OR Spectrum (2005) 27: 633–652
DOI: 10.1007/s00291-004-0191-5

OR Spectrum

An interactive method for multiple objective linear fractional programming problems[*]

João Paulo Costa

Faculdade de Economia da Universidade de Coimbra,
Av. Dias da Silva, 165, 3004-512 Coimbra, Portugal, and
INESC –Instituto de Engenharia de Sistemas e Computadores, Portugal
(e-mail: jpaulo@fe.uc.pt)

Abstract. Multiple objective linear fractional programming (MOLFP) is an important field of research. Using some branch and bound techniques, we have developed a new interactive method for MOLFP that drastically reduces the computational effort needed, while providing guidance for the decision maker in the choice of his/her preferred solutions. The basic idea of the computation phase of the algorithm is to optimize one of the fractional objective functions while constraining the others. Several linear programming problems, organized in a tree structure, are generated as the search evolves. The whole idea is simple and it results in a fast and very intuitive approach to exploring the non-dominated set of solutions in MOLFP, and eventually to finding the preferred solution.

Keywords: Fractional programming – Multiple objective programming – Interactive methods

1 Introduction

Fractional programming has been attracting the attention of a fair amount of research all over the world. A bibliography with 386 entries presented by Stancu-Minasian (1999), mainly covering the years from 1990 to ‘96, gives a clear idea of the amount of work that has been done in the field in recent years. It should be noted that this bibliography is an extension of four other bibliographies previously published by the same author that cover previous periods. Schaible (1981) has published a comprehensive review of the work in fractional programming, outlining some of its major developments. Stancu-Minasian’s textbook (1997) contains the

[*] This work was supported by the Fundação para a Ciência e Tecnologia and FEDER, project POCTI/32405/GES/2000.

state-of-the-art theory and practice of fractional programming, allowing the reader to quickly become acquainted with what has been done in the field.

The main reason for the interest in and importance of fractional programming stems from the fact that various problems, ranging from engineering to economics, consider the optimization of a ratio between physical and/or economic linear functions. In many linear applications, for example, the programming models could better fit the 'real problem' if we used linear fractional goals (see Kornbluth and Steuer, 1981a; Schaible, 1981; Schaible and Ibaraki, 1983; Craven, 1988; Goedhart and Spronk, 1995; Stancu-Minasian, 1997; among others).

The explicit use of several criteria or objectives can also make a model to better fit most decision situations where several conflicting, or at least different, points of view have to be considered. These situations usually need not only mathematical tools and techniques to support the determination of one solution (or a small set of solutions), but they also need some decision aid in order to help elicit the decision maker's (DM) preferences. Interactive methods are usually chosen to support both these needs, when a multiobjective linear programming (MOLP) problem is used to model the decision situation.

Although the interactive character of a method may have distinct, particular purposes (ranging from a free learning search to targeted and fast convergent searches), it can still be modelled by the repetition of two basic phases: i) the dialogue with the DM (seeking more information about preferences), and ii) the computation of non-dominated solutions (using the newly-elicited information). Reference may be made to Steuer (1986) for a review of the more representative existing MOLP interactive methods, while noticing the existence of improved methods or versions of the same methods.

The computation phase of a MOLP method is very easy when compared with the complexity and computational burden usually needed to compute one non-dominated solution in multiple objective linear fractional programming (MOLFP). Knowing that in an interactive method the computation phase can be repeated several times (more or less times, depending on the particular method) one becomes aware of the difficulty of developing this kind of method in MOLFP; it is necessary to use a simple and straightforward process in the computation phase.

This paper presents a new interactive method for computing non-dominated solutions and finding the preferred one in MOLFP. As in MOLP, we have two sequential phases (dialogue/computation) performed repeatedly. A special effort has been made to develop an easy and fast process for the computation phase, while keeping the elicitation of preference information as natural and comfortable (for the DM) as possible and guiding the DM in his/her search for the preferred solution. The basic idea of the computation phase of the algorithm is to optimize one of the fractional objective functions while constraining the others. Several linear programming problems, organized in a tree structure, are generated as the search progresses. The whole idea is simple and it results in a fast and very intuitive approach to the exploration of the non-dominated set of solutions in MOLFP and eventually to finding the preferred solution. The proposed method can also be applied to other kinds of multiple objective programming problems. For instance, in the case of MOLP the method can be applied directly and the computations are even easier. Other types of problem may require significant adaptations, focusing on the computation phase.

Kornbluth and Steuer (1981a) can be considered the seminal work on MOLFP. In this paper they describe its main characteristics, difficulties and differences from MOLP. They also present one method to compute all the non-dominated (and weakly non-dominated) solutions of a MOLFP problem. This method is complicated by having to know the ‘break point’ of the ‘broken edges’, which is the point where a feasible edge changes from dominated to non-dominated or vice-versa. Benson (1985) presents another approach to computing these break-points. Nykowski and Zolkiewski (1985) describe a procedure for choosing the DM’s preferred non-dominated solution based on the method of Kornbluth and Steuer (1981a). In it, they first generate the set of weakly non-dominated solutions and then work on it to choose the preferred solution.

Goal programming ideas have also been applied to MOLFP, resulting in the development of interactive methods. Kornbluth and Steuer (1981b) present the mechanism of a method, based on weighted goal programming and needing the Kornbluth and Steur (1981a) method to generate all the weakly non-dominated vertices of a given feasible sub-region. By choosing a non-dominated solution the DM implicitly chooses a sub-region of the weight space. The objective function values of the selected solution can be used to further reduce the feasible region and the corresponding weight space. This mechanism results in complex computations because it needs to generate all the weakly non-dominated vertices of a region. Goedhart and Spronk (1995) present an interactive goal programming method, where the DM sets minimum goals for each objective function. The maximum value that each objective function can achieve, taking those minimum goals into account, are then computed (straightforwardly) and presented to the DM. Then s/he can change the minimum goals so as to compute the next range of values. The presentation of this method does not include a proof of convergence to a non-dominated solution. Other approaches have also been applied to MOLFP, e.g. the trade-off satisficing method (Nakayama, 1991). This is a general method whose application to MOLFP requires the use of an iterative algorithm for solving a linear fractional min-max problem. Nakayama (1991) uses a Dinkelbach type algorithm which implies the resolution of a sequence of auxiliary linear programming problems. Ferland and Potvin (1985) compare the performance of several algorithms of this kind for solving the so called generalized fractional programming problem.

The paper is structured as follows. First we present the notation and formulation of a MOLFP problem. We also describe a straightforward process that has been used to compute a monocriterion linear fractional programming problem, because this is the calculation base of the method presented in this paper. Section 3 formally develops the algorithm and gives a general idea of its usage. We also supply a proof of the convergence of the algorithm. The next section presents an illustration and discusses certain issues of the method’s usage. Finally, in the last section we present the conclusions.

2 The MOLFP

In this paper we formulate the multiple objective linear fractional programming (MOLFP) problem, like the one below:

$$\begin{array}{l} \max \left\{ z_1 = \dfrac{c^1 x + \alpha_1}{d^1 x + \beta_1} \right\} \\ \ldots \\ \max \left\{ z_p = \dfrac{c^p x + \alpha_p}{d^p x + \beta_p} \right\} \\ \text{s.t.}: \ x \in S = \{x \in R^n | Ax = b, \ x \geq 0, \ b \in R^m\} \end{array}$$

where c^k, $d^k \in R^n$, and $\alpha_k, \beta_k \in R, k = 1, ..., p$ and $d^k x + \beta_k > 0, k = 1, ..., p$ and $x \in S$.

We want to find the DM's preferred solution among the set of all non-dominated solutions defined by this problem. We will differentiate between the weakly non-dominated solutions – a point $x' \in S$ is weakly non-dominated if and only if there does not exist another point $x \in S$ such that $z_k(x) \geq z_k(x')$, for all $k = 1, ..., p$ – and the non-dominated solutions – a point $x' \in S$ is non-dominated if and only if there does not exist another point $x \in S$ such that $z_k(x) \geq z_k(x')$, for $k = 1, ..., p$, and $z_k(x) > z_k(x')$ for at least one k.

Kornbluth and Steuer (1981a) present and illustrate several aspects of a MOLFP problem that distinguish it from a MOLP problem, and make MOLFP a far more complex problem to deal with:

- The set of non-dominated solutions does not need to be closed.
- Some interior solutions may be non-dominated while others, from the same hyper-face, can be dominated.
- Two non-dominated vertices (in the simplex method sense) do not have to be connected by a path of non-dominated edges.
- There may exist some edges that are partially non-dominated and partially dominated.

They also give two other results:

- The set of all weakly non-dominated solutions is closed.
- If the feasible set is bounded, the set of weakly non-dominated vertices is connected by a finite number of weakly non-dominated line segments.

We can consider three main techniques – widely used by practitioners and researchers in multiobjective programming – for computing a non-dominated solution by transforming the multiple criteria in a single criterion problem.

i) The weighted sum of the objective functions formulated as:

$$\begin{array}{l} \max \left\{ \lambda_1 \dfrac{c^1 x + \alpha_1}{d^1 x + \beta_1} + \ \cdots \ + \lambda_p \dfrac{c^p x + \alpha_p}{d^p x + \beta_p} \right\} \\ \text{s.t.} \quad x \in S \end{array}$$

where $\lambda \in R^p$ is defined according to the preferences of the DM. Usually $\sum_{k=1}^{p} \lambda_k = 1$ in order to normalise the weights and $\lambda_k > 0, k = 1, ..., p$, in order to prevent the result from being a weakly non-dominated solution.

We can see that the linear numerator and denominator of the fractional criteria turn out to be (as a rule) polynomials of degree p in the single criterion problem. This transformation of the MOLFP to a single criterion problem therefore leads to a problem that is very hard to solve by the techniques currently available. Konno and Yamashita (1999) present an algorithm that computes, with a pre-defined error, the solution of this problem. The algorithm uses a convex approximation of the objective function and solves a series of non-linear programs, each time enhancing the objective function approximation. Costa and Lourenço (2001a,b) present another algorithm for computing the solution of this problem, also with a pre-defined error, but using branch and bound linear techniques.

ii) The reference point approach:
The following achievement scalarizing function (or a similar one) can be defined:

$$s(q,z) = \max_{k=1,\ldots,p} (q_k - z_k) - \varepsilon \sum_{k=1}^{p} z_k$$

where $q \in R^p$ is a reference point in the criterion space and is defined according to the DM's preferences, the parameter $\varepsilon \in R^+$ is used to prevent the result from being a weakly non-dominated solution and must be sufficiently small (to allow for the possibility of determining all the interesting non-dominated solutions).

The scalarized problem can be formulated as:

$$\begin{aligned}
&\min \left\{ \delta - \varepsilon \sum_{k=1}^{p} \frac{c^k x + \alpha_k}{d^k x + \beta_k} \right\} \\
&\text{s.t.}: \quad \delta \geq \left(q_k - \frac{c^k x + \alpha_k}{d^k x + \beta_k} \right), k = 1, \cdots, p \\
&\qquad\quad x \in S \\
&\qquad\quad \delta \in R
\end{aligned}$$

We can see that the objective function turns out to be as complicated as in the weighted sum approach. Moreover (even if we consider $\varepsilon = 0$, disregarding the weakly non-dominated solutions issue), the auxiliary constraints, corresponding to the objective functions, are not linear because there are some terms with the variable δ multiplied by x. Thus, the scalarized problem in MOLFP is also a very difficult problem to solve. Metev and Gueorguieva (2000) report several characteristics of the scalarizing function and of the auxiliary problem that allows non-linear algorithms to be used to give local maximums.

iii) Constraining the objective functions:
In this technique we maximize one of the objective functions while constraining the others; that is, we solve the problem:

$$\begin{aligned}
&\max \left\{ \frac{c^i x + \alpha_i}{d^i x + \beta_i} \right\} \\
&\text{s.t.}: \quad \frac{c^k x + \alpha_k}{d^k x + \beta_k} \geq \bar{z}_k \Leftrightarrow (c^k - \bar{z}_k d^k)x \geq \bar{z}_k \beta_k - \alpha_k; k = 1, \cdots, p; k \neq i \\
&\qquad\quad x \in S
\end{aligned}$$

Where $\bar{z}_k \in R$ and is defined according to the preferences of the DM.

In the set of alternative optima of this problem there will be at least one non-dominated solution or the set is empty (the other solutions will be weakly non-dominated). The computations required to solve this problem are far easier than those required by the earlier approaches. There is a variable change technique developed by Charnes and Cooper (1962) (and used by Steuer, 1986; Goedhart and Spronk, 1995; among others) that turns this problem into a linear program, making it very simple.

Consider the following single criterion problem:

$$\max\left\{z = \frac{cx+\alpha}{dx+\beta}\right\}$$
$$\text{s.t.}: \quad x \in S = \{x \in R^n : Ax = b, x \geq 0, b \in R^m\}$$

where $c, d \in R^n$; $\alpha, \beta \in R$ and $\forall x \in S, dx + \beta > 0$.

We define the new variables:

$$t = \frac{1}{dx+\beta} \text{ and } y = xt.$$

Substituting the variables we arrive at the following linear program:

$$\begin{aligned} &\max\{z = cy + \alpha t\} \\ &\text{s.t.}: \quad Ay - bt = 0 \\ &\qquad\quad dy + \beta t = 1 \\ &\qquad\quad y \in R^n, y \geq 0, t \in R, t > 0 \end{aligned}$$

So, having achieved an easy calculus technique, two problems remain in the objective function constraining technique: which function to maximize and how to define the limits, $\bar{z}_k$, on the other objective functions, in order to find the DM's preferred non-dominated solution. The method presented in this paper uses this calculus technique and has been specially designed to determine the objective function to maximize and the limits on the others, in an interactive way. The dialogue with the DM consists of presenting some typical non-dominated solutions, of which the DM must choose one to continue the search for other solutions that are closer to his/her preferences. It is on the basis of this chosen non-dominated solution that the method determines the function to maximize and the new limits on the other objective functions. The next section gives full details of the method.

3 The interactive method

In this section we present a new interactive method for computing the preferred non-dominated solution of a MOLFP. The method has two sequential phases (computation/dialogue) performed repeatedly.

Section 2 shows that a simple linear programming problem can be used to obtain the solution. This is done for each objective function, initially without constraining the objective functions, thus computing the pay-off table (Step 2.1 of the method). One more non-dominated solution, approximately in the middle of the non-dominated region, is also computed (Step 2.2). This ‘middle’ solution is computed with the help of an auxiliary constraint dividing into two equal parts the objective function variation interval whose the range is the biggest in the pay-off table, and, by maximizing the objective function that corresponds to the pay-off table solution, having the minimum value for the objective function to constrain. The computed $(p+1)$ non-dominated solutions characterize the non-dominated region and are presented to the DM (Step 3). S/he must choose the non-dominated solution closest to her/his preferences in order to proceed (or the method stops if the DM is already satisfied). The method uses this solution to define the objective function’s lower limits for use in the next iteration, making several sub-sets of the non-dominated region (Step 4). This process is then repeated for each sub-set. That is, for each sub-set $(p+1)$, non-dominated solutions are calculated and presented to the DM and s/he must choose the solution to proceed. The method is formalized as follows.

The method:

Step 1 – Starting
$g = 1; h = 1. g^h$ is the label of each sub-set of the feasible region. g represents the current level of search and h the sub-set associated with the improvement of objective function h, relative to the current chosen non-dominated solution.

$S(g^h)$ is the feasible region associated with the sub-set g^h. Initially $S(g^h) = S(1^1) = S$.

Step 2 – Characterizing the sub-set g^h

Step 2.1 – Computing the pay-off table of the sub-set g^h
Compute $z^k = z(x^{*k}); k = 1, \ldots, p$; where x^{*k} is non-dominated and optimizes the program $(k = 1, ..., p)$:

$$\begin{aligned} \max \quad & z_k(x) \\ \text{s.t.}: \quad & x \in S(g^h) \end{aligned}$$

Step 2.2 – Computing the solution in the ‘middle’ of the non-dominated region
The index of the objective function to constrain, denoted r, corresponds to the one having the biggest column range in the pay-off table, computed in Step 2.1:

$$r = k : \max_{k=1,\ldots,p} \left\{ \varDelta z_k = \left(z_k^k - \min_{k \neq j} \left\{ z_k^j \right\} \right) \right\}$$

The index of the objective function to maximize, denoted by v, corresponds to the row of the solution that has the minimum value in the pay-off table in column (r) to constrain:

$$v = j : \min_{j=1,\ldots,p} z_r^j$$

Compute $z^{p+1} = z(x^*)$, where x^* is a non-dominated solution that optimizes:

$$\begin{aligned} \max \quad & z_v\left(x\right) \\ s.t.: \quad & x \in S\left(g^h\right) \\ & z_r\left(x\right) \geq z_r^v + {}^1\!/_2 \varDelta z_r \end{aligned}$$

Step 3 – Choosing the solution to proceed or to stop
The DM chooses an incumbent solution z^* from the non-dominated solutions computed in the last step. If the DM is satisfied with this solution the method stops. If not, the DM sets $\gamma(g^h) \in R^+$. By default γ is a small positive number, but the DM can change it. This parameter will define how far from z^* the next generated solutions will be. It also defines the part of the non-dominated region that the DM does not want to search any more. A small γ will make the search proceed more slowly than a bigger γ, but it should be settled as the minimum amount of increase of the objective functions values that will make the next z* more interesting for the DM, in order to make sure that the DM will not jump potentially interesting solutions.

Note that it is possible to set different values for γ according to the amount of improvement that the DM wants for each objective function. In this case, $\gamma \in R^p$, $\gamma_k \geq 0, k = 1,\ldots,p$, and $\gamma_k > 0$ for at least one k. Some of the γ can be set to zero, meaning that the DM is satisfied with the value of the corresponding objective function and does not want to search for solutions with a bigger value on those objective functions. For simplicity, the method as presented will use the same value of γ for all objective functions.

Step 4 – Dividing the region $S(g^h)$

$$I = \left\{ i : z_i^i \geq z_i^* + \gamma\left(g^h\right) \right\}$$

$$S\left((g+1)^i\right) = S\left(g^h\right) \cap \left\{ x \in R^n : z_i \geq z_i^* + \gamma\left(g^h\right) \right\};\ i \in I$$

There will be as many sub-sets of the non-dominated (feasible) region as the elements of the set I. In the case where $I = \emptyset$ then return to Step 3 to low $\gamma(g^h)$ or to stop.

In a g^h sub-set, the parameter g indicates the level of search and the h indicates which objective function is improved (relative to the last chosen solution), by the amount $\gamma(g^h)$, in the non-dominated solutions of the sub-set.

Step 5 – Return to Step 2 with $g = g + 1$ and $h \in I$.
In this method several linear programming problems, organized in a tree structure are generated during the evolution of the search. Figure 1 presents the general idea

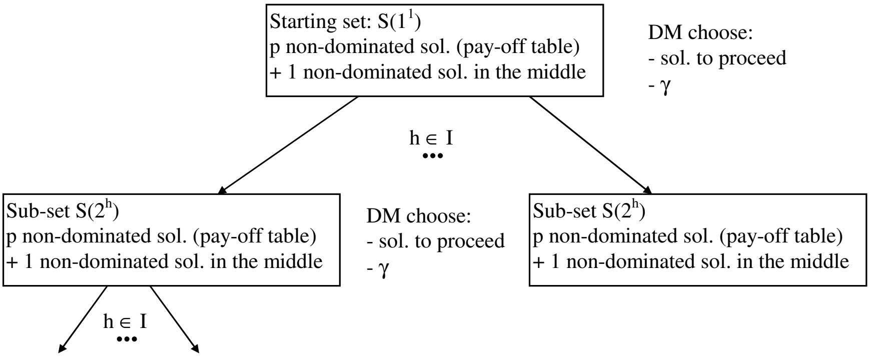
Starting set: S(1¹)
p non-dominated sol. (pay-off table)
+ 1 non-dominated sol. in the middle
DM choose:
- sol. to proceed
- γ
h ∈ I
•••
Sub-set S(2ʰ)
p non-dominated sol. (pay-off table)
+ 1 non-dominated sol. in the middle
DM choose:
- sol. to proceed
- γ
Sub-set S(2ʰ)
p non-dominated sol. (pay-off table)
+ 1 non-dominated sol. in the middle
h ∈ I
•••

Fig. 1. Representing the tree search

of this tree search. The whole idea is simple, it results in a fast and very intuitive approach to exploring the non-dominated set of solutions in MOLFP and eventually to finding the DM's preferred solution.

3.1 Convergence of the method

First, we will formalise the preferences of the DM according to the weighted sum of the objective function approach. Then we will establish a set of rules to be followed by the DM while interacting with the algorithm. After that, we will prove that the non-dominated solution computed by the algorithm running under those DM's rules is (one of) the achievable solution(s) that maximizes the preferences of the DM. This proves the convergence of the method. Nevertheless, in order to show that the interactive method can also be applied under the reference point approach, we will also formalise the preferences of the DM according to that approach, adjust the set of rules to be followed by the DM and again supply the proof of convergence of the method.

We must make it clear that we do not think this is the 'right way' of using the algorithm. Moreover, we acknowledge that this kind of usage is contradictory to the design of the method, the aim of which is not only to provide a solution in keeping with the DM's preferences, but to support the DM in building and adjusting his/her preferences while searching the non-dominated region.

3.1.1 Objective function weighted sum

Consider a DM with the following pre-defined value function, explicitly given as the weighted sum of the objective functions.

$$\begin{aligned}&\max\left\{\lambda_1\frac{c^1x+\alpha_1}{d^1x+\beta_1}+\cdots+\lambda_p\frac{c^px+\alpha_p}{d^px+\beta_p}\right\}\\&s.t.:\quad x\in S\end{aligned}$$

where $\lambda_p \in R^p$ and $\sum_{k=1}^{p}\lambda_k = 1$ and $\lambda_k > 0, k = 1, \ldots, p$.

The set of rules to be followed by the DM in order to control the algorithm (Step 3) is:

(i) Always choose the solution from the 'middle' of the non-dominated sub-region under consideration;

(ii) Always set $\gamma = 0$;

(iii) Ignore (that is, do not sub-divide) the sub-regions $s(g^h)$ where the weighted sum of their ideal point, z^{IP} , is lower than the weighted sum of at least one of the non-dominated solutions computed so far (the ideal point is the point of the objective function space whose co-ordinates are equal to the maximum that can be achieved by each objective function in the feasible region):

$$\exists z_k : \sum_{k=1}^{p}\lambda_k z_k^{IP} < \sum_{k=1}^{p}\lambda_k z_k$$

(iv) When the objective function ranges of the pay-off table of all the active sub-regions are lower than a pre-specified error value, stop the algorithm and choose the computed non-dominated solution that maximizes the weighted-sum of the objective functions.

Property 1 Condition to ignore a sub-region – rule (iii).

Assume that z^{IP} is the ideal point of the sub-region A of S and that z^1 is a solution of the sub-region B of S. If $\sum_{k=1}^{p} \lambda_k z_k^{IP} < \sum_{k=1}^{p} \lambda_k z_k^1$ holds, then the non-dominated solution, z^*, that maximizes $\sum_{k=1}^{p} \lambda_k z_k(x), x \in S$, does not belong to the sub-region A.

Proof. Assume that z^* belongs to the sub-region A and that

$$\sum_{k=1}^{p} \lambda_k z_k^* \geq \sum_{k=1}^{p} \lambda_k z_k^1 > \sum_{k=1}^{p} \lambda_k z_k^{IP} . \quad \text{Then } \sum_{k=1}^{p} \lambda_k \left(z_k^* - z_k^{IP}\right) > 0 .$$

The last expression means that there is at least one k' such that $z_{k'}^* > z_{k'}^{IP}$, and so z^{IP} would not be the ideal point of the sub-region A. □

Property 2 Sub-division of a sub-region – rule (ii).

Assume that z^1 is a non-dominated solution of S and that Z is the non-dominated solution set in the objective function space, $Z = \{z : z(x)$ is non-dominated and $x \in S\}$. So, by definition, $z^1 \in Z$.

If $Z^k = \{z : z(x)$ is non-dominated, $x \in S$ and $z_k(x) \geq z_k^1\}$ then $Z = \cup_{k=1}^{p} Z^k$.

Proof. Assume that z^2 belongs to Z but does not belong to any of the Z^k, $k = 1,\ldots,p$. Thus $z_k^2 < z_k^1$, for all $k = 1,\ldots,p$, or else z^2 would belong to at least one of the $Z^k, k = 1,\ldots,p$. If $z_k^2 < z_k^1$ for all $k = 1,\ldots,p$, then z^2 is dominated by z^1 and cannot belong to Z. □

Property 3 Conclusion.

If the DM follows the above set of rules when interacting with the algorithm, the method will compute a non-dominated solution that maximizes the preferences of the DM.

Proof. This property follows directly from the other two: the only non-dominated solutions that are ignored are the ones that belong to the sub-regions, which can be proven not to contain the solution that maximizes the preferences of the DM.

3.1.2 Reference points

Consider a DM that, now, wants to 'come close' to the reference point $q \in R^p$ in the sense of the following achievement scalarizing function:

$$s(q, z) = \max_{k=1,\ldots,p} (q_k - z_k) - \varepsilon \sum_{k=1}^{p} z_k .$$

The parameter $\varepsilon \in R^+$ is used to prevent the result from being a weakly non-dominated solution and it must be small enough (to allow for the possibility of determining all the interesting non-dominated solutions). Reference point (Wierzbicki,

1980) methods and techniques can be considered as 'generalized goal programming'. Their essence is that a reference point is a goal, but the sense of 'coming close' to it does not mean the minimization of a distance, but the minimization of an achievement scalarising function. The characteristics of this function enable it to provide non-dominated solutions, even if the goals are fixed below them (that is, 'better' solutions than the settled goals if they are feasible).

Let us adjust the rules (iii) and (iv) presented in the last sub-section which are to be followed by the DM in order to control the algorithm (Step 3):

(iii) Ignore (that is, do not sub-divide) the sub-regions $s(g^h)$ for which the following condition holds:

$$\left[\max_{k=1,\ldots,p}\left(q_k - z_k^{IP}\right) - \varepsilon\sum_{k=1}^{p} z_k^{IP}\right] > \left[\max_{k=1,\ldots,p}\left(q_k - z_k^{1}\right) - \varepsilon\sum_{k=1}^{p} z_k^{1}\right]$$

where, z^{IP} , is the ideal point of the sub-region to be ignored and z^1 is one of the non-dominated solutions computed so far.

(iv) When the objective function ranges of the pay-off table of all the active sub-regions are lower than a pre-specified error value, stop the algorithm and choose the computed non-dominated solution that minimizes the achievement scalarizing function.

Property 4 Condition for ignoring a sub-region – rule (iii).

Consider that z^{IP} is the ideal point of region A, from S, and z^1 is one solution of region B from S. If the following condition holds, then the non-dominated solution, z^*, minimizing $s(q,z), x \in S$, does not belong to region A.

$$\left[\max_{k=1,\ldots,p}\left(q_k - z_k^{IP}\right) - \varepsilon\sum_{k=1}^{p} z_k^{IP}\right] > \left[\max_{k=1,\ldots,p}\left(q_k - z_k^{1}\right) - \varepsilon\sum_{k=1}^{p} z_k^{1}\right]$$

Proof. Consider that z^* belongs to A, then if z^* minimizes $s(q,z)$ over S:

$$\max_{k=1,\ldots,p}\left(q_k - z_k^{*}\right) - \varepsilon\sum_{k=1}^{p} z_k^{*} \le \max_{k=1,\ldots,p}\left(q_k - z_k^{1}\right) - \varepsilon\sum_{k=1}^{p} z_k^{1}$$

$$< \max_{k=1,\ldots,p}\left(q_k - z_k^{IP}\right) - \varepsilon\sum_{k=1}^{p} z_k^{IP}$$

That is:

$$\max_{k=1,\ldots,p}\left(q_k - z_k^{*}\right) - \varepsilon\sum_{k=1}^{p} z_k^{*} < \max_{k=1,\ldots,p}\left(q_k - z_k^{IP}\right) - \varepsilon\sum_{k=1}^{p} z_k^{IP}$$

$$\max_{k=1,\ldots,p}\left(q_k - z_k^{*}\right) - \max_{k=1,\ldots,p}\left(q_k - z_k^{IP}\right) + \varepsilon\sum_{k=1}^{p}\left(z_k^{IP} - z_k^{*}\right) < 0$$

Consider i, without loss of generality, such that $q_i - z_i^* \ge q_k - z_k^*\ ,\ k = 1,\ldots,p$ and remember that $z_k^{IP} \ge z_k^*;\ k = 1,\ldots,p$, because z^{IP} is the ideal point of A. Then $q_i - z_i^* \ge q_k - z_k^{IP}\ ,\ k = 1,\ldots,p$ and ${}_{k=1,\ldots,p}(q_k - z_k^*) \ge \max_{k=1,\ldots,p}(q_k - z_k^{IP})$. Concluding: $\max_{k=1,\ldots,p}(q_k - z_k^*) -_{k=1,\ldots,p} \left(q_k - z_k^{IP}\right) \ge 0$.

Due to this last equation, it is necessary that $\varepsilon \sum_{k=1}^{p} (z_k^{IP} - z_k^*) < 0$. This implies that there is at least one k, lets say k', for which $z_{k'}^* > z_{k'}^{IP}$ and so z^{IP} could not be the ideal point of A. □

Properties 2 and 3 can also be applied in this sub-section.

4 Illustrative example

In this section we will illustrate the interactive method presented in the previous section. We will use the example given by Kornbluth and Steuer (1981a):

$$\begin{aligned}
&\max\left\{z_1 = \frac{x_1 - 4}{-x_2 + 3}\right\}\\
&\max\left\{z_2 = \frac{-x_1 + 4}{x_2 + 1}\right\}\\
&\max\{z_3 = -x_1 + x_2\}\\
&\text{s.t.}:\\
&\qquad -x_1 + 4x_2 \leq 0\\
&\qquad x_1 - {}^1\!/_2 x_2 \leq 4\\
&\qquad x_1, x_2 \geq 0
\end{aligned}$$

S is the feasible region defined by the constraints of the problem. Figure 2 presents the feasible region – the triangle defined by the points A, B and C – and the non-dominated region – the bold (dashed) points – of the problem. The points B and D are weakly non-dominated solutions.

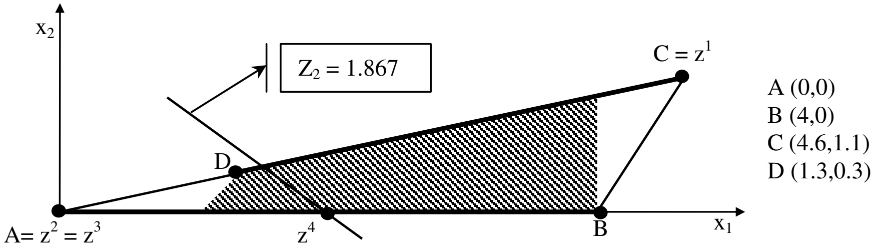

Fig. 2. Feasible and non-dominated region for the problem with $S(1^1)$

Step 1 – Starting.
$g = 1; h = 1.$

Step 2.1 – Computing the pay-off table for $S(1^1) = S$.
To compute

$$\begin{aligned}
&\max z_1\\
&\text{s.t.}: \quad x \in S(1^1),
\end{aligned}$$

Table 1. Four non-dominated solutions characterizing the problem $S(1^1)$

		z_1	z_2	z_3
Pay-off	z^1	0.308	−0.267	−3.429
	z^2	−1.333	4	0
	z^3	−1.333	4	0
Middle	z^4	−0.622	1.867	−2.133

we solve the problem:

$$\begin{aligned}&\max\{y_1 - 4t\}\\ &\text{s.t.}:\ -y_1 + 4y_2 \leq 0\\ &\qquad y_1 - {}^1\!/\!_2 y_2 - 4t \leq 0\\ &\qquad -y_2 + 3t = 1\\ &\qquad y_1, y_2, t \geq 0\end{aligned}$$

where $t = \frac{1}{-x_2+3}$; $y_1 = x_1 t$ and $y_2 = x_2 t$.

There are no alternative optima and so the optimum of the problem is a non-dominated solution, z^1: $y_1 = 2.462$; $y_2 = 0.615$; $t = 0.538$; $x_1 = 4.6$; $x_2 = 1.1$; $z_1 = 0.308$; $z_2 = -0.267$ and $z_3 = -3.429$.

Repeating the same process over the other objective functions we compute the pay-off table for $S(1^1)$ presented in Table 1. These solutions are also presented in Figure 2.

Step 2.2 – Computing the mid-point solution.

$$\Delta z_1 = 0.308 - (-1.333) = 1.638; \quad \Delta z_2 = 4.267; \quad \Delta z_3 = 3.429$$

Hence, r equals 2. v will be equal to 1 because the minimum of the z_2 column of Table 1 is (-0.267).

We will solve the problem:

$$\begin{aligned}&\max z_1\\ &\text{s.t.}:\quad x \in S(1^1); z_2 \geq 1.867.\end{aligned}$$

The non-dominated solution resulting from this problem, z^4, is given in Table 1 and is the mid-point solution of the non-dominated region of $S(1^1)$. The auxiliary constraint $(z_2 \geq 1.867)$ is depicted in Figure 2.

Step 3 – Choosing the solution to proceed.
The DM chooses to continue the search through z^4 (Table 1) and sets $\gamma = 0.2$. Note that it is possible to set different values for γ according to the amount of improvement that the DM wants for each objective function. In this case, $\gamma \in R^3$, $\gamma_k \geq 0, k = 1, 2, 3$. Some of the γ can be set to zero, meaning that the DM is satisfied with the value of the corresponding objective function and does not want to search for solutions with a bigger value in those objective functions. For simplicity,

Table 2. Four non-dominated solutions characterizing the problem $S(2^1)$

		z_1	z_2	z_3
Pay-off	z^1	0.308	−0.267	−3.429
	z^2	−0.422	1.266	−2.734
	z^3	−0.422	0.535	−2.292
Middle	z^4	−0.167	0.5	−3.5

the method as presented used the same value of γ for all objective functions and the illustrative example will follow the same rule.

This first choice of the DM is the middle solution from the non-dominated region of the MOLFP problem. This means that the DM does not want a solution near the maximum of any of the objective functions but a more balanced solution. Nevertheless, s/he is not satisfied and sets $\gamma = 0.2$ in order to explore other solutions that are greater than z^4 (Table 1) by the amount of at least 0.2 in one of the objective functions. Therefore the DM is excluding from further searches the part of the non-dominated region that surrounds z^4 and that is defined by γ.

Step 4 – Dividing the region $S(1^1)$.

$I = \{1, 2, 3\}$. We will have 3 sub-sets:
$S(2^1) = S(1^1) \cap \{x \in R^n | z_1 \geq -0.422\}$
$S(2^2) = S(1^1) \cap \{x \in R^n | z_2 \geq 2.067\}$
$S(2^3) = S(1^1) \cap \{x \in R^n | z_3 \geq -1.933\}$

Figure 3 presents the constraints defining the sub-sets. Note that the parameter $\gamma = 0.2$ sets the sub-region that will not be searched any more.

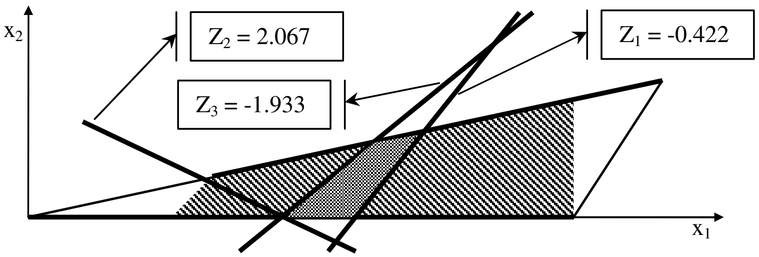

Fig. 3. The constraints defining the sub-sets $S(2^1)$, $S(2^2)$ and $S(2^3)$

Step 5 – We return to Step 2.

Step 2 – $S(2^1); S(2^2); S(2^3)$.
The non-dominated solutions, for $S(2^1)$, computed at this step are presented in Table 2.

Table 3. Four non-dominated solutions characterizing the problem $S(2^2)$

		z_1	z_2	z_3
Pay-off	z^1	−0.689	2.067	−1.933
	z^2	−1.333	4	0
	z^3	−1.333	4	0
Middle	z^4	−1.012	3.035	−0.965

Table 4. Four non-dominated solutions characterizing the problem $S(2^3)$

		z_1	z_2	z_3
Pay-off	z^1	−0.604	0.865	−1.933
	z^2	−1.333	4	0
	z^3	−1.333	4	0
Middle	z^4	−0.812	2.435	−1.565

Note that for computing the solution z^4 we solve the following program:

$$\begin{aligned}
&\max\{y_1 - 4t\}\\
&\text{s.t.}:\ -y_1 + 4y_2 \le 0\\
&\qquad y_1 - {}^1\!/\!_2 y_2 - 4t \le 0\\
&\qquad -y_2 + 3t = 1\\
&\qquad y_1 - 0.422y_2 - 2.734t \ge 0 \quad (Partition\ const.)\\
&\qquad y_1 + 0.5y_2 - 3.5t \le 0 \quad (Middle\ const.)\\
&\qquad y_1, y_2, t \ge 0
\end{aligned}$$

The non-dominated solutions, for $S(2^2)$ and $S(2^3)$, computed at this step, are presented in Tables 3 and 4, respectively.

Step 3 – Choosing the solution to proceed.
The DM chooses to continue the search through the solution z^4 of the sub-set $S(2^1)$ (Table 2) with $\gamma = 0.1$. This second choice of the DM is the middle solution of the sub-set $S(2^1)$, i.e. the sub-set where the non-dominated solutions have the first objective function greater than the last choice of z^* by the amount of at least 0.2.

Note that the method formally presented in Section 3 determines the computation of the characteristic solutions of all the sub-sets resulting $S(2^1)$, $S(2^2)$ and $S(2^3)$. It is clear that another possibility would be to ask the DM, in Step 3, which objective function s/he would like to improve (by at least the amount of γ) and to generate only the corresponding sub-set in Step 4. This modification of the method would result in a version with fewer computations and fewer non-dominated solutions to present to the DM, but with more questions to ask to the DM. The version presented in Section 3 may be better for a more exploratory (almost free) search while the alternative version seems to be better suited to more targeted searches where a fast convergence to the preferred non-dominated solution is desirable. A computer implementation of the method should offer both versions.

By choosing the middle solution of $S(2^1) - z^4$ of Table 2 – the DM improves the first objective function by more than the amount of the parameter γ. That is $\gamma(1^1) = 0.2$ was fixed at a conservative value. It is clear that the improvement in the first objective function was achieved by decreasing the other objective functions. The DM is not yet satisfied and wants to explore beyond the neighbourhood of z^4 from $S(2^1)$, defined by $\gamma(2^1)$. S/he fixes $\gamma(2^1)$ to the value of 0.1.

Step 4 – Dividing the region $S(2^1)$.

$I = \{1, 2, 3\}$
$S(3^1) = S(2^1) \cap \{x \in R^n | z_1 \geq -0.067\}$
$S(3^2) = S(2^1) \cap \{x \in R^n | z_2 \geq 0.6\}$
$S(3^3) = S(2^1) \cap \{x \in R^n | z_3 \geq -3.4\}$

Figure 4 presents the new constraints defining the sub-sets. In this figure we only depict the sub-set $S(2^1)$.

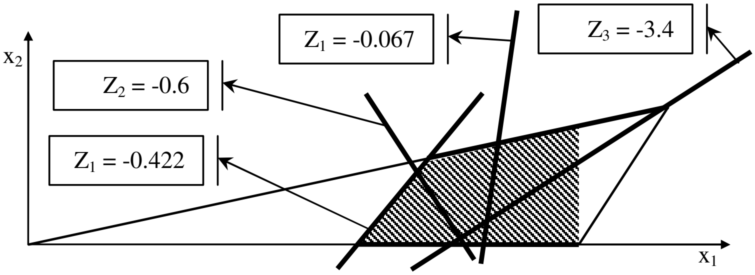

Fig. 4. The constraints defining the sub-sets $S(3^1)$, $S(3^2)$ and $S(3^3)$

Step 5 – We return to Step 2.

Step 2 – $S(3^1)$; $S(3^2)$; $S(3^3)$.
The non-dominated solutions, for $S(3^1)$, $S(3^2)$ and $S(3^3)$, computed at this step are presented in Tables 5, 6 and 7, respectively.

Step 3 – Choosing the solution to proceed.
The DM chooses to continue the search through the solution z^1 from the sub-set $S(3^2)$ (Table 6) with $\gamma = 0.2$.

The third choice of the DM is the non-dominated solution that maximizes the first objective function in sub-set $S(3^2)$. So, the DM chooses to improve his/her last chosen solution in the second objective function – this is the meaning of choosing $S(3^2)$ – by the amount of $\gamma(2^1) = 0.1$, giving up as little as possible of the first objective function – this is the meaning of choosing z^1. It seems that the DM feels that s/he did lost too much in objective function 2 with his/her choice in the previous

Table 5. Four non-dominated solutions characterizing the problem $S(3^1)$

		z_1	z_2	z_3
Pay-off	z^1	0.308	−0.267	−3.429
	z^2	−0.067	0.201	−3.799
	z^3	−0.067	0.069	−2.898
Middle	z^4	−0.067	0.114	−3.348

Table 6. Four non-dominated solutions characterizing the problem $S(3^2)$

		z_1	z_2	z_3
Pay-off	z^1	−0.2	0.6	−3.4
	z^2	−0.422	1.266	−2.734
	z^3	−0.422	0.6	−2.354
Middle	z^4	−0.298	0.6	−2.878

Table 7. Four non-dominated solutions characterizing the problem $S(3^3)$

		z_1	z_2	z_3
Pay-off	z^1	0.286	−0.25	−3.4
	z^2	−0.422	1.266	−2.734
	z^3	−0.422	0.535	−2.292
Middle	z^4	−0.183	0.508	−3.4

interaction. This choice corresponds to a tuning of the non-dominated solution according to the DM's preferences. Nevertheless, the DM remains dissatisfied with the solution z^1 from the sub-set $S(3^2)$ (Table 6) and wants to continue the search by exploring beyond its neighbourhood, defined by $\gamma(3^2) = 0.2$.

Step 4 – Dividing the region $S(3^2)$.

$$I = \{2, 3\}; \quad S(4^2) = S(3^2)\cap\}x \in R^n|z_2 \geq 0.8\};$$

$$S(4^3) = S(3^2) \cap \{x \in R^n|z_3 \geq -3.2\}$$

Figure 5 presents the new constraints defining the sub-sets. In this figure we only depict the sub-set $S(3^2)$.

Step 5 – We return to Step 2.

Step 2 – $S(4^2); S(4^3)$.
The non-dominated solutions, for $S(4^2)$ and $S(4^3)$, computed at this step are presented in Tables 8 and 9, respectively.

Step 3 – Choosing the solution to proceed.

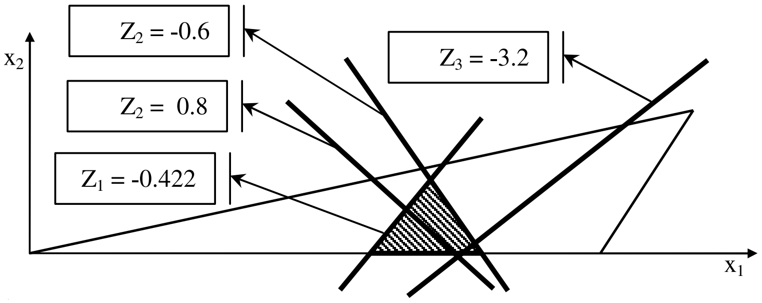

Fig. 5. The constraints defining the sub-sets $S(4^2)$ and $S(4^3)$

Table 8. Four non-dominated solutions characterizing the problem $S(4^2)$

		z_1	z_2	z_3
Pay-off	z^1	-0.267	0.8	-3.2
	z^2	-0.422	1.266	-2.734
	z^3	-0.422	0.8	-2.514
Middle	z^4	-0.339	0.6	-2.857

Table 9. Four non-dominated solutions characterizing the problem $S(4^3)$

		z_1	z_2	z_3
Pay-off	z^1	-0.235	0.6	-3.2
	z^2	-0.422	1.266	-2.734
	z^3	-0.422	0.6	-2.354
Middle	z^4	-0.341	0.6	-2.678

The DM chooses the solution z^1 from the sub-set $S(4^3)$ as the final solution, and the algorithm stops. The fourth and last choice of the DM is the non-dominated solution that maximizes the first objective function over the sub-set $S(4^3)$. So the DM now chooses to improve the non-dominated solution in objective function 3 by exactly the amount of $\gamma(3^2) = 0.2$ (see Table 5), losing as little as possible in the first objective function. The DM is once more tuning the non-dominated solution, like s/he did in the last interaction, but in another objective function. The DM is now satisfied with the non-dominated solution and the method stops.

Table 10 presents the solutions that the DM did choose on the route to finding his/her preferred solution. This illustrative example shows that the method proposed in this paper is very easy and intuitive to use, and allows both a fast convergence to the preferred non-dominated solution and a sensitive tuning at the end of the search. Furthermore, the computations that it requires are very easy and straightforward (linear programming problems), especially if we use some post-optimal techniques

Table 10. The solutions that the DM did choose during the search for his/her preferred solution

g^h	z	z_1	z_2	z_3	γ
1^1	z^4	−0.622	1.867	−2.133	0.2
2^1	z^4	−0.167	0.5	−3.5	0.1
3^2	z^1	−0.2	0.6	−3.4	0.2
4^3	z^1	−0.235	0.6	−3.2	Stop

(not shown in this example) for introducing a new constraint in the simplex tableau when moving from a level of the search tree (an interaction) to the next one (next interaction). A free learning search of the non-dominated region of a MOLFP is also very easy to perform with the help of this method. The DM just needs to continue the search thorough several sub-sets of the non-dominated region at the same level of search, building a wider search tree.

The possible incoherence of the choices of the DM can also be dealt with by going back in the search tree to a node before the mistake was made and by proceeding with the search through other non-dominated solutions, or the same one again, but with another value for the parameter γ.

5 Conclusions

This paper has presented a new interactive method for computing the preferred non-dominated solution in multiple objective linear fractional programming (MOLFP). We have two sequential phases performed repeatedly: i) the dialogue with the DM, where s/he chooses the non-dominated solution that better fits her/his preferences and the amount of improvement s/he wants, and ii) the computation of other non-dominated solutions, using the newly-elicited information. The basic idea of the computation phase of the algorithm is to optimize one of the fractional objective functions while constraining the others. Several linear programming problems, organized in a tree structure, are generated as the search evolves.

The method proposed in this paper is very easy and intuitive to use and allows for both a fast convergence to the preferred non-dominated solution and a sensitive tuning at the end of the search. Moreover the computations that it requires are quite straightforward and simple (linear programming problems), especially if we use some post-optimal techniques for introducing a new constraint in the simplex tableau when moving from a level of the search tree (an interaction) to the next one (next interaction). A free learning search of the non-dominated region of a MOLFP can also be very easily performed with the help of this method. The DM just needs to continue the search through several sub-sets of the non-dominated region at the same level of search, building a wider search tree.

The whole idea is simple and it results in a fast and very intuitive approach to exploring the non-dominated set of solutions in MOLFP, and eventually to find the preferred solution. We have also proved the convergence of the method.

References

Benson HP (1985) Finding certain weakly-efficient vertices in multiple objective linear fractional programming. Management Science 31(2): 240–245

Charnes A, Cooper W (1962) Programming with linear fractional functionals. Naval Research Logistics Quarterly 9: 181–186

Costa JP, Lourenço J (2001a) Computing the weighted sum in multiple objective linear fractional programming. Proceedings of the 6th International Conference of the Decision Science Institute, (CD-ROM) 8–11 July, 2001, Chihuahua, Mexico

Costa JP, Lourenço J (2001b) Cálculo da soma pesada em programação linear fraccionária multiobjectivo (In Portuguese). Proceedings of SPOLM 2001 – IV Simpósio de Pesquisa Operacional da marinha, V Simpósio da Logística da Marinha, CASNAV – Centro de Análise de Sistemas Navais, 18–19 December, 2001, Rio de Janeiro, Bazil, pp 564–576

Craven BD (1988) Fractional programming. Heldermann, Berlin

Ferland JA, Potvin J-Y (1985) Generalized fractional programming: algorithms and numerical experimentation. European Journal of Operational Research 20: 92–101

Goedhart M, Spronk J (1995) Financial planning with fractional goals. European Journal of Operational Research 82: 111–124

Konno H, Yamashita H (1999) Minimizing sums and products of linear fractional functions over a polytope. Naval Research Logistics 46: 583–596

Kornbluth J, Steuer R (1981a) Multiple objective linear fractional programming. Management Science 27(9): 1024–1039

Kornbluth J, Steuer R (1981b) Goal programming with linear fractional criteria. European Journal of Operational Research 8: 58–65

Metev B, Gueorguieva D (2000) A simple method for obtaining weakly efficient points in multiobjective linear fractional programming problems. European Journal of Operational Research 126: 386–390

Nakayama H (1991) Satisficing trade-off method for problems with multiple linear fractional objectives and its applications. In: Lewandowski A, Volkovich V (eds) Multiobjective problems of mathematical programming. Proceedings of the International Conference held in Ialta, October 26 – November 2, 1988, Lecture Notes in Economics and Mathematical Systems, vol 351, pp 42–50. Springer, Berlin Heidelberg New York

Nykowski I, Zolkiewski Z (1985) A compromise procedure for the multiple objective linear fractional programming problem. European Journal of Operational Research 19: 91–97

Schaible S (1981) Fractional programming: applications and algorithms. European Journal of Operational Research 7: 111–120

Schaible S, Ibaraki T (1983) Fractional programming. European Journal of Operational Research 12: 325–338

Stancu-Minasian IM (1997) Fractional programming: theory, methods and applications. Kluwer, Dordrecht

Stancu-Minasian IM (1999) A fifth bibliography of fractional programming. Optimization 45: 343–367

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

Wierzbicki A (1980) The use of reference points in multiobjective optimization. In: Fandel G, Gal T (eds) Multiple criteria decision making, theory and application, pp 468–486. Springer, Berlin Heidelberg New York