Ann Oper Res
DOI 10.1007/s10479-010-0831-x

Improving the computational efficiency in a global formulation (GLIDE) for interactive multiobjective optimization

Francisco Ruiz · Mariano Luque · Kaisa Miettinen

Abstract In this paper, we present a new general formulation for multiobjective optimization that can accommodate several interactive methods of different types (regarding various types of preference information required from the decision maker). This formulation provides a comfortable implementation framework for a general interactive system and allows the decision maker to conveniently apply several interactive methods in one solution process. In other words, the decision maker can at each iteration of the solution process choose how to give preference information to direct the interactive solution process, and the formulation enables changing the type of preferences, that is, the method used, whenever desired. The first general formulation, GLIDE, included eight interactive methods utilizing four types of preferences. Here we present an improved version where we pay special attention to the computational efficiency (especially significant for large and complex problems), by eliminating some constraints and parameters of the original formulation. To be more specific, we propose two new formulations, depending on whether the multiobjective optimization problem to be considered is differentiable or not. Some computational tests are reported showing improvements in all cases. The generality of the new improved formulations is supported by the fact that they can accommodate six interactive methods more, that is, a total of fourteen interactive methods, just by adjusting parameter values.

Keywords Multiobjective programming · Multiple objectives · Interactive methods · Reference point methods · Classification · Marginal rates of substitution · Global system · Pareto optimality

F. Ruiz · M. Luque
University of Malaga, Calle Ejido 6, 29071 Malaga, Spain

F. Ruiz
e-mail: rua@uma.es

M. Luque
e-mail: mluque@uma.es

K. Miettinen (✉)
Department of Mathematical Information Technology, University of Jyväskylä, P.O. Box 35 (Agora), 40014 University of Jyväskylä, Finland
e-mail: kaisa.miettinen@jyu.fi

Published online: 07 January 2011

1 Introduction

Multiobjective optimization problems involve multiple conflicting objectives to be optimized simultaneously. Having conflicting objectives means that it is not possible to find a feasible solution where all the objectives could reach their individual optima but one must find the most satisfactory compromise between the objectives. These compromise solutions, where none of the objectives can be improved without impairing at least one of the others, are often referred to as Pareto optimal, efficient or noninferior solutions. Some preference information, typically coming from a human decision maker (DM), is required to determine which of them should be the most preferred solution, to be called the final solution.

The role of multiobjective optimization methods can be seen as supporting the DM in finding and identifying the most preferred solution. Many different approaches and methods have been proposed in the literature for this purpose (see, e.g., Chankong and Haimes 1983; Hwang and Masud 1979; Miettinen 1999; Sawaragi et al. 1985; Steuer 1986; Chinchuluun and Pardalos 2007). For example, in Hwang and Masud (1979), Miettinen (1999), methods are classified according to the role of the DM in a priori, a posteriori and interactive methods, where preferences are specified before, after or during the solution process, respectively. The fourth class is for methods where no DM is involved. Many methods convert the multiple objectives and the preference information into a scalarized single objective optimization problem which can be solved with any appropriate optimization method and the efficiency of the resulting solutions can be proven for many scalarizing functions.

Interactive methods (Miettinen 1999; Miettinen and Hakanen 2009; Miettinen et al. 2008) have become popular because they, for example, allow the DM to learn about the problem and one's preferences during the solution process and to concentrate on such efficient solutions that are interesting (thus decreasing the amount of efficient solutions to be compared, minimizing the cognitive load on the DM). In Miettinen and Hakanen (2009), the main steps of a general interactive method are briefly summarized as follows: (1) Initialize (e.g. calculate ranges of efficient solutions and show the corresponding objective function values to the DM), (2) generate an efficient starting point (some neutral compromise solution or solution given by the DM), (3) ask for preference information from the DM, (4) generate new efficient solution(s) according to the preferences and show it/them and possibly some 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 iterating, if the DM wants so. Otherwise, go to step (3).

Widely used ways of specifying preference information are the following four types: specification of desirable objective function values known as aspiration or reference levels, classification of objective functions according to how their values should change (i.e., to improve, impair or maintain the current value), selection of one from a small set of efficient solutions and specification of marginal rates of substitution (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). Typically, by selecting a method, one fixes the type of preference information used, but it may be desirable to change the type of preferences during the solution process. For this reason, global formulations that can accommodate methods of different kinds can be useful for real decision processes. In this line, Gardiner and Steuer (1994a, 1994b) suggested a unified algorithm which contains from nine to thirteen different methods, although its implementation turns out to be a hard task. In Ogryczak and Lahoda (2006), goal programming implementation techniques are used to modelize the aspiration-reservation-based decision support approach, and in Romero (1993), a general optimization structure is proposed for several variants of goal programming as well as some other methods.

In this paper, we concentrate on interactive approaches for continuous problems and discuss a global formulation including many interactive methods as its special cases. By giving a global formulation for interactive methods we enable creating a system where different interactive methods and, thus, different types of preference information, can be utilized without the need of implementing each of them separately. Instead, different methods can be obtained by simply adjusting the parameters of the global formulation. In our earlier work (Luque et al. 2011), we developed a global formulation for multiobjective optimization, which can accommodate eight interactive methods. This allows us to carry out a computational implementation of a general interactive procedure in an easy and comfortable way. To be more specific, in Luque et al. (2011), we defined a general scalarized formulation called GLIDE (GLobal Interactive Decision Environment) as

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

where, by changing the values of the parameters α, ρ, ω_i^h, q_i^h, μ_i^h, ε_i^h, s_ε and $\Delta\varepsilon_i^h$, we can obtain the (intermediate) single objective scalarized problems used by eight different interactive methods to generate the efficient solution(s) of the next iteration. Behind (1), the original multiobjective optimization problem has k objective functions f_i to be minimized subject to $\mathbf{x} \in S$.

When solving real-life applications, the objective and constraint function evaluations may be time-consuming and they may, for example, be derived from modelling and simulation tools. For instance, a simulator may have to solve a system of partial differential equations whenever an objective or constraint function value is needed. Such simulation-based optimization problems are examples of cases where computational cost plays an important role. Examples of such simulation-based problems are optimal shape design of ultrasonic transducers (Heikkola et al. 2006), radiotherapy treatment planning (Ruotsalainen et al. 2009), heat exchanger network synthesis (Laukkanen et al. 2010) and separation of glucose and fructose (Hakanen et al. 2007). When using an interactive method, the iterative nature of the solution process requires many optimization problems to be solved. Thus, computational savings per each iteration accumulate and even small savings per iteration do count.

Unfortunately, the formulation in (1) does not pay any special attention to computational efficiency. Because the interactive nature of the solution process suffers if the DM must wait for the next efficient solution to be generated, one should aim at minimizing computational cost. To this end, in this paper, we introduce an improved formulation, GLIDE-II, by reformulating GLIDE and eliminating some constraints and parameters.

Because taking the special characteristics of the problem in question into account allows using more efficient optimization methods, it is advantageous to pay attention, e.g., to differentiability aspects. For this reason, we here propose two different formulations of GLIDE-II. If the original problem is differentiable, we can formulate the general scalarized problem so that it preserves differentiability. This means that the problem can be solved using single objective solvers utilizing differentiability. Besides, this formulation also preserves other properties (for example, convexity) of the original problem. On the other hand, the nondifferentiable formulation of GLIDE-II has a reduced number of constraints. So, this formulation is advisable to be used for nondifferentiable problems and it can be solved with any appropriate nondifferentiable or global solver (whatever is needed).

In Luque et al. (2011), we presented a flexible global interactive solution scheme which allows the DM to choose the kind of information one wants to provide at each iteration of the interactive solution process. Our GLIDE and GLIDE-II formulations cover interactive methods using the four interaction styles described above to specify preference information to direct the search. Even though GLIDE-II can be the core of any interactive multiobjective optimization system that needs to accommodate different interactive methods, we want to emphasize that the global solution scheme presented in Luque et al. (2011) can be used with GLIDE-II instead of GLIDE. In this global scheme, the DM can conveniently change the type of preference information between iterations if so desired. Implementation of such a system is an answer to the challenge proposed in Kaliszewski (2004) about creating environments where different methods are present and where the DM can freely select the method and switch between methods. When using separate methods such a flexible solution approach is not possible. Examples mentioned above represent some application areas where multiobjective optimization problems can be encountered and where availability of more efficient methods can significantly increase the utilization of multiobjective optimization methods.

Furthermore, we have generalized GLIDE-II by pointing out six more interactive methods which can be covered by GLIDE-II formulations. This makes it possible to cover fourteen interactive methods in total. Therefore, this enables using GLIDE-II in a synchronous way, like in Miettinen and Mäkelä (2006). That is, with the same preference information provided by the DM, several solutions can be obtained by applying the different methods covered by our global formulation to offer the DM more solutions reflecting the preferences expressed. Thanks to the computational efficiency of GLIDE-II, it is fast to generate them.

The rest of this paper is organized as follows. In Sect. 2, we introduce the main concepts and notations used. Then we can describe and discuss two versions of GLIDE-II, the improved global interactive multiobjective formulation in Sect. 3. We report results of some computational experiments with various test problems concentrating on comparing the efficiency of GLIDE and GLIDE-II in Sect. 4 and conclude in Sect. 5. Finally, the Appendix is devoted to six more interactive methods that GLIDE-II can cover.

2 Concepts and notations

We consider *multiobjective optimization problems* of the form

$$\begin{aligned} &\text{minimize} \quad \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})\} \\ &\text{subject to} \quad \mathbf{x} \in S, \end{aligned} \tag{2}$$

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

In multiobjective optimization, objective vectors are regarded as 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 (2) if there does not exist another $\mathbf{x} \in S$ such that $f_i(\mathbf{x}) < f_i(\mathbf{x}')$ for all $i = 1, \dots, k$. The corresponding objective vectors $\mathbf{f}(\mathbf{x})$ are called *(weakly) nondominated objective vectors*. Note that the set of nondominated solutions is a subset of weakly nondominated solutions.

Let us assume that for problem (2) the set of nondominated objective vectors contains more than one vector. From a computational point of view, it is often useful to know the ranges of objective vectors in the set of nondominated solutions. We calculate the *ideal objective vector* $\mathbf{z}^\star = (z_1^\star, \dots, z_k^\star)^T \in \mathbb{R}^k$ by minimizing each objective function individually in the feasible region, that is, $z_i^\star = \min_{\mathbf{x} \in S} f_i(\mathbf{x}) = \min_{\mathbf{x} \in E} f_i(\mathbf{x})$ 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 components of 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. (For details, see, e.g., Miettinen 1999 and references therein. Lately, some approaches for more reliable nadir vector generation have been proposed e.g. in Deb and Miettinen 2010; Deb et al. 2010.) Alternatively, we can ask the DM to specify worst possible objective function values and regard this as the nadir point. However, when specifying the worst objective function values, one must keep in mind that it will not be possible to go beyond these values during the solution process.

Furthermore, sometimes a *utopian objective vector* $\mathbf{z}^{\star\star} = (z_1^{\star\star}, \dots, z_k^{\star\star})^T$ is defined as a vector strictly better than the ideal objective vector. To this end, we set $z_i^{\star\star} = z_i^\star - \varepsilon$ $(i = 1, \dots, k)$, 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 therefore, a *decision maker* (DM) has 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 Sect. 1). In what follows, if h denotes the current iteration, we denote the current efficient solution by $\mathbf{x}^h$, and its corresponding nondominated image in the objective space by $\mathbf{f}^h$. When using a reference point or classification scheme, the point consisting of the reference levels (or aspiration levels) is referred to as a *reference point* $\hat{\mathbf{q}}^h = (\hat{q}_1^h, \dots, \hat{q}_k^h)$.

3 GLIDE-II: an improved global interactive multiobjective formulation

In this section, we propose a new formulation for the GLobal Interactive Decision Environment (GLIDE) for multiobjective problems, which has been designed in order to improve the computational efficiency of (1). By introducing two new index sets, the number of constraints and parameters will be reduced (the actual reductions will depend on the specific interactive method considered). Depending on the differentiability of problem (2), we define a formulation called (GLIDE II-Dif), which allows us to maintain the differentiability of the problem, and another one called (GLIDE II-NDif), designed for nondifferentiable problems, whose number of constraints is reduced by incorporating the constraints defined by the minimax distance into the scalarizing function.

For the multiobjective optimization problem (2), the (GLIDE II-Dif) formulation is defined as follows

$$\text{(GLIDE II-Dif)} \quad \left\{ \begin{array}{lll} \text{minimize} & \alpha + \rho \sum_{i=1}^{k} \omega_i^h (f_i(\mathbf{x}) - q_i^h) & \\ \text{subject to} & \mu_i^h (f_i(\mathbf{x}) - q_i^h) \leq \alpha & \text{for } i \in I_\alpha^h \\ & f_i(\mathbf{x}) \leq \varepsilon_i^h + s_\varepsilon \cdot \Delta\varepsilon_i^h & \text{for } i \in I_\varepsilon^h \\ & \mathbf{x} \in S, & \end{array} \right. \tag{3}$$

and the (GLIDE II-NDif) formulation as

$$\text{(GLIDE II-NDif)} \quad \left\{ \begin{array}{lll} \text{minimize} & \max_{i \in I_\alpha^h} \{\mu_i^h (f_i(\mathbf{x}) - q_i^h)\} + \rho \sum_{i=1}^{k} \omega_i^h (f_i(\mathbf{x}) - q_i^h) & \\ \text{subject to} & f_i(\mathbf{x}) \leq \varepsilon_i^h + s_\varepsilon \cdot \Delta\varepsilon_i^h & \text{for } i \in I_\varepsilon^h \\ & \mathbf{x} \in S, & \end{array} \right. \tag{4}$$

where $\mathbf{x} \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$ are the variables. Besides, there are a series of real parameters (α, $\rho \geq 0$, $\omega_i^h \geq 0$, q_i^h, $\mu_i^h \geq 0$, ε_i^h, $s_\varepsilon \geq 0$ and $\Delta\varepsilon_i^h$) and two index sets, I_α^h and I_ε^h, which are subsets of $\{1, \dots, k\}$. It must be noted that problems (3) and (4) are equivalent. Their optimal solution will be denoted by $\mathbf{x}^{h+1}$ and the corresponding objective vector by $\mathbf{f}^{h+1} = \mathbf{f}(\mathbf{x}^{h+1})$.

By changing the values of the parameters (as will be described in Tables 1–7), formulations (3) and (4) can be transformed into the (intermediate) single objective problems used by eight different interactive methods to generate the solution of the next iteration, and thus, the (weak, proper) efficiency of the corresponding optimal solution is guaranteed as in each original method. In addition, six new methods have been incorporated in the formulation as demonstrated in the Appendix. Nevertheless, the following theorem gives general results about the efficiency of the optimal solutions of problems (3) and (4), depending on the values of some parameters.

Theorem 1 *Let $\mathbf{x}^{h+1}$ be an optimal solution of problem* (3) (*or* (4)). *Then*, *the following statements hold*:

(i) *If $\mathbf{f}(\mathbf{x}^{h+1})$ is the unique optimal solution of* (3) (*or* (4)) *in the objective space*, *then $\mathbf{x}^{h+1}$ is an efficient solution of problem* (2).

(ii) *If $\rho > 0$*, *and*
   (ii-a) *If $\omega_i^h > 0$ $(i = 1, \dots, k)$*, *then $\mathbf{x}^{h+1}$ is an efficient solution of problem* (2).
   (ii-b) *If there exist $i, j \in \{1, \dots, k\}$*, *such that $\omega_i^h > 0$ and $\omega_j^h = 0$*, *then $\mathbf{x}^{h+1}$ is a weakly efficient solution of problem* (2).

(iii) *If $I_\alpha^h \neq \emptyset$*, *and $\mu_i^h > 0$* (*for $i \in I_\alpha^h$*), *then $\mathbf{x}^{h+1}$ is a weakly efficient solution of problem* (2).

Proof Given that problems (3) and (4) are equivalent, we will prove the theorem for (4). Therefore, let us assume that $\mathbf{x}^{h+1}$ is an optimal solution of problem (4).

First, let us assume that there exists a decision vector $\mathbf{x}^* \in S$ such that $f_i(\mathbf{x}^*) \leq f_i(\mathbf{x}^{h+1})$, for all $i = 1, \dots, k$, and $f_l(\mathbf{x}^*) < f_l(\mathbf{x}^{h+1})$, for some $l \in \{1, \dots, k\}$. Then,

$$f_i(\mathbf{x}^*) \leq f_i(\mathbf{x}^{h+1}) \leq \varepsilon_i^h + s_\varepsilon \cdot \Delta\varepsilon_i^h \quad (i \in I_\varepsilon^h),$$

which means that $\mathbf{x}^*$ is feasible for (4). Besides, the two following inequalities hold:

$$\max_{i\in I_\alpha^h}\{\mu_i^h(f_i(\mathbf{x}^*)-q_i^h)\}\leq\max_{i\in I_\alpha^h}\{\mu_i^h(f_i(\mathbf{x}^{h+1})-q_i^h)\},$$

$$\rho\sum_{i=1}^k\omega_i^h(f_i(\mathbf{x}^*)-q_i^h)\leq\rho\sum_{i=1}^k\omega_i^h(f_i(\mathbf{x}^{h+1})-q_i^h),$$

which implies that $\mathbf{f}(\mathbf{x}^*)=\mathbf{f}(\mathbf{x}^{h+1})$. This proves (i) directly. On the other hand, if $\rho>0$ and $\omega_i^h>0$, for all $i\in\{1,\dots,k\}$, then the second inequality is strict, and this contradicts the optimality of $\mathbf{x}^{h+1}$ and thus proves (ii-a).

Second, let us assume that there exists a decision vector $\mathbf{x}^*\in S$ such that $f_i(\mathbf{x}^*)<f_i(\mathbf{x}^{h+1})$, for all $i=1,\dots,k$. Following the same reasoning as before, $\mathbf{x}^*$ is feasible for (4) and verifies the previous inequalities. Besides, if $\rho>0$ and $\omega_i^h>0$, for some $i\in\{1,\dots,k\}$, then the second inequality is strict, which contradicts the fact that $\mathbf{x}^{h+1}$ is an optimal solution of problem (4), and thus proves (ii-b). In the same way, if $I_\alpha^h\neq\emptyset$ and $\mu_i^h>0$, for all $i\in I_\alpha^h$ then the first inequality is strict and thus proves (iii). This completes the proof of the theorem. □

It is important to point out that all the efficient solutions of problem (2) can be obtained using this global scalarized formulation with adequate values for the parameters. For example, if we set $I_\alpha^h=\{1,\dots,k\}$, $I_\varepsilon^h=\emptyset$ and $\mu_i^h>0$ $(i=1,\dots,k)$, we get the achievement scalarizing function defined in Wierzbicki (1980), which covers the whole efficient set (i.e. every efficient solution can be obtained by setting an appropriate reference point).

As it was pointed out before, in Luque et al. (2011) we have considered eight different interactive methods, which have been classified according to their iteration style. Next, we will specify the values that have to be given to the parameters of models (3) and (4), in order to get the intermediate problems of the eight methods considered (Tables 1–7). Later, in the Appendix, we will specify parameters for six more methods (in Tables 12–18), which are also supported by the formulations introduced. Thus, (3) and (4) include a total of fourteen interactive methods as special cases.

3.1 Reference level methods

In this section, we assume that the DM wishes to specify reference levels (also known as aspiration levels) to be reached by each objective function. Let us denote by $\hat{\mathbf{q}}^h=(\hat{q}_1^h,\dots,\hat{q}_k^h)$ the reference point given by the DM (consisting of reference or aspiration levels). Formulations (3) and (4) can include, among others, the following reference point based interactive methods.

3.1.1 Reference point method

The reference point method utilizing an achievement scalarizing function was proposed in Wierzbicki (1980). The intermediate solutions of this method are obtained by solving the

Table 1 Reference point method

Index sets	$I_{\alpha}^{h} = \{1, \dots, k\}$	$I_{\varepsilon}^{h} = \emptyset$	
Weights	$\omega_i^h = 1$	$\mu_i^h = \dfrac{1}{z_i^{\text{nad}} - z_i^{\star\star}}$	$\rho > 0$
Reference levels	$q_i^h = \hat{q}_i^h$		
Objective bounds	–	–	–

Here $i = 1, \dots, k$

following problem

$$\begin{array}{ll} \text{minimize} & \alpha + \rho \displaystyle\sum_{i=1}^{k} (f_i(\mathbf{x}) - \hat{q}_i^h) \\ \text{subject to} & \dfrac{f_i(\mathbf{x}) - \hat{q}_i^h}{z_i^{\text{nad}} - z_i^{\star\star}} \le \alpha \quad (i = 1, \dots, k) \\ & \mathbf{x} \in S, \end{array}$$

or, equivalently,

$$\begin{array}{ll} \text{minimize} & \displaystyle\max_{i=1,\dots,k} \left\{ \frac{f_i(\mathbf{x}) - \hat{q}_i^h}{z_i^{\text{nad}} - z_i^{\star\star}} \right\} + \rho \sum_{i=1}^{k} (f_i(\mathbf{x}) - \hat{q}_i^h) \\ \text{subject to} & \mathbf{x} \in S. \end{array}$$

These problems are achieved in formulations (3) and (4), by considering the parameters and index sets indicated in Table 1.

We can generate more solutions with the same preference information by considering perturbations of the reference point, as suggested in Wierzbicki (1980). This just means changing the values of $\hat{q}_j^h$, in Table 1.

3.1.2 GUESS method

In the GUESS method, defined in Buchanan (1997), the problem solved at each iteration is the following:

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

or, if the nondifferentiable scheme is used,

$$\begin{array}{ll} \text{minimize} & \displaystyle\max_{i=1,\dots,k} \left\{ \frac{f_i(\mathbf{x}) - \hat{q}_i^h}{z_i^{\text{nad}} - \hat{q}_i^h} \right\} \\ \text{subject to} & \mathbf{x} \in S. \end{array}$$

These problems can be obtained from formulations (3) and (4), by considering the parameters and index sets indicated in Table 2.

Table 2 *GUESS* method

Here $i = 1, \dots, k$

Index sets	$I_\alpha^h = \{1, \dots, k\}$	$I_\varepsilon^h = \emptyset$	
Weights	$\omega_i^h = 0$	$\mu_i^h = \frac{1}{z_i^{\text{nad}} - \hat{q}_i^h}$	$\rho = 0$
Reference levels	$q_i^h = \hat{q}_i^h$		
Objective bounds	–	–	–

Let us add that there exist methods which use more or less elaborated achievement scalarizing functions, including piecewise linear functions like Grauer et al. (1984) or Lewandowski et al. (1989). In order to include these variants in GLIDE-II, however, modifications of the formulation (including new sets of constraints) would be necessary. On the other hand, double reference point schemes utilizing both aspiration levels and reservation levels (describing objective values that should be achieved according to the DM) for each objective can also be found in the literature. But again, the formulation, given e.g. in Wierzbicki et al. (2000) is not compatible with GLIDE-II. Because we want our model to be as simple as possible, we do not claim that we can handle all types of achievement scalarizing functions. Instead, GLIDE-II covers only such reference point based methods that use linear rescaling and/or bounding of objective values, allowing minmax and/or linear aggregation.

3.2 Classification methods

Given the current nondominated objective vector $\mathbf{f}^h$, the DM may wish to classify the objective functions into several classes, according to how the current solution should be improved. The following classification based interactive methods are considered.

3.2.1 *NIMBUS method*

The NIMBUS method was proposed in Miettinen (1999), Miettinen and Mäkelä (1995), Miettinen and Mäkelä (2006). Here we consider the formulation given in Miettinen and Mäkelä (2006). Given the current nondominated objective vector $\mathbf{f}^h$, the DM has the possibility of expressing desired changes in the current objective values by classifying the objective functions into up to five classes. For each objective function f_i, the DM can decide to:

- improve it as much as possible ($i \in I_h^<$),
- improve it to a certain aspiration level $\hat{q}_i^h$ ($i \in I_h^\leq$),
- regard it as satisfactory at its present value ($i \in I_h^=$),
- allow to relax it to a certain level $\hat{q}_i^h$ ($i \in I_h^\geq$),
- allow it to change freely ($i \in I_h^\diamond$).

Obviously, the union of all classes must be $\{1, \dots, k\}$, $I_h^< \cup I_h^\leq \neq \emptyset$ and $I_h^\geq \cup I_h^\diamond \neq \emptyset$. Besides, it must be verified that $\hat{q}_i^h < f_i^h$ for all $i \in I_h^\leq$ and $\hat{q}_i^h > f_i^h$ for all $i \in I_h^\geq$. For more details, see the references above.

Table 3 *NIMBUS* method

Index sets	$I_\alpha^h = I_h^< \cup I_h^\leq$	$I_\varepsilon^h = I_h^< \cup I_h^\leq \cup I_h^= \cup I_h^\geq$	
Weights	$\omega_i^h = \dfrac{1}{z_i^{\text{nad}} - z_i^{\star\star}}\ (i = 1, \ldots, k)$	$\mu_i^h = \dfrac{1}{z_i^{\text{nad}} - z_i^{\star\star}}$ for $i \in I_h^< \cup I_h^\leq$	$\rho > 0$
Reference levels	$q_i^h = z_i^*$ for $i \in I_h^<$ $q_i^h = \hat{q}_i^h$ for $i \in I_h^\leq$		
Objective bounds	$\varepsilon_i^h = f_i^h$ for $i \in I_h^< \cup I_h^\leq \cup I_h^=$ $\varepsilon_i^h = \hat{q}_i^h$ for $i \in I_h^\geq$	$\Delta\varepsilon_i^h = 0\ (i = 1, \ldots, k)$	$s_\varepsilon = 0$

A new NIMBUS solution is obtained by solving the problem

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

or, equivalently,

$$\begin{aligned}
\text{minimize} \quad & \max_{\substack{i \in I_h^< \\ j \in I_h^\leq}} \left\{ \frac{f_i(\mathbf{x}) - z_i^*}{z_i^{\text{nad}} - z_i^{**}}, \frac{f_j(\mathbf{x}) - \hat{q}_j^h}{z_j^{\text{nad}} - z_j^{**}} \right\} + \rho \sum_{i=1}^{k} \frac{f_i(\mathbf{x})}{z_i^{\text{nad}} - z_i^{\star\star}} \\
\text{subject to} \quad & f_i(\mathbf{x}) \leq f_i^h \quad (i \in I_h^< \cup I_h^\leq \cup I_h^=) \\
& f_i(\mathbf{x}) \leq \hat{q}_i^h \quad (i \in I_h^\geq) \\
& \mathbf{x} \in S.
\end{aligned}$$

Table 3 contains the parameters that have to be used in formulations (3) and (4), in order to get the problems described above.

3.2.2 STEP and STOM methods

In the STEP method, proposed in Benayoun et al. (1971), the DM has the possibility of classifying all objective functions into three classes at each iteration. Given the current objective values $\mathbf{f}^h$, these classes are:

- functions to be improved ($i \in I_h^\leq$),
- functions to be kept at their current values ($i \in I_h^=$) and
- functions allowed to be relaxed to a certain level $\hat{q}_i^h$ ($i \in I_h^\geq$).

Table 4 STEP method

Index sets	$I^h_\alpha = I^{\leq}_h$	$I^h_\varepsilon = \{1, \ldots, k\}$	
Weights	$\omega^h_i = 0\ (i = 1, \ldots, k)$	$\mu^h_i = \dfrac{z^{\text{nad}}_i - z^{\star\star}_i}{\max\{\lvert z^{\text{nad}}_i\rvert, \lvert z^{\star\star}_i\rvert\}}$ for $i \in I^{\leq}_h$	$\rho = 0$
Reference levels	$q^h_i = z^{\star\star}_i$ for $i \in I^{\leq}_h$		
Objective bounds	$\varepsilon^h_i = f^h_i$ for $i \in I^{\leq}_h \cup I^{=}_h$ $\varepsilon^h_i = \hat{q}^h_i$ for $i \in I^{\geq}_h$	$\Delta\varepsilon^h_i = 0\ (i = 1, \ldots, k)$	$s_\varepsilon = 0$

The union of all classes must be equal to $\{1, \ldots, k\}$, $I^{\leq}_h \neq \emptyset$ and $I^{\geq}_h \neq \emptyset$. Besides, it must be verified that $\hat{q}^h_i > f^h_i$ for all $i \in I^{\geq}_h$.

Although the original STEP method was designed for linear problems, it was modified in Eschenauer et al. (1990) to handle nonlinear problems. In this case, the single objective optimization problems solved are

$$\begin{aligned}
&\text{minimize} && \alpha \\
&\text{subject to} && \frac{z^{\text{nad}}_i - z^{\star\star}_i}{\max\{\lvert z^{\text{nad}}_i\rvert, \lvert z^{\star\star}_i\rvert\}}(f_i(\mathbf{x}) - z^{\star\star}_i) \leq \alpha \quad (i \in I^{\leq}_h) \\
& && f_i(\mathbf{x}) \leq\ f^h_i \quad (i \in I^{\leq}_h \cup I^{=}_h) \\
& && f_i(\mathbf{x}) \leq\ \hat{q}^h_i \quad (i \in I^{\geq}_h) \\
& && \mathbf{x} \in S,
\end{aligned}$$

or, in the nondifferentiable case,

$$\begin{aligned}
&\text{minimize} && \max_{i \in I^{\leq}_h} \left\{ \frac{z^{\text{nad}}_i - z^{\star\star}_i}{\max\{\lvert z^{\text{nad}}_i\rvert, \lvert z^{\star\star}_i\rvert\}}(f_i(\mathbf{x}) - z^{\star\star}_i) \right\} \\
&\text{subject to} && f_i(\mathbf{x}) \leq\ f^h_i \quad (i \in I^{\leq}_h \cup I^{=}_h) \\
& && f_i(\mathbf{x}) \leq\ \hat{q}^h_i \quad (i \in I^{\geq}_h) \\
& && \mathbf{x} \in S.
\end{aligned}$$

These problems can be obtained in formulations (3) and (4), by considering the parameters and index sets indicated in Table 4.

The *satisficing trade-off method* (STOM) (Nakayama and Sawaragi 1984), can also be obtained, like the GUESS method, by setting $q^h_i = z^{\star\star}_i$ and $\mu^h_i = \frac{1}{\hat{q}^h_i - z^{\star\star}_i}$ $(i = 1, \ldots, k)$. All the other parameter values and index sets are the same ones as in the GUESS method (Table 2). The original STOM offers the possibility (under certain assumptions) to calculate reference levels for objectives in $I^{\geq}_h$, using sensitivity analysis (see Nakayama and Sawaragi 1984).

3.3 Generating methods

In some cases, the DM may want to see several nondominated solutions at each iteration, and to choose his/her most preferred one. In these cases, the methods produce several solutions representing the zone of the nondominated set under study. This is the case of the Tchebycheff method.

Table 5 *Tchebycheff* method

Index sets	$I_\alpha^h = \{1, \dots, k\}$	$I_\varepsilon^h = \emptyset$	
Weights	$\omega_i^h = 1$	μ_i^h (random)	$\rho > 0$
Reference levels	$q_i^h = z_i^{\star\star}$		
Objective bounds	–	–	–

Here $i = 1, \dots, k$

3.3.1 Tchebycheff method

In the Tchebycheff method (Steuer and Choo 1983), nondominated solutions are obtained by solving, for several values of the vector of weights $\boldsymbol{\mu}^h$, the following problem:

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

or, equivalently,

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

In order to get these problems in formulations (3) and (4), one must consider the parameters and index sets indicated in Table 5. The reader is referred to Steuer and Choo (1983) for a detailed explanation of how to choose $\boldsymbol{\mu}^h$.

3.4 Tradeoff methods

Given the current nondominated objective vector $\mathbf{f}^h$, the DM may wish to specify opinions related to local tradeoffs or marginal rates of substitution (MRS) for the current values. In this case, the DM is asked to choose a reference objective function f_r, and then provide MRSs m_{ri}^h comparing each objective function f_i 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 in the value of objective function f_i that can exactly offset a given amount, Δf_r^h, to be worsened of the reference objective f_r. Then, these amounts allow us to approximate the MRSs as

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

It must be noted that, unlike the previously considered methods, tradeoff based methods cannot be applied, in general, to any multiobjective problem. In particular, they generally require to calculate the optimal Karush-Kuhn-Tucker (KKT) multipliers at the current solution. To this end, some *regularity condition* (constraint qualification) has to be satisfied at every iteration. Let us denote by λ_{ri}^h these optimal KKT multipliers. For more details about how to calculate these values, see Luque et al. (2011). Besides, each specific method may require other conditions (for example, convexity). The user should verify in the orig-

Table 6 SPOT method

Index sets	$I_\alpha^h = \emptyset$	$I_\varepsilon^h = \{1, \ldots, k\} \setminus \{r\}$	
Weights	$\omega_i^h = 0$ $\omega_r^h = 1$	–	$\rho = 1$
Reference levels	$q_r^h = 0$		
Objective bounds	$\varepsilon_i^h = f_i^h$	$\Delta\varepsilon_i^h = \lambda_{ri}^h - m_{ri}^h$	s_ε (varied)

Here $i = 1, \ldots, k$ and $i \neq r$

inal source of each method whether it can be applied to his/her problem. The MRS based methods considered are the following.

3.4.1 SPOT method

The SPOT method was proposed in Sakawa (1982). Originally, this method uses a proxy function (whose estimation goes beyond the reach of the GLIDE formulation) to determine the step length at each iteration. Here, we use an adaptation of the method where this proxy function is not necessary. In this case, after the DM has specified the MRS values, SPOT solves the following problem

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

where several values for s_ε are set and, in this way, different solutions are obtained. The DM is then asked to choose one of them. In this case, the problems for both cases (differentiable and nondifferentiable) are identical, and they can be obtained from formulations (3) and (4) if we set the parameters as indicated in Table 6.

3.4.2 PROJECT method

Proposed in Luque et al. (2009), the PROJECT method is a modification of the GRIST method described in Yang (1999). The method is based on the projection of the utility gradient (obtained from trade-off information given by the DM) onto the tangent hyperplane of the efficient set at the current objective vector $\mathbf{f}^h$. By identifying an ascending direction in the DM's utility, new values for the objectives are obtained. If these values are not attainable, they form a new reference point $\hat{\mathbf{q}}^h$, and an achievement scalarizing function is used for maintaining local tradeoffs while a new efficient solution is sought. This efficient solution is obtained by solving the problem

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

or, equivalently,

$$\begin{aligned} &\text{minimize} && \max_{i=1,\ldots,k} \left\{ \frac{f_i(\mathbf{x}) - \hat{q}_i^h}{|\hat{q}_i^h - f_i^h|} \right\} \\ &\text{subject to} && \mathbf{x} \in S. \end{aligned}$$

Table 7 PROJECT method

Index sets	$I_{\alpha}^{h} = \{1, \ldots, k\}$	$I_{\varepsilon}^{h} = \emptyset$	
Weights	$\omega_i^h = 0$	$\mu_i^h = \frac{1}{\lvert \hat{q}_i^h - f_i^h \rvert}$	$\rho = 0$
Reference levels	$q_i^h = \hat{q}_i^h$		
Objective bounds	–	–	–

Here $i = 1, \ldots, k$

Table 8 Number of additional constraints in each formulation

	GLIDE II-Dif	GLIDE II-NDif
Reference point	k	0
GUESS	k	0
STOM	k	0
NIMBUS	$k - \text{card}(I_h^{\diamond}) + \text{card}(I_h^{<} \cup I_h^{\leq})$	$k - \text{card}(I_h^{\diamond})$
STEP	$k + \text{card}(I_h^{\leq})$	k
Tchebycheff	k	0
SPOT	$k - 1$	$k - 1$
PROJECT	k	0

The values given in Table 7 have to be given to the parameters and index sets of formulations (3) and (4) to get these problems.

3.5 Number of additional constraints

In the original formulation of GLIDE (1), $2k$ constraints are added to the original constraints of the multiobjective optimization model (2), for every interactive method considered in this paper. GLIDE-II allows us to reduce this number of additional constraints. The actual reduction depends on the method used, as shown in Table 8, where we can see the number of additional constraints for each method considered, both for the differentiable (3) (GLIDE II-Dif) and the nondifferentiable (4) (GLIDE II-NDif) formulations. As we can see, some of these reductions are significant. For example in the reference point method, only k additional constraints are used in formulation (3) and no additional constraint in formulation (4). One can expect the problems to be easier to solve (in terms of computational efficiency) when the number of constraints is smaller. Some computational experiments are carried out in Sect. 4 to prove this statement.

4 Computational experiments

In order to compare the performance of the GLIDE-II formulation (in terms of computational efficiency), to the original GLIDE model, we have carried out computational experiments using several multiobjective optimization problems. The aim of these experiments is to show that significant reduction can be achieved in function evaluations of the corresponding scalarizing function, when using the GLIDE-II formulation to solve the scalarized problems associated to each method. For the experiments, we have used the twelve test problems given in Miettinen et al. (2006) (P1 to P12), a quadratic problem, which has been randomly generated using the scheme given in Goldfarb and Idnani (1983) (P13), and the

Table 9 Dimensions of the test problems

Problem	Variables	Objectives	Lin. constr.	Nonlin. constr.
P1	3	3	0	1
P2	2	5	0	0
P3	2	3	0	0
P4	2	3	1	0
P5	3	5	1	1
P6	2	7	0	0
P7	3	3	0	1
P8	3	6	0	1
P9	3	3	3	0
P10	10	3	3	5
P11	2	3	0	1
P12	2	5	0	0
P13	50	7	20	0
P14	30	2	0	0

nondifferentiable problem given in Fonseca and Fleming (1995) (P14). Table 9 shows the dimensions (number of variables, number of objective functions, number of linear constraints and number of nonlinear constraints) of the fourteen tests problems.

All the convex problems (P4, P7, P9, P11 and P13) are differentiable and they have been solved using the differentiable formulation (3), and the rest have been solved using the nondifferentiable formulation (4). Besides, we have also solved all of them using the original formulation (1).

In order to simulate the different styles of specifying preference information, we have solved each test problem using the eight methods described in the previous section. The settings of the experiments are the following:

1. We have carried out 10 iterations for all methods except for the Tchebycheff method, where only 5 iterations have been carried out due to its higher computational cost (higher number of intermediate single optimization problems for each iteration).
2. For the methods based on reference levels (reference point method and GUESS) and classification methods (NIMBUS and STEP), we have randomly generated ten reference points between the ideal and nadir values.
3. For the generating method (Tchebycheff), both the weight vectors and the objective vector chosen by the DM at each step have been randomly generated.
4. For the tradeoff based methods (SPOT and PROJECT), both the MRS values and the nondominated vector chosen by the DM at each step have also been determined in a random fashion.

The implementation has been done in the C++ language, using the interactive system PROMOIN (Caballero et al. 2002), with each of the formulations considered. Next, we present the results obtained in these experiments.

4.1 Differentiable case

In order to solve the convex intermediate problems in the differentiable case, we have used a solver of the NAG (Numerical Algorithms Group) library (NAG 2000) that uses a sequential

Table 10 Differentiable case—reduction in function evaluations

Problem	Ref.p.	GUESS	NIMBUS	STEP	Tchebycheff	SPOT	PROJECT
P4	100%	100%	90%	91%	94%	99%	83%
P7	79%	86%	94%	86%	73%	98%	82%
P9	95%	99%	88%	83%	95%	91%	98%
P11	60%	66%	57%	96%	96%	98%	87%
P13	86%	44%	94%	44%	64%	91%	90%

quadratic programming (SQP) method to find an optimal solution. The optimal solution of this solver must satisfy the Karush-Kuhn-Tucker conditions to the accuracy requested by an optional parameter (we have considered 10^{-6}) for all problems). For more details about the precision of this solver see NAG (2000) and, and for an overview of SQP methods see Fletcher (2000) and Gill et al. (1981).

In Table 10, we show the reduction, in terms of the number of evaluations of the scalarizing functions, achieved when using the GLIDE II-Dif formulation (3), instead of the GLIDE formulation (1). For each test problem we have calculated the mean number of evaluations per iteration, using both formulations, and we show the percentage of evaluations that GLIDE-II needs, when comparing them to GLIDE (that is, 100% means the same number of evaluations, 50% means that GLIDE-II needs half the number of the evaluations GLIDE needs, and so on).

As it can be seen, in most of the cases there has been a reduction of evaluations (only for two instances of problem P4 the number is the same). These reductions can be significant in certain cases. It must be noted that the number of evaluations was never very high for these convex problems, using the algorithms mentioned above (the highest number of evaluations for a single problem in these experiments was 143).

4.2 Nondifferentiable case

Although all the multiobjective optimization problems considered are differentiable, we have chosen to solve the nonconvex ones using the nondifferentiable formulation, in order to compare it with GLIDE. Of course, they can also be solved using the differentiable formulation together with a differentiable global solver. In order to solve the single objective nondifferentiable problems, we have used the LGO Solver System for Continuous Global Optimization (Pinter 2001, 2006). This solver uses a Branch and Bound + Local Search scheme for some problems, and a Global Adaptive Random Search + Local Search scheme for others. The stopping criterion is the improvement tolerance of the single objective function, which is applied in the local search phase. We have used the recommended default value (10^{-8}) for all problems except for P2, where we had to relax it (10^{-4}).

In this case, we have not considered the tradeoff based methods in the computational tests, because the SPOT method assumes the convexity of the problem, and the PROJECT method needs, at least, differentiability. The results are displayed in Table 11.

Again, the use of the GLIDE II formulation did always produce (except in one instance of problem P5) a reduction in the evaluations of scalarizing functions. In some cases, the reductions are really significant (see, for example, problem P2). When using the global solver for nonconvex problems, the total number of evaluations can be much higher than in the convex and differentiable case (it was over one million for LGO in some cases), and thus, the reductions are more important. In general, given that the number of additional constraints

Table 11 Nondifferentiable case—reduction in function evaluations

Problem	Ref.p.	GUESS	NIMBUS	STEP	Tchebycheff
P1	96%	89%	83%	86%	71%
P2	27%	29%	48%	67%	69%
P3	69%	57%	61%	70%	66%
P5	98%	100%	91%	68%	52%
P6	76%	60%	78%	68%	40%
P8	77%	71%	90%	90%	88%
P10	97%	91%	90%	66%	91%
P12	98%	98%	99%	98%	98%
P14	92%	94%	98%	96%	98%

in the nondifferentiable formulation of GLIDE is higher (see Table 8), the reductions with GLIDE-II are also more significant than those of the differentiable case.

5 Conclusions

Interactive methods are useful and realistic multiobjective optimization techniques and this is why the number of such methods is continuously growing. However, they have two important drawbacks when using them in real applications. Firstly, the question of which method should be chosen is not trivial. Different methods require different types of preference information from the DM, and generate efficient solutions in a different way. Secondly, there are rather few practical implementations of the methods available. Many authors have already stressed the need for systems that incorporate different types of interactive techniques. In this paper, we have introduced a general formulation that can accommodate fourteen interactive methods.

From the technical point of view, this formulation provides a comfortable implementation framework for a general interactive system. The global formulation is complemented with tables with the values of the parameters of GLIDE-II 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. Besides, this implementation allows the decision maker to choose how to give preference information to the system, and enables changing it anytime during the solution process. This change-of-method option provides a very flexible framework for the decision maker.

The GLIDE-II formulation presented in this paper is an evolution of the original GLIDE formulation. By introducing some index sets in the parameters of the model, we have eliminated some constraints (the specific number depends on the method), and this results in a higher computational efficiency of the formulation. We have carried out several computational experiments in order to prove this. GLIDE-II has got two different versions (differentiable and nondifferentiable). We have shown that the reduction of constraints in the nondifferentiable case is higher, and so is the reduction of function evaluations in this case, when compared to GLIDE. Let us mention that whenever using a scalarizing based method, one has to select an appropriate single objective solver to solve it. Thus, selecting either the differentiable or nondifferentiable version of GLIDE-II does not decrease the generality of the approach—the choice simply is a step in selecting the solver used.

Moreover, we have included six more interactive methods in the GLIDE-II formulation, and this brings the number of methods covered by our general formulation up to fourteen.

This fact enables GLIDE-II to be the core part of the implementations of interactive decision making systems. Consideration of different general formulations capable of capturing different scalarizing functions (including piecewise linear ones) is a topic of future research.

Development of GLIDE and GLIDE-II and using them as a general interactive solution framework enables solving multiobjective optimization problems in a flexible way. When combined with the computational efficiency provided by GLIDE-II, this enables convenient solution processes for the DM when the type of preference information used does not have to be fixed before the solution process. Many real-life problems in various application fields require efficient and user-friendly solution approaches and GLIDE-II paves the way in finding acceptable solution methods and satisfactory solutions to the problems.

Acknowledgements This research was partly supported by the Andalusian Regional Ministry of Innovation, Science and Enterprises (SEJ-445 and P09-FQM-5001) and by the Spanish Ministry of Education and Science (MTM2009-07646).

Appendix

Apart from the eight methods shown in the paper, many other interactive methods can be accommodated under the GLIDE-II formulation. In this appendix, we show the values of the parameters for one more reference point based method (visual interactive approach), two classification based methods (modified reference point method and reference direction algorithm), two trade-off based methods (interactive surrogate worth tradeoff method and weighting method) and a method which works with both aspiration and reservation levels (MICA).

For simplicity, in each method we just formulate the problems to be solved and we present the table of parameters. Readers are referred to the source publications for further details.

Modified reference point method

The modified reference point method (Vassilev et al. 2001) is a classification based method proposed for integer convex multiobjective problems, which solves the following problem in order to obtain each efficient solution:

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

Table 12 Modified reference point method

Index sets	$I^h_\alpha = I^{\le}_h \cup I^{\ge}_h$	$I^h_\varepsilon = I^{=}_h$	
Weights	$\omega^h_i = 0\ (i = 1, \dots, k)$	$\mu^h_i = \dfrac{1}{f^h_i - \hat{q}^h_i}$ for $i \in I^{\le}_h$ $\mu^h_i = \dfrac{1}{\hat{q}^h_i - f^h_i}$ for $i \in I^{\ge}_h$	$\rho = 0$
Reference levels	$q^h_i = \hat{q}^h_i$ for $i \in I^{\le}_h$ $q^h_i = f^h_i$ for $i \in I^{\ge}_h$		
Objective bounds	$\varepsilon^h_i = f^h_i$ for $i \in I^{=}_h$	$\Delta\varepsilon^h_i = 0\ (i = 1, \dots, k)$	$s_\varepsilon = 0$

or, equivalently,

$$\begin{array}{ll} \text{minimize} & \displaystyle\max_{\substack{i \in I^{\le}_h \\ j \in I^{\ge}_h}} \left\{ \frac{f_i(\mathbf{x}) - \hat{q}^h_i}{f^h_i - \hat{q}^h_i}, \frac{f_j(\mathbf{x}) - f^h_j}{\hat{q}^h_j - f^h_j} \right\} \\ \text{subject to} & f_i(\mathbf{x}) \le f^h_i \quad (i \in I^{=}_h) \\ & \mathbf{x} \in S. \end{array}$$

The values of the parameters and index sets are displayed in Table 12.

Visual interactive approach

The visual interactive approach (VIA) is presented in Korhonen and Laakso (1986), and the intermediate problem is

$$\begin{array}{ll} \text{minimize} & \alpha \\ \text{subject to} & \dfrac{f_i(\mathbf{x}) - (f^h_i + s_q(\hat{q}^h_i - f^h_i))}{z^{\text{nad}}_i - z^{\star\star}_i} \le \alpha \quad (i = 1, \dots, k) \\ & \mathbf{x} \in S, \end{array}$$

or, equivalently,

$$\begin{array}{ll} \text{minimize} & \displaystyle\max_{i=1,\dots,k} \left\{ \frac{f_i(\mathbf{x}) - (f^h_i + s_q(\hat{q}^h_i - f^h_i))}{z^{\text{nad}}_i - z^{\star\star}_i} \right\} \\ \text{subject to} & \mathbf{x} \in S, \end{array}$$

where s_q is a step-length. In general, we can consider several values of s_q and solve the corresponding problems. However, the special case of linear multiobjective problems can be solved by parametric linear programming (for more details, see Korhonen and Wallenius 1988).

The values of the parameters and index sets are shown in Table 13.

Reference direction algorithm

The reference direction algorithm (Vassilev and Narula 1993) is also a classification based method proposed for integer linear multiobjective problems, which uses the intermediate

Table 13 Visual interactive approach

Index sets	$I_\alpha^h = \{1, \dots, k\}$	$I_\varepsilon^h = \emptyset$	
Weights	$\omega_i^h = 0$	$\mu_i^h = \dfrac{1}{z_i^{\text{nad}} - z_i^{\star\star}}$	$\rho = 0$
Reference levels	$q_i^h = f_i^h + s_q(\hat{q}_i^h - f_i^h)$		
Objective bounds	–	–	–

Here $i = 1, \dots, k$

Table 14 Reference direction algorithm

Index sets	$I_\alpha^h = I_h^{\le}$	$I_\varepsilon^h = I_h^{=} \cup I_h^{\ge}$	
Weights	$\omega_i^h = 0$ $(i = 1, \dots, k)$	$\mu_i^h = \frac{1}{f_i^h - \hat{q}_i^h}$ for $i \in I_h^{\le}$	$\rho = 0$
Reference levels	$q_i^h = \hat{q}_i^h$ for $i \in I_h^{\le}$		
Objective bounds	$\varepsilon_i^h = f_i^h$ for $i \in I_h^{=}$ $\varepsilon_i^h = \hat{q}_i^h$ for $i \in I_h^{\ge}$	$\Delta\varepsilon_i^h = 0$ for $i \in I_h^{=}$ $\Delta\varepsilon_i^h = f_i^h - \hat{q}_i^h$ for $i \in I_h^{\ge}$	s_ε

problem:

$$\begin{aligned} \text{minimize} \quad & \alpha \\ \text{subject to} \quad & \frac{f_i(\mathbf{x}) - \hat{q}_i^h}{f_i^h - \hat{q}_i^h} \le \alpha \quad (i \in I_h^{\le}) \\ & f_i(\mathbf{x}) \le \ f_i^h \quad (i \in I_h^{=}) \\ & f_i(\mathbf{x}) \le \ \hat{q}_i^h + s_\varepsilon(f_i^h - \hat{q}_i^h) \quad (i \in I_h^{\ge}) \\ & \mathbf{x} \in S, \end{aligned}$$

or, equivalently,

$$\begin{aligned} \text{minimize} \quad & \max_{i \in I_h^{\le}} \left\{ \frac{f_i(\mathbf{x}) - \hat{q}_i^h}{f_i^h - \hat{q}_i^h} \right\} \\ \text{subject to} \quad & f_i(\mathbf{x}) \le \ f_i^h \quad (i \in I_h^{=}) \\ & f_i(\mathbf{x}) \le \ \hat{q}_i^h + s_\varepsilon(f_i^h - \hat{q}_i^h) \quad (i \in I_h^{\ge}) \\ & \mathbf{x} \in S. \end{aligned}$$

The corresponding values of the parameters and index sets can be found in Table 14.

ISWT method

In the interactive surrogate worth tradeoff (*ISWT*) (Chankong and Haimes 1978), the DM is shown tradeoffs among objectives, which the DM is asked to evaluate by assigning each of them a value val_i^h in a given scale. Given these values, the intermediate problems take the

Table 15 ISWT method

Index sets	$I^h_\alpha = \emptyset$	$I^h_\varepsilon = \{1, \ldots, k\} \setminus \{r\}$	
Weights	$\omega^h_i = 0$ $\omega^h_r = 1$	–	$\rho = 1$
Reference levels	–		
Objective bounds	$\varepsilon^h_i = f^h_i$	$\Delta\varepsilon^h_i = \mathrm{val}^h_i \cdot \lambda^h_{ri} \cdot \lvert f^h_i \rvert$	s_ε (varied)

Here $i = 1, \ldots, k$ and $i \neq r$

Table 16 Weighting method

Index sets	$I^h_\alpha = \emptyset$	$I^h_\varepsilon = \emptyset$	
Weights	ω^h_i $(i = 1, \ldots, k)$	–	$\rho = 1$
Reference levels	–		
Objective bounds	–	–	–

following form:

$$\begin{aligned}
&\text{minimize} && f_r(\mathbf{x})\\
&\text{subject to} && f_i(\mathbf{x}) \leq f^h_i + s_\varepsilon \cdot \mathrm{val}^h_i \cdot \lambda^h_{ri} \cdot |f^h_i| \quad (i = 1, \ldots, k,\ i \neq r)\\
& && \mathbf{x} \in S,
\end{aligned}$$

where λ^h_{ri} are the Karush-Kuhn-Tucker multipliers associated to the current solution. The values of the parameters and index sets are indicated in Table 15.

Weighting method

In the weighting method (Gass and Saaty 1955; Zadeh 1963), solutions are obtained by solving the following problem

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

for given weights ω^h_i (typically summing up to one). This problem can be obtained from GLIDE-II using the parameters and index sets given in Table 16. One should note that this problem can find any efficient solution only for convex problems (see, e.g. Miettinen 1999).

MICA method

The modified interactive Chebyshev method (MICA) (Luque et al. 2010) has been proposed for convex multiobjective problems, in order to deal with double reference point schemes (i.e. with both reservation and aspiration levels). Given a reference point consisting of aspiration levels $\hat{\mathbf{q}}^h$ and a reservation vector $\hat{\boldsymbol{\varepsilon}}^h$, at the first stage of the method some auxiliary

Table 17 MICA method—auxiliary problems (first level)

Index sets	$I_\alpha^h = \emptyset$	$I_\varepsilon^h = \{1, \dots, k\} \setminus \{r\}$	
Weights	$\omega_i^h = 0$ $\omega_r^h = 1$	–	$\rho = 1$
Reference levels	–		
Objective bounds	$\varepsilon_i^h = \hat{\varepsilon}_i^h$	$\Delta\varepsilon_i^h = 0$	s_ε=0

Here $i = 1, \dots, k$ and $i \neq r$

Table 18 MICA method—auxiliary problems (second level)

Index sets	$I_\alpha^h = \emptyset$	$I_\varepsilon^h = \{1, \dots, k\}$	
Weights	$\omega_i^h = 1$ $(i = 1, \dots, k)$	–	$\rho = 1$
Reference levels	–		
Objective	$\varepsilon_i^h = \hat{\varepsilon}_i^h \quad (i = 1, \dots, k, i \neq r)$	$\Delta\varepsilon_i^h = 0 \quad (i = 1, \dots, k)$	s_ε=0
bounds	$\varepsilon_r^h = f_r(\mathbf{x}^*)$		

programs $P_r(\hat{\boldsymbol{\varepsilon}}^h)$ for $r = 1, \dots, k$ are solved:

$$\begin{array}{ll} \text{lex minimize} & f_r(\mathbf{x}), \displaystyle\sum_{i=1}^{k} f_i(\mathbf{x}) \\ \text{subject to} & f_i(\mathbf{x}) \geq \hat{\varepsilon}_i^h \quad (i = 1, \dots, k, \ i \neq r) \\ & \mathbf{x} \in S. \end{array}$$

The first level of the lexicographic problem can be obtained from GLIDE-II using the parameters and index sets given in Table 17.

Let be $\mathbf{x}^*$ the optimal solution of the first priority level. Then, the problem corresponding to the second priority level can be obtained from GLIDE-II using the parameters and index sets given in Table 18.

Making use of the optimal solutions of the auxiliary problems, MICA builds several families of weights $\boldsymbol{\mu}^h$, and the intermediate problem of the Tchebycheff method (see Sect. 3.3.1) is solved for each of them. Therefore, Table 5 can be used to get the corresponding problems in formulations (3) and (4) by considering the aspiration levels $\hat{\mathbf{q}}^h$ instead of the utopian vector $\mathbf{z}^{\star\star}$. The reader is referred to Luque et al. (2010) for a detailed explanation of how to construct $\boldsymbol{\mu}^h$.

References

Benayoun, R., de Montgolfier, J., Tergny, J., & Laritchev, O. (1971). Linear 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*, 202–206.

Caballero, R., Luque, M., Molina, J., & Ruiz, F. (2002). PROMOIN: an interactive system for multiobjective programming. *International Journal of Information Technology & Decision Making*, *1*(4), 635–656.

Chankong, V., & Haimes, Y. Y. (1978). The interactive surrogate worth trade-off (ISWT) method for multiobjective decision-making. In S. Zionts (Ed.), *Multiple criteria problem solving* (pp. 42–67). Berlin: Springer.

Chankong, V., & Haimes, Y. Y. (1983). *Multiobjective decision making theory and methodology*. New York: Elsevier Science.

Chinchuluun, A., & Pardalos, P. M. (2007). A survey of recent developments in multiobjective optimization. *Annals of Operations Research*, *154*(1), 29–50.

Deb, K., & Miettinen, K. (2010). Nadir point estimation using evolutionary approaches: better accuracy and computational speed through focused search. In M. Ehrgott, B. Naujoks, T. J. Stewart, & J. Wallenius (Eds.), *Multiple criteria decision making for sustainable energy and transportation systems* (pp. 339–354). Berlin/Heidelberg: Springer.

Deb, K., Miettinen, K., & Chaudhuri, S. (2010). Towards an estimation of nadir objective vector using a hybrid of evolutionary and local search approaches. *IEEE Transactions on Evolutionary Computation*, *14*(6), 821–841.

Eschenauer, H. A., Osyczka, A., & Schäfer, E. (1990). Interactive multicriteria optimization in design process. In H. Eschenauer, J. Koski, & A. Osyczka (Eds.), *Multicriteria design optimization procedures and applications* (pp. 71–114). Berlin: Springer.

Fletcher, R. (2000). *Practical methods of optimization* (2nd ed.). New York: Wiley.

Fonseca, C. M., & Fleming, P. J. (1995). An overview of evolutionary algorithms in multi-objective optimization. *Evolutionary Computation*, *3*(1), 1–16.

Gardiner, L. R., & Steuer, R. E. (1994a). Unified interactive multiple objective programming. *European Journal of Operational Research*, *74*(3), 391–406.

Gardiner, L. R., & 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.

Gass, S., & Saaty, T. (1955). The computational algorithm for the parametric objective function. *Naval Research Logistics Quaterly*, *2*(1–2), 39–45.

Gill, P. E., Murray, W. W., & Wright, M. H. (1981). *Practical optimization*. London/New York: Academic Press.

Goldfarb, D., & Idnani, A. (1983). A numerically stable dual method for solving strictly convex quadratic problems. *Mathematical Programming*, *27*, 1–33.

Grauer, M., Lewandowski, A., & Wierzbicki, A. P. (1984). DIDASS—theory, implementation and experiences. In M. Grauer & A. P. Wierzbicki (Eds.), *Interactive decision analysis* (pp. 22–30). Berlin: Springer.

Hakanen, J., Kawajiri, Y., Miettinen, K., & Biegler, L. T. (2007). Interactive multi-objective optimization for simulated moving bed processes. *Control and Cybernetics*, *36*(2), 283–302.

Heikkola, E., Miettinen, K., & Nieminen, P. (2006). Multiobjective optimization of an ultrasonic transducer using NIMBUS. *Ultrasonics*, *44*(4), 368–380.

Hwang, C. L., & Masud, A. S. M. (1979). *Multiple objective decision making—methods and applications: a state-of-the-art survey*. Berlin: Springer.

Kaliszewski, I. (2004). Out of the mist—towards decision-maker-friendly multiple criteria decision making support. *European Journal of Operational Research*, *158*(2), 293–307.

Korhonen, P., & Laakso, J. (1986). A visual interactive method for solving the multiple criteria problem. *European Journal of Operational Research*, *24*(2), 277–287.

Korhonen, P., & Wallenius, J. (1988). A Pareto race. *Naval Research Logistics*, *35*(6), 615–623.

Laukkanen, T., Tveit, T.-M., Ojalehto, V., Miettinen, K., & Fogelholm, C.-J. (2010). An interactive multi-objective approach to heat exchanger network synthesis. *Computers and Chemical Engineering*, *34*(6), 943–952.

Lewandowski, A., Kreglewski, T., Rogowski, T., & Wierzbicki, A. P. (1989). Didass—theory, implementation and experiences. In A. Lewandowski & A. P. Wierzbicki (Eds.), *Aspiration based decision support systems: theory, software and applications* (pp. 21–47). Berlin: Springer.

Luque, M., Yang, J. B., & Wong, B. Y. H. (2009). PROJECT method for multiobjective optimization based on the gradient projection and reference point. *IEEE Transactions on Systems, Man and Cybernetics—Part A: Systems and Humans*, *39*(4), 864–879.

Luque, M., Ruiz, F., & Steuer, R. E. (2010). Modified interactive Chebyshev algorithm (MICA) for convex multiobjective programming. *European Journal of Operational Research*, *204*(3), 557–564.

Luque, M., Ruiz, F., & Miettinen, K. (2011). Global formulation for interactive multiobjective optimization. *OR Spectrum*, *33*(1), 27–48.

Miettinen, K. (1999). *Nonlinear multiobjective optimization*. Boston: Kluwer Academic.

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: Singapore.

Miettinen, K., & Mäkelä, M. M. (1995). Interactive bundle-based method for nondifferentiable multiobjective optimization: NIMBUS. *Optimization*, *34*(3), 231–246.

Miettinen, K., & Mäkelä, M. M. (2006). Synchronous approach in interactive multiobjective optimization. *European Journal of Operational Research*, *170*(7–8), 909–922.

Miettinen, K., Mäkelä, M. M., & Kaario, K. (2006). Experiments with classification-based scalarizing functions in interactive multiobjective optimization. *European Journal of Operational Research*, *175*(2), 931–947.

Miettinen, K., Ruiz, F., & Wierzbicki, A. (2008). Introduction to multiobjective optimization: interactive approaches. In J. Branke, K. Deb, K. Miettinen, & R. Słowiński (Eds.), *Multiobjective optimization: interactive and evolutionary approaches* (pp. 27–57). Berlin/Heidelberg: Springer.

NAG (2000). *Numerical algorithm group limited: NAG C library manual. Mark 6*. Oxford: NAG.

Nakayama, H., & Sawaragi, Y. (1984). Satisficing trade-off method for multiobjective programming. In M. Grauer & A. P. Wierzbicki (Eds.), *Interactive decision analysis* (pp. 113–122). Berlin: Springer.

Ogryczak, W., & Lahoda, S. (2006). Aspiration/reservation-based decision support—a step beyond goal programming. *Journal of Multi-Criteria Decision Analysis*, *1*(2), 101–117.

Pinter, J. D. (2001). *Computational global optimization in nonlinear systems: an interactive tutorial*. Atlanta: Lionheart.

Pinter, J. D. (2006). *Nonlinear optimization with MPL/LGO: introduction and user's guide*. Technical report, Maximal Software and PCS.

Romero, C. (1993). Extended lexicographic goal programming: a unified approach. *Omega*, *29*(1), 63–71.

Ruotsalainen, H., Boman, E., Miettinen, K., & Tervo, J. (2009). Nonlinear interactive multiobjective optimization method for radiotherapy treatment planning with Boltzmann transport equation. *Contemporary Engineering. Sciences*, *2*(9), 391–422.

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., & Tanino, T. (1985). *Theory of multiobjective optimization*. Orlando: Academic Press.

Steuer, R. E. (1986). *Multiple criteria optimization: theory, computation and application*. New York: Wiley.

Steuer, R. E., & Choo, E. U. (1983). An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming*, *26*(1), 326–344.

Vassilev, V., & Narula, S. C. (1993). A reference direction algorithm for solving multiple objective integer linear programming problems. *Journal of the Operational Research Society*, *44*(12), 1201–1209.

Vassilev, V., Narula, S. C., & Gouljashki, V. G. (2001). An interactive reference direction algorithm for solving multi-objective convex nonlinear integer programming problems. *International Transactions in Operational Research*, *8*(4), 367–380.

Wierzbicki, A. P. (1980). The use of reference objectives in multiobjective optimization. In G. Fandel & T. Gal (Eds.), *Multiple criteria decision making, theory and applications* (pp. 468–486). Berlin: Springer.

Wierzbicki, A. P., Makowski, M., & Wessels, J. (Eds.) (2000). *Model-based decision support methodology with environmental applications*. Dordrecht: Kluwer Academic.

Yang, J. B. (1999). Gradient projection and local region search for multiobjective optimization. *European Journal of Operational Research*, *112*(2), 432–459.

Zadeh, L. (1963). Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*, *8*(1), 59–60.