



Mariano Luque – Francisco Ruiz – Kaisa Miettinen

GLIDE – GENERAL FORMULATION FOR INTERACTIVE MULTIOBJECTIVE OPTIMIZATION

Mariano Luque* – Francisco Ruiz* – Kaisa Miettinen**

GLIDE – GENERAL FORMULATION FOR INTERACTIVE MULTIOBJECTIVE OPTIMIZATION

*University of Malaga,

**Helsinki School of Economics,
Dept. of Business Technology

June
2007

HELSINGIN KAUPPAKORKEAKOULU
HELSINKI SCHOOL OF ECONOMICS
PL 1210
FI-00101 HELSINKI
FINLAND

© Mariano Luque, Francisco Ruiz, Kaisa Miettinen and
Helsinki School of Economics

ISSN 1235-5674
(Electronic working paper)
ISBN 978-952-488-154-8

Helsinki School of Economics -
HSE Print 2007

GLIDE – General Formulation for Interactive Multiobjective Optimization

Mariano Luque, Francisco Ruiz
University of Málaga, Calle Ejido 6, E-29071 Málaga, Spain
mluque@uma.es, rua@uma.es

Kaisa Miettinen
Helsinki School of Economics, P.O. Box 1210
FI-00101 Helsinki, Finland
kaisa.miettinen@hse.fi

Abstract

Interactive methods are commonly regarded as one of the most useful and realistic multiobjective optimization techniques and for this reason a great number of different interactive methods have been developed. However, we can say that they have two important drawbacks when it comes to using them in real applications. On the one hand, the question of which method should be chosen is anything but trivial. On the other hand, there is a relatively small number of practical implementations of the methods. In this paper, we introduce a general formulation that can accommodate several interactive methods. This provides a comfortable implementation framework for a general interactive system. Besides, this implementation allows the decision maker to choose the way (s)he wants to give information to the system, and makes it possible to change it anytime during the solution process. This is in fact a change-of-method option which provides a very flexible framework for the decision maker.

Keywords: Multiple criteria decision making, multiple objectives, interactive methods, preference information

1 Introduction

During the years, many methods have been suggested for solving multiobjective optimization problems (see, e.g., Hwang and Masud (1979); Miettinen (1999); Sawaragi *et al.* (1985)) where the aim is to find the most preferred solution in the

presence of several conflicting objectives. The task of multiobjective optimization methods is to support a decision maker (DM) in formulating her/his preferences and identifying the best of many Pareto optimal solutions. Different multiobjective optimization methods use a variety of forms of preference information and versatile techniques in generating Pareto optimal solutions. Because the task is to support a decision maker, so-called interactive approaches have turned out to be very useful and applicable. For this reason, many interactive methods, based on different solution philosophies, have been developed.

In interactive multiobjective optimization methods, a solution pattern is formed and repeated several times and at each iteration the DM can direct the search towards such Pareto optimal solutions that (s)he is interested in. The DM gets to see some information about the feasible Pareto optimal solutions available (and maybe some other information describing the trade-offs in the problem) and can specify preference information in order to find more preferable solutions. The benefit of using an interactive method is that the DM has a chance to learn about the interrelationships between the objectives and what kind of solutions are feasible for the problem. This helps in having more realistic expectations about the solutions. On the other hand, the DM can concentrate on such solutions that are interesting to her/him and, thus, it typically is sufficient to produce a relatively small amount of Pareto optimal solutions, which means low computational costs.

Because there are many interactive methods available, it is not always easy to select the method to be used. As pointed out by Kaliszewski (2004), it would be good to create environments (with a common interface) where different methods were present and the DM could freely select the method (and the way of specifying preference information) as well as switch between methods. There have been some attempts of formulating this kind of an environment. For example, Gardiner and Steuer (1994a,b) suggested a unified algorithm which incorporates nine to thirteen different methods. However, the implementation of this unified algorithm is not a straightforward task. This conclusion is supported by Kaliszewski (2004) with the statement that even though the idea was appealing, it has remained too complex for DMs. Romero (2001) presented another attempt to prepare a general optimization structure where several variants of goal programming as well as some other methods can be covered. However, the usefulness and benefits of this approach have not been demonstrated or discussed from the implementation point of view. One more generalized formulation is given in Vasileva *et al.* (2005) incorporating thirteen different subproblems used in different methods but the formulation is rather complex. As far as implementations are concerned, PROMOIN (Caballero *et al.*, 2002) is an example of an environment where several interactive methods are included but there is no general problem formulation and each method must be implemented separately. Furthermore, the synchronous NIMBUS method (Miettinen and Mäkelä, 2006) and its implementations WWW-NIMBUS[®] (<http://nimbus.it.jyu.fi/>) and IND-NIMBUS[®]

(Miettinen, 2006) contain scalarizing functions of different methods but the way to specify preferences is the same for all of them.

In this paper, we introduce a general formulation that covers eight interactive multiobjective optimization methods representing three (or actually four) different method types. The advantage of this formulation is its simple and compact structure which enables easy implementation. Furthermore, the framework presented allows the DM to conveniently change the style of expressing preference information, that is, changing the method used. However, the DM is not supposed to know different multiobjective optimization methods and their specificities but can concentrate on the actual problem to be solved and must only decide which kinds of preferences (s)he could provide in order to direct the solution process to a desired direction so that (s)he can identify the most preferred solution. Based on the preference type used, the general interactive solution scheme will choose the most appropriate method(s) for each case. The flexible possibility of changing the method means that the DM is not restricted to one way of specifying preferences. In different phases of the solution process the DM may wish to approach the problem in different ways and this is now possible. For example, at the early stages, in the so-called learning phase, the DM may wish to get a general overview of the solutions available and later on, once an interesting region of solutions has been identified, the DM may wish to fine-tune one's preferences in a smaller neighborhood. Our framework supports the DM in this and the DM has easy access to methods representing different solution philosophies.

It has often been wondered in the literature why relatively few real life applications of multiobjective optimization have been reported. One possible explanation is that computer implementations of the plethora of methods developed are not always easily accessible. Another explanation could be that they and their user interfaces are too complex for real DMs (as mentioned, e.g., in Kaliszewski (2004)). The benefit of our framework is that it is easy to implement. Thus, we can say that one of our motivations in this paper is to bridge the gap between methods developed and their real-world applicability.

In practice, our interactive solution scheme is used so that at each iteration of the solution process, the DM can decide the type of preferences (s)he wishes to specify: a reference point (or a classification of objective functions), selecting from a set of Pareto optimal solutions or marginal rates of substitution (MRSs). Our formulation contains several methods in each method type and problems corresponding to the type selected are formed (by setting appropriate variable values in the general problem formulation) and solved and the DM can decide whether to consider one such solution or more solutions based on the same preference information. Then, the DM can fine-tune one's preferences and get new solutions or continue by changing the type of preference information specified.

Let us point out that if the DM so desires, after having specified preferences in some form and obtained solution(s), it is possible to deduce what preferences of some other type could have produced the same solution, see Luque *et al.* (2007a).

This property can easily be included in our solution framework.

The rest of this paper is organized as follows. We discuss some concepts and notations in Section 2 and introduce our general problem framework, that is, GLIDE, the global formulation for interactive multiobjective optimization and the interactive solution scheme based on GLIDE in Section 3. Section 4 is devoted to an example demonstrating how our formulation can be applied and, finally, we conclude in Section 5.

2 Concepts and Notations

We consider *multiobjective optimization problems* of the form

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

involving k (≥ 2) conflicting *objective functions* $f_i : S \rightarrow \mathbb{R}$ that we want to minimize simultaneously. The *decision variables* $\mathbf{x} = (x_1, \dots, x_n)^T$ belong to the nonempty compact *feasible region* $S \subset \mathbb{R}^n$. *Objective vectors* in *objective space* \mathbb{R}^k consist of *objective values* $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))^T$ and the image of the feasible region is called the *feasible objective region* $Z = \mathbf{f}(S)$.

In multiobjective optimization, objective vectors are optimal if none of their components can be improved without deteriorating at least one of the others. More precisely, a decision vector $\mathbf{x}' \in S$ is said to be *efficient* if there does not exist another $\mathbf{x} \in S$ such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x}')$ for all $i = 1, \dots, k$ and $f_j(\mathbf{x}) < f_j(\mathbf{x}')$ for at least one index j . On the other hand, a decision vector $\mathbf{x}' \in S$ is said to be *weakly efficient* for problem (1) if there does not exist another $\mathbf{x} \in S$ such that $f_i(\mathbf{x}) < f_i(\mathbf{x}')$ for all $i = 1, \dots, k$ and \mathbf{x}' is said to be *properly efficient* if unbounded trade-offs are not allowed. The corresponding objective vectors $\mathbf{f}(\mathbf{x})$ are called (*weakly/properly*) *nondominated objective vectors*. Note that the set of properly nondominated solution is a subset of nondominated solutions which is a subset of weakly nondominated solutions. In what follows, we denote the current nondominated solution by \mathbf{f}^h .

Let us assume that for problem (1) the set of nondominated objective vectors contains more than one vector. Because it is often useful to know the ranges of objective vectors in the nondominated set, we calculate the *ideal objective vector* $\mathbf{z}^* = (z_1^*, \dots, z_k^*)^T \in \mathbb{R}^k$ by minimizing each objective function individually in the feasible region, that is, $z_i^* = \min_{\mathbf{x} \in S} f_i(\mathbf{x}) = \min_{\mathbf{x} \in E} f_i(\mathbf{x})$ for all $i = 1, \dots, k$, where E is the set of efficient solutions. This gives lower bounds for the objectives. The upper bounds, that is, the *nadir objective vector* $\mathbf{z}^{\text{nad}} = (z_1^{\text{nad}}, \dots, z_k^{\text{nad}})^T$, can be defined as $z_i^{\text{nad}} = \max_{\mathbf{x} \in E} f_i(\mathbf{x})$ for all $i = 1, \dots, k$. In practice, the nadir objective vector is usually difficult to obtain. Its components can be approximated using a pay-off table but in general this kind of an estimate is not necessarily too good (see, e.g., Miettinen (1999) and references therein). As

an alternative, we can ask from the DM the worst possible objective values (s)he could consider and use them as components of the nadir objective vector. In this way, the solution process is not disturbed by the possibly weak approximation obtained from the pay-off table.

Furthermore, sometimes a *utopian objective vector* $\mathbf{z}^{**} = (z_1^{**}, \dots, z_k^{**})^T$ is defined as a vector strictly better than the ideal objective vector. Then we set $z_i^{**} = z_i^* - \varepsilon$ ($i = 1, \dots, k$), where $\varepsilon > 0$ is a small real number. This vector can be considered instead of an ideal objective vector in order to avoid the case where ideal and nadir values are equal or very close to each other. In what follows, we assume that the set of nondominated objective vectors is bounded and that we have global estimates of the ranges of nondominated solutions available.

All nondominated solutions can be regarded as equally desirable in the mathematical sense and we need a *decision maker* (DM) to identify the most preferred one among them. A DM is a person who can express preference information related to the conflicting objectives. In this paper, it will be assumed that the solution process is carried out using some interactive method (the general features of this class of methods have been described in Section 1). Among other issues, these methods differ from each other in the kind of information asked from the DM at each iteration.

In this paper, we consider different styles of specifying preference information: reference levels (that is, levels that are regarded as desirable for the DM for each objective function; or classification, that is, a division of the objectives into classes depending on whether the DM wishes to improve, impair or maintain the current value), just choosing one solution among several ones, or MRSs, that is, the amount of decrement in the value of one objective function that compensates an infinitesimal increment in the value of another one, while the values of all the other objectives remain unaltered. In this paper we introduce a global formulation which can be transformed, by changing some parameters, into interactive methods belonging to all these three classes.

3 GLIDE: A Global Formulation for Interactive Multiobjective Optimization

We want to define a global formulation for interactive multiobjective optimization techniques. For this purpose, we define a general scalarized formulation called the GLocal Interactive Decision Environment (GLIDE), which has been designed so that it can generate different interactive techniques by changing the values of its parameters. Although eight different methods obtained from GLIDE are reported in this paper, the global formulation can accommodate other interactive procedures as well. For the multiobjective optimization problem (1), the GLIDE problem is defined as follows

$$(\text{GLIDE}) \left\{ \begin{array}{l} \text{minimize} \quad \alpha + \rho \sum_{i=1}^k \omega_i^h (f_i(\mathbf{x}) - q_i^h) \\ \text{subject to} \quad \mu_i^h (f_i(\mathbf{x}) - q_i^h) \leq \alpha \quad (i = 1, \dots, k) \\ \quad \quad \quad f_i(\mathbf{x}) \leq \varepsilon_i^h + s_\varepsilon \cdot \Delta \varepsilon_i^h \quad (i = 1, \dots, k) \\ \quad \quad \quad \mathbf{x} \in S, \end{array} \right. \quad (2)$$

where $\mathbf{x} \in \mathbb{R}^k$ and $\alpha \in \mathbb{R}$ are the variables. In addition, we have parameters $(\alpha, \rho, \omega_i^h, q_i^h, \mu_i^h, \varepsilon_i^h, s_\varepsilon$ and $\Delta \varepsilon_i^h)$ which are set depending on the kind of information the DM is willing to provide, and consequently, on the corresponding interactive method. The optimal solution obtained will be denoted by \mathbf{x}^{h+1} and the corresponding objective vector by $\mathbf{f}^{h+1} = f(\mathbf{x}^{h+1})$.

As previously mentioned, this formulation can be transformed into the (intermediate) single objective problems solved by eight different interactive methods, and therefore, the (weak, proper) efficiency of its optimal solution is guaranteed just as in the corresponding methods. Anyway, the following theorem collects general results about the efficiency of the optimal solutions of problem (2), depending on the values of some parameters.

Theorem 1 *Let \mathbf{x}^{h+1} be an optimal solution of problem (2). Then, the following statements hold:*

- (i) *If $\rho > 0$, $\omega_i^h > 0$ and $\mu_i^h > 0$ ($i = 1, \dots, k$), then \mathbf{x}^{h+1} is an efficient solution of problem (1).*
- (ii) *If $\rho = 0$ or all $\omega_i^h = 0$, and $\mu_i^h > 0$ ($i = 1, \dots, k$), then \mathbf{x}^{h+1} is a weakly efficient solution of problem (1). Moreover, if \mathbf{x}^{h+1} is a unique solution, then it is efficient.*
- (iii) *If $\rho > 0$, $\omega_i^h > 0$ and $\mu_i^h = 0$ ($i = 1, \dots, k$), then \mathbf{x}^{h+1} is efficient. If for any j we have $\omega_j^h = 0$, then \mathbf{x}^{h+1} is a weakly efficient solution of problem (1).*

Proof.

- (i) Let us suppose that $\rho > 0$, $\omega_i^h > 0$ and $\mu_i^h > 0$ for all i and that \mathbf{x}^{h+1} is not efficient. Then, there exists a feasible solution $\mathbf{x}^* \in S$ such that $f_i(\mathbf{x}^*) = f_i^* \leq f_i^{h+1}$ ($i = 1, \dots, k$) with some j such that $f_j^* < f_j^{h+1}$. Therefore, we have for all $i = 1, \dots, k$

$$\begin{aligned}
f_i^* &\leq f_i^{h+1} \leq \varepsilon_i^h + s_\varepsilon \cdot \Delta \varepsilon_i^h, \\
\mu_i^h (f_i^* - q_i^h) &\leq \mu_i^h (f_i^{h+1} - q_i^h) \leq \alpha \text{ and} \\
\rho \sum_{i=1}^k \omega_i^h (f_i^* - q_i^h) &< \rho \sum_{i=1}^k \omega_i^h (f_i^{h+1} - q_i^h),
\end{aligned}$$

which implies that \mathbf{x}^* is feasible for problem (2), and the corresponding value of the objective function is better than the objective value of \mathbf{x}^{h+1} , which contradicts the fact that \mathbf{x}^{h+1} is an optimal solution of (2). Thus, \mathbf{x}^{h+1} must be efficient.

- (ii) If $\rho = 0$ or all $\omega_i^h = 0$, and $\mu_i^h > 0$ ($i = 1, \dots, k$), then problem (2) is equivalent to solving a minmax problem with a restricted feasible region. If \mathbf{x}^{h+1} is not weakly efficient, then there exists a feasible solution $\mathbf{x}^* \in S$, such that $f_i(\mathbf{x}^*) < f_i(\mathbf{x}^{h+1})$ ($i = 1, \dots, k$).

Therefore, for each $i = 1, \dots, k$, we have:

$$\begin{aligned} f_i^* &< f_i^{h+1} \leq \varepsilon_i^h + s_\varepsilon \cdot \Delta \varepsilon_i^h, \\ \mu_i^h (f_i^* - q_i^h) &< \mu_i^h (f_i^{h+1} - q_i^h) \leq \alpha. \end{aligned}$$

Thus, \mathbf{x}^* is feasible for problem (2), and the corresponding value of the objective function is strictly better than the objective value of \mathbf{x}^{h+1} (as in case (i)). Because of this contradiction, \mathbf{x}^{h+1} must be weakly efficient. Using the same reasoning it follows that if \mathbf{x}^{h+1} is a unique optimal solution of (2), then it is efficient.

- (iii) If $\rho > 0$, $\omega_i^h > 0$ and $\mu_i^h = 0$ ($i = 1, \dots, k$), the first block of constraints of problem (2) is actually a non-negativity condition on α . Therefore, at the optimal solution α must be zero, and thus, problem (2) is equivalent to solving

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^k \omega_i^h f_i(\mathbf{x}) \\ \text{subject to} \quad & f_i(\mathbf{x}) \leq \varepsilon_i^h + s_\varepsilon \cdot \Delta \varepsilon_i^h \quad (i = 1, \dots, k) \\ & \mathbf{x} \in S, \end{aligned}$$

which is equivalent to a weighted sum problem with a restricted feasible region, whose additional constraints are the same as in the previous case. Therefore, any feasible solution with better or at least the same objective values as in \mathbf{x}^{h+1} (with at least one strictly better value) would be feasible for (2), and it would have a better objective function value. Here we again have a contradiction and, thus, \mathbf{x}^{h+1} must be efficient. Analogously, if for some j we have $\omega_j^h = 0$, then \mathbf{x}^{h+1} is weakly efficient.

It is important to point out that all the efficient (or properly efficient) solutions of problem (1) can be obtained using this global scalarized formulation with adequate values for the parameters. For example, if we set $\mu_i^h > 0$ and $\varepsilon_i^h + s_\varepsilon \cdot \Delta \varepsilon_i^h \geq z_i^{nad}$ ($i = 1, \dots, k$), we get the achievement scalarizing function defined by Wierzbicki (1980).

Besides giving a general problem formulation, we also want to design a flexible global interactive solution scheme which allows the DM to choose the kind of information (s)he wants to provide at each iteration. This will make the decision process easier for the DM, who does not have to follow the same preference elicitation pattern during the whole solution process. Rather than that, the scheme is adapted to the way the DM prefers to provide preference information in, by changing the interactive method. To this end, based on the GLIDE formulation, we propose the global interactive solution scheme described in Figure 1. Given the current solution, the DM can choose what kind of information (s)he wants to provide: *just choose a solution among several ones, reference levels or marginal rates of substitution*. The discontinuous line leading to the MRS option means that this case cannot be applied to all kinds of problems, as will be specified later in Section 3.3. Then the DM is asked to provide the corresponding preference information, and the parameters for GLIDE are set accordingly. Namely, if the reference point option is chosen, the reference points based methods (REF) or the classification methods (CLASS) can be used, as will be described in Section 3.1; if the DM wants to choose a solution among several ones, the corresponding methods (SAMPLE) are used as will be described in Section 3.2; and if the MRS option is chosen, then some methods of this type can be used, as will be described in Section 3.3.

Once the parameters have been set, problem (2) is solved. If several solutions are generated, the DM must choose one of them. After this, (s)he is asked whether (s)he is satisfied with this solution. If so, the solution process finishes. Otherwise, the DM is given can decide whether (s)he wants to obtain more solutions without having to provide extra information. If (s)he wishes so, problem (2) is solved again with internally updated parameters values. To this end, several methods of the selected type are used. Finally, if the DM wishes to provide new kind of information, a new iteration is carried out.

In the implementation of this global formulation, it can be useful to allow the DM to save interesting solutions found during the solution process in a database (see, e.g., Jaszkiwicz and Slowiński (1999); Miettinen and Mäkelä (2006)). In other words, the DM can save solutions that seem good candidates as a final solution although (s)he is not quite satisfied yet. Also, given two already saved solutions, it could be interesting for the DM to generate intermediate solutions between them. This can be done by considering as reference points several points belonging to the segment which joins both solutions, and adjusting the parameters as described in Section 3.1.

In what follows, we describe the different types of specifying preference information that we consider and the corresponding interactive methods in more detail.

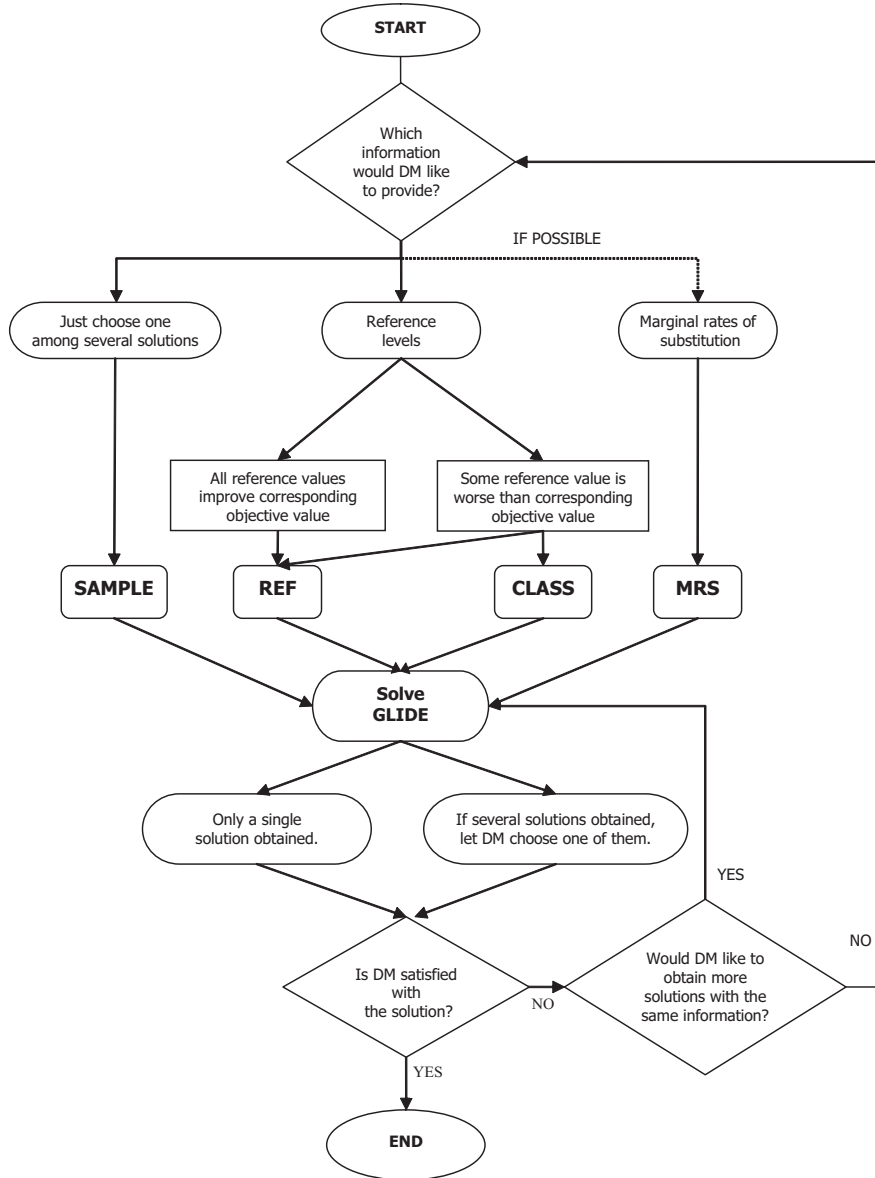


Figure 1: Flowchart of the global interactive solution scheme for multiobjective optimization.

3.1 Reference Levels

If the DM wishes to specify *reference levels* (also known as *aspiration levels*), given the current nondominated objective vector \mathbf{f}^h , the DM must provide such objective function values that (s)he wants to reach. The point consisting of these reference levels is referred to as a *reference point* $\hat{\mathbf{q}}^h = (\hat{q}_1^h, \dots, \hat{q}_k^h)$.

Because the current solution is nondominated, it is not possible to improve all objective function values simultaneously. However, the DM can specify a

reference point without taking this specifically into account. Then, once a new solution is generated with a reference point based method, the DM can see which objective values did actually impair in order to let the others improve. However, if the DM wants to control the search more by specifying which objective functions are allowed to impair, we can utilize this information by using a classification based method. Thus, based on what kind of values the DM specifies, we differentiate two separate cases:

1. If each reference level improves the corresponding objective value, that is, $\hat{q}_i^h \leq f_i^h$ ($i = 1, \dots, k$), then a reference point based method (REF) will be used.
2. Otherwise, that is, if there exists j such that $\hat{q}_j^h > f_j^h$, then a classification based method (CLASS) is chosen.

Note that based on the reasoning above, if classification based methods can be used, then also reference point based methods can be used but not vice versa (Miettinen and Mäkelä, 2002).

3.1.1 REF - Generate first solution.

If the DM has specified reference levels such that a reference point based method is to be used, we use the *reference point method* (Wierzbicki, 1980) to generate the first new nondominated solution to be shown to the DM. To this end, let us set the parameters for problem (2) as shown in Table 1. Parameter ρ is the so-called augmentation coefficient¹.

Table 1: REF – parameters for the reference point method. (Here $i = 1, \dots, k$.)

| | | | |
|-------------------------|-------------------------------|--|---------------------|
| <i>Weights</i> | $\omega_i^h = 1$ | $\mu_i^h = \frac{1}{z_i^{nad} - z_i^{**}}$ | $\rho > 0$ |
| <i>Reference levels</i> | $q_i^h = \hat{q}_i^h$ | | |
| <i>Objective bounds</i> | $\varepsilon_i^h = z_i^{nad}$ | $\Delta\varepsilon_i^h = 0$ | $s_\varepsilon = 0$ |

Using these parameters, problem (2) is equivalent to solving the following problem:

$$\begin{aligned}
 & \text{minimize} && \alpha + \rho \sum_{i=1}^k (f_i(\mathbf{x}) - \hat{q}_i^h) \\
 & \text{subject to} && \frac{f_i(\mathbf{x}) - \hat{q}_i^h}{z_i^{nad} - z_i^{**}} \leq \alpha && (i = 1, \dots, k) \\
 & && \mathbf{x} \in S.
 \end{aligned} \tag{3}$$

¹These augmented achievement scalarizing functions produce properly efficient solutions (see, e.g., Miettinen (1999); Wierzbicki (1986), for more details). It has also been shown that the augmentation terms may improve computational efficiency (Miettinen *et al.*, 2006).

It must be pointed out that in this case it is also possible for the DM to express preferences regarding the achievement of the reference levels. This possibility, which can accelerate in many cases the convergence of the method, is explained in full detail in Luque *et al.* (2006), and from the implementation point of view, it only involves assigning different values to the weights μ_i^h .

3.1.2 REF - Generate more solutions.

If the DM wants to generate more solutions with the same preference information (that is, the same reference levels), we can consider a perturbation of the reference point, as suggested by Wierzbicki (1980), for each $j \in \{1, \dots, k\}$ as follows

$$\begin{aligned} \Delta q_j^{h,j} &= \|\hat{\mathbf{q}}^h - \mathbf{f}^{h,0}\|_2 \\ \Delta q_i^{h,j} &= 0 \quad (i = 1, \dots, k, \quad i \neq j), \end{aligned}$$

where $\mathbf{f}^{h,0}$ is the objective vector corresponding to the optimal solution of (3). In this way, k new solutions can be generated by solving problem (2) with $\mathbf{q}^h = \hat{\mathbf{q}}^h + \Delta \mathbf{q}^{h,j}$ ($j = 1, \dots, k$) and keeping the other parameters as in Table 1.

If the DM still wants to see more solutions, the *GUESS method* (Buchanan, 1997) can be used by setting the parameters for problem (2) as shown in Table 2.

Table 2: REF – parameters for the GUESS method. (Here $i = 1, \dots, k$.)

| | | | |
|-------------------------|-------------------------------|---|---------------------|
| <i>Weights</i> | $\omega_i^h = 0$ | $\mu_i^h = \frac{1}{z_i^{nad} - \hat{q}_i^h}$ | $\rho = 0$ |
| <i>Reference levels</i> | $q_i^h = \hat{q}_i^h$ | | |
| <i>Objective bounds</i> | $\varepsilon_i^h = z_i^{nad}$ | $\Delta \varepsilon_i^h = 0$ | $s_\varepsilon = 0$ |

In this case, problem (2) is equivalent to solving the problem

$$\begin{aligned} &\text{minimize} && \alpha \\ &\text{subject to} && \frac{f_i(\mathbf{x}) - \hat{q}_i^h}{z_i^{nad} - \hat{q}_i^h} \leq \alpha \quad (i = 1, \dots, k) \\ &&& \mathbf{x} \in S. \end{aligned}$$

Note that there is no augmentation term in the original method but in our formulation we can easily generate properly efficient solutions (instead of weakly efficient ones) by setting $\rho > 0$ and $\omega_i^h = \mu_i^h$ ($i = 1, \dots, k$).

3.1.3 CLASS - Generate first solution.

If the DM specifies a reference point such that a *classification* based method can be used, we use the *NIMBUS method* (Miettinen and Mäkelä, 1995, 2006) to

generate the first solution. Let us define three classes of indices: $I_h^< = \{i \mid \hat{q}_i^h < f_i^h\}$, $I_h^= = \{i \mid \hat{q}_i^h = f_i^h\}$, and $I_h^> = \{i \mid \hat{q}_i^h > f_i^h\}$, and let us set the parameters for problem (2) as shown in Table 3.

Table 3: CLASS – parameters for the NIMBUS method.

| | | | |
|-------------------------|--|--|---------------------|
| <i>Weights</i> | $\omega_i^h = \frac{1}{z_i^{nad} - z_i^{**}}$ ($i=1, \dots, k$) | $\mu_i^h = \frac{1}{z_i^{nad} - z_i^{**}}$ for $i \in I_h^<$ | $\rho > 0$ |
| <i>Reference levels</i> | $q_i^h = \hat{q}_i^h$ for $i \in I_h^<$ | $\mu_i^h = 1$ for $i \in I_h^= \cup I_h^>$ | |
| <i>Objective</i> | $\varepsilon_i^h = \hat{q}_i^h$ for $i \in I_h^>$ | $q_i^h = z_i^{nad} + 1$ for $i \in I_h^= \cup I_h^>$ | |
| <i>bounds</i> | $\varepsilon_i^h = f_i^h$ for $i \in I_h^< \cup I_h^=$ | $\Delta\varepsilon_i^h = 0$ ($i = 1, \dots, k$) | $s_\varepsilon = 0$ |

Given the parameter values of Table 3, problem (2) can be expressed as

$$\begin{aligned}
& \text{minimize} && \alpha + \rho \sum_{i=1}^k \frac{f_i(\mathbf{x})}{z_i^{nad} - z_i^{**}} \\
& \text{subject to} && \frac{f_i(\mathbf{x}) - \hat{q}_i^h}{z_i^{nad} - z_i^{**}} \leq \alpha && (i \in I_h^<) \\
& && f_i(\mathbf{x}) \leq f_i^h && (i \in I_h^= \cup I_h^=) \\
& && f_i(\mathbf{x}) \leq \hat{q}_i^h && (i \in I_h^>) \\
& && \mathbf{x} \in S.
\end{aligned} \tag{4}$$

However, it may be necessary to clarify the reasoning behind the equivalence with problem (2) by showing an intermediate phase. Given that

$$\begin{aligned}
\mu_i^h(f_i(\mathbf{x}) - q_i^h) &= 1 \cdot (f_i(\mathbf{x}) - (z_i^{nad} + 1)) \leq -1 \text{ for } i \in I_h^= \cup I_h^> \text{ and} \\
\mu_j^h(f_j(\mathbf{x}) - q_j^h) &= \frac{f_j(\mathbf{x}) - q_j^h}{z_j^{nad} - z_j^{**}} \geq -1 \text{ for } j \in I_h^<,
\end{aligned}$$

then it follows that:

$$\text{for any } i \in I_h^= \cup I_h^> \text{ and } j \in I_h^< \text{ we have } \mu_i^h(f_i(\mathbf{x}) - \hat{q}_i^h) \leq \mu_j^h(f_j(\mathbf{x}) - \hat{q}_j^h) \leq \alpha$$

and, therefore, the optimal value for α is achieved at some constraint belonging to $I_h^<$. Therefore, the constraints regarding $I_h^= \cup I_h^>$ can be eliminated from problem (2), and thus, we have the equivalent problem (4).

Note that in the original NIMBUS method, there are two more classes: $I_h^<$ and $I_h^>$ for functions to be improved as much as possible and functions that are allowed to change freely for a while, respectively. Anyway, if a reference level \hat{q}_i^h is set equal to the corresponding ideal objective value z_i^* , then this index could

be considered as belonging to $I_h^<$. In the same way, if any reference level \hat{q}_j^h is set equal to the corresponding nadir objective value z_i^{nad} , then this index could be considered as belonging to I_h^\diamond . For further details, see Miettinen (1999); Miettinen and Mäkelä (1995, 1999, 2000, 2006).

3.1.4 CLASS - Generate more solutions.

The GLIDE formulation can accommodate two other classification based methods: STEM and STOM. In order to obtain the *STEP method (STEM)* (Benayoun *et al.*, 1971), let us define classes $I_h^<$, $I_h^=$, and $I_h^>$ as in the NIMBUS method, and let us set the parameters for problem (2) as shown in Table 4.

Table 4: CLASS – Parameters for the STEP method.

| | | | |
|-------------------------|--|--|---------------------|
| <i>Weights</i> | $\omega_i^h = 0$ ($i = 1, \dots, k$) | $\mu_i^h = \frac{z_i^{nad} - z_i^{**}}{\max\{ z_i^{nad} , z_i^{**} \}}$ for $i \in I_h^<$ | $\rho = 0$ |
| <i>Reference levels</i> | $q_i^h = z_i^{**}$ for $i \in I_h^<$ | $\mu_i^h = 1$ for $i \in I_h^= \cup I_h^>$ | |
| <i>Objective</i> | $\varepsilon_i^h = \hat{q}_i^h$ for $i \in I_h^>$ | $q_i^h = z_i^{nad}$ for $i \in I_h^= \cup I_h^>$ | |
| <i>bounds</i> | $\varepsilon_i^h = f_i^h$ for $i \in I_h^< \cup I_h^=$ | $\Delta\varepsilon_i^h = 0$ ($i = 1, \dots, k$) | $s_\varepsilon = 0$ |

Using a similar reasoning as in the previous method, problem (2) takes the following equivalent form:

$$\begin{aligned}
& \text{minimize} && \alpha \\
& \text{subject to} && \frac{z_i^{nad} - z_i^{**}}{\max\{|z_i^{nad}|, |z_i^{**}|\}} (f_i(\mathbf{x}) - z_i^{**}) \leq \alpha \quad (i \in I_h^<) \\
& && f_i(\mathbf{x}) \leq f_i^h \quad (i \in I_h^< \cup I_h^=) \\
& && f_i(\mathbf{x}) \leq \hat{q}_i^h \quad (i \in I_h^>) \\
& && \mathbf{x} \in S,
\end{aligned}$$

which corresponds to STEM modified for nonlinear problems (Eschenauer *et al.*, 1990). (Note that the original method was for linear problems only.)

Finally, the *satisficing trade-off method (STOM)* by Nakayama and Sawaragi (1984) can also be obtained from the GLIDE formulation by setting $q_i^h = z_i^{**}$ and $\mu_i^h = \frac{1}{\hat{q}_i^h - z_i^{**}}$ ($i = 1, \dots, k$). All the other parameter values are the same as in the GUESS method in Table 2. (Under certain assumptions, the original STOM offers a possibility to calculate reference levels for objectives in $I_h^>$ using sensitivity analysis, see Nakayama and Sawaragi (1984). However, we do not utilize it here because in our formulation we assume that the DM specifies all reference levels.)

Let us point out that if the DM wants to see more nondominated solutions after the solutions of NIMBUS, STEM and STOM have been generated, we can use the reference point based methods described earlier.

3.2 SAMPLE - Several Solutions

3.2.1 SAMPLE - Generate first solution.

If the DM wants only to compare a set of nondominated solutions, we use the *Chebyshev method* by Steuer and Choo (1983). To this end, the DM is asked to decide the number of solutions, NS , (s)he wishes to be shown. Then, a sample of weight vectors (with all coordinates ranging between 0 and 1) will be randomly generated. For each vector of weights $\boldsymbol{\mu}^h$, let us set the parameters for problem (2) as shown in Table 5.

Table 5: SAMPLE – Parameters for the Chebyshev method. (Here $i = 1, \dots, k$.)

| | | | |
|-------------------------|-------------------------------|-----------------------------|---------------------|
| <i>Weights</i> | $\omega_i^h = 1$ | μ_i^h (random) | $\rho > 0$ |
| <i>Reference levels</i> | $q_i^h = z_i^{**}$ | | |
| <i>Objective bounds</i> | $\varepsilon_i^h = z_i^{nad}$ | $\Delta\varepsilon_i^h = 0$ | $s_\varepsilon = 0$ |

This problem is equivalent to solving:

$$\begin{aligned}
 &\text{minimize} && \alpha + \rho \sum_{i=1}^k (f_i(\mathbf{x}) - z_i^{**}) \\
 &\text{subject to} && \mu_i^h (f_i(\mathbf{x}) - z_i^{**}) \leq \alpha \quad (i = 1, \dots, k) \\
 &&& \mathbf{x} \in S,
 \end{aligned}$$

and a corresponding objective vector is obtained. (Actually, more than NS solutions can be generated and then filtered in order to obtain the NS most different objective vectors to be shown to the DM; see Steuer and Choo (1983) for further details about the random generation of weights and the filtering process).

If the DM wants to generate more concentrated solutions (for example solutions focused on an area around a given previously saved solution), the Chebyshev method offers the possibility to reduce the weights space using a reduction factor. Again, see Steuer and Choo (1983) for further details on this option. Finally, if the DM wants to generate another set of solutions, a new random generation process can be carried out using the same method (and therefore, the same parameters but new random weights).²

²If the random process is well designed and there are enough different solutions, the new solutions obtained are expected to be different from the previous ones.

3.3 MRS - Marginal Rates of Substitution

If the DM desires to specify marginal rates of substitution, then, given the current objective vector \mathbf{f}^h , the DM must choose a reference objective function f_r , and then provide MRSs m_{ri}^h comparing each objective function to f_r ($i = 1, \dots, k$, $i \neq r$). This information can be approximated in the following way. Starting from \mathbf{f}^h , the DM is required to provide the amount Δf_i^h to be improved on the value of objective function f_i that can exactly offset the given amount Δf_r^h to be worsened of the reference objective f_r . Then, these amounts allow us to approximate the MRSs as

$$m_{ri}^h = \frac{\Delta f_r^h}{\Delta f_i^h} \quad (i = 1, \dots, k).$$

As previously mentioned, MRS methods cannot be applied to any kind of problem. In what follows, we will state the conditions under which the different methods are assured to work properly. It is strongly recommended not to use this kind of methods unless one can somehow be sure that the corresponding conditions are fulfilled. To begin with, optimal Karush-Kuhn-Tucker (KKT) multipliers have to be calculated at the current solution. To this end, some *regularity condition* (constraint qualification) has to be satisfied at every iteration.

The optimal KKT multipliers can be calculated in two different ways. In option 1, it is also necessary that *second order optimality conditions* (Chankong and Haimes, 1983) are satisfied at the current solution. If this holds, the following problem is solved:

$$\begin{aligned} & \text{minimize} && f_r(\mathbf{x}) \\ & \text{subject to} && f_i(\mathbf{x}) \leq f_i^h \quad (i = 1, \dots, k, \quad i \neq r) \\ & && \mathbf{x} \in S, \end{aligned} \tag{5}$$

which is obtained from problem (2) by setting the parameters as shown in Table 6.

Table 6: MRS - Parameters for the calculation of the optimal KKT multipliers, option 1. (Here $i = 1, \dots, k$.)

| | | | |
|-------------------------|--|------------------------------|---------------------|
| <i>Weights</i> | $\omega_i^h = 0 \quad (i \neq r)$ | $\mu_i^h = 0$ | $\rho = 1$ |
| | $\omega_r^h = 1$ | | |
| <i>Reference levels</i> | $q_i^h = 0$ | | |
| <i>Objective</i> | $\varepsilon_i^h = f_i^h \quad (i \neq r)$ | $\Delta \varepsilon_i^h = 0$ | $s_\varepsilon = 0$ |
| <i>bounds</i> | $\varepsilon_r^h = z_r^{\text{nad}}$ | | |

Let us denote by λ_{ri}^h the optimal KKT multipliers corresponding to the constraints set on objective functions (i.e., $f_i(\mathbf{x}) \leq f_i^h$, $i \neq r$). Furthermore, we set $\lambda_{rr}^h := 1$.

Option 2 to obtain the optimal KKT multipliers, which does not assume second order optimality conditions (Yang, 1999), is by solving the following minmax problem

$$\begin{aligned}
& \text{minimize} && \alpha \\
& \text{subject to} && \frac{1}{f_i^h - z_i^{**}}(f_i(\mathbf{x}) - z_i^*) \leq \alpha \quad (i = 1, \dots, k) \\
& && \mathbf{x} \in S,
\end{aligned} \tag{6}$$

which can also be obtained from problem (2) by considering the parameter values given in Table 7. Again, we denote by λ_{ri}^h the optimal KKT multipliers corresponding to the constraints $(1/(f_i^h - z_i^{**}))(f_i(\mathbf{x}) - z_i^*) \leq \alpha$ ($i = 1, \dots, k$).

Table 7: MRS - Parameters for the calculation of the optimal KKT multipliers, option 2. (Here $i = 1, \dots, k$.)

| | | | |
|-------------------------|-------------------------------|--|---------------------|
| <i>Weights</i> | $\omega_i^h = 0$ | $\mu_i^h = \frac{1}{f_i^h - z_i^{**}}$ | $\rho = 0$ |
| <i>Reference levels</i> | $q_i^h = z_i^*$ | | |
| <i>Objective bounds</i> | $\varepsilon_i^h = z_i^{nad}$ | $\Delta\varepsilon_i^h = 0$ | $s_\varepsilon = 0$ |

3.3.1 MRS - Generate first solution.

The first solution based on MRSs is an adaptation of the *SPOT method* (Sakawa, 1982). This adaptation does not make use of the proxy function (whose estimation goes beyond the reach of the GLIDE formulation). This method is applicable if *the objective functions are twice continuously differentiable and convex*, and *the feasible set is convex*. Besides, the existence of a continuously differentiable implicit value function of the DM, which is strictly decreasing and concave, is also assumed. Let us set the parameters for problem (2) as shown in Table 8.

Table 8: MRS - Parameters for the SPOT method. (Here $i = 1, \dots, k$.)

| | | | | |
|-------------------------|--|-------------------------------|--|--------------------------|
| <i>Weights</i> | $\omega_i^h = 0$ ($i \neq r$) | $\omega_r^h = 1$ | $\mu_i^h = 0$ | $\rho = 1$ |
| <i>Reference levels</i> | $q_i^h = 0$ | | | |
| <i>Objective bounds</i> | $\varepsilon_i^h = f_i^h$ ($i \neq r$) | $\varepsilon_r^h = z_r^{nad}$ | $\Delta\varepsilon_i^h = \lambda_{ri}^h - m_{ri}^h$ ($i \neq r$) | s_ε (varied) |
| | | | $\Delta\varepsilon_r^h = 0$ | |

Then, problem (2) is equivalent to solving the following problem:

$$\begin{aligned}
& \text{minimize} && f_r(\mathbf{x}) \\
& \text{subject to} && f_i(\mathbf{x}) \leq f_i^h + s_\varepsilon \cdot (\lambda_{ri}^h - m_{ri}^h) \quad (i = 1, \dots, k, i \neq r) \\
& && \mathbf{x} \in S.
\end{aligned} \tag{7}$$

Several values for s_ε are set and, in this way, different solutions are obtained by solving problem (7). The DM is then asked to choose one of them. The variation interval for s_ε must be chosen in such a way that problem (7) is assured to have feasible solutions. To this end, it is necessary to calculate an upper bound for s_ε , denoted by \bar{s}_ε . Let \bar{s}_ε be the smallest value of s_ε such that some coordinate of $\varepsilon_i^h + s_\varepsilon \cdot \Delta\varepsilon_i^h$ reaches its corresponding ideal value. If problem (7), for $s_\varepsilon = \bar{s}_\varepsilon$, has feasible solutions, then \bar{s}_ε is an appropriate upper bound. If not, we reduce \bar{s}_ε , multiplying it by a number between 0 and 1 (usually, this number is chosen between 0.5 and 1), and we check the feasibility again. The process continues until a valid value for \bar{s}_ε is found. Then, several values between 0 and \bar{s}_ε are selected and the corresponding problem (7) is solved.

3.3.2 MRS - Generate more solutions.

Finally, the algorithm used to generate alternative solutions under the MRS option is a modification of the *GRIST* method (Yang, 1999), called *PROJECT* (Luque *et al.*, 2007b). In this method, *the objective functions are assumed to be continuously differentiable, and the efficient set is assumed to be connected.* Given the current objective vector and the MRSs between the objective functions, an approximation of the normalized value function gradient is calculated and a normal vector to the current objective vector is given by the optimal KKT multipliers mentioned above. The projection of this vector onto the tangent plane of the efficient set can provide an ascending direction in the DM's value function that better approximates the feasible region. Starting from \mathbf{f}^h and in the direction of this projection vector, several points are generated and shown the DM. To generate these points, we calculate the last point whose coordinates are equal to or smaller than the corresponding ideal objective values (i.e., the first point with at least one coordinate being equal to the corresponding ideal objective value). Then we calculate several points between this point and the current objective vector and show them to the DM who is supposed to choose one. Let us denote it by $\hat{\mathbf{q}}^h$. This point chosen by the DM, which is infeasible in the majority of cases, is used to generate a solution which obeys the MRSs by solving the following problem:

$$\begin{aligned}
& \text{minimize} && \alpha \\
& \text{subject to} && \frac{f_i(\mathbf{x}) - \hat{q}_i^h}{|\hat{q}_i^h - f_i^h|} \leq \alpha \quad (i = 1, \dots, k) \\
& && \mathbf{x} \in S,
\end{aligned} \tag{8}$$

which can be obtained from the GLIDE formulation by setting the parameters as shown in Table 9. For further details about the determination of the normal vector and other features of the algorithm, see Yang (1999); Yang and Li (2002).

Table 9: MRS – Parameters for the PROJECT method. (Here $i = 1, \dots, k$.)

| | | | |
|-------------------------|-------------------------------|---|---------------------|
| <i>Weights</i> | $\omega_i^h = 0$ | $\mu_i^h = \frac{1}{ \hat{q}_i^h - f_i^h }$ | $\rho = 0$ |
| <i>Reference levels</i> | $q_i^h = \hat{q}_i^h$ | | |
| <i>Objective bounds</i> | $\varepsilon_i^h = z_i^{nad}$ | $\Delta \varepsilon_i^h = 0$ | $s_\varepsilon = 0$ |

4 Example

Let us illustrate with an example how our interactive solution scheme with GLIDE can be utilized. The problem considered has been described in Narula and Weistroffer (1989) and also considered in Miettinen and Mäkelä (1997) and briefly described in Miettinen *et al.* (2006). We have a pollution problem of a river involving four objective functions and two variables in the form

$$\begin{aligned}
& \text{maximize} && f_1(\mathbf{x}) = 4.07 + 2.27x_1 \\
& \text{maximize} && f_2(\mathbf{x}) = 2.60 + 0.03x_1 + 0.02x_2 + \frac{0.01}{1.39 - x_1^2} + \frac{0.30}{1.39 - x_2^2} \\
& \text{maximize} && f_3(\mathbf{x}) = 8.21 - \frac{0.71}{1.09 - x_1^2} \\
& \text{minimize} && f_4(\mathbf{x}) = -0.96 + \frac{0.96}{1.09 - x_2^2} \\
& \text{subject to} && 0.3 \leq x_1, x_2 \leq 1.0.
\end{aligned}$$

In this problem, a fishery and a city of medium size are polluting water in the river and, thus, there is both industrial and municipal waste in the river. The city is situated downstream from the fishery. There exist two treatment plants, one in the fishery and another in the city. The waste is described in pounds of biochemical oxygen demanding material (BOD) and the two decision variables represent the proportional amounts of BOD removed from water in the treatment plants.

Here the first two objective functions describe the quality of water after the fishery and after the city, respectively. Cleaning the water in the fishery decreases the return on investments and, thus the third objective function measures the percent return on investment at the fishery. Finally, cleaning the water in the city means addition to the tax rate in the city and this is represented in the last objective function. Therefore, the first three objective functions are to be maximized, while the last one is to be minimized. The ideal and nadir values of the four objective functions are displayed in Table 10. It can be observed that f_2 (quality of the water after the city) has the smallest variation range.

Table 10: Ideal and nadir values.

| <i>Objective</i> | | <i>Ideal</i> | <i>Nadir</i> |
|------------------|----------|--------------|--------------|
| f_1 | maximize | 6.34 | 4.75 |
| f_2 | maximize | 3.44 | 2.85 |
| f_3 | maximize | 7.50 | 0.32 |
| f_4 | minimize | 0.00 | 9.71 |

Iteration 1. When we start solving the problem, let us assume that the DM wishes to get an overall picture of the problem with four different nondominated solutions. Therefore, the option 'choosing one among several solution' is used, and the following nondominated solutions are obtained by solving the GLIDE problem (2) with the parameters given in Table 5 (with four sets of random weights):

$$(5.56, 2.87, 7.13, 0.00), (6.00, 2.88, 6.27, 0.00), \\ (5.80, 3.02, 6.82, 0.95), (6.21, 3.15, 4.75, 2.00).$$

Among these solutions, the DM chooses the last one: (6.21, 3.15, 4.75, 2.00). In relative terms, here the value of f_1 (water quality after the fishery) is the closest one to its ideal value.

Iteration 2. Next, the DM wishes to give MRSs³, taking f_1 as the reference objective. If this objective is relaxed by 0.1 units, the DM says that this is compensated by an improvement of 0.3 units in f_2 , or an improvement of 0.1 units in f_3 , or an improvement of 0.05 units in f_4 . This implies that the DM regards f_2 as (locally) less important than objective 1, objective 3 as (locally) equally important than objective 1, and f_4 as (locally) more important than objective one. Therefore, problem (2) is solved using the parameters shown in Table 8 for four step sizes, and the following solutions are shown to the DM:

$$(6.21, 3.15, 4.75, 2.00), (6.18, 3.09, 5.05, 1.38) \\ (6.15, 3.03, 5.35, 0.87), (6.12, 2.95, 5.65, 0.39),$$

³Note that for illustrative purposes we here use this option even though the problem does not satisfy the assumptions specified in Section 3.3.1.

who chooses the second one: (6.18, 3.09, 5.05, 1.38).

As it can be seen, the environmental objectives f_1 and f_2 have experimented a light decrease, while the economical ones (f_3 and f_4) have been improved.

Iteration 3. Let us now assume that the DM wishes to set very optimistic reference levels for every objective, as follows:

$$\hat{q}_1^h = 6.2, \quad \hat{q}_2^h = 3.2, \quad \hat{q}_3^h = 6.0, \quad \hat{q}_4^h = 1.0.$$

Therefore, the reference point option is selected, and the parameters for problem (2) are set as shown in Table 1. As a consequence, the following solution is obtained: (6.16, 3.14, 5.29, 1.97).

In this iteration, the quality of the water after the fishery (f_1) has slightly decreased and the addition to the tax rate (f_4) has increased significantly. On the other hand, both the quality of the water after the city (f_2) and the return on investments (f_3) have increased.

Iteration 4. Now the DM wishes to improve the economical objectives at the price of relaxing a little bit in the environmental ones. Therefore, the DM sets the following reference levels:

$$\hat{q}_1^h = 6.0, \quad \hat{q}_2^h = 3.0, \quad \hat{q}_3^h = 6.0, \quad \hat{q}_4^h = 1.5.$$

In this case, the classification based option is chosen and problem (2) is solved using the parameters shown in Table 3. The solution obtained is the following one: (6.00, 3.05, 6.28, 1.10).

As it can be seen, the quality of the water after the fishery is at its reference level and the quality of water after the city is slightly better than the reference level. On the other hand, the economical objectives have been improved beyond their reference levels. Let us assume that the DM decides to stop with this solution, which implies to remove the 85.02% of the BOD at the treatment plant of the fishery, and the 79.01% of the BOD at the treatment plant of the city. A graphical representation of the four iterations (with the solutions selected) can be seen in Figure 2, where the bigger the horizontal bars are, the better the value of the corresponding objective is.

Finally, it must be pointed out that the purpose of this example is to illustrate the performance and flexibility of the global interactive solution scheme, and that is why four iterations have been carried out, all of them with different preference elicitation styles. Obviously, the DM does not have to change the type of preference information so often, (s)he can ask for several solutions based on the information already specified and continue the solution process as long as desired.

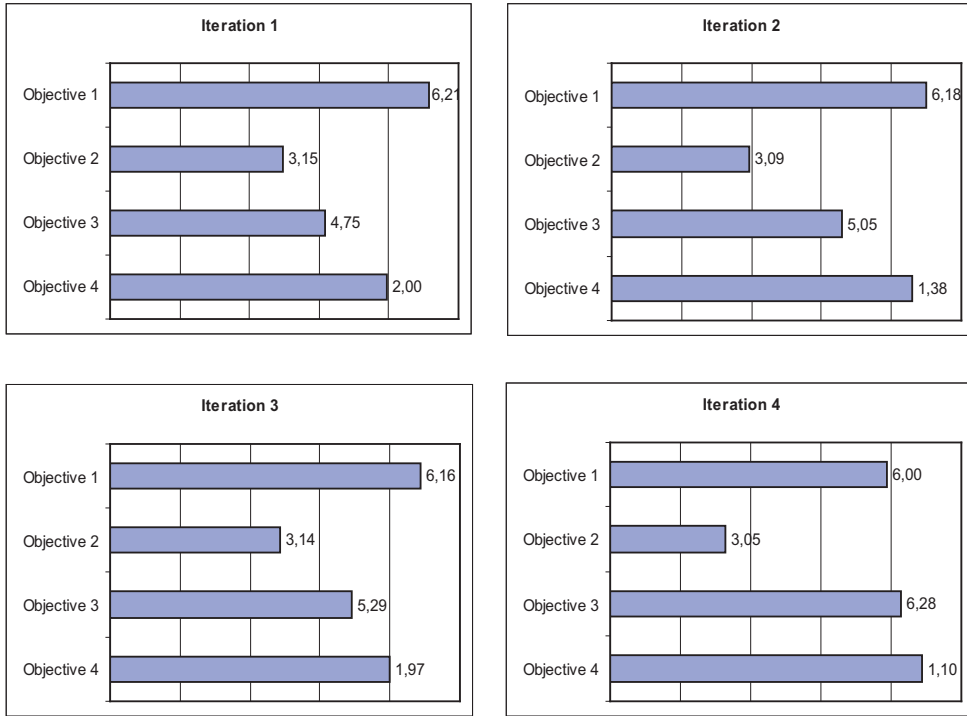


Figure 2: Graphical display of the four iterations.

5 Conclusions

In this paper, a global formulation which can accommodate several interactive multiobjective optimization methods has been introduced. The compact structure of this formulation takes the form of a general optimization problem with a set of parameters that have to be changed in order to obtain the different interactive methods supported. To be more specific, in this paper we show how the reference point, GUESS, NIMBUS, STEM, STOM, Chebyshev, SPOT and PROJECT methods can be derived from this general formulation (besides, other interactive methods like Klamroth and Miettinen (2007) could also be considered by choosing the corresponding parameters).

It is clear that one of the main features that an interactive decision aid tool must have in order to successfully tackle real problems is flexibility. This flexibility is understood as the ability of the system to adapt the solution process to the DM's wishes allowing her/him to choose the most appropriate way of providing information any time during the process. To this end, it is necessary to implement different interactive techniques which utilize different kinds of preference information. On the other hand, one of the weak points of the interactive multiobjective optimization field is the very limited number of practical and easy-to-use implementations.

We would like to emphasize the compactness of the formulation proposed,

which makes it different from the other general counterparts previously published. The GLIDE formulation has been designed with two main aims. First, providing a comfortable decision aid environment for the DM, and second, aiding potential programmers to implement the interactive system.

From the point of view of the DM, a general algorithm has been built where (s)he only has to decide what type of preference information (reference levels, choosing a solution among some efficient solutions or MRSs) does (s)he want to provide at each iteration. The algorithm chooses the corresponding method and produces the solution(s). Given that more than one interactive method of each class is supported by the global formulation, it is also possible to generate several different solutions with the same information if the DM so wishes.

From the point of view of the programmer, the global formulation is complemented with tables with the values of the parameters of GLIDE for each of the methods considered. This provides a simple implementation framework that makes it easier to create an interactive system based on the GLIDE formulation.

Finally, a simple hypothetical illustrative example has been used to show the performance of the global formulation and of the interactive solution scheme proposed. Although a preliminary implementation has been developed in order to test the algorithm and to solve the example, our future research trends include a more versatile implementation of an interactive system based on GLIDE with a flexible and graphical user interface, and its application to real life problems.

Acknowledgements

This research was partly supported by the Andalusian Regional Ministry of Innovation, Science and Enterprises (PAI group SEJ-445) and has also been supported by the Spanish Ministry of Education and Science (MTM2006-01921).

References

- Benayoun, R., de Montgolfier, J., Tergny, J., and Laritchev, O. (1971). Programming with multiple objective functions: Step method (STEM). *Mathematical Programming*, **1**(3), 366–375.
- Buchanan, J. T. (1997). A naïve approach for solving MCDM problems: The GUESS method. *Journal of the Operational Research Society*, **48**(2), 202–206.
- Caballero, R., Luque, M., Molina, J., and Ruiz, F. (2002). Promoin: an interactive system for multiobjective programming. *International Journal on Information Technologies and Decision Making*, **1**, 635–656.
- Chankong, V. and Haimes, Y. Y. (1983). *Multiobjective Decision Making: Theory and Methodology*. North-Holland, New York.

- Eschenauer, H. A., Osyczka, A., and Schäfer, E. (1990). Interactive multicriteria optimization in design process. In H. Eschenauer, J. Koski, and A. Osyczka, editors, *Multicriteria Design Optimization Procedures and Applications*, pages 71–114, Berlin, Heidelberg. Springer-Verlag.
- Gardiner, L. and Steuer, R. E. (1994a). Unified interactive multiple objective programming. *European Journal of Operational Research*, **74**(3), 391–406.
- Gardiner, L. and Steuer, R. E. (1994b). Unified interactive multiple objective programming: An open architecture for accommodating new procedures. *Journal of the Operational Research Society*, **45**(12), 1456–1466.
- Hwang, C. L. and Masud, A. S. M. (1979). *Multiple Objective Decision Making – Methods and Applications: A State-of-the-Art Survey*. Springer-Verlag, Berlin.
- Jaszkiewicz, A. and Slowiński, R. (1999). The ‘light beam search’ approach – an overview of methodology and applications. *European Journal of Operational Research*, **113**, 300–314.
- Kaliszewski, I. (2004). Out of the mist – towards decision-maker-friendly multiple criteria decision making support. *European Journal of Operational Research*, **158**, 293–307.
- Klamroth, K. and Miettinen, K. (2007). Integrating approximation and interactive decision making in multicriteria optimization. *Operations Research*. to appear.
- Luque, M., Miettinen, K., Eskelinen, P., and Ruiz, F. (2006). Three different ways for incorporating preference information in interactive reference point based methods. Working Paper W-410, Helsinki School of Economics, Helsinki. <http://hsepubl.lib.hse.fi/pdf/wp/w410.pdf>.
- Luque, M., Caballero, R., Molina, J., and Ruiz, F. (2007a). Equivalent information for multiobjective interactive procedures. *Management Science*, **53**(1), 125–134.
- Luque, M., Yang, J. B., and Wong, B. Y. H. (2007b). Project: A new interactive method based on the gradient projection and reference point techniques. Mbs working paper series, Manchester Business School, Manchester.
- Miettinen, K. (1999). *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston.
- Miettinen, K. (2006). IND-NIMBUS for demanding interactive multiobjective optimization. In T. Trzaskalik, editor, *Multiple Criteria Decision Making '05*, pages 137–150, Katowice. The Karol Adamiecki University of Economics in Katowice.

- Miettinen, K. and Mäkelä, M. M. (1995). Interactive bundle-based method for nondifferentiable multiobjective optimization: NIMBUS. *Optimization*, **34**(3), 231–246.
- Miettinen, K. and Mäkelä, M. M. (1997). Interactive method NIMBUS for non-differentiable multiobjective optimization problems. In J. Climaco, editor, *Multicriteria Analysis*, pages 310–319. Springer-Verlag.
- Miettinen, K. and Mäkelä, M. M. (1999). Comparative evaluation of some interactive reference point-based methods for multi-objective optimisation. *Journal of the Operational Research Society*, **50**(9), 949–959.
- Miettinen, K. and Mäkelä, M. M. (2000). Interactive multiobjective optimization system WWW-NIMBUS on the Internet. *Computers & Operations Research*, **27**(7–8), 709–723.
- Miettinen, K. and Mäkelä, M. M. (2002). On scalarizing functions in multiobjective optimization. *OR Spectrum*, **24**(2), 193–213.
- Miettinen, K. and Mäkelä, M. M. (2006). Synchronous approach in interactive multiobjective optimization. *European Journal of Operational Research*, **170**(3), 909–922.
- 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), 931–947.
- Nakayama, H. and Sawaragi, Y. (1984). Satisficing trade-off method for multiobjective programming. In M. Grauer and A. P. Wierzbicki, editors, *Interactive Decision Analysis*, pages 113–122. Springer-Verlag.
- Narula, S. C. and Weistroffer, H. R. (1989). A flexible method for nonlinear multicriteria decisionmaking problems. *IEEE Transactions on Systems, Man, and Cybernetics*, **19**(4), 883–887.
- Romero, C. (2001). Extended lexicographic goal programming: A unified approach. *Omega*, **29**, 63–71.
- Sakawa, M. (1982). Interactive multiobjective decision making by the sequential proxy optimization technique: SPOT. *European Journal of Operational Research*, **9**(4), 386–396.
- Sawaragi, Y., Nakayama, H., and Tanino, T. (1985). *Theory of Multiobjective Optimization*. Academic Press, Orlando, FL.

- Steuer, R. E. and Choo, E. U. (1983). An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming*, **26**, 326–344.
- Vassileva, M., Miettinen, K., and Vassilev, V. (2005). Generalized scalarizing problem for multicriteria optimization. IIT Working Papers IIT/WP-205, Institute of Information Technologies, Bulgaria.
- Wierzbicki, A. P. (1980). 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. Springer-Verlag.
- Wierzbicki, A. P. (1986). On the completeness and constructiveness of parametric characterizations to vector optimization problems. *OR Spectrum*, **8**(2), 73–87.
- Yang, J. B. (1999). Gradient projection and local region search for multiobjective optimization. *European Journal of Operational Research*, **112**, 432–459.
- Yang, J. B. and Li, D. (2002). Normal vector identification and interactive trade-off analysis using minimax formulation in multiobjective optimisation. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, **32**(3), 305–319.