Modified Interactive Chebyshev Algorithm (MICA) for Convex Multiobjective Programming

Mariano Luque
University of Málaga
Calle Ejido 6, 29071 Málaga, Spain
mluque@uma.es

Francisco Ruiz
University of Málaga
Calle Ejido 6, 29071 Málaga, Spain
rua@uma.es

Ralph E. Steuer
University of Georgia
Terry College of Business, Athens, Georgia 30602-6253, USA
rsteuer@uga.edu

Forthcoming in EJOR: DOI 10.1016/j.ejor.2009.11.011

Abstract

In this paper, we describe an interactive procedure algorithm for convex multiobjective programming based upon the Tchebycheff method, Wierzbicki's reference point approach, and the procedure of Michalowski and Szapiro. At each iteration, the decision maker (DM) has the option of expressing his or her objective-function aspirations in the form of a reference criterion vector. Also, the DM has the option of expressing minimally acceptable values for each of the objectives in the form of a reservation vector. Based upon this information, a certain region is defined for examination. In addition, a special set of weights is constructed. Then with the weights, the algorithm of this paper is able to generate a group of efficient solutions that provides for an overall view of the current iteration's certain region. By modification of the reference and reservation vectors, one can "steer" the algorithm at each iteration. From a theoretical point of view, we prove that none of the efficient solutions obtained using this scheme impair any reservation value for convex problems. The behavior of the algorithm is illustrated by means of graphical representations and an illustrative numerical example.

Keywords: multiobjective programming, interactive procedures, Tchebycheff method, reference point methods, aspiration criterion vectors, reservation levels

1 Introduction

When facing a real decision problem, a decision maker (DM) must often deal with several conflicting objectives. In such cases, the traditional optimization approach, in which a single objective is optimized subject to a given set of constraints, is no longer applicable. Instead, a multiobjective model is to be formulated and solved. Because of the rarity of solutions that optimize all objectives simultaneously, multiobjective programming utilizes efficient solutions. These are solutions from which no objective can be improved without deteriorating at least one of the others. Being "trade-off efficient" in this way, the set of all efficient solutions is precisely the set of all candidates for optimality. But as for which is to be optimal, this is for the DM to decide, and this often involves a contemplative process.

As outlined in Hwang and Masud (1979), procedures for solving multiobjective decision problems can be grouped into three categories depending upon whether preference information is elicited from the DM before, after, or during the solution process. In the "before" category are *a priori* methods. In these methods, after eliciting information from the DM, an optimization problem is solved to compute a solution. A difficulty of a priori methods is that it is hard to know in advance with sufficient accuracy the information required by the optimization problem for it to produce a *final* solution (an optimal solution or a solution close enough to one to qualify in its stead). Also, with these methods, there is the question about being able to recognizej a final solution even when confronted with one without knowing more about the efficient set. In the "after" category are *a posteriori* or, as called by Cohon (1985), *generating* methods. In these methods, a comprehensive set of efficient solutions (or in the best case the whole efficient set) is generated and shown to the DM. Then, the DM is to choose his or her most preferred solution from the set. The drawback of these methods is that usually a great number of efficient solutions has to be generated, and it can be extremely hard for the DM to manage all of the information.

In the "during" category are interactive procedures. Interactive procedures are designed to overcome the difficulties encountered in a priori and a posteriori methods. In interactive procedures, phases of information elicitation are interleaved with phases of computation. In the beginning, the information exchanged between the DM and procedure is general, but then becomes more local in character as the procedure continues. In this way, interactive procedures have two main features: (a) they help a DM learn about a problem while solving it and (b) they put to work iteratively any new insights gained during the solution process to help the DM navigate to a final solution. Prominent interactive procedures include STEM by Benayoun et al. (1971), the Zionts-Wallenius procedure (1976), Wierzbicki's reference point method (1980), interactive goal programming (see Spronk (1981) for instance), the Tchebycheff method of Steuer and Choo (1983), Pareto Race by Korhonen and Wallenius (1988), and the bi-reference procedure of Michalowski and Szapiro (1992).

Others that can be mentioned are the light beam search method of Jaszkiewicz and Słowinski (1999), Miettinen's NIMBUS (1999), and the normal vector identification approach of Yang and Li (2002).

The real forces behind so many interactive procedures have been the many different types of problems that lend themselves to multiple criteria analysis and the fact, due to the differences in the procedures, that the procedure to use in a given instance is typically application and decision-making style of the user dependent. Consequently, there has been a need for today's many interactive procedures. From the cognitive point of view, interactive procedures basically differ from one another in the way information is asked of the DM at each iteration. Four styles can be differentiated. One asks the DM to specify local tradeoffs or marginal rates of substitution between the objectives. Another asks the DM to select from several solution candidates at each iteration. A third asks the DM to specify target or aspiration levels for the different objectives, and a fourth asks the DM to classify the objectives, for instance, as to which are to be improved, which are permitted to become relaxed, and which are to be held at their current levels on the next iteration. Naturally, researchers have thought of consolidating procedures. There have been attempts to create global formulations such as by Gardiner and Steuer (1994) and Luque et al. (2009), and also to design combined implementations such as by Antunes et al. (1992) and Caballero et al. (2002). But mostly these have involved *procedure-switching*. By procedure-switching, we mean giving to the user the option to switch to any procedure on any iteration. For example, a user may choose to start with one procedure on the first iteration, switch to another for the second iteration, switch to a third for the third iteration, and so forth. But this significantly increases the cognitive burden when, if anything, we should be going in the opposite direction.

Primarily motivated by the Tchebycheff method of Steuer and Choo (1982), the reference point method of Wierzbicki (1980), and the bi-reference point procedure of Michalowski and Szapiro (1992), we present the modified interactive Chebychev algorithm (MICA) of this paper.[1] In one sense, MICA is similar to the Tchebycheff method in that it conducts multiple probing and uses "oversampling/filtering" techniques when developing each iteration's group of solutions to be presented to the DM. But MICA departs from the Tchebycheff procedure in the way the neighborhoods of the efficient set that are to be examined at each iteration are defined, shifted, contracted, and sampled. In the Tchebycheff method, the neighborhoods are defined, shifted, and contracted by manipulating subsets in weight space. Unfortunately, being in weight space, these manipulations are not very intuitive, and there is no intention of pursuing them further here. Rather, the neighborhoods to be explored in MICA are designed to be controlled by an iteratively adjustable aspiration criterion vector and an iteratively adjustable reservation vector. In this way, the aspiration and reservation vectors of a given iteration define the "frame" that contains the neighborhood of the efficient set to be explored on that iteration. And by adjusting the vectors, the neighborhoods can be shifted and contracted one iteration to the next in search of a final solution.

[1]We use the term Chebychev to stress MICA's relationship to, yet differences from, the Tchebycheff method of 1982

With regard to the sampling of the neighborhoods, it is to be pointed out that MICA possesses a special technical feature. The technical feature involves the way in which the weight vectors used to sample the neighborhoods are generated. As shown, they are specially generated so as to ensure that no reservation level of any objective is ever violated during the sampling process without ever having to include any reservation level in the constraint set of the program used to carry out the sampling operations.

There is also another item that arises in the paper. It stems from the number of procedures that currently comprise the field of interactive multiobjective programming. Even though the possession of many procedures is generally considered a strength of interactive multiobjective programming, things could well be different in the the future with the field ultimately becoming dominated by a much smaller number of procedures, each capturing the power of several of today's procedures without incurring a cognitive burden greater than any of the procedures singly. As we will see, in taking a step in this direction, MICA shows that at least some of this is possible.

The paper is organized as follows. Section 2 sets forth the problem to be addressed and reviews some background concepts. Because MICA draws upon features from the Tchebycheff method, Wierzbicki's aspiration criterion vector method, and the bi-reference procedure of Michalowski and Szapiro, these procedures are overviewed in Section 3. The basic philosophy of MICA is outlined in Section 4 along with details about how weight vectors can be generated so not to impair any reservation levels in the sampling process. A step-by-step description of MICA is given in Section 5. An example illustrating the operation of MICA comprises Section 6, and Section 7 brings the paper to a close with concluding remarks. A proof of the main theorem of the paper is given in the Appendix.

2 Formulation and Background Concepts

MICA is designed for the solution of the (convex) multiobjective problem

$$\begin{aligned} \max \quad & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x})) \\ & \text{subject to} \;\; \mathbf{x} \in X \end{aligned} \tag{1}$$

in which all f_i are continuous and (of course) concave, and $X \subset \mathbb{R}^n$, the feasible region in *decision space*, is closed, bounded and (of course) convex. Being a multiobjective problem, there is also $Z \subset \mathbb{R}^k$, the feasible region in *criterion space*, where $Z = \{\, \mathbf{z} = \mathbf{f}(\mathbf{x}) \mid \mathbf{x} \in X\}$. In decision space, $\bar{\mathbf{x}} \in X$ is *efficient* if and only if there does not exist another $\mathbf{x} \in X$ such that $\mathbf{f}(\mathbf{x}) \geq \mathbf{f}(\bar{\mathbf{x}})$ and $\mathbf{f}(\mathbf{x}) \neq \mathbf{f}(\bar{\mathbf{x}})$. Then, in criterion space, criterion vector $\tilde{\mathbf{z}} \in Z$ is *nondominated* if and only if there exists an $\tilde{\mathbf{x}} \in X$ such that $\tilde{\mathbf{z}} = \mathbf{f}(\tilde{\mathbf{x}})$ and $\tilde{\mathbf{x}}$ is efficient. The set of all efficient points is called the *efficient set* and is designated E. The set of all nondominated criterion vectors is called the *nondominated set*. Also, in decision space, $\bar{\mathbf{x}} \in X$ is *weakly* efficient if and only if there does not exist another $\mathbf{x} \in X$ such that $\mathbf{f}(\mathbf{x}) > \mathbf{f}(\bar{\mathbf{x}})$. Then, in criterion space, criterion vector $\tilde{\mathbf{z}} \in Z$ is *weakly* nondominated if and only if there exists an $\tilde{\mathbf{x}} \in X$ such that $\tilde{\mathbf{z}} = \mathbf{f}(\tilde{\mathbf{x}})$ and $\tilde{\mathbf{x}}$ is weakly efficient. Note that the set of all weakly efficient points subsumes all efficient points.

Ideal and *nadir* criterion vectors, $\mathbf{z}^*$ and $\mathbf{z}^{nad}$, whose components are given by

$$z_i^* = \max_{\mathbf{x} \in E} f_i(\mathbf{x}) \quad (i = 1, \ldots, k) \tag{2}$$

$$z_i^{nad} = \min_{\mathbf{x} \in E} f_i(\mathbf{x}) \quad (i = 1, \ldots, k) \tag{3}$$

are often of interest in multiobjective programming. One use of them, should $\mathbf{z}^{nad}$ be available, would be to form the intervals $[z_i^{nad}, z_i^*]$, $1 \leq i \leq k$, so as to frame a problem in the sense that no optimal solution will have any component outside its specified interval. Unfortunately, in many problems, nadir criterion vectors with all components known to be correct are difficult to obtain. While there is now the special algorithm by Alves and Costa (2009) for exactly computing nadir criterion values in multiobjective linear programming, in other cases, heuristics are often the rule as discussed, for instance, in Ehrgott and Tenfelde-Podehl (2003) and Deb, Chaudhuri and Miettinen (2006). Thus, often, a user will have no other choice but to use tentative values or approximations instead.

A further use, but of just $\mathbf{z}^*$, is to slightly adjust $\mathbf{z}^*$ to form a $\mathbf{z}^{**}$ ideal criterion vector that strictly dominates it. Most of the time this is overkill but is done to ensure that each nondominated criterion vector always has associated with it a $\boldsymbol{\lambda}$-vector

$$\boldsymbol{\lambda} \in \Lambda = \{\boldsymbol{\lambda} \in \mathbb{R}^k \mid \lambda_i \in (0,1), \sum_{i=1}^{k} \lambda_i = 1\}$$

that makes the nondominated criterion vector *uniquely* computable. By this we mean that when the $\boldsymbol{\lambda}$-vector is used in the Tchebycheff scalarizing program[2]

$$\begin{aligned} & \min \ \ \alpha \\ \text{subject to} \ \ & \lambda_i(z_i^{**} - f_i(\mathbf{x})) \leq \alpha \quad (i = 1, \ldots, k) \\ & z_i = f_i(\mathbf{x}) \quad (i = 1, \ldots, k) \\ & \mathbf{x} \in X \end{aligned} \tag{4}$$

where $\alpha \in \mathbb{R}$, or in its lexicographic variant, the *lexicographic* Tchebycheff sampling program

$$\begin{aligned} & \text{lex min} \ \ \alpha, -\sum_{i=1}^{k} f_i(\mathbf{x}) \\ \text{subject to} \ \ & \lambda_i(z_i^{**} - f_i(\mathbf{x})) \leq \alpha \quad (i = 1, \ldots, k) \\ & z_i = f_i(\mathbf{x}) \quad (i = 1, \ldots, k) \\ & \mathbf{x} \in X \end{aligned} \tag{5}$$

the nondominated criterion vector in question alone results.

[2] a variant of the scalarizing functional $s(\mathbf{q}, \mathbf{f}(\mathbf{x}), \boldsymbol{\lambda}) = \max_{i=1,\ldots,k} \{\lambda_i (q_i - f_i(\mathbf{x}))\}$ originally presented in Wierzbicki (1977) where $\mathbf{q} < \mathbf{z}^{**}$

Note that (4) is the first optimization stage of (5). As for the solutions returned by (4) or, identically, the first optimization stage of (5), all are weakly efficient. But at least one among them is efficient. Thus, when the solution to (4) or the first optimization stage of (5) is unique, it is efficient (i.e., has a nondominated criterion vector). But when there are multiple solutions to (4) or the first optimization stage of (5), the second optimization stage of (5) finds from among them an efficient point (i.e., one that is a member of E).

Formulations (4) and (5) work is as follows. Consider a $\boldsymbol{\lambda} \in \Lambda$ and the line going through $\mathbf{z}^{**}$ that is unbounded in both directions

$$-\left(\frac{1}{\lambda_1}, \ldots, \frac{1}{\lambda_k}\right) \quad \text{and} \quad +\left(\frac{1}{\lambda_1}, \ldots, \frac{1}{\lambda_k}\right) \tag{6}$$

Now imagine a *translated* non-negative orthant in $\mathbb{R}^k$ with its vertex (origin) attached to the line so that the non-negative orthant can slide up and down the line. Then the minimization of α in (4) or the first optimization stage of (5) moves the non-negative orthant as far as it can up the line (i.e., in the $+$ direction) to its position of last intersection with Z. If there is only one point of last intersection between Z and the translated non-negative orthant, the point is a member of E and we are done. But suppose there are more than one. Then, invoking the second optimization stage of (5) discards all points of last intersection whose criterion vector components do not sum to the greatest value, thus leaving behind only efficient points. In this way, the lexicographic Tchebycheff sampling program (5) always returns an efficient point whereas the Tchebycheff scalarizing program (4), because of alternative optima, may sometimes return just a weakly efficient point.

As for one more item, there is no guarantee that the criterion vector $\tilde{\mathbf{z}}$ returned by (4) or (5) is at the vertex of the last intersecting non-negative orthant. But if the components of the $\boldsymbol{\lambda}$ used in (4) or (5) were to satisfy

$$\lambda_i = \frac{1}{(z_i^{**} - \tilde{z}_i)} \left[\sum_{j=1}^{k} \frac{1}{(z_i^{**} - \tilde{z}_j)} \right]^{-1} \qquad (i = 1, \ldots, k) \tag{7}$$

the $\tilde{\mathbf{z}}$ in question would occur at the vertex of the last intersecting orthant. Because of this, the $\boldsymbol{\lambda}$-vector defined by (7) is called the *T-vertex* $\boldsymbol{\lambda}$-vector defined by $\tilde{\mathbf{z}}$ and $\mathbf{z}^{**}$.

3 Overview of Influencing Procedures

To make the paper as self-contained as possible and better see the influences of Wierzbicki's reference point method, the Tchebycheff method, and the procedure of Michalowski and Szapiro on MICA, these three procedures are outlined.

Using a $\mathbf{z}^{**}$, the basic reference point method of Wierzbicki begins by asking the DM to specify an initial aspiration criterion vector $\mathbf{q}^1 < \mathbf{z}^{**}$. Using the T-vertex $\boldsymbol{\lambda}$-vector defined by $\mathbf{q}^1$ and $\mathbf{z}^{**}$, the Tchebycheff scalarizing program (4) is solved for criterion vector solution $\mathbf{z}^1$. Commencing the second iteration, in the light of $\mathbf{z}^1$, the DM is asked to specify a second aspiration criterion vector $\mathbf{q}^2 < \mathbf{z}^{**}$. Using the T-vertex $\boldsymbol{\lambda}$-vector defined by $\mathbf{q}^2$ and $\mathbf{z}^{**}$, the Tchebycheff scalarizing program is solved for criterion vector solution $\mathbf{z}^2$. On the third iteration, in the light of $\mathbf{z}^2$, the DM is asked to specify a third aspiration criterion vector $\mathbf{q}^3 < \mathbf{z}^{**}$, and so forth. The method ends when the DM stops specifying new $\mathbf{q}$-vectors.

The Tchebycheff method begins by settling on the number of solutions P to be presented to the DM at each iteration. Commencing the first iteration, ρP dispersed $\boldsymbol{\lambda}$-vectors, where ρ is an *oversampling factor* like 2 or 3, are obtained from $\Lambda^1 = \{\boldsymbol{\lambda} \in \Lambda \mid \lambda_i \in [\ell_i^1, \mu_i^1]\}$. Being that this is the first iteration, in the specification of Λ^1, all intervals $[\ell_i^1, \mu_i^1] = [0, 1]$. Then the lexicographic Tchebycheff sampling program (5) is solved for each $\boldsymbol{\lambda}$ with the P most different of the resulting (nondominated) criterion vectors being presented to the DM for the selection of the solution of the first iteration $\mathbf{z}^1$. After tightening the intervals about the T-vertex $\boldsymbol{\lambda}$-vector defined by $\mathbf{z}^1$ and $\mathbf{z}^{**}$ to form the $[\ell_i^2, \mu_i^2]$ of the second iteration, ρP dispersed $\boldsymbol{\lambda}$-vectors are obtained from $\Lambda^2 = \{\boldsymbol{\lambda} \in \Lambda \mid \lambda_i \in [\ell_i^2, \mu_i^2]\}$. Then the lexicographic Tchebycheff sampling program is solved for each $\boldsymbol{\lambda}$ with the P most different of the resulting criterion vectors being presented to the DM for the selection of the solution of the second iteration $\mathbf{z}^2$. After tightening the intervals about the T-vertex $\boldsymbol{\lambda}$-vector defined by $\mathbf{z}^2$ and $\mathbf{z}^{**}$ to form the $[\ell_i^3, \mu_i^3]$ of the third iteration, the procedure keeps repeating until the $[\ell_i, \mu_i]$ interval widths are reduced to some pre-determined value.

After creating a $\mathbf{z}^{**}$ ideal criterion vector and ascertaining $\mathbf{z}^{nad}$ (or a substitute), the bi-reference procedure begins by asking the DM for an initial "worst" outcome vector $\boldsymbol{\varepsilon}^1$, $\mathbf{z}^{nad} \leq \boldsymbol{\varepsilon}^1 < \mathbf{z}^{**}$. Using the T-vertex $\boldsymbol{\lambda}$-vector defined by $\boldsymbol{\varepsilon}^1$ and $\mathbf{z}^{**}$, the lexicographic Tchebycheff sampling program with $\mathbf{z} \geq \boldsymbol{\varepsilon}^1$ augmenting the constraint set is solved for the criterion vector solution of the first iteration $\mathbf{z}^1$. After examining $\mathbf{z}^1$ and following some rules, the DM is asked to specify a "worst" outcome vector $\boldsymbol{\varepsilon}^2$ for the second iteration. Using the T-vertex $\boldsymbol{\lambda}$-vector defined by $\boldsymbol{\varepsilon}^2$ and $\mathbf{z}^{**}$, the lexicographic Tchebycheff sampling program with $\mathbf{z} \geq \boldsymbol{\varepsilon}^2$ augmenting the constraint set is solved for the criterion vector solution of the second iteration $\mathbf{z}^2$. After examining $\mathbf{z}^2$ and following the rules, the DM is asked to specify a "worst" outcome vector $\boldsymbol{\varepsilon}^3$ for the third iteration, and so forth. The procedure ends when the differences between the components of two successive resulting criterion vectors fall below some pre-set tolerance.

4 Theoretical Foundations of the Algorithm

This section overviews the operation of MICA. Basically, on each iteration, MICA uses an aspiration criterion vector and a reservation vector (MICA's counterpart of the bi-reference procedure's worst outcome vector) to define the portion of the nondominated set to be examined. Then it conducts multiple probes of the designated region to develop P representatives of it for presentation to the DM for the selection of the most preferred of the group.

More specifically, after creating a $\mathbf{z}^{**}$ ideal criterion vector, ascertaining $\mathbf{z}^{nad}$ (or a substitute such as from a payoff table if a better method is not available), and deciding on the number of criterion vectors P to be presented to the DM at each iteration, MICA begins by setting aspiration vector $\mathbf{q}^1$ to $\mathbf{z}^{**}$ and reservation vector

$\boldsymbol{\varepsilon}^1$ to $\mathbf{z}^{nad}$. With this intending to place the whole nondominated set within the frame, MICA then multi-probes, in its first iteration, the (whole) nondominated set in exactly the same fashion as done in the first iteration of the Tchebycheff method. That is, ρP dispersed $\boldsymbol{\lambda}$-vectors are obtained from Λ and the lexicographic Tchebycheff sampling program (5) is solved for each of them. Then from the P most different of the resulting (nondominated) criterion vectors, the DM selects his or her most preferred for the solution of the first iteration $\mathbf{z}^1$.

On the second iteration, in the light of $\mathbf{z}^1$, the DM specifies aspiration vector $\mathbf{q}^2 \leq \mathbf{z}^{**}$ and reservation vector $\boldsymbol{\varepsilon}^2 \geq \mathbf{z}^{nad}$. At this point, ρP *new* $\boldsymbol{\lambda}$-vectors are computed. Exactly how the "new" $\boldsymbol{\lambda}$-vectors are computed, which is key to MICA, is discussed after the next few paragraphs. Then, to multi-probe the portion of the nondominated set falling within the frame defined by $\boldsymbol{\varepsilon}^2$ and $\mathbf{q}^2$, the lexicographic Tchebycheff sampling program is solved for each of them with the DM selecting from the P most different of the resulting criterion vectors the solution of the second iteration $\mathbf{z}^2$. All subsequent iterations follow the pattern of this (the second) iteration until the DM wishes to conclude.

Notice that in MICA the focus at each iteration h is on the portion of the efficient set whose criterion vectors are in the intervals $[\varepsilon_i^h, q_i^h]$. These ranges can be specified in at least two ways. One is to specify the ranges directly and then induce from them $\mathbf{q}^h$ and $\boldsymbol{\varepsilon}^h$. The other is to specify $\mathbf{q}^h$ and $\boldsymbol{\varepsilon}^h$ first and then deduce from them the ranges.

At the beginning of each iteration $h \geq 2$, after the reservation vector of the iteration has been specified, MICA solves the following series of *auxiliary* programs

$$\begin{aligned} \text{lex max} \quad & f_r(\mathbf{x}), \sum_{i=1}^{k} f_i(\mathbf{x}) && (P_r(\boldsymbol{\varepsilon}^h)) \\ \text{subject to} \quad & f_i(\mathbf{x}) \geq \varepsilon_i^h \quad (i = 1, \ldots, k, \ i \neq r) \\ & \mathbf{x} \in X \end{aligned}$$

$(r = 1, \ldots, k)$. The auxiliary problems have several uses. One is to assure that the reservation vector specified for the iteration is indeed feasible according to the following definition.

Definition 1. Given (1), reservation vector $\boldsymbol{\varepsilon}^h$ is *feasible* if and only if there exists an $\mathbf{x} \in X$ such that $\mathbf{f}(\mathbf{x}) \geq \boldsymbol{\varepsilon}^h$.

The feasibility of any candidate reservation vector $\boldsymbol{\varepsilon}^h$ is determined by solving any auxiliary program, such as $P_1(\boldsymbol{\varepsilon}^h)$. Should the auxiliary program be infeasible or have an optimal first-stage objective function value less that ε_1^h, then another reservation vector must be sought.

As for notation in the auxiliary programs, let $(\mathbf{x}^{hr}, \mathbf{z}^{hr})$ denote the solution of $P_r(\boldsymbol{\varepsilon}^h)$ with $\mathbf{z}^{hr} = \mathbf{f}(\mathbf{x}^{hr})$. Should a DM not wish to specify a reservation value for an objective on a given iteration, the nadir value for the objective (or the best estimate that we have for it) is to be used instead.

We now look at two cases. In the first, as in Figure 1, $\mathbf{q}^h$ strictly dominates all $\mathbf{z}^{hr}$. Then the T-vertex $\boldsymbol{\lambda}$-vectors defined by $\mathbf{q}^h$ and the $\mathbf{z}^{hr}$, denoted $\boldsymbol{\lambda}^{hr}$, are each strictly positive. Note that if we let the $\mathbf{u}^{hr}$ denote the vectors that connect the $\mathbf{z}^{hr}$ to $\mathbf{q}^h$, then their directions

$$\left(\frac{1}{\lambda_1^{hr}}, \ldots, \frac{1}{\lambda_k^{hr}}\right) \qquad (r = 1, \ldots, k) \tag{8}$$

are strictly positive, too.

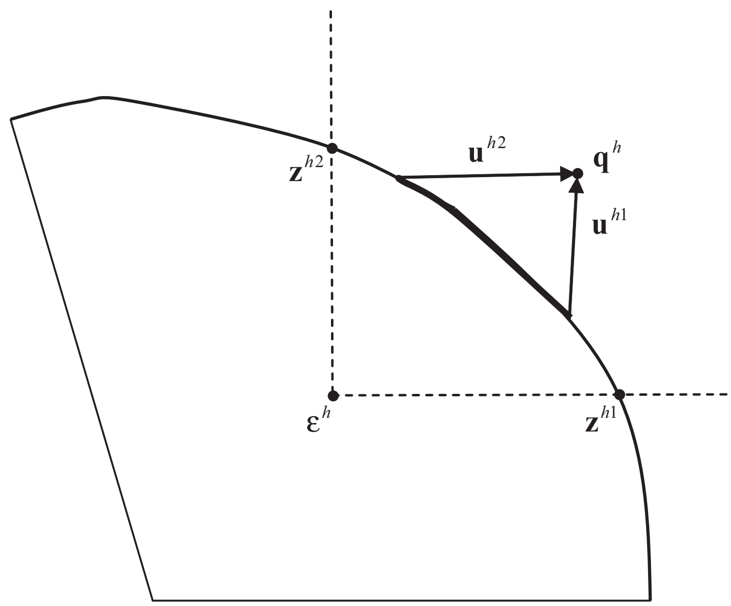

Figure 1: Case one, when $\mathbf{q}^h$ strictly dominates all $\mathbf{z}^{hr}$. The portion of the efficient set operating within the intervals defined by $\boldsymbol{\varepsilon}^h$ and $\mathbf{q}^h$ is indicated by the thick line.

In the second case, as in Figure 2, there is an r for which $\mathbf{q}^h \not> \mathbf{z}^{hr}$. Then the $\boldsymbol{\lambda}^{hr}$ defined by $\mathbf{q}^h$ and $\mathbf{z}^{hr}$ for this r would not be strictly positive. Thus, for such an r, we construct a "modified" $\boldsymbol{\lambda}^{hr}$. Toward this end, we form index sets

$$\begin{aligned} I^{hr} &= \{i \mid z_i^{hr} \geq q_i^h\} \\ J^{hr} &= \{i \mid z_i^{hr} < q_i^h\} \end{aligned} \tag{9}$$

and then reset the components of $\boldsymbol{\lambda}^{hr}$ to

$$\lambda_i^{hr} = \begin{cases} \frac{1}{\delta}\, C^{hr} & \text{if } i \in I^{hr} \\ \frac{1}{(q_i^h - z_i^{hr})}\, C^{hr} & \text{if } i \in J^{hr} \end{cases} \tag{10}$$

where δ is a small, but numerically significant, positive number, that satisfies the following relation:

$$\delta \leq \min_{i=1,\ldots,k} \{q_i^h - z_i^{hr}\} \tag{11}$$

and

$$C^{hr} = \left[\sum_{j \in J^{hr}} \frac{1}{(q_j^h - z_j^{hr})} + \text{card}(I^{hr}) \frac{1}{\delta} \right]^{-1}$$

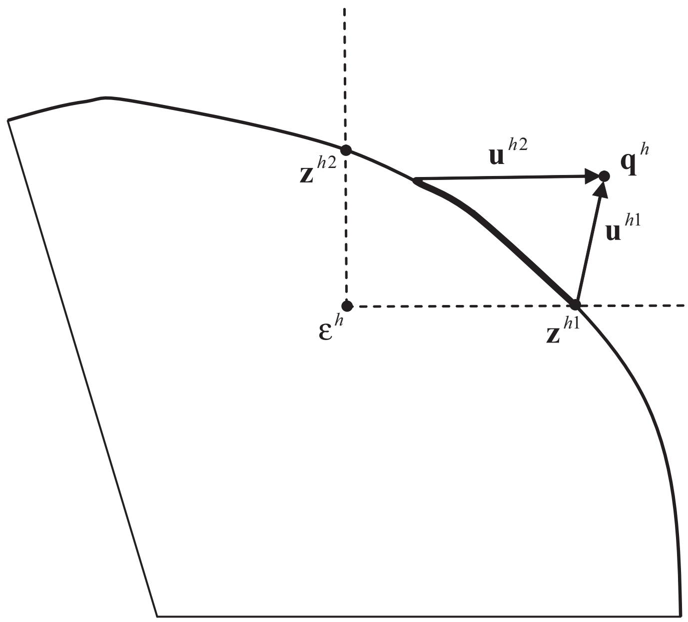

Figure 2: Case two, when there exists an r for which $\mathbf{q}^h \ngtr \mathbf{z}^{hr}$. The (small) slope of $\mathbf{u}^{h2}$ is given by δ.

Reforming

$$\left(\frac{1}{\lambda_1^{hr}}, \ldots, \frac{1}{\lambda_k^{hr}}\right) \tag{12}$$

with the modified $\boldsymbol{\lambda}$-vector for each r such that $\mathbf{q}^h \ngtr \mathbf{z}^{hr}$, all resulting $\mathbf{u}^{hr}$ are strictly positive as their components (see Figure 2) are given by

$$u_i^{hr} = \begin{cases} \delta & \text{if } i \in I^{hr} \\ (q_i^h - z_i^{hr}) & \text{if } i \in J^{hr} \end{cases}$$

Thus, we have the following general rule. For each r such that $\mathbf{q}^h > \mathbf{z}^{hr}$, we construct $\boldsymbol{\lambda}^{hr}$ is the usual way, and for each r such that $\mathbf{q}^h \ngtr \mathbf{z}^{hr}$, we construct $\boldsymbol{\lambda}^{hr}$ in modified fashion.

Once all $\boldsymbol{\lambda}^{hr}$ have been constructed, the next step, motivated by the Tchebycheff method, is to generate a collection of weight vectors that are convex combinations of the $\boldsymbol{\lambda}^{hr}$. For showing P solutions to the DM at each iteration, this is done as follows. ρP dispersed weight vectors, designated $\{\boldsymbol{\omega}^1, \ldots, \boldsymbol{\omega}^{\rho P}\}$, each with all non-negative components that sum to one, are generated using, for instance, the LAMBDA/FILTER strategy described in Steuer and Choo (1983).

Then, for each $\boldsymbol{\omega}^j$, we build $\bar{\boldsymbol{\lambda}}^j = (\bar{\lambda}_1^j, \ldots, \bar{\lambda}_k^j)$ as follows:

$$\bar{\lambda}_i^j = \left(\sum_{r=1}^{k} \omega_r^j \frac{1}{\lambda_i^{hr}}\right)^{-1} \quad (i = 1, \ldots, k) \tag{13}$$

That is,

$$\frac{1}{\bar{\lambda}_i^j} = \sum_{r=1}^{k} \omega_r^j \frac{1}{\lambda_i^{hr}} \quad (i = 1, \ldots, k)$$

After employing each $\bar{\boldsymbol{\lambda}}^j$ in the lexicographic Tchebycheff sampling program, ρP solutions are obtained. Filtering to obtain the P most different, we have the solutions to be shown to the DM.

The question now is this: Might any of these solutions be in violation of the reservation values? Theorem 1 proves that none of the solutions obtained using the described scheme impairs the reservation levels. In Theorem 1, all of the following *conditions* are assumed:

1. X is a convex set, and all the objective functions f_i are concave.
2. $\boldsymbol{\varepsilon}^h$ is a feasible reservation vector, and $\mathbf{q}^h$ is such that $\mathbf{q}^h > \boldsymbol{\varepsilon}^h$.
3. Vectors $\mathbf{z}^{hr}$ $(r = 1, \dots, k)$ are obtained by solving the auxiliary problems and all corresponding $\boldsymbol{\lambda}^{hr}$ are computed via (10).
4. Each $\boldsymbol{\omega}^j$ vector is such that $\omega_r^j \geq 0, (r = 1, \dots, k)$ and $\sum_{r=1}^k \omega_r^j = 1$.
5. Each $\bar{\boldsymbol{\lambda}}^j$ is computed as per (13).

Theorem 1. *Consider problem* (1) *for which condition* 1 *holds. Let* $\boldsymbol{\varepsilon}^h$, $\mathbf{q}^h$ *and* $\bar{\boldsymbol{\lambda}}^j$ *satisfy conditions* 2 to 5. *For any* $\bar{\boldsymbol{\lambda}}^j$, *let* $(\bar{\mathbf{x}}, \bar{\mathbf{z}})$ *be the solution returned by the lexicographic Tchebycheff sampling program. Then* $\bar{\mathbf{z}} \geq \boldsymbol{\varepsilon}^h$.

Proof. See the Appendix. □

This result shows that all solutions obtained using the proposed scheme satisfy the reservation values in the convex case.

While MICA is described with all objectives in maximization form, without loss of generality, any mix of maximization and minimization objectives can be handled. In the event of minimization objectives, one needs to note the following:

a. For each minimization objective, the optimizations of (2) and (3) are to be reversed.
b. For each minimization objective i, recognize that z_i^{nad} is to be its maximum value over the efficient set and that $[z_i^*, z_i^{nad}]$ is to be its interval for framing a problem.
c. Whereas each maximization objective has its z_i^* augmented by a small positive value to form z_i^{**}, each minimization objective has its z_i^* augmented by a small negative value to form its z_i^{**}.
d. For each minimization objective i, constraint $\lambda_i(z_i^{**} - f_i(\mathbf{x})) \leq \alpha$ in (4) and (5) is converted to $\lambda_i(f_i(\mathbf{x}) - z_i^{**}) \leq \alpha$.

5 MICA Algorithm

The modified interactive Chebyshev algorithm is now described in step-by-step fashion.

Step 1. Initialization. Given problem (1), specify a $\mathbf{z}^{**}$ ideal vector, what is to be used for $\mathbf{z}^{nad}$ (either its exact specification or a good substitute), and the number of solutions P to be shown to the DM at each iteration. Let $\mathbf{q}^1 = \mathbf{z}^{**}$ and $\boldsymbol{\varepsilon}^1 = \mathbf{z}^{nad}$. Also, from Λ, obtain ρP dispersed weight vectors

$$\bar{\boldsymbol{\lambda}}^1, \bar{\boldsymbol{\lambda}}^2, \ldots, \bar{\boldsymbol{\lambda}}^{\rho P}$$

Let $h = 1$.

Step 2. Sampling the efficient set. For each $\bar{\boldsymbol{\lambda}}^j$, $1 \leq j \leq \rho P$, solve the lexicographic Tchebycheff sampling problem (5). Filter the resulting ρP criterion vectors to obtain the P most different for presentation to the DM.

Step 3. Termination rule. If after selecting the most preferred of the P criterion vectors the DM wishes to cease iterating, go to Step 8.

Step 4. Adjusting reservation vector. Let $h = h+1$. The DM is asked if the current reservation vector is to be updated. If so, the DM specifies adjusted reservation vector $\boldsymbol{\varepsilon}^h$. If not, let $\boldsymbol{\varepsilon}^h = \boldsymbol{\varepsilon}^{h-1}$ and go to Step 6.

Step 5. Solve auxiliary problems. Solve problems $P_r(\boldsymbol{\varepsilon}^h)$, $1 \leq r \leq k$, to check the feasibility of $\boldsymbol{\varepsilon}^h$ and obtain criterion vectors $\mathbf{z}^{h1}, \ldots, \mathbf{z}^{hk}$ as described in Section 4.

Step 6. Adjusting aspiration criterion vector. The DM is asked if the current aspiration vector is to be updated. If so, the DM specifies adjusted aspiration vector $\mathbf{q}^h$. If not, let $\mathbf{q}^h = \mathbf{q}^{h-1}$.

Step 7. Generate weight vectors. Form the vectors $\boldsymbol{\lambda}^{hr}$, $1 \leq r \leq k$, as in (10). Randomly select ρP weight vectors $\boldsymbol{\omega}^j$ such that $\omega_r^j \geq 0, (r = 1, \ldots, k)$ and $\sum_{r=1}^{k} \omega_r^j = 1$. For each $\boldsymbol{\omega}^j$, construct its $\bar{\boldsymbol{\lambda}}^j$ as in (13). Go to step 2.

Step 8. Stop with $(\mathbf{x}^h, \mathbf{z}^h)$ as the final solution.

Notice in Step 3 that termination is under the control of the DM. The procedure concludes when the DM decides, not when forced to by some mechanical condition. In Steps 4 and 6, the portion of the efficient set contained in any frame can be targeted by the appropriate specification of reservation and aspiration criterion vectors. Also, in Steps 4 and 6, it is possible to conduct more than one iteration with the same reservation and aspiration vectors. Because of the randomness embedded in Step 7, one might want to do something like this, for instance, at the end just to confirm that a better solution cannot be found. Also, there is no reason why the ρ's in Step 2 and 7 can't be different. It just depends upon the degree of resolution desired in carrying out each step.

6 Illustrative Example

In this section, we employ a numerical example to illustrate MICA. The example has been adapted from a problem in Caballero et al. (2002) which deals with a hierarchical multiobjective model. The problem, which is convex, is as follows

$$
\begin{array}{ll}
\text{minimize} & f_1(\mathbf{x}) = (x_1 - 2)^2 + 2(x_2 - 3)^2 \\
\text{minimize} & f_2(\mathbf{x}) = x_3^2 + 2x_4^2 \\
\text{minimize} & f_3(\mathbf{x}) = 2(x_5 - 5)^2 + (x_6 - 1)^2 \\
\text{minimize} & f_4(\mathbf{x}) = 3x_7^2 + x_8^2 \\
\text{subject to} & x_3 + x_4 + x_2 - x_5 = 2 \\
& x_1^2 + x_2^2 \leq 5 \\
& x_1 - x_3 \leq 0 \\
& x_2 - x_4 \leq 0 \\
& x_1 + x_2 + x_5 + x_7 - 2x_8 = 0 \\
& x_7^2 + x_8^2 \leq 8 \\
& x_1, x_2, x_5, x_6 \geq 0 \\
& x_3 \geq 1, \quad x_8 \geq 0.5
\end{array}
$$

Given the minimization objectives, in this problem the z_i^* and z_i^{nad} reflect the minimum and maximum criterion values over the efficient set. In the case of this problem, they have been extracted from the payoff table and are shown in Table 1 although in practice additional work might be conducted to see if better nadir values can't be obtained. We now simulate MICA with $\rho = 2$, and $\delta = 10^{-1}$.

Objective		$\mathbf{z}^*$	$\mathbf{z}^{nad}$
1	min	3.00	14.73
2	min	1.50	21.47
3	min	0.00	50.00
4	min	0.25	8.31

Table 1: Payoff table obtained minimum and maximum criterion values over the efficient set

Iteration 1

Step 1. Let $P = 4$, $\mathbf{q}^1 = \mathbf{z}^{**} = (2.99, 1.49, -0.01, 0.24)$ and $\boldsymbol{\varepsilon}^1 = \mathbf{z}^{nad} = (14.73, 21.47, 50.00, 8.31)$. Then obtain $\rho P = 8$ dispersed weight vectors from Λ, denoted $\bar{\boldsymbol{\lambda}}^j$, $1 \leq j \leq 8$, and let $h = 1$.

Step 2. Solving the lexicographic Tchebycheff sampling problem for each $\bar{\boldsymbol{\lambda}}^j$ and filtering the 8 resulting criterion vectors to obtain the $P = 4$ most different, the DM is shown:

(6.30, 11.03, 3.11, 8.24) (7.35, 6.07, 15.86, 3.58)
(4.21, 7.61, 9.86, 8.05) (12.78, 14.22, 4.36, 4.52)

Step 3. Assuming the DM chooses the second from the group, $\mathbf{z}^1 = (7.35, 6.07, 15.86, 3.58)$. The DM does not wish to stop.

Iteration 2

Steps 4 and 5. $h = 2$. Assume that the DM decides to change the reservation vector and proposes $(7, 8, 13, 4)$. Solving the first auxiliary problem, this reservation vector is found not to be feasible. Attempting another in the form of $(7.5, 8.5, 14, 5)$ to make sure objective 3 still improves, this reservation vector is found to be feasible. Running the auxiliary problems with $\boldsymbol{\varepsilon}^2 = (7.5, 8.5, 14, 5)$ to calculate the $\mathbf{z}^{2r}$, we have

$$\mathbf{z}^{21} = (5.4, 6.63, 14, 5) \quad \mathbf{z}^{22} = (7.49, 6.32, 14, 4.78)$$
$$\mathbf{z}^{23} = (7.5, 8.26, 8.18, 5) \quad \mathbf{z}^{24} = (7.5, 6.63, 14, 3.75)$$

Step 6. Wishing to set demanding aspiration levels on objectives 1, 3 and 4, the DM specifies $\mathbf{q}^2 = (5, 7, 10, 2)$.

Step 7. Calculating the $\boldsymbol{\lambda}^{2r}$ by means of (10), we have

$$\boldsymbol{\lambda}^{21} = (0.193, 0.763, 0.019, 0.025) \quad \boldsymbol{\lambda}^{22} = (0.037, 0.908, 0.023, 0.033)$$
$$\boldsymbol{\lambda}^{23} = (0.035, 0.069, 0.868, 0.029) \quad \boldsymbol{\lambda}^{24} = (0.036, 0.891, 0.022, 0.051)$$

After randomly generating $\rho P = 8$ dispersed $\boldsymbol{\omega}^j$ vectors with $\omega_r^j \geq 0, (r = 1, \ldots, k)$ and $\sum_{r=1}^k \omega_r^j = 1$, their $\bar{\boldsymbol{\lambda}}^j$ are computed in accordance with (13). Go to Step 2.

Step 2. Solving the lexicographic Tchebycheff sampling problem for each of the $\bar{\boldsymbol{\lambda}}^j$ and filtering the 8 resulting criterion vectors to obtain the $P = 4$ most different, the DM is shown:

$$(7.13, 7.09, 12.60, 4.15) \quad (6.54, 7.22, 12.23, 4.52)$$
$$(7.16, 7.28, 11.68, 4.32) \quad (6.75, 6.97, 12.94, 4.28)$$

Step 3. Assuming the DM chooses the fourth from the group, $\mathbf{z}^2 = (6.75, 6.97, 12.94, 4.28)$. In this way, objectives 1 and 3 are improved, but at the expense of the other two. The DM wishes to go on iterating.

Iteration 3

Step 4. $h = 3$. Suppose the DM wishes to leave the reservation vector as is. Thus $\boldsymbol{\varepsilon}^3 = \boldsymbol{\varepsilon}^2 = (7.5, 8.5, 14, 5)$. Go to Step 6.

Step 6. Wishing to improve objectives 3 and 4 and at the expense of some impairment in the first objective, the DM specifies $\mathbf{q}^3 = (6, 7, 9, 1)$.

Step 7. Calculating the $\boldsymbol{\lambda}^{3r}$ by means of (10), we have

$$\boldsymbol{\lambda}^{31} = (0.489, 0.489, 0.010, 0.012) \quad \boldsymbol{\lambda}^{32} = (0.060, 0.898, 0.018, 0.024)$$
$$\boldsymbol{\lambda}^{33} = (0.057, 0.068, 0.854, 0.021) \quad \boldsymbol{\lambda}^{34} = (0.059, 0.890, 0.018, 0.032)$$

After randomly generating $\rho P = 8$ dispersed $\boldsymbol{\omega}^j$ vectors with $\omega_r^j \geq 0, (r = 1, \ldots, k)$ and $\sum_{r=1}^k \omega_r^j = 1$, their $\bar{\boldsymbol{\lambda}}^j$ are computed in accordance with (13). Go to Step 2.

Step 2. Solving the lexicographic Tchebycheff sampling problem for each of the $\bar{\boldsymbol{\lambda}}^j$ and filtering the 8 resulting criterion vectors to obtain the $P = 4$ most different, the DM is shown:

$$\begin{array}{cc} (7.06, 7.04, 12.75, 4.16) & (6.74, 7.26, 12.13, 4.43) \\ (6.72, 6.91, 13.12, 4.27) & (6.61, 7.03, 12.76, 4.39) \end{array}$$

Steps 3 and 8. Assuming the DM chooses the first from the group, $\mathbf{z}^3 = (7.06, 7.04, 12.75, 4.16)$. Deciding to conclude with this solution, the DM goes to Step 8 for termination.

In this example, different types of iterations have been carried out to illustrate the different possibilities of the algorithm. For reasons of simplicity, the number of solutions shown to the DM was 4, although larger numbers would be preferable in practice.

7 Conclusions

In this paper, we have designed an algorithm that combines multiple probing from the Tchebycheff method, aspiration criterion vectors from the work of Wierzbicki, and reservation vectors from Michalowski and Szapiro. The purpose is to capture the broad-based powers of these procedures within the context of a single-pattern user interface with a low cognitive burden. In MICA, all that is required at each iteration is to select from a small group of solutions and then specify criterion value intervals for the next iteration, information that is natural and couldn't be much easier to provide.

Throughout its history, interactive multiobjective programming has taken pride in the diverse spectrum of different procedures that it has had at its disposal for solving multiobjective programming problems. However, this may just be a symptom that the field is still in its early stages. In the future it could well be that there are fewer prevailing procedures with each capturing the power of several of today's procedures while still keeping the user's side of the interface single pattern and relatively plain vanilla. With the algorithm of this paper showing some universality in this regard, it is our belief that the structure of MICA and the simplicity of its interface lend credence to this prognostication about the future.

The theoretical results proved in this paper assure that the efficient solutions generated by the algorithm always satisfy the reservation levels set by the DM in convex cases. In the non-convex case, there are no guarantees, but we think that the algorithm might even run well in the majority of non-convex cases. This will be the subject of future investigations.

Appendix

In this appendix, we prove Theorem 1 (which was stated in Section 4).

Theorem 1. *Consider problem* (1) *for which condition* 1 *holds. Let* $\boldsymbol{\varepsilon}^h$, $\mathbf{q}^h$ *and* $\bar{\boldsymbol{\lambda}}^j$ *satisfy conditions* 2 *to* 5. *For any* $\bar{\boldsymbol{\lambda}}^j$, *let* $(\bar{\mathbf{x}}, \bar{\mathbf{z}})$ *be the solution returned by the lexicographic Tchebycheff sampling program. Then* $\bar{\mathbf{z}} \geq \boldsymbol{\varepsilon}^h$.

Proof. Since any $\mathbf{x}$ that is an optimal solution of (5) for a given $\bar{\boldsymbol{\lambda}}^j$ is also an optimal solution of (4) for the same $\bar{\boldsymbol{\lambda}}^j$, then:

$$\forall \mathbf{x} \in X, \quad s(\mathbf{q}^h, \bar{\mathbf{z}}, \bar{\boldsymbol{\lambda}}^j) \leq s(\mathbf{q}^h, \mathbf{f}(\mathbf{x}), \bar{\boldsymbol{\lambda}}^j).$$

Let us define a vector $\boldsymbol{\beta}^h = (\beta_1^h, \ldots, \beta_k^h) \in \mathbb{R}^k$ in the following way:

$$\beta_r^h = \frac{\omega_r^j}{C^{hr}} \left(\sum_{s=1}^{k} \frac{\omega_s^j}{C^{hs}} \right)^{-1} \qquad r = 1, \ldots, k. \tag{14}$$

Given that, for all $r = 1, \ldots, k$, $\omega_r^j > 0$ and $C^{hr} > 0$, then:

$$\beta_r^h > 0 \quad (r = 1, \ldots, k), \qquad \text{and} \qquad \sum_{r=1}^{k} \beta_r^h = 1.$$

Since X is convex, then $\sum_{r=1}^k \beta_r^h \mathbf{x}^{hr} \in X$, and given that all the functions f_i are concave, we have:

$$f_i \left(\sum_{r=1}^{k} \beta_r^h \mathbf{x}^{hr} \right) \geq \sum_{r=1}^{k} \beta_r^h z_i^{hr} \quad (i = 1, \ldots, k) \Rightarrow$$
$$\Rightarrow s \left(\mathbf{q}^h, \mathbf{f} \left(\sum_{r=1}^{k} \beta_r^h \mathbf{x}^{hr} \right), \bar{\boldsymbol{\lambda}}^j \right) \leq s \left(\mathbf{q}^h, \sum_{r=1}^{k} \beta_r^h \mathbf{z}^{hr}, \bar{\boldsymbol{\lambda}}^j \right).$$

Taking again into account that $\sum_{r=1}^k \beta_r^h \mathbf{x}^{hr} \in X$ and given that $\bar{\boldsymbol{z}}$ is an optimal solution of problem (4) with weights $\bar{\boldsymbol{\lambda}}^j$, then

$$s(\mathbf{q}^h, \bar{\mathbf{z}}, \bar{\boldsymbol{\lambda}}^j) \leq s \left(\mathbf{q}^h, \mathbf{f} \left(\sum_{r=1}^{k} \beta_r^h \mathbf{x}^{hr} \right), \bar{\boldsymbol{\lambda}}^j \right) \leq s \left(\mathbf{q}^h, \sum_{r=1}^{k} \beta_r^h \mathbf{z}^{hr}, \bar{\boldsymbol{\lambda}}^j \right) \Rightarrow$$
$$\Rightarrow \max_{i=1,\ldots,k} \left\{ \bar{\lambda}_i^j (q_i^h - \bar{z}_i) \right\} \leq \max_{i=1,\ldots,k} \left\{ \bar{\lambda}_i^j \left(q_i^h - \sum_{r=1}^{k} \beta_r^h z_i^{hr} \right) \right\}. \tag{15}$$

Next, we are going to find a bound on the right hand side of inequality (15). Since $\sum_{r=1}^k \beta_r^h = 1$, and taking into account the definition of $\bar{\boldsymbol{\lambda}}^j$ given in (13), we have:

$$\max_{i=1,\ldots,k} \left\{ \bar{\lambda}_i^j \left(q_i^h - \sum_{r=1}^{k} \beta_r^h z_i^{hr} \right) \right\} = \max_{i=1,\ldots,k} \left\{ \frac{1}{\sum_{r=1}^k \omega_r^j \frac{1}{\lambda_i^{hr}}} \sum_{r=1}^{k} \beta_r^h (q_i^h - z_i^{hr}) \right\} \tag{16}$$

For each $i = 1, \ldots, k$, let us denote:

$$J_i = \left\{ r \in \{1, \ldots, k\} \mid i \in J^{hr} \right\}, \quad I_i = \left\{ r \in \{1, \ldots, k\} \mid i \in I^{hr} \right\}.$$

Note that $J_i \cap I_i = \emptyset$, and $J_i \cup I_i = \{1, \ldots, k\}$. Therefore,

$$\sum_{r=1}^{k} \beta_r^h (q_i^h - z_i^{hr}) = \sum_{r \in J_i} \beta_r^h (q_i^h - z_i^{hr}) + \sum_{r \in I_i} \beta_r^h (q_i^h - z_i^{hr}).$$

On the other hand, from (10), it follows that:

$$\forall i \in I_i \quad q_i^h - z_i^{hr} \leq 0 < \delta = \frac{C^{hr}}{\lambda_i^{hr}},$$

$$\forall i \in J_i \quad q_i^h - z_i^{hr} = \frac{C^{hr}}{\lambda_i^{hr}}.$$

Substituting these relations into equation (16), and taking into account the definition of β_r^h given in (14), we obtain:

$$\max_{i=1,\ldots,k} \left\{ \bar{\lambda}_i^j \left(q_i^h - \sum_{r=1}^{k} \beta_r^h z_i^{hr} \right) \right\} \leq \max_{i=1,\ldots,k} \left\{ \frac{1}{\sum_{r=1}^{k} \omega_r^j \frac{1}{\lambda_i^{hr}}} \sum_{r=1}^{k} \beta_r^h \frac{C^{hr}}{\lambda_i^{hr}} \right\} =$$

$$= \max_{i=1,\ldots,k} \left\{ \frac{1}{\sum_{r=1}^{k} \omega_r^j \frac{1}{\lambda_i^{hr}}} \sum_{r=1}^{k} \frac{\omega_r^j}{C^{hr}} \left(\sum_{s=1}^{k} \frac{\omega_s^j}{C^{hs}} \right)^{-1} \frac{C^{hr}}{\lambda_i^{hr}} \right\} = \left(\sum_{s=1}^{k} \frac{\omega_s^j}{C^{hs}} \right)^{-1}.$$

Next, we shall apply this bound in relation (15):

$$\max_{i=1,\ldots,k} \left\{ \bar{\lambda}_i^j (q_i^h - \bar{z}_i) \right\} \leq \left(\sum_{s=1}^{k} \frac{\omega_s^j}{C^{hs}} \right)^{-1} \Rightarrow$$

$$\Rightarrow q_i^h - \bar{z}_i \leq \left(\sum_{s=1}^{k} \frac{\omega_s^j}{C^{hs}} \right)^{-1} \frac{1}{\bar{\lambda}_i^j} = \left(\sum_{s=1}^{k} \frac{\omega_s^j}{C^{hs}} \right)^{-1} \sum_{r=1}^{k} \omega_r^j \frac{1}{\lambda_i^{hr}} \quad (i = 1, \ldots, k).$$

From the definition of λ_i^{hr} given in (10), we obtain:

$$q_i^h - \bar{z}_i \leq \left(\sum_{s=1}^{k} \frac{\omega_s^j}{C^{hs}} \right)^{-1} \left(\sum_{r \in J_i} \omega_r^j \frac{q_i^h - z_i^{hr}}{C^{hr}} + \sum_{r \in I_i} \omega_r^j \frac{\delta}{C^{hr}} \right) =$$

$$= \sum_{r \in J_i} \frac{\omega_r^j}{C^{hr}} \left(\sum_{s=1}^{k} \frac{\omega_s^j}{C^{hs}} \right)^{-1} (q_i^h - z_i^{hr}) + \sum_{r \in I_i} \frac{\omega_r^j}{C^{hr}} \left(\sum_{s=1}^{k} \frac{\omega_s^j}{C^{hs}} \right)^{-1} \delta \quad (i = 1, \ldots, k).$$

Taking into account the definition of β_r^h given in (14), it follows that

$$q_i^h - \bar{z}_i \leq \sum_{r \in J_i} \beta_r^h (q_i^h - z_i^{hr}) + \sum_{r \in I_i} \beta_r^h \delta \quad (i = 1, \ldots, k).$$

Taking into account the definition of problems $P_r(\boldsymbol{\varepsilon}^h)$, it is obvious that $\mathbf{z}^{hr} \geq \boldsymbol{\varepsilon}^h$ $(r = 1, \ldots, k)$. Therefore, given that δ satisfies relation (11), we obtain:

$$q_i^h - \bar{z}_i \leq \sum_{r \in J_i} \beta_r^h (q_i^h - z_i^{hr}) + \sum_{r \in I_i} \beta_r^h (q_i^h - \varepsilon_i^h) \leq$$

$$\leq \sum_{r \in J_i} \beta_r^h (q_i^h - \varepsilon_i^h) + \sum_{r \in I_i} \beta_r^h (q_i^h - \varepsilon_i^h) = \sum_{r=1}^{k} \beta_r^h (q_i^h - \varepsilon_i^h) \quad (i = 1, \ldots, k).$$

Finally, given that $\sum_{r=1}^{k} \beta_r^h = 1$, we have

$$q_i^h - \bar{z}_i \leq \sum_{r=1}^{k} \beta_r^h (q_i^h - \varepsilon_i^h) = (q_i^h - \varepsilon_i^h) \sum_{r=1}^{k} \beta_r^h = q_i^h - \varepsilon_i^h \quad (i = 1, \ldots, k),$$

and therefore,

$$\bar{z}_i \geq \varepsilon_i^h \quad (i = 1, \ldots, k),$$

and this completes the proof. □

References

[1] M. J. Alves and J. P. Costa. An exact method for computing the nadir values in multiple objective linear programming. *European Journal of Operational Research*, 198(2):637–466, 2009.

[2] C. H. Antunes, M. J. Alves, A. L. Silva, and J. N. Climaco. An integrated MOLP method base package – a guided tour of TOMMIX. *Computers & Operations Research*, 19(7):609–625, 1992.

[3] R. Benayoun, J. de Montgolfier, J. Tergny, and O. Laritchev. Linear programming with multiple objective functions: Step method (STEM). *Mathematical Programming*, 1(3):366–375, 1971.

[4] R. Caballero, T. Gómez, M. Luque, F. Miguel, and F. Ruiz. Hierarchical generation of pareto optimal solutions in large-scale multiobjective systems. *Computers & Operations Research*, 29(11):1537–1558, 2002.

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

[6] J. L. Cohon. Multicriteria programming: Brief review and application. In J. S. Gero, editor, *Design Optimization*, pages 163–191. Academic Press, 1985.

[7] K. Deb, S. Chaudhuri, and K. Miettinen. Towards estimating nadir objective vector using evolutionary approaches. In M. Keijzer et al., editor, *Proceedings of the 8th Annual Genetic and Evolutionary Computation Conference*, pages 643–650. Volume 1, The Association of Computing Machinery, New York, 2006.

[8] M. Ehrgott and D. Tenfelde-Podehl. Computation of ideal and nadir values and implications for their use in MCDM methods. *European Journal of Operational Research*, 151(1):119–131, 2003.

[9] L. R. Gardiner and R. E. Steuer. Unified interactive multiple objective programming. *European Journal of Operational Research*, 74(3):391–406, 1994.

[10] C. L. Hwang and A. S. M. Masud. *Multiple Objective Decision Making – Methods and Applications: A State-of-the-Art Survey*. Lecture Notes in Economics and Mathmatical Systems, Vol. 164, Springer Verlag, Berlin, 1979.

[11] A. Jaszkiewicz and R. Słowinski. The light beam search approach -an overview of methodology and applications. *European Journal of Operational Research*, 113(2):300–314, 1999.

[12] P. Korhonen and J. Wallenius. A Pareto Race. *Naval Research Logistics*, 35(6):615–623, 1988.

[13] M. Luque, F. Ruiz, and K. Miettinen. Global formulation for interactive multiobjective optimization. *OR Spectrum*, to appear, 2009.

[14] W. Michalowski and T. Szapiro. A bi-reference procedure for interactive multiple objective programming. *Operations Research*, 40:247–258, 1992.

[15] K. M. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer, Boston, 1999.

[16] J. Spronk. *Interactive Multiple Goal Programming: Applications to Financial Management*. Martinus Nijhoff Publishing, Boston, 1981.

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

[18] A. P. Wierzbicki. Basic properties of scalarizing functionals for multiobjective optimization. *Mathematische Operationsforschung und Statistik*, 8:55–60, 1977.

[19] A. P. Wierzbicki. The use of reference objectives in multiobjective optimization. In G. Fandel and T. Gal, editors, *Lecture Notes in Economics and Mathematical Systems,* vol. 177, pages 468–486. Springer-Verlag, Berlin, 1980.

[20] J. B. Yang and D. Li. Normal vector identification and interactive tradeoff analysis using minimax formulation in multiobjective optimisation. *IEEE Transactions on Systems, Man, and Cybernetics*, 32(3):305–319, 2002.

[21] S. Zionts and J. Wallenius. An interactive programming method for solving the multiple criteria problem. *Management Science*, 22(6):652–663, 1976.