Computers & Operations Research 39 (2012) 1673–1681

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

A two-slope achievement scalarizing function for interactive multiobjective optimization

Mariano Luque [a,*], Kaisa Miettinen [b], Ana B. Ruiz [a], Francisco Ruiz [a]

[a] University of Málaga, Calle Ejido 6, 29071 Málaga, Spain
[b] Department of Mathematical Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland

ARTICLE INFO

Available online 10 October 2011

Keywords:
Multiobjective programming
Interactive methods
Reference point methods
Pareto optimality

ABSTRACT

The use of achievement (scalarizing) functions in interactive multiobjective optimization methods is very popular, as indicated by the large number of algorithmic and applied scientific papers that use this approach. Key parameters in this approach are the reference point, which expresses desirable objective function values for the decision maker, and weights. The role of the weights can range from purely normalizing to fully preferential parameters that indicate the relative importance given by the decision maker to the achievement of each reference value. Technically, the influence of the weights in the solution generated by the achievement scalarizing function is different, depending on whether the reference point is achievable or not. Besides, from a psychological point of view, decision makers also react in a different way, depending on the achievability of the reference point. For this reason, in this work, we introduce the formulation of a new achievement scalarizing function with two different weight vectors, one for achievable reference points, and the other one for unachievable reference points. The new achievement scalarizing function is designed so that an appropriate weight vector is used in each case, without having to carry out any a priori achievability test. It allows us to reflect the decision maker's preferences in a better way as a part of an interactive solution method, and this can cause a quicker convergence of the method. The computational efficiency of this new formulation is shown in several test examples using different reference points.

1. Introduction

Many real life problems involve dealing with several criteria, which must be maximized or minimized simultaneously. Such problems are called multiobjective optimization problems when both the criteria and the constraints that determine the feasible set of alternatives can be mathematically expressed by functions. Because the criteria, also known as objective functions, typically are conflicting, it is impossible to find a solution where all the objectives can reach their individual optima simultaneously. Instead, we can identify compromise solutions, that is, so-called Pareto optimal or nondominated points, where none of the objectives can get a better value without deteriorating at least one of the other objectives.

Many methods have been developed for solving multiobjective optimization problems during the years. They can be classified in three classes according to the role of the decision maker (DM) in the solution process (see, e.g., [6,11]). In the so-called *a posteriori methods* a representation of nondominated points is first generated and displayed to the DM who then is supposed to select the best of them as the final solution. The difficulty here is that it may be cognitively difficult for the DM to analyze all the provided solutions. Alternatively, the DM can specify desires and hopes before the solution process in the so-called *a priori methods*. The drawback here is that it may be difficult for the DM to set expectations on a realistic level before getting to know the problem. Finally, the third group comprises of *interactive methods*. The idea behind these algorithms is the gradual incorporation of the DM's preferences during the interactive and iterative solution process.

Interactive multiobjective optimization methods have been widely studied and used in real applications (see, e.g., [11,16] and references therein). In them, a solution pattern is formulated and repeated iteratively, and the DM takes actively part in the solution process by specifying and refining his/her preference information. There are many interactive methods and basically they differ from each other in what kind of information is asked for and shown to the DM at each iteration, as well as in the way the solutions are calculated. Examples of different types of preference information asked from the DM include marginal rates of substitution, surrogate values for trade-offs, classification of

* Corresponding author. Tel.: +34 952131173; fax: +34 952132061.
E-mail addresses: mluque@uma.es (M. Luque), kaisa.miettinen@jyu.fi (K. Miettinen), abruiz@uma.es (A.B. Ruiz), rua@uma.es (F. Ruiz).

0305-0548/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2011.10.002

objective functions and reference points. For further details, see [5,6,10,11,23] and references therein.

One of the ways to generate nondominated points in interactive methods is the use of an achievement scalarizing function. The popularity of the achievement scalarizing functions [24] in the framework of interactive methods is unquestionable. It is normally used within two different ways of specifying preference information. In reference point-based approaches (see [2,7,8,18,25]), the DM gives a reference value to each objective (these values constitute a so-called reference point), while in classification-based approaches (see [1,14]), the DM classifies the objectives into different categories (objectives to be improved, objectives than can be worsened, etc.). However, methods based on classification are closely related to reference point-based methods because a reference point can be formed once a classification has been made [14]. Once a reference point is given, the achievement scalarizing function is optimized to find the nondominated point that is, in some sense, closest to the reference point. For an overview of achievement scalarizing functions, see [13].

Regarding the interactive reference point-based methods, the main difference between them is the form of the weighting coefficients used in the achievement scalarizing function. In [12,13,21], wide studies of the performance of some of these methods (including classification based procedures), and of different weights, are presented and the following conclusion is reached: solutions obtained using different reference point-based methods (or classification-based ones) are, in fact, different. As a consequence, a synchronous approach is proposed in [14] where some of these solutions are calculated at each iteration and shown to the DM, who chooses the most preferred one according to his/her preferences.

Although different achievement scalarizing functions have been developed (for example, an additive achievement scalarizing function in [22]), the most widely used ones so far are extensions of the L_∞-distance (Chebychev distance). When the reference point is unachievable, these achievement scalarizing functions minimize the L_∞-distance between the reference point and the feasible set. In other words, the maximum (unwanted) deviation between the coordinates of the reference point and the feasible set is minimized. On the other hand, when the reference point is achievable, these achievement scalarizing functions minimize the maximum value of the negative differences between the coordinates of the reference point and the nondominated set, which is equivalent to maximizing the minimum deviation between the coordinates of the reference point and the feasible set in the objective space.

Several studies support the idea that it is better to use different vectors of weights, depending on the reference point given by the DM. In [21], it is shown that the effect of the weights on the relations between the reference values and the corresponding components of the optimal solution are completely different depending on whether the reference point is achievable or not. Furthermore, in [3] some reference point-based methods are compared, and experiments with real DMs are carried out to determine which solutions they prefer. In fact, solutions obtained by two different methods are compared: STOM [18] (Satisfying Trade-Off Method), where the reference point is projected onto the nondominated objective set, in the direction joining the ideal point with the reference point, and GUESS [2] (a naïve approach) where the reference point is projected onto the nondominated objective set, in the direction joining the nadir vector with the reference point. The conclusion is that if the reference point is achievable, a higher percentage of DMs (55.2%) prefers the *STOM solution* rather than the *GUESS solution*, while if the reference point is unachievable, a higher percentage of DMs (61%) prefers the *GUESS solution* instead of the *STOM solution*.

Different methods use different vectors of weights, which are hardly ever controlled by the DM in any way. But studies reflect that, when the reference point is achievable, the DMs tend to prefer the solutions obtained with certain weights, while when the reference point is unachievable, they often prefer others. On the other hand, the DM does not necessarily know in advance whether the reference point (s)he gives is achievable or not. Therefore, in this paper we suggest an achievement scalarizing function which automatically chooses a vector of weights for achievable reference points, and a different one for unachievable reference points.

Another possible application of an achievement scalarizing function which automatically adjusts itself according to the achievability of reference points can be found in [9], where it is shown that the use of preferential weights in achievement scalarizing functions allows us to obtain solutions that are more satisfactory to the DM, and speeds up the convergence of the algorithm. Several alternatives are provided for considering such preferential weights. In one of them, it is necessary to determine whether the reference point is achievable or not, because different weights are used in each case. This implies to first solve a subproblem in order to check the achievability of the reference point before the actual achievement scalarizing function can be solved.

As said, in this paper, we propose a new achievement scalarizing function with two different weight vectors: one is automatically used for unachievable reference points and the other one for achievable reference points. The advantage of this function is that we do not need to test whether the reference point is achievable or not before optimizing the achievement scalarizing function. Instead, the optimization process itself guarantees that the appropriate weight vector will be used in each case. This means that, for example, the previously mentioned preferential weights problem, proposed in [9], can be solved in a single problem, or following [3], we can consider the STOM weights for achievable reference points and the GUESS weights for unachievable reference points, again solving only a single problem per iteration of the interactive method.

To the authors' knowledge, an achievement scalarizing function corresponding to the one proposed here cannot be found in the literature. The most closely related function is described in [26], where an achievement scalarizing function with three different vectors of weights is proposed for a double reference point approach (involving both desirable and acceptable reference values, that is, aspiration and reservation levels, respectively). This function is defined in a branch-wise fashion, so that different weights are used depending on the relative position of the objective vector with respect to the reservation and the aspiration levels. Nevertheless, in this case the achievability of the reference point is not the issue (both the reservation and the aspiration levels can be achievable or not, without altering the corresponding weights), and the function defined implies an if–then formulation every time the achievement scalarizing function is evaluated.

The remainder of this paper is organized as follows. In Section 2, we introduce the main concepts and notations used. The new achievement scalarizing function is defined in Section 3, demonstrating the efficiency of the solutions obtained, and that an appropriate vector of weights is used in each case. In Section 4, the case of differentiable problems is analyzed. Some computational tests show the performance of our new achievement scalarizing function for both differentiable and nondifferentiable cases in Section 5 and finally, some conclusions are drawn in Section 6.

2. Formulation and background concepts

We consider *multiobjective optimization problems* of the form

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