small scalar. The solution of problem (6.4) is Pareto optimal and, as said, different Pareto optimal solutions can be generated by setting a different reference point (Miettinen, 1999).

In the reference point method, the DM specifies a reference point and the corresponding solution of (6.4) is shown to him/her. In addition, the DM is shown  $k$  other solutions obtained by slightly shifting the reference point in each coordinate direction. Thus, the DM can compare  $k+1$  Pareto optimal solutions close to the reference point. Then the DM can set a new reference point (i.e., adjust the reference point according to his/her preferences) and the solution process continues as long as the DM wants to. When the Pareto optimal solutions are generated, the DM learns more about the possibilities and limitations of the problem and, therefore, can use more appropriate reference points.

Because of the intuitive character of reference points, it is advisable to use the achievement scalarizing function even when it is not possible to use an interactive approach. This means that the DM expresses his/her hopes in the form of a reference point and the solution of (6.4) is then shown to him/her. In this way, a reference point based approach can be used as an *a priori* method. It is also possible to use reservation levels representing objective values that should be achieved (besides aspiration levels). For further details, see Wierzbicki *et al.* (2000).

Another reference point based method is the 'light beam search' (Jaszkiewicz and Slowinski, 1999). It uses a similar achievement function as the reference point method but combined with tools of multiattribute decision analysis (designed for comparing a discrete set of solutions). Besides a reference point, the DM must supply thresholds for objective functions describing indifference and preference in objective values. This information is used to derive outranking relations between solutions. As a result, incomparable or indifferent solutions are not displayed to the DM.

### 6.3.2. Classification-Based Methods

As discussed, moving from one Pareto optimal solution to another implies trading off. In other words, to move to another Pareto optimal solution where some objective function gets a better value, some other objective function must be allowed to get worse. This is the starting point of classification-based methods where the DM studies a Pareto optimal solution and says what kind of changes in the objective function values would

lead to a more preferred solution. Larichev (1992) has shown that for DMs classification is a cognitively valid way of expressing preference information.

Classification is an intuitive way for the DM to direct the solution process because no artificial concepts are used. Objective function values are as such meaningful and understandable for the DM. The DM can express hopes about improved solutions and directly see and compare how well the hopes could be realized.

To be more specific, when classifying objective functions the DM indicates which function values should improve, which ones are acceptable and which are allowed to get worse. In addition, amounts of improvement or impairments are asked from the DM. There exist several classification-based interactive multi-objective optimization methods. They use different numbers of classes and generate new solutions in different ways.

Let us point out that expressing preference information as a reference point (Miettinen and Mäkelä, 2002; Miettinen *et al.*, 2006) is closely related to classification. However, when classification assumes that some objective function must be allowed to get worse, a reference point can be set without considering the current solution. Even though it is not possible to improve all objective function values of a Pareto optimal solution simultaneously, the DM can still express preferences without paying attention to this fact and then see what kind of solutions are feasible. On the other hand, when using classification, the DM is more in control and selects functions to be improved and specifies amounts of impairment for the others.

Next, we briefly introduce the satisficing trade-off method and then describe the NIMBUS method in some more detail. We pay more attention to NIMBUS (and software implementing it) because we shall refer to it later when discussing applications.

#### 6.3.2.1. *Satisficing Trade-Off Method*

The satisficing trade-off method (STOM) (Nakayama and Sawaragi, 1984) is based on the classification of objective functions at the current Pareto optimal solution into the three classes described earlier. A reference point can be formed based on this information.

For functions whose values the DM wants to improve, (s)he also has to specify desirable aspiration levels. If some function has an acceptable value, it is set as the corresponding aspiration level. Under some assumptions, it is possible to calculate how much impairment should be allowed in the other objective functions in order to attain the desired improvements. This is

called automatic trade-off (Nakayama, 1995). In this way, the DM has to specify less information. Once all components of a reference point have been set, one can solve a scalarized problem

$$\text{minimize } \max_{i=1,\dots,k} \left[ \frac{f_i(\mathbf{x}) - z_i^{**}}{\bar{z}_i - z_i^{**}} \right] + \rho \sum_{i=1}^k \frac{f_i(\mathbf{x})}{\bar{z}_i - z_i^{**}} \quad (6.5)$$

subject to  $\mathbf{x} \in S$ ,

where  $\rho > 0$  is a relatively small scalar. Then, the solution of (6.5) (which is guaranteed to be Pareto optimal, see Miettinen (1999)) is shown to the DM and (s)he can stop or classify the objective functions again. The DM can easily learn about the problem by comparing the hopes expressed in the classification and the Pareto optimal solution obtained.

If it is not possible to use automatic trade-off, classifying the objective functions or setting a reference point are almost the same, as discussed earlier. The only difference is that here the reference point is set such that some objective functions must be allowed to get impaired values. Let us finally mention that STOM has been applied to many engineering problems, e.g., in Nakayama (1995); Nakayama and Furukawa (1985); Nakayama and Sawaragi (1984).

### 6.3.2.2. The NIMBUS Method

The NIMBUS method (Miettinen, 1999; Miettinen and Mäkelä, 1995, 1999, 2000, 2006) is an interactive method based on classification of the objective functions into up to five classes. To be more specific, the DM is asked to specify how the current Pareto optimal solution  $\mathbf{f}(\mathbf{x}^h)$  should be improved by classifying the objective functions into classes where the functions  $f_i$

- should be improved as much as possible ( $i \in I^{imp}$ ),
- should be improved until a specified aspiration level  $\bar{z}_i$  ( $i \in I^{asp}$ ),
- are satisfactory at the moment ( $i \in I^{sat}$ ),
- can impair till a specified bound  $\varepsilon_i$  ( $i \in I^{bound}$ ) and
- can change freely ( $i \in I^{free}$ ).

A classification is feasible if at least one of the objective functions is allowed to get worse. Then the original multi-objective optimization problem is converted into a scalarized problem using the classification information specified. The solution of the scalarized problem reflects how well the hopes expressed in the classification could be achieved.

There exist several variants of NIMBUS (Miettinen, 1999; Miettinen and Mäkelä, 1995, 1999, 2000, 2006). Here we concentrate on the synchronous version (Miettinen and Mäkelä, 2006), where several scalarizing functions can be used based on a classification once expressed. Because they take the preference information into account in slightly different ways (Miettinen and Mäkelä, 2002), the DM can learn more about different solutions satisfying his/her hopes and choose the one that best obeys his/her preferences. An example of the scalarized problems used is

$$\begin{aligned} & \text{minimize} && \max_{i \in I^{imp}, j \in I^{asp}} \left[ \frac{f_i(\mathbf{x}) - z_i^*}{z_i^{nad} - z_i^{**}}, \frac{f_j(\mathbf{x}) - \bar{z}_j}{z_j^{nad} - z_j^{**}} \right] + \rho \sum_{i=1}^k \frac{f_i(\mathbf{x})}{z_i^{nad} - z_i^{**}} \\ & \text{subject to} && f_i(\mathbf{x}) \leq f_i(\mathbf{x}^h) \text{ for all } i \in I^{imp} \cup I^{asp} \cup I^{sat}, \\ & && f_i(\mathbf{x}) \leq \varepsilon_i, \text{ for all } i \in I^{bound}, \mathbf{x} \in S, \end{aligned} \quad (6.6)$$

where  $\rho > 0$  is a relatively small scalar. The three other scalarized problems used in the synchronous NIMBUS method and further details are given in Miettinen and Mäkelä (2006).

Once the DM has classified the objective functions, (s)he can decide how many Pareto optimal solutions (between one and four) based on this information (s)he wants to see and compare. Then, as many scalarized problems are formed and solved and the new solutions are shown to the DM together with the current solution. If the DM has found the most preferred solution, the solution process stops. Otherwise, the DM can select a solution as a starting point of a new classification or ask for a desired number of intermediate (Pareto optimal) solutions between any two solutions generated so far. The DM can also save any interesting solutions to a database and return to them later. All the solutions considered are Pareto optimal. For details of the algorithm, see Miettinen and Mäkelä (2006).

In the initialization phase of the NIMBUS method, the ranges in the Pareto optimal set, that is, the ideal and the nadir objective vectors are computed to give the DM some information about the possibilities of the problem. The starting point of the solution process can be specified by the DM or it can be a neutral compromise solution located approximately in the middle of the Pareto optimal set. To get it, we set  $(\mathbf{z}^{nad} + \mathbf{z}^{**})/2$  as a reference point and solve (6.4).

In NIMBUS, the DM iteratively expresses his/her desires and learns about the feasible solutions available for the problem considered. Unlike some other methods based on classification, the success of the solution process does not depend entirely on how well the DM manages in specifying

the classification and the appropriate parameter values. It is important that the classification is not irreversible. Thus, the DM is free to go back or explore intermediate points. (S)he can easily get to know the problem and its possibilities by specifying, e.g., loose upper bounds and examining intermediate solutions. NIMBUS has been successfully applied, e.g., in the fields of optimal control and optimal design (Hämäläinen *et al.*, 2003; Heikkola *et al.*, 2006; Madetoja *et al.*, 2006; Miettinen *et al.*, 1998).

As far as software is concerned, the interactive nature of the solution process naturally sets its own requirements (Hakanen, 2006). First of all, a good graphical user-interface (GUI) is needed in order to enable the interaction between the DM and the method. In addition, visualizations of the solutions obtained must be available for the DM to compare and evaluate the solutions generated. With interactive methods, more than three objective functions can easily be considered, which sets more requirements on the visualization when compared to, e.g., visualizing the Pareto optimal set for bi-objective problems.

Currently, the NIMBUS method has two implementations: WWW-NIMBUS<sup>®</sup> and IND-NIMBUS<sup>®</sup>. The WWW-NIMBUS<sup>®</sup> system (Miettinen and Mäkelä, 2000, 2006) has been operating via the Internet at <http://nimbus.it.jyu.fi> since 1995 and can be used free of charge for teaching and academic purposes. Only a browser is required for using WWW-NIMBUS<sup>®</sup> and, therefore, the user has always the latest version available. All the computation is performed in the server computer at the University of Jyväskylä.

As far as using WWW-NIMBUS<sup>®</sup> is concerned, one can create an account of one's own or visit the system as a guest. Once an account has been created, it is possible to save problems and solutions in the system. WWW-NIMBUS<sup>®</sup> takes the user from one web page to another. The problem to be solved can be input by filling a web form (or as a subroutine). The system first asks for the name and the dimensions of the problem. In the second web page, the user can type in the formulas of each objective and constraint function as well as ranges (and initial values) for variables.

There are different single-objective optimizers available for solving the scalarized problems formed and the user can decide after each classification which optimizer to use or use the default one. The proximal bundle method (Mäkelä and Neittaanmäki, 1992) is a local optimizer and needs initial values for variables as well as (sub)gradients for functions. (The system can generate the latter automatically.) Alternatively, it is possible to use two variants of (global) real-coded genetic algorithms that differ from each other

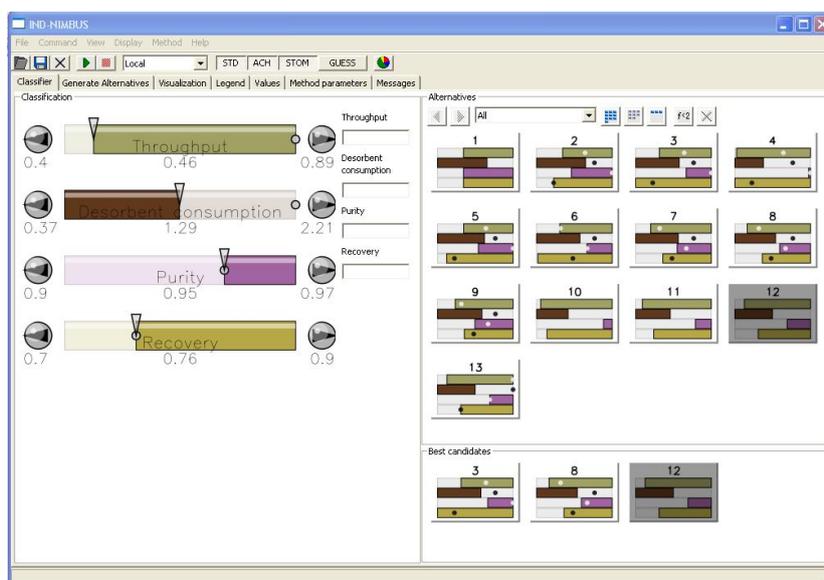


Fig. 6.1. A screenshot of IND-NIMBUS<sup>®</sup>.

in constraint handling (Miettinen *et al.*, 2003b). These optimizers can also handle mixed-integer problems. A hybrid of a global and a local optimizer can also be used. There are also different visualizations available to aid the user in analyzing and comparing different Pareto optimal solutions. The system has a tutorial that guides the user through the different phases of the interactive solution process. In addition, each web page has a separate help available.

IND-NIMBUS<sup>®</sup> (Miettinen, 2006; Ojalehto *et al.*, 2007) is a commercial implementation of the NIMBUS method developed for solving industrial multi-objective optimization problems (<http://ind-nimbus.it.jyu.fi/>). IND-NIMBUS<sup>®</sup> is available for Linux and MS-Windows operating systems. A screenshot of IND-NIMBUS<sup>®</sup> can be seen in Fig. 6.1. The bars on the left represent the current values of objective functions and the DM can classify the functions by clicking with a mouse or by specifying desired function values. The window on the right shows Pareto optimal solutions generated so far and interesting solutions can be saved in the lower right corner as best candidates.

Problems formulated with various simulators or modeling tools can be connected with IND-NIMBUS<sup>®</sup> and different underlying single-objective

optimizers can be used depending on the properties of the problem considered. IND-NIMBUS<sup>®</sup> has been connected, e.g., to the BALAS<sup>®</sup> process simulator (<http://virtual.vtt.fi/virtual/balas/>), developed at the VTT Technical Research Center of Finland. In addition to the single-objective optimizers available in WWW-NIMBUS<sup>®</sup>, e.g., the IPOPT optimizer (Wächter and Biegler, 2006) has been used in IND-NIMBUS<sup>®</sup>.

In what follows, we call the combination of IND-NIMBUS<sup>®</sup> and some modeling tool or a simulator by the name *IND-NIMBUS<sup>®</sup> process design tool*. In Section 6.5 we discuss how it has been applied in some chemical process design problems.

### 6.3.3. *Some Other Interactive Methods*

A natural way of developing new methods is hybridizing ideas of different existing ones. It is, e.g., fruitful to hybridize ideas of *a posteriori* and interactive methods. In this way, the DM can both get a general overview of the possibilities and limitations of the problem and direct the search to a desired direction in order to find the most preferred solution. An example of such a method is introduced in Miettinen *et al.* (2003a), where NIMBUS (see Subsection 6.3.2.2) is hybridized with the feasible goals method (Lotov *et al.*, 2004). The latter generates visual interactive displays of the feasible objective vectors which helps the DM in understanding what kinds of solutions are available. Then it is easier to make classifications for NIMBUS.

Another hybrid is described in Klamroth and Miettinen (2007), where an adaptive approximation method (Klamroth *et al.*, 2002) approximating the Pareto optimal set is hybridized with reference point ideas. This means that the approximation is made more accurate only in those parts of the Pareto optimal set that the DM is interested in. Finally, let us mention one more hybrid method where reference points and achievement scalarizing functions are hybridized in EMO, see Thiele *et al.* (2007). On a general level, the idea is the same as in the previous hybrid but here the achievement scalarizing function is incorporated in the fitness evaluation and the interactive algorithm is different. Other ideas of handling preferences in EMO are surveyed in Coello (2000).

## 6.4. Interactive Approaches in Chemical Process Design

Multi-objective optimization has been applied to problems in chemical engineering frequently during the last 25 years (see, e.g., Andersson (2000);

Chakraborty and Linninger (2002); Clark and Westerberg (1983); Kawajiri and Biegler (2006a); Ko and Moon (2002); Lim *et al.* (1999); Subramani *et al.* (2003)) as described in Chapter 2. Usually, only two objective functions have been considered and the Pareto optimal set has been approximated by using either the weighting method or the  $\varepsilon$ -constraint method. In other words, the simplest methods have been used and sometimes the authors have not realized that they have been using a multi-objective optimization method. As mentioned, regardless of their simplicity, these methods have serious drawbacks. Recently, EMO methods have become popular in solving chemical engineering problems, but still only two or three objectives have been considered maybe due to the limitations of EMO approaches discussed earlier (Bhaskar *et al.*, 2000; Rajesh *et al.*, 2001; Roosen *et al.*, 2003; Subramani *et al.*, 2003; Tarafder *et al.*, 2005; Zhang *et al.*, 2002).

Interactive multi-objective optimization methods have considerable advantages over the methods mentioned above. However, they have been used very rarely in chemical engineering. For example, interactive methods can not be found in the survey of Marler and Arora (2004) and they are only briefly mentioned in Andersson (2000) and Bhaskar *et al.* (2000). This might be due to the lack of knowledge of interactive methods or the lack of appropriate interactive multi-objective optimization software. The few examples of interactive multi-objective optimization in chemical engineering include Grauer *et al.* (1984) and Umeda and Kuriyama (1980).

In what follows, we describe and summarize research on multi-objective optimization in chemical engineering reported in Hakanen (2006) and Hakanen *et al.* (2004, 2005, 2006a,b, 2007). These studies have focused on offering chemical engineering an efficient and practical way of handling all the necessary aspects of the problem, that is, to be able to simultaneously consider several conflicting objective functions that affect the behaviour of the problem considered. Therefore, they have been solved using the interactive NIMBUS method.

### 6.5. Applications of Interactive Approaches

Interactive multi-objective optimization can successfully be applied in chemical process design problems. For example, encouraging experiences related to papermaking and sugar industries have been obtained in Hakanen *et al.* (2004, 2005, 2006a,b, 2007). The reported solutions of these industrial problems are based on utilizing the NIMBUS method and IND-NIMBUS<sup>®</sup> and involving DMs having experience and knowledge about the problems in

question. In this section, we shortly describe problems related to simulated moving bed processes, water allocation in a paper mill and a heat recovery system design. The interactive solution process is described in more detail for the first problem in order to give an idea of the interaction between the method and the DM. Other problems are described on a more general level with further references.

### 6.5.1. Simulated moving bed processes

Simulated moving bed (SMB) processes are related to the separation of chemical products. Efficient purification techniques are crucial in chemical process industries. Liquid chromatographic separation has been widely used for products with an extremely high boiling point, or thermally unstable products such as proteins. In liquid chromatographic processes, a small amount of feed mixture is supplied to an end of a column which is packed with adsorbent particles, and then pushed to the other end with desorbent (water, organic solvent, or mixture of these).

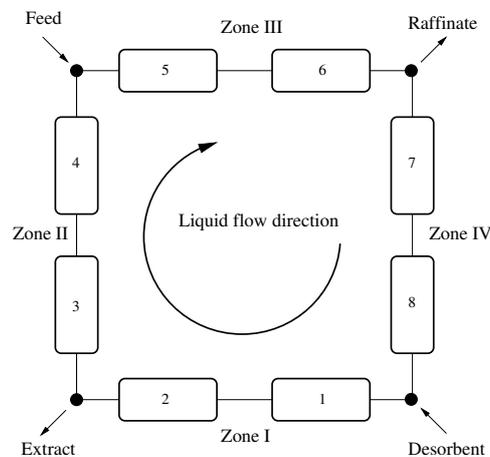


Fig. 6.2. A schematic diagram of an SMB process.

An SMB process is a realization of a continuous and counter-current operation of a liquid chromatographic separation process and it emerged from the industry in the 1960's (Ruthven and Ching, 1989). An SMB unit consists of multiple columns which are packed with adsorbent particles. The columns are connected to each other making a circulation loop, see Fig. 6.2 (with eight columns). The feed mixture is inserted into the process

in the upper left corner, while desorbent input is in the lower right corner. The two products, raffinate and extract, are collected in the upper right corner and the lower left corner, respectively. Feed mixture and desorbent are supplied between columns continuously. At the same time, the two products are withdrawn from the loop also continuously. The two inlet and two outlet streams are switched in the direction of the liquid flow at a regular interval. Because of the four inlet/outlet streams, the SMB loop has four liquid velocity zones as shown in Fig. 6.2.

The SMB model consists of partial differential equations (PDEs) for the concentrations of chemical components, restrictions for the connections between different columns and cyclic steady-state constraints (Kawajiri and Biegler, 2006b). Previously, the SMB processes have been usually optimized with respect to one objective only. Recently, multi-objective optimization has been applied in periodic separation processes (Ko and Moon, 2002), in gas separation and in SMB processes (Subramani *et al.*, 2003). Ko and Moon used a modified sum of weighted objective functions to obtain a representation of the Pareto optimal set. Their approach is valid for two objective functions only. On the other hand, Subramani *et al.* applied EMO to a problem where they had two or three objective functions.

In order to accelerate the process optimization, Kawajiri and Biegler (2006b) have developed an efficient full discretization approach combined with a large-scale nonlinear programming method for the optimization of SMBs. More recently, they have extended this approach to a superstructure SMB formulation and used the  $\varepsilon$ -constraint method to solve the bi-objective problem, where throughput and desorbent consumption were optimized (Kawajiri and Biegler, 2006a).

We can say that interactive methods have not been used to optimize SMB processes and, usually, only one or two objective functions have been considered. The advantages of interactive multi-objective optimization in SMB processes has been demonstrated in Hakanen *et al.* (2006b, 2007) for the separation of fructose and glucose (the values of the parameters in the SMB model used come from Hashimoto *et al.* (1983); Kawajiri and Biegler (2006b)). In Hakanen *et al.* (2006b, 2007), the problem formulation consists of four objective functions: maximize throughput (T, [m/h]), minimize consumption of solvent in the desorbent stream (D, [m/h]), maximize product purity (P, [%]), and maximize recovery of the valuable component in the product (R, [%]). In Hakanen *et al.* (2007), a standard formulation of the SMB model is used while a superstructure formulation of SMBs is used in Hakanen *et al.* (2006b). The superstructure formulation is a more general

way to represent SMBs and it can produce novel SMB operating schemes (Kawajiri and Biegler, 2006a).

Fig. 6.3 shows the differences between standard and superstructure SMB configurations. The standard configuration has only one fixed place for each input and output stream whereas the superstructure SMB allows more diverse configurations because the input and output streams can be placed in some of the alternative positions shown in Fig. 6.3.

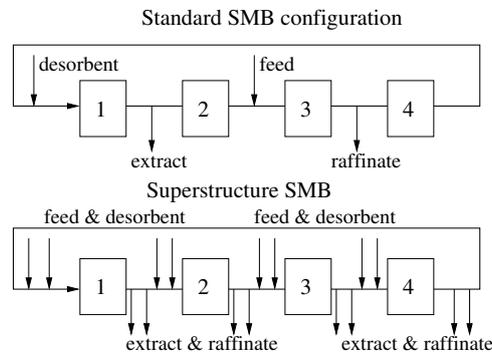


Fig. 6.3. A schematic diagram of the standard and the superstructure SMB processes.

For the PDE model of the SMB process, full discretization was used, that is, both temporal and spatial variables were discretized leading to a huge system of algebraic equations. The standard SMB optimization problem has 33 997 decision variables and 33 992 equality constraints while the superstructure SMB optimization problem has 34 102 decision variables and 34 017 equality constraints. Note that there are many more degrees of freedom in the superstructure formulation (altogether 85) than in the standard SMB formulation (5 degrees of freedom).

The SMB process is a dynamic process operating on periodic cycles which makes it a challenging optimization problem. The IPOPT optimizer (Wächter and Biegler, 2006) was used within the IND-NIMBUS<sup>®</sup> software (as an underlying optimizer) to produce new Pareto optimal solutions. The IPOPT optimizer was chosen because it has been developed for solving large scale optimization problems.

In what follows, we describe the interactive solution process for the standard SMB process using the IND-NIMBUS<sup>®</sup> process design tool. For further details, see Hakanen *et al.* (2007). The aim here is to give an understanding of the nature of an interactive solution process. The

DM involved was an expert in SMB processes. First, in the initialization phase, the ranges in the Pareto optimal set were computed as  $\mathbf{z}^* = (0.891, 0.369, 97.2, 90.0)^T$  and  $\mathbf{z}^{\text{nad}} = (0.400, 2.21, 90.0, 70.0)^T$ . A neutral compromise solution  $\mathbf{f}(\mathbf{x}^1) = (0.569, 1.58, 92.5, 76.9)^T$  was the starting point for the interactive solution process. Remember that the objective functions represented throughput (T), consumption of desorbent (D), purity (P) and recovery (R) and their values are here presented in objective vectors in this order (T, D, P, R). Note that the second objective function was minimized while the others were maximized.

In  $\mathbf{f}(\mathbf{x}^1)$ , the DM wanted to improve purity and throughput while desorbent consumption and recovery were allowed to deteriorate till specified levels. Therefore, the DM made the classification  $I^{\text{imp}}=\{\text{P}\}$ ,  $I^{\text{asp}}=\{\text{T}\}$ ,  $\bar{z}_T = 0.715$ ,  $I^{\text{bound}}=\{\text{D,R}\}$  with  $\epsilon_D = 1.78$  and  $\epsilon_R = 74.5$ . The DM wanted to get four new solutions and they were  $\mathbf{f}(\mathbf{x}^2) = (0.569, 1.56, 93.3, 74.5)^T$ ,  $\mathbf{f}(\mathbf{x}^3) = (0.553, 1.43, 94.8, 70.0)^T$ ,  $\mathbf{f}(\mathbf{x}^4) = (0.412, 1.07, 97.0, 70.0)^T$  and  $\mathbf{f}(\mathbf{x}^5) = (0.570, 1.52, 93.9, 72.4)^T$ . All the new solutions had a better purity than  $\mathbf{f}(\mathbf{x}^1)$  but the bounds in the classification for D and R did not allow throughput to improve as much as the DM would have liked. Among the new solutions, he selected  $\mathbf{f}(\mathbf{x}^3)$  as the basis of the next classification.

Next, he wanted to explore trade-off between improving recovery and letting desorbent consumption deteriorate (purity and throughput were satisfactory at the moment). The classification  $I^{\text{asp}}=\{\text{R}\}$ ,  $\bar{z}_R = 0.796$ ,  $I^{\text{sat}}=\{\text{P,T}\}$ ,  $I^{\text{bound}}=\{\text{D}\}$  with  $\epsilon_D = 1.78$  was made and three different solutions were obtained:  $\mathbf{f}(\mathbf{x}^6) = (0.497, 1.41, 93.9, 77.2)^T$ ,  $\mathbf{f}(\mathbf{x}^7) = (0.481, 1.36, 94.2, 77.3)^T$  and  $\mathbf{f}(\mathbf{x}^8) = (0.515, 1.46, 93.5, 77.1)^T$ . The new solutions had a better recovery but it could only be achieved at the expense of throughput and purity. The DM liked  $\mathbf{f}(\mathbf{x}^7)$  best because of the recovery and desorbent consumption. However, the purity was not so good.

In order to get a better understanding of the effects of the purity, the DM wanted to generate three intermediate solutions between  $\mathbf{f}(\mathbf{x}^4)$  with the best purity and  $\mathbf{f}(\mathbf{x}^7)$ . The new solutions obtained were  $\mathbf{f}(\mathbf{x}^9) = (0.426, 1.14, 96.3, 72.8)^T$ ,  $\mathbf{f}(\mathbf{x}^{10}) = (0.443, 1.21, 95.6, 74.8)^T$  and  $\mathbf{f}(\mathbf{x}^{11}) = (0.461, 1.29, 95.0, 76.2)^T$ . The DM found  $\mathbf{f}(\mathbf{x}^{11})$  to be very well balanced between all the objectives and selected it as the final, most preferred solution. The solution process was thus terminated. To summarize, we can say that the DM could conveniently direct the solution process according to his preferences and obtain a satisfactory solution without too much cognitive burden. The information exchanged was intuitive and understandable for the DM. Let us point out that, altogether, only eleven

Pareto optimal solutions were generated and, thus, the computational cost was rather low.

Considering an SMB design problem with four objective functions was a novel approach because, previously, only two or three objective functions had been considered. This enabled full utilization of the properties of the SMB model without any unnecessary simplifications. In addition, the DM obtained more thorough understanding of the interrelationships of different objectives considered and, thus, learned more about the problem.

Even better solutions can be found by using the superstructure formulation of an SMB process (when compared to the standard formulation), see Hakanen *et al.* (2006b). Although producing Pareto optimal solutions is somewhat more time consuming for the superstructure formulation (because of more complicated formulas used), the model can describe the problem better and the DM could find a very satisfactory solution, as described in Hakanen *et al.* (2006b).

### **6.5.2. Water allocation problem**

Next, we consider an application related to the paper making process and study a water allocation problem for an integrated plant containing a thermomechanical pulping plant and a paper mill. For details of this problem with three objectives, see Hakanen *et al.* (2004).

The water management in paper making is guided by the need to produce paper efficiently. The process requires fresh water to keep the disturbing contaminants on a level that is acceptable for both paper quality and machine runnability. In modern mills, the water consumption has been pressed down to 5-10 m<sup>3</sup> per a ton of paper by matching water sources (e.g. filtrates) with water sinks (e.g. dilution duties) as illustrated in Fig. `water_allocation.jpg` on the CD attached. First, wood is processed into pulp (with refining, screening, washing and bleaching). Then, the pulp is led to the paper machine. During the paper making process, (fresh) water is needed for various diluting duties in the washers and screening.

The upper part of Fig. `water_allocation.jpg` represents the thermomechanical pulping plant and the lower part represents the paper mill. The goal is to minimize the amount of fresh water taken into the process and also to minimize the amount of dissolved organic material in critical parts of the process by determining the right recycling of water. These objectives are clearly conflicting because if the amount of fresh water is reduced, then more organic material is accumulated into the water in the process.

On the other hand, if the amount of fresh water is increased, then more organic material exits the process in waste water and the concentration of the organic material decreases.

The problem has three objective functions. The first of them describes the concentration of dissolved organic material in the white water of the paper machine. (White water is water that is removed from paper web in the paper machine.) This objective has an impact on the use of chemicals and quality of paper produced. The second objective describes the concentration of dissolved organic material in the pulp entering the bleaching process. This influences the pulp brightness and the use of bleaching chemicals. The third objective describes the amount of fresh water taken into the paper making process. The problem has two inequality constraints that restrict the consistency of the pulp going to the bleaching press and the washing press. The eight decision variables are of two types: splitters and valves.

As described, the water allocation problem is a multi-objective optimization problem by nature. Previously, this problem has typically been solved using the  $\varepsilon$ -constraint method by optimizing the consumption of fresh water while turning the other two objectives into inequality constraints. However, this approach can produce only one solution at a time corresponding to the upper bounds set for the new inequality constraints. It can also be quite difficult to set correct upper bounds to find the most desirable solution without knowing the behaviour of the problem well enough and the roles of the objective functions and the constraints may be varied. If an interactive approach, like the IND-NIMBUS<sup>®</sup> process design tool, is used instead, different solutions can be generated according to the preferences of the DM and the study of the interrelationships of the different objective functions is more flexible.

The consistencies of dissolved organic material in the white water of the paper machine and in the pulp entering the bleaching have no exact upper bounds which makes using the  $\varepsilon$ -constraint method poorly justified. Therefore, the selection of the best solution is not self-evident but the role of the DM is emphasized and, thus, a need for an interactive solution method is obvious. In Hakanen *et al.* (2004), the DM wanted primarily to study the effect of the first two objectives on the fresh water consumption because the approach was more flexible for this purpose than the  $\varepsilon$ -constraint method (which had been used earlier). The solution process with the IND-NIMBUS<sup>®</sup> process design tool provided a better understanding of the interrelationships of the objective functions when compared to the previous

studies. The DM was able to rigorously study these relationships with various levels for dissolved organic materials and what kind of overall effect they have on the fresh water consumption. For a detailed description of the interactive solution process, see Hakanen *et al.* (2004).

### **6.5.3. Heat recovery system design**

Finally, we discuss an example of designing a heat recovery system for the process water system of a paper mill (see Fig. `heat_recovery.jpg` on the CD attached). The problem involving four objective functions has been described and solved in Hakanen *et al.* (2005, 2006a). We consider a virtual fine paper mill operating in a climate typical to northern latitudes, where ambient temperature varies according to the season.

The aim is to organize the heat management of the process water system in the most efficient way. A special characteristic of this optimization problem is to consider the effect of seasonal changes in the climate that affect heat management. For example, fresh water taken into the process is much colder in the winter than in the summer. In practice, fresh water taken into the paper making process needs to be heated to the process temperature 60 °C. The heat sources available are the effluent from the process (at around 50 °C), which needs to be cooled down to around 37 °C to be suitable for an effluent treatment process, and dryer exhaust, which is moist air at the temperature 85 °C. In addition, steam can be used for the final heating of process water and effluent temperature can be controlled by external cooling or heating.

The design task is to determine the amount of heat recovered from the heat sources of the process to the heat sinks, that is, to estimate the size of heat exchangers and the amount of external energy needed for heating or cooling of the effluent coming from the paper machine. Both summer and winter scenarios are included in the model by combining two parallel process models for summer and winter conditions that are solved simultaneously. Ambient air/water temperatures are 20/20 °C and -5/2 °C for summer and winter, respectively.

If the heat management was designed only according to winter conditions, the sizing of the heat recovery system would be too large resulting in large investment costs. On the other hand, if it was designed only for summer conditions, the energy consumption would be very high, because the heat recovery system would then be too small-sized. The higher the degree of heat recovery, the less external energy is required to satisfy the needs

of the process. On the other hand, the size of heat exchangers (and hence investment costs) rises with an increased degree of heat recovery. Seasonal changes add to the complexity of the problem, since a recovery system designed for winter conditions can be oversized for summer conditions and lead to a need for external cooling, for instance.

As mentioned, the main trade-off here is between running costs, that is energy, and investment costs. Typically, this trade-off is handled with single objective optimization by formulating an objective function that consists of annualised energy and investment costs with estimated amortisation time and interest rate for the capital. The cooling or heating of effluent before treatment can be primarily either energy or investment cost. In case of heating or cooling with water, the running costs will dominate. However, in many cases, the use of water for cooling is not possible, and then a cooling facility is needed, which in the design phase is mainly an investment cost.

Instead of trying to estimate all relevant aspects to get a single objective function, we can formulate four separate objective functions to be minimized: steam needed for heating water for both summer and winter conditions, estimation of area for heat exchangers (heat exchange from effluent and dryer exhaust in winter conditions, which represents the maximum values for the exchangers), and the amount of cooling or heating needed for the effluent. The first two objectives tell how much, on the average, we need to provide steam for the system and give also an estimation of the size of a steam distribution system needed. The third objective describes the amount of heat exchange area needed. Once the area is known, we can estimate investment costs more accurately from real vendor data rather than using a general cost correlation. The fourth objective, the amount of energy needed to regulate the temperature of the effluent, is really an indication of the goodness of the design, since a value deviating from zero indicates either an additional investment (i.e., a cooling tower) or need for, e.g., steam. Finally, the three decision variables are the area of the effluent heat exchanger and the approach temperatures of the dryer exhaust heat exchangers for both summer and winter operations.

As said, traditionally, this type of a problem has been formulated as a single objective optimization problem hiding the interrelationships between the objectives. Then, monetary values have to be assigned *a priori* to energy and investments with a large degree of uncertainty in the correlations. Our approach eliminates these uncertainties and leaves it to the DM to assess the costs and their uncertainties *a posteriori* when the required energy and material flows are much better defined. Having four objective functions

causes no troubles for an interactive method like NIMBUS and the problem can conveniently be solved, new insight into the problem obtained and a satisfactory solution found. For a detailed description of the interactive solution process with NIMBUS, see Hakanen *et al.* (2005, 2006a).

## 6.6. Conclusions

We have introduced some interactive methods for multi-objective optimization problems and discussed their advantages. Interactive approaches allow the DM to learn about the problem considered and the interrelationships in it. In that way, (s)he gets deeper understanding of the phenomena in question. Because the DM can direct the search for the most preferred solution, only solutions that are interesting to him/her are generated which means savings in computation time. For computationally demanding problems, this may be a significant advantage. It is important that interactive methods can be applied to problems having more than two objective functions and, thus, the true nature of the problem can be taken into account.

If the problem considered has only two objective functions, methods generating a representation of the Pareto optimal set, like EMO approaches can be applied because it is simple to visualize the solutions on a plane. However, when the problem has more than two objectives, the visualization is no longer trivial and interactive approaches offer a viable alternative to solve the problem without artificial simplifications. Because interactive methods rely heavily on the preference information specified by the DM, it is important to select such a user-friendly method where the style of specifying preferences is convenient for the DM. In addition, the specific features of the problem to be solved must be taken into consideration.

We have shown with three applications how interactive multi-objective optimization can be utilized in chemical process design and demonstrated the benefits an interactive approach can offer. In all the cases, it was possible to solve the problems in their true multi-objective character and an efficient tool was created to support the DM (or designer) in the decision making problem.

Besides describing the potential of interactive methods, we have also discussed some properties of widely used methods because their shortcomings do not seem to be generally known. However, it is important that when selecting a method, the limitations set by the method are known. Otherwise, the solutions obtained may not give a truthful impression about the problem in question.

## References

- Andersson, J. (2000). A survey of multiobjective optimization in engineering design, Tech. Rep. LiTH-IKP-R-1097, Department of Mechanical Engineering, Linköping University.
- Bhaskar, V., Gupta, S. and Ray, A. (2000). Applications of multiobjective optimization in chemical engineering. *Reviews in Chemical Engineering* **16**, pp. 1–54.
- Branke, J., Deb, K., Miettinen, K. and Slowinski, R. (eds.) (2008). *Multiobjective Optimization: Interactive and Evolutionary Approaches*, Lecture Notes in Computer Science, State-of-the-Art Survey (Springer-Verlag), to appear.
- Buchanan, J. T. (1986). Multiple objective mathematical programming: A review, *New Zealand Operational Research* **14**, pp. 1–27.
- Chakraborty, A. and Linninger, A. (2002). Plant-wide waste management. 1. synthesis and multiobjective design, *Industrial & Engineering Chemistry Research* **41**, pp. 4591–4604.
- Chankong, V. and Haimes, Y. Y. (1983). *Multiobjective Decision Making: Theory and Methodology* (Elsevier Science Publishing, New York).
- Clark, P. and Westerberg, A. (1983). Optimization for design problems having more than one objective, *Computers & Chemical Engineering* **7**, pp. 259–278.
- Coello, C. (2000). Handling preferences in evolutionary multiobjective optimization: A survey, in *Proceedings of the 2000 Congress on Evolutionary Computation* (IEEE Service Center, Piscataway, NJ), pp. 30–37.
- Das, I. and Dennis, J. E. (1997). A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems, *Structural Optimization* **14**, 1, pp. 63–69.
- Edgeworth, F. Y. (1881). *Mathematical Psychics: An Essay on the Application of Mathematics to the Moral Sciences* (C. Kegan Paul & Co., London), University Microfilms International (Out-of-Print Books on Demand), 1987.
- Gass, S. and Saaty, T. (1955). The computational algorithm for the parametric objective function, *Naval Research Logistics Quarterly* **2**, pp. 39–45.
- Grauer, M., Lewandowski, A. and Wierzbicki, A. (1984). Multiple-objective decision analysis applied to chemical engineering, *Computers & Chemical Engineering* **8**, pp. 285–293.
- Haimes, Y. Y., Lasdon, L. S. and Wismer, D. A. (1971). On a bicriterion formulation of the problems of integrated system identification and system

- optimization, *IEEE Transactions on Systems, Man, and Cybernetics* **1**, pp. 296–297.
- Haines, Y. Y., Tarvainen, K., Shima, T. and Thadathil, J. (1990). *Hierarchical Multiobjective Analysis of Large-Scale Systems* (Hemisphere Publishing Corporation, New York).
- Hakanen, J. (2006). *On Potential of Interactive Multiobjective Optimization in Chemical Process Design*, Jyväskylä Studies in Computing 68 (PhD thesis, University of Jyväskylä, Jyväskylä).
- Hakanen, J., Hakala, J. and Manninen, J. (2006a). An integrated multiobjective design tool for process design, *Applied Thermal Engineering* **26**, pp. 1393–1399.
- Hakanen, J., Kawajiri, Y., Miettinen, K. and Biegler, L. (2006b). Interactive multiobjective optimization of superstructure SMB processes, Talk in the MOPGP07 Conference, Tours, France.
- Hakanen, J., Kawajiri, Y., Miettinen, K. and Biegler, L. (2007). Interactive multiobjective optimization for simulated moving bed processes, *Control & Cybernetics* **36**, 2.
- Hakanen, J., Miettinen, K. and Mäkelä, M. (2004). An application of multiobjective optimization to process simulation, in P. Neittaanmäki, T. Rossi, S. Korotov, E. Onate, J. Periaux and D. Knörzer (eds.), *Proceedings of ECCOMAS 2004*, Vol. II (Jyväskylä, Finland), CD-ROM.
- Hakanen, J., Miettinen, K., Mäkelä, M. and Manninen, J. (2005). On interactive multiobjective optimization with NIMBUS in chemical process design, *Journal of Multicriteria Decision Analysis* **13**, pp. 125–134.
- Hämäläinen, J., Miettinen, K., Tarvainen, P. and Toivanen, J. (2003). Interactive solution approach to a multiobjective optimization problem in paper machine headbox design, *Journal of Optimization Theory and Applications* **116**, 2, pp. 265–281.
- Hashimoto, K., Adachi, S., Noujima, H. and Maruyama, H. (1983). Models for the separation of glucose/fructose mixture using a simulated moving bed adsorber, *Journal of Chemical Engineering of Japan* **16**, 5, pp. 400–406.
- Heikkola, E., Miettinen, K. and Nieminen, P. (2006). Multiobjective optimization of an ultrasonic transducer using NIMBUS, *Ultrasonics* **44**, pp. 368–380.
- Hwang, C. L. and Masud, A. S. M. (1979). *Multiple objective decision making – methods and applications: A state-of-the-art survey* (Springer-Verlag).
- Jaszkiewicz, A. and Slowinski, R. (1999). The ‘light beam search’ approach – an overview of methodology and applications, *European Journal of Operational Research* **113**, pp. 300–314.
- Kawajiri, Y. and Biegler, L. (2006a). A nonlinear programming superstructure for optimal dynamic operations of simulated moving bed processes, *Industrial & Engineering Chemistry Research* **45**, pp. 8503–8513.
- Kawajiri, Y. and Biegler, L. (2006b). Optimization strategies for simulated moving bed and powerfeed processes, *AIChE Journal* **52**, 4, pp. 1343–1350.
- Klamroth, K. and Miettinen, K. (2007). Integrating approximation and interactive decision making in multicriteria optimization, *Operations Research To appear*.

## References

31

- Klamroth, K., Tind, J. and Wiecek, M. M. (2002). Unbiased approximation in multicriteria optimization, *Mathematical Methods of Operations Research* **56**, pp. 413–437.
- Ko, D. and Moon, I. (2002). Multiobjective optimization of cyclic adsorption processes, *Industrial & Engineering Chemistry Research* **41**, 1, pp. 93–104.
- Koopmans, T. (1951). Analysis and production as an efficient combination of activities, in T. Koopmans (ed.), *Activity Analysis of Production and Allocation: Proceedings of a Conference* (John Wiley & Sons, New York), pp. 33–97, Yale University Press, New Haven, London, 1971.
- Kuhn, H. and Tucker, A. (1951). Nonlinear programming, in J. Neyman (ed.), *Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probability* (University of California Press, Berkeley, LA), pp. 481–492.
- Larichev, O. (1992). Cognitive validity in design of decision aiding techniques, *Journal of Multi-Criteria Decision Analysis* **1**, 3, pp. 127–138.
- Lim, Y., Floquet, P., Joulia, X. and Kim, S. (1999). Multiobjective optimization in terms of economics and potential environment impact for process design and analysis in a chemical process simulator, *Industrial & Engineering Chemistry Research* **38**, pp. 4729–4741.
- Lotov, A. V., Bushenkov, V. A. and Kamenev, G. K. (2004). *Interactive Decision Maps. Approximation and Visualization of Pareto Frontier* (Kluwer Academic Publishers, Boston).
- Madetoja, E., Miettinen, K. and Tarvainen, P. (2006). Issue related to the computer realization of a multidisciplinary and multiobjective optimization system, *Engineering and Computing* **22**, pp. 33–46.
- Mäkelä, M. and Neittaanmäki, P. (1992). *Nonsmooth Optimization: Analysis and Algorithms with Applications to Optimal Control* (World Scientific Publishing Co., Singapore).
- Marler, R. and Arora, J. (2004). Survey of multi-objective optimization methods for engineering, *Structural and Multidisciplinary Optimization* **26**, 6, pp. 369–395.
- Miettinen, K. (1999). *Nonlinear Multiobjective Optimization* (Kluwer Academic Publishers, Boston).
- Miettinen, K. (2006). IND-NIMBUS for demanding interactive multiobjective optimization, in T. Trzaskalik (ed.), *Multiple Criteria Decision Making '05* (The Karol Adamiecki University of Economics in Katowice, Katowice), pp. 137–150.
- Miettinen, K., Lotov, A. V., Kamenev, G. K. and Berezkin, V. E. (2003a). Integration of two multiobjective optimization methods for nonlinear problems, *Optimization Methods and Software* **18**, 1, pp. 63–80.
- Miettinen, K. and Mäkelä, M. (1995). Interactive bundle-based method for non-differentiable multiobjective optimization: NIMBUS, *Optimization* **34**, pp. 231–246.
- Miettinen, K. and Mäkelä, M. (1999). Comparative evaluation of some interactive reference point-based methods for multi-objective optimisation, *Journal of the Operational Research Society* **50**, 9, pp. 949–959.
- Miettinen, K. and Mäkelä, M. (2000). Interactive multiobjective optimization

- system WWW-NIMBUS on the Internet, *Computers & Operations Research* **27**, pp. 709–723.
- Miettinen, K. and Mäkelä, M. (2002). On scalarizing functions in multiobjective optimization, *OR Spectrum* **24**, pp. 193–213.
- Miettinen, K. and Mäkelä, M. (2006). Synchronous approach in interactive multiobjective optimization, *European Journal of Operations Research* **170**, 3, pp. 909–922.
- Miettinen, K., Mäkelä, M. and Toivanen, J. (2003b). Numerical comparison of some penalty-based constraint handling techniques in genetic algorithms, *Journal of Global Optimization* **27**, pp. 427–446.
- Miettinen, K. and Mäkelä, M. M. (2002). On scalarizing functions in multiobjective optimization, *OR Spectrum* **24**, 2, pp. 193–213.
- Miettinen, K., Mäkelä, M. M. and Kaario, K. (2006). Experiments with classification-based scalarizing functions in interactive multiobjective optimization, *European Journal of Operational Research* **175**, 2, pp. 931–947.
- Miettinen, K., Mäkelä, M. M. and Männikkö, T. (1998). Optimal control of continuous casting by nondifferentiable multiobjective optimization, *Computational Optimization and Applications* **11**, 2, pp. 177–194.
- Nakayama, H. (1995). Aspiration level approach to interactive multi-objective programming and its applications, in P. Pardalos, Y. Siskos and C. Zopounidis (eds.), *Advances in Multicriteria Analysis* (Kluwer Academic Publishers), pp. 147–174.
- Nakayama, H. and Furukawa, K. (1985). Satisficing trade-off method with an application to multiobjective structural design, *Large Scale Systems* **8**, pp. 47–57.
- Nakayama, H. and Sawaragi, Y. (1984). Satisficing trade-off method for multiobjective programming, in M. Grauer and A. Wierzbicki (eds.), *Interactive Decision Analysis* (Springer-Verlag), pp. 113–122.
- Ojalehto, V., Miettinen, K. and Mäkelä, M. (2007). Interactive software for multiobjective optimization: IND-NIMBUS, *WSEAS Transactions on Computers* **6**, 1, pp. 87–94.
- Pareto, V. (1896). *Cours d'Economie Politique* (Rouge, Lausanne).
- Pareto, V. (1906). *Manuale di Economia Politica* (Piccola Biblioteca Scientifica, Milan), translated into English by Ann S. Schwier (1971), *Manual of Political Economy*, MacMillan, London.
- Podinovski, V. V. (1994). Criteria importance theory, *Mathematical Social Sciences* **27**, 3, pp. 237–252.
- Rajesh, J., Gupta, S., Rangaiah, G. and Ray, A. (2001). Multi-objective optimization of industrial hydrogen plants, *Chemical Engineering Science* **56**, pp. 999–1010.
- Roosen, P., Uhlenbruck, S. and Lucas, K. (2003). Pareto optimization of a combined cycle power system as a decision support tool for trading off investment vs. operating cost, *International Journal of Thermal Sciences* **42**, pp. 553–560.
- Roy, B. and Mousseau, V. (1996). A theoretical framework for analysing the notion of relative importance of criteria, *Journal of Multi-Criteria Decision*

- Analysis* **5**, 2, pp. 145–159.
- Ruthven, D. and Ching, C. (1989). Counter-current and simulated counter-current adsorption separation processes, *Chemical Engineering Science* **44**, pp. 1011–1038.
- Sawaragi, Y., Nakayama, H. and Tanino, T. (1985). *Theory of Multiobjective Optimization* (Academic Press, Orlando, FL).
- Statnikov, R. B. (1999). *Multicriteria Design: Optimization and Identification* (Kluwer Academic Publishers, Dordrecht).
- Steuer, R. E. (1986). *Multiple Criteria Optimization: Theory, Computation, and Application* (John Wiley & Sons, New York).
- Stewart, T. J. (1992). A critical survey on the status of multiple criteria decision making theory and practice, *OMEGA* **20**, 5–6, pp. 569–586.
- Subramani, H., Hidayat, K. and Ray, A. (2003). Optimization of reactive SMB and varicol systems, *Computers & Chemical Engineering* **27**, pp. 1883–1901.
- Tabucanon, M. T. (1988). *Multiple Criteria Decision Making in Industry* (Elsevier Science Publishers, Amsterdam).
- Tanner, L. (1991). Selecting a text-processing system as a qualitative multiple criteria problem, *European Journal of Operational Research* **50**, pp. 179–187.
- Tarafder, A., Rangaiah, G. and Ray, A. (2005). Multiobjective optimization of an industrial styrene monomer manufacturing process, *Chemical Engineering Science* **60**, pp. 347–363.
- Thiele, L., Miettinen, K., Korhonen, P. J. and Molina, J. (2007). A preference-based interactive evolutionary algorithm for multiobjective optimization, Working Papers W-412, Helsinki School of Economics, Helsinki.
- Umeda, T. and Kuriyama, T. (1980). Interactive solution to multiple criteria problems in chemical process design, *Computers & Chemical Engineering* **4**, pp. 157–165.
- Vanderpooten, D. and Vincke, P. (1989). Description and analysis of some representative interactive multicriteria procedures, *Mathematical and Computer Modelling* **12**, pp. 1221–1238.
- Vincke, P. (1992). *Multicriteria Decision-Aid* (John Wiley & Sons, Chichester).
- Wächter, A. and Biegler, L. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, *Mathematical Programming* **106**, pp. 25–57.
- Wierzbicki, A. (1982). A mathematical basis for satisficing decision making, *Mathematical Modelling* **3**, 25, pp. 391–405.
- Wierzbicki, A., Makowski, M. and Wessels, J. (eds.) (2000). *Model-Based Decision Support Methodology with Environmental Applications* (Kluwer Academic Publishers, Dordrecht).
- Zadeh, L. (1963). Optimality and non-scalar-valued performance criteria, *IEEE Transactions on Automatic Control* **8**, pp. 59–60.
- Zhang, Z., Hidayat, K. and Ray, A. (2002). Multiobjective optimization of SMB and varicol process for chiral separation, *AIChE Journal* **48**, pp. 2800–2816.

**Exercises**

See exercises in `MiettinenHakanenExercises.pdf` on the CD attached.

## Index

- $\varepsilon$ -constraint method, 8
- 'Light beam search', 12
- A posteriori methods, 2
- A priori methods, 2
- Achievement (scalarizing) function, 11
- Analyst, 5
- Aspiration level, 6
- Classification, 12
- Classification-based methods, 12
- Classifying objective functions, 13
- Convex multi-objective optimization, 7
- Decision maker (DM), 2, 5
- Decision variable, 4
- Feasible goals method, 18
- Feasible region, 4
- Final solution, 5
- Heat recovery system design, 26
- Hybrid methods, 18
- Ideal objective vector, 4
- IND-NIMBUS, 16
- Interactive methods, 2, 7
- IPOPT optimizer, 22
- Learning phase, 11
- Most preferred solution, 5
- Multiple criteria decision making (MCDM), 2
- Nadir objective vector, 5
- NIMBUS method, 18
- No-preference methods, 2
- Objective function, 3
- Objective value, 4
- Objective vector, 4
- Paper making, 24, 26
- Pareto optimality
  - global, 5
  - local, 5
- Preference information, 5, 9, 10, 13
- Reference point, 6, 13
- Reference point method, 11
- Satisficing trade-off method (STOM), 13
- Scalarization, 5
- Scalarizing function, 5, 10
- Simulated moving bed (SMB) processes, 20
- Solution process, 5
- Sugar separation, 21
- Superstructure SMB, 21
- Surveys of methods, 2
- Trade-off, 5
- Water allocation problem, 24
- Weak Pareto optimality, 4
- Weighting method, 6

WWW-NIMBUS, 16