

# Computing non-dominated solutions in MOLFP

João Paulo Costa \*

*Faculdade de Economia da Universidade de Coimbra and INESC Coimbra, Av. Dias da Silva 165, 3004-512 Coimbra, Portugal*

Received 1 December 2004; accepted 1 November 2005

Available online 15 May 2006

---

## Abstract

In this paper we present a technique to compute the maximum of a weighted sum of the objective functions in multiple objective linear fractional programming (MOLFP). The basic idea of the technique is to divide (by the approximate ‘middle’) the non-dominated region in two sub-regions and to analyze each of them in order to discard one if it can be proved that the maximum of the weighted sum is in the other. The process is repeated with the remaining region. The process will end when the remaining regions are so little that the differences among their non-dominated solutions are lower than a pre-defined error. Through the discarded regions it is possible to extract conditions that establish weight indifference regions. These conditions define the variation range of the weights that necessarily leads to the same non-dominated solution. An example, illustrating the concept, is presented. Some computational results indicating the performance of the technique are also presented.

© 2006 Elsevier B.V. All rights reserved.

*Keywords:* Multiobjective linear fractional programming; Sum of linear ratios; Weight indifference regions

---

## 1. Introduction

The computation required by a multiobjective linear programming (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). The weighted sum of the functions is probably the widest multicriteria approach used in practice to aggregate the objective functions according to the preferences of the decision maker (DM). This aggregation leads to a fractional function where the linear numerator

and denominator of each objective function turns out to be (in the general case) polynomials, the degree of which equals the number of objective functions. Thus, this transformation of the MOLFP in to a single criterion problem leads to a very difficult problem to solve by the nowadays existing techniques. Schaible and Shi (2003) consider it one of the most difficult fractional problems encountered so far – it is much more removed from convex programming than other multiratio problems. Schaible and Shi (2003) provide a survey of applications and various algorithmic approaches for this problem.

In this paper we present a new technique (a new and faster version of the algorithm presented in Costa and Lourenço, 2001) for computing the maximum of the weighted sum of the linear fractional

---

\* Tel.: +351 239 790 586.

E-mail address: [jpaulo@fe.uc.pt](mailto:jpaulo@fe.uc.pt)

objective functions. This means to compute the non-dominated solution of the MOLFP problem associated with a given weight vector for the objective functions. The basic idea of the technique is to divide (by the approximate ‘middle’) the non-dominated region in two sub-regions and to analyze each of them in order to discard one if it can be proved that the maximum of the weighted sum is in the other. The process is repeated with the remaining region. It is not always possible to discard one of the regions and so the process must be repeated for both, building a search tree. In most problems it is only after a certain level of the search tree that we can start to discard regions. The process will end when the remaining regions are so little that the differences among their non-dominated solutions are lower than a pre-defined error.

One region can be discarded when the value of the weighted sum of its ideal point is worst than the value of the weighted sum of a non-dominated solution belonging to another region not yet discarded. This condition can be used for computing the weight indifference regions. Considering all the discarded regions we build a set of constraints defining the weight variation ranges that necessarily leads to the same non-dominated solution. It is possible that weight vectors not fulfilling the set of constraints also lead to the same non-dominated solution, but at least we will have information about part of the indifference region. This information is very useful in multiobjective, because in most applications the decision makers (DM’s) will not be satisfied with just one run of the technique. They will want to know what happens if they change the weight vector. Knowing part of the weight indifference region will ease the computation burden.

The needed amount of calculus of the technique is the corresponding to the computation of the ideal point for each region that is created. To compute the ideal point of one region we need to solve a linear programming problem for each objective function. In this paper we present some computational results for several performance tests. We can conclude that the technique performs very well when compared with the nowadays existing techniques. Kuno (2002) seems to be the existing technique that proved to have the highest performance. Nevertheless, it is very difficult to compare techniques, because several of the existing ones attempt also to solve other kinds of problems. Schaible and Shi (2003) present a good survey about the techniques and try to compare them. We would like to empha-

size the work of Falk and Palocsay (1992), Konno and Yamashita (1999), Konno and Abe (1999), Konno and Fukaishi (2000), Freund and Jarre (2001), Kuno (2002), Phuong and Tuy (2003), Dai et al. (2005) because they constitute chronologic marks of the work in the field. Nevertheless, the break through of the technique presented in this paper was achieved by the explicit consideration of a sum of linear ratios as a multiobjective problem: probably there is much to gain in the design of new hybrid techniques incorporating concepts from several fields.

Kornbluth and Steuer (1981a) can be considered the seminal work in MOLFP. In their paper they present its main characteristics, difficulties and differences from MOLP. They also present one method to compute all the weakly non-dominated solutions of a MOLFP problem. This method is complex due to the necessity of knowing the ‘break point’ of the ‘broken edges’, which is the point where a feasible edge changes from dominated to non-dominated or vice versa. In another work of the same authors (Kornbluth and Steuer, 1981b) they extend goal programming in order to include linear fractional criteria, resulting in the core mechanism of an interactive MOLFP method. Costa (2005) attempts to make a small survey of the existing interactive MOLFP methods while presenting a new interactive method. 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. He also provides a good introduction to MOLFP. Stancu-Minasian (1997, Chapter 6) extends these concepts.

The paper will proceed as follows. Firstly, we present the used notation and formulation of the problems to be solved. Secondly we give an outline of the new computation technique. After that, we formally define the technique. In Section 5, we present the results supporting the technique and the building process of the weight indifference region. Section 6 illustrates the concepts with a small example and Section 7 presents some computational results. Finally, in the last section, we present some conclusions.

## 2. Notation

In this paper we formulate the multiple objective linear fractional programming (MOLFP) problems in the following way:

$$\begin{aligned} & \max \left\{ z_1 = \frac{c^1 x + z_1}{d^1 x + \beta_1} \right\} \\ & \dots \\ & \max \left\{ z_p = \frac{c^p x + z_p}{d^p x + \beta_p} \right\} \\ & \text{s.t. } x \in S = \{x \in R^n | Ax \leq b, x \geq 0, b \in R^m\}, \end{aligned}$$

where  $c^k, d^k \in R^n, A \in R^{m \times n}$  and  $\alpha_k, \beta_k \in R, k = 1, \dots, p$  and  $\forall_k, x \in S: d^k x + \beta_k > 0$ .

We will differentiate between 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  $z_k(x) > z_k(x')$ , for all  $k = 1, \dots, p$  – and 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  $k = 1, \dots, p$ , and  $z_k(x) > z_k(x')$  for at least one  $k$ . We will use the some denomination – non-dominated – in both the decision and objective spaces. That is, the image,  $z'$ , in the objective space of the non-dominated solution  $x' \in S$  will also be called ‘non-dominated’.

The weighted sum of the objective functions can be formulated as

$$\begin{aligned} & \max \left\{ \lambda_1 \frac{c^1 x + z_1}{d^1 x + \beta_1} + \dots + \lambda_p \frac{c^p x + z_p}{d^p x + \beta_p} \right\} \\ & \text{s.t. } x \in S, \end{aligned}$$

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 normalize the weights and  $\lambda_k > 0, k = 1, \dots, p$ , in order to prevent the result from being a weakly non-dominated solution. The maximum of a weighted sum of linear fractional functions is a non-dominated solution (if all the weights are strictly bigger than zero), but in general not all of the non-dominated solutions can be computed using a weighted sum of the linear fractional functions. That is because in MOLFP the non-dominated region can be non-convex.

The ideal point,  $z^*$ , is the point of the objective functions’ space whose coordinates are equal to the maximum that can be achieved separately by each objective function in the feasible region.  $z^*$  is computed through the determination of the pay-off table, that is, computing  $z^k = z(x^{*k}); k = 1, \dots, p$ ; where  $x^{*k}$  is non-dominated and optimizes the program:

$$\begin{aligned} & \max z_k(x) \\ & \text{s.t. } x \in S. \end{aligned}$$

There is a variable change technique (Charnes and Cooper, 1962) that turns a linear fractional problem

into a plain linear program. Consider the following single criterion problem:

$$\begin{aligned} & \max \left\{ z = \frac{cx+z}{dx+\beta} \right\} \\ & \text{s.t. } x \in S = \{x \in R^n | Ax = b, x \geq 0, b \in R^m\}, \end{aligned}$$

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

We define the new variables:

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

Making the variables substitution we arrive to the following linear program:

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

This variable change is extensively used by the technique in order to compute the maximum of each objective function.

### 3. Outline of the technique

The technique is a new and faster version of the technique presented in Costa and Lourenço (2001) for computing the maximum of the weighted sum of linear fractional objective functions. This means to compute the non-dominated solution of the MOLFP problem associated with a given weight vector for the objective functions. The basic idea of the technique is to divide (by the approximate ‘middle’) the non-dominate region in two sub-regions and to analyze each of them in order to discard one if it can be proved that the maximum of the weighted sum is in the other. The process is repeated with the remaining region.

The division of the non-dominated region – Step 4, Section 4 – is by the approximate middle and not by exactly the middle, because in order to know the range of the region we compute the pay-off table of the problem. As one of the anonymous referees noticed, the pay-off table of a multiobjective problem with more than two objective functions does not always indicate the minimum of the criterion values over the non-dominated region. Iserman and Steuer (1987), Reeves and Reid (1988) provide a deep discussion about this issue. This particularity tends to work in favor of the technique’s performance. We tested other forms of dividing the region, namely using the weighted range of the

pay-off table and the range of the weighted sum of the solutions of the pay-off table. Besides that, we also tried to divide not by the middle (that is, 50%–50%) but using 30%–70% and other possibilities. The form presented in this paper proved to be the best one.

It is also relevant to note that when analyzing a region, that is, when computing the pay-off table corresponding to that region, it is not necessary to perform all the calculus. If we have the pay-off table of one region and if we divide it in two sub-regions we only need to compute half of the pay-off table solutions of the two new sub-regions. The solutions of the precedent pay-off table will be present in the pay-off tables of the two new sub-regions – Step 5, Section 4. This, of course, doubles the speed of the technique.

It is not always possible to discard one of the regions and so the process must be repeated for both, building a search tree. In most problems it is only after a certain level of the search tree that we can start to discard regions. The process will end when the remaining regions are so little that the differences among their non-dominated solutions are lower than a pre-defined error. One region can be discarded when the value of the weighted sum of its ideal point is worst than the value of the weighted sum of a non-dominated solution belonging to another region not yet discarded – Corollary 1 of Section 5. In order to speed up the computations, an incumbent non-dominated solution is maintained: the solution that has the best weighted sum computed so far. This incumbent solution is used to make the necessary comparisons with the ideal point of the regions that are potentially to discard.

Having a tree to search a criterion to choose the next region to divide is necessary. The technique chooses the one having the best ideal point – Step 3, Section 4. Other criteria were tested: the region having the lower index and the region having the best weighted solution. The criterion presented in this paper proved to be the best.

When there are no more regions to divide, because the remains have the range, of their pay-off table objective functions, lower than the error the technique stops.

**4. The technique**

*Step 1 – Initializing*

$\varepsilon$  is the pre-defined error.

$Q = \emptyset$  is the set of region indexes the variation range of which is lower than the pre-defined error.

$n = 0$  is the index of the current region.

$M$  is the index set of regions that can be further sub-divided.

$M = \{n\}$  is the initial region index set.

$g = 0$  is the region counter.

$z^I = (-\infty, -\infty, \dots, -\infty)$  is the incumbent solution.

$n^I = n$  is the incumbent region index.

$S(n)$  is the feasible region of the  $n$ th region;

$S(0) = S$ .

$\lambda$  is the given weight vector.

*Step 2 – Analyzing the first region*

Computing the pay-off table of the  $n$ th region.

For  $k = 1, \dots, p$

compute  $z_k(x^{*k}) = \max z_k(x)$  s.t.  $x \in S(n)$ ;

and, for  $j = 1, \dots, p$

$$z_j^{kn} = z_j(x^{*k}).$$

*Note:* The solutions can be weakly non-dominated.

$z^{*n} = (z_1^{1n}, z_2^{2n}, \dots, z_p^{pn})$  is the ideal point of region  $n = 0$ .

Initializing the incumbent solution.

For  $j = 1, \dots, p$  do:

If  $\sum_{k=1}^p \lambda_k z_k^{jn} > \sum_{k=1}^p \lambda_k z_k^I$  then  $z^I \leftarrow z^{jn}$ .

*Step 3 – Choosing the next region*

If there are regions that can be further divided the algorithm chooses the one having the best ideal point to proceed, i.e., it defines the new  $n$ .

If  $M \neq \emptyset$  and  $\sum_{k=1}^p \lambda_k z_k^{kn} = \max_{m \in M} \sum_{k=1}^p \lambda_k z_k^{km}$  then

$$M \leftarrow M \setminus \{n\};$$

If not

If there is no other region to further divide it is necessary to recalculate the pay-off tables of the not discarded regions (the ones remaining in  $Q$ ), taking in to consideration that we do not want weakly non-dominated solutions.

For all  $q \in Q$  do:

Compute  $z^{kq} = z(x^{*k})$ ,  $k = 1, \dots, p$ , where  $x^{*k}$  is non-dominated and optimizes the program:

$$\max z_k(x) \text{ s.t. } x \in S(q)$$

End of For all  $q$ .

The non-dominated solution,  $\bar{z}$ , that maximizes the weighted sum of the objective functions is the one that maximizes:

$$\max_{q \in Q; j=1, \dots, p} \sum_{k=1}^p \lambda_k z_k^{jq}.$$

The algorithm stops.

End of If not.

**Step 4 – Sub-dividing a region**

The index of the objective function to constrain, denoted by  $r$ , in order to sub-divide the region  $n$ , corresponds to the one having the largest range in the pay-off table, i.e.:

$$\Delta z_r^n = \max_{k=1, \dots, p} \left\{ \Delta z_k^n = \left( z_k^{kn} - \min_{k \neq j} \{ z_k^{jn} \} \right) \right\}.$$

Creating two new regions:

$$g \leftarrow g + 1;$$

$$S(g) = S(n) \cap \{ x \in R^n | z_r(x) \geq z_r^n - \frac{1}{2} \Delta z_r^n \}.$$

$$g \leftarrow g + 1;$$

$$S(g) = S(n) \cap \{ x \in R^n | z_r(x) \leq z_r^n - \frac{1}{2} \Delta z_r^n \}.$$

**Step 5 – Analyzing the two new regions**

Computing the pay-off table of the two new regions. Note that the maximum of each objective function in the previous region must belong to one of the new regions, and so there is no need to compute it.

For  $k = 1, \dots, p$  do:

If  $z_r^{kn} \geq z_r^n - \frac{1}{2} \Delta z_r^n$  then

$$z_k^{(g-1)} = z_k^{kn},$$

$$z_k^{kg} = \max z_k(x) \quad \text{s.t. } x \in S(g).$$

Note: The solutions can be weakly efficient.

If not

$$z_k^{kg} = z_k^{kn},$$

$$z_k^{k(g-1)} = \max z_k(x) \quad \text{s.t. } x \in S(g-1).$$

Note: The solutions can be weakly efficient.

End of If

End of For  $k = 1, \dots, p$ .

If it is an interesting region (a region where it is still possible the subsistence of the searched non-dominated solution – that is, a region that fulfils the following condition) it will be further analyzed, otherwise it will just be ignored (discarded).

If  $\sum_{k=1}^p \lambda_k z_k^{*g} \geq \sum_{k=1}^p \lambda_k z_k^1$  then

Being an interesting region the incumbent solution will be tested against the solutions of the region's pay-off table.

For  $j = 1, \dots, p$  do:

If  $\sum_{k=1}^p \lambda_k z_k^{jg} > \sum_{k=1}^p \lambda_k z_k^1$  then  $z^1 \leftarrow z^{jg}$ ;  $n^1 \leftarrow g$ .  
End of For  $j = 1, \dots, p$ .

Being an interesting region it will either be further divided (if the range of one of the pay-off table objective functions is bigger than the error) or classified as a region to search for the non-dominated solution in the end.

If  $(\exists k, j \ k = 1, \dots, p; j = 1, \dots, p | z_k^{*g} - z_k^{jg} > \varepsilon)$  then

$$M \leftarrow M + \{g\}$$

If not  $Q \leftarrow Q + \{g\}$

End of If

The same as above for the region  $(g-1)$ .

If  $\sum_{k=1}^p \lambda_k z_k^{*(g-1)} \geq \sum_{k=1}^p \lambda_k z_k^1$  then

For  $j = 1, \dots, p$  do:

If  $\sum_{k=1}^p \lambda_k z_k^{j(g-1)} > \sum_{k=1}^p \lambda_k z_k^1$  then

$$z^1 \leftarrow z^{j(g-1)}$$

$$n^1 \leftarrow (g-1)$$

End of If

End of For  $j = 1, \dots, p$ .

If  $(\exists k, j \ k = 1, \dots, p; j = 1, \dots, p | z_k^{*(g-1)} - z_k^{j(g-1)} > \varepsilon)$  then

$$M \leftarrow M + \{(g-1)\}$$

If not  $Q \leftarrow Q + \{(g-1)\}$

End of If.

**Step 6 – Discarding regions**

If the region did change, the incumbent solution also changed, and so it pays to check if we can discard some more regions.

If  $(n^1 = g)$  or  $(n^1 = (g-1))$  then

For all the  $m \in M$  do

If  $\sum_{k=1}^p \lambda_k z_k^{*m} < \sum_{k=1}^p \lambda_k z_k^1$  then  $M \leftarrow M \setminus \{m\}$ ;

End of For all  $m$ .

For all the  $q \in Q$  do

If  $\sum_{k=1}^p \lambda_k z_k^{*q} < \sum_{k=1}^p \lambda_k z_k^1$  then  $Q \leftarrow Q \setminus \{q\}$ ;

End of For all  $q$ .

End of If.

**Return to Step 3.**

### 5. Convergence and indifference regions

Let us introduce the set:

$$A = \left\{ (\lambda_1, \dots, \lambda_p) : \lambda_i > 0, i = 1, \dots, p, \sum_{j=1}^p \lambda_j = 1 \right\}.$$

**Theorem** (Condition for discarding a region (Step 6)). *Consider that  $z^*$  is the ideal point of region  $A$  of  $S$  and  $z^1$  that can be achieved in region  $B$  of  $S$ . If  $\sum_{k=1}^p \lambda_k z_k^* < \sum_{k=1}^p \lambda_k z_k^1$ ,  $\lambda \in A$ , holds then the non-dominated solution,  $\bar{z}$ , that maximizes  $\sum_{k=1}^p \lambda_k z_k(x)$ ,  $x \in S$ , cannot be achieved in region  $A$ .*

**Proof.** Consider that  $\bar{z}$  can be achieved in region  $A$ . Thus

$$\sum_{k=1}^p \lambda_k \bar{z}_k \geq \sum_{k=1}^p \lambda_k z_k^1 > \sum_{k=1}^p \lambda_k z_k^*$$

that is  $\sum_{k=1}^p \lambda_k (\bar{z}_k - z_k^*) > 0$ .

This last expression means that there is at least one  $k'$  to which  $\bar{z}_{k'} > z_{k'}^*$  and so  $z^*$  would not be the ideal point of region  $A$ . This proves the theorem.  $\square$

**Corollary 1.** *Let  $\bar{z}$  be a non-dominated solution,  $z^*$  the ideal point of region  $A \subset S$  and  $\lambda \in A$ . If  $\sum_{k=1}^p \lambda_k z_k^* < \sum_{k=1}^p \lambda_k \bar{z}_k$  and  $S^*$  is the set of optimal solutions of the linear program*

$$\max \sum_{k=1}^p \lambda_k z_k(x), \quad \text{s.t. } x \in S,$$

then  $S^* \subset S \setminus A$ .

The proof of the corollary is a direct consequence of the theorem.

Let  $N$  be the non-dominated solution set in the objective function space:  $N = \{z: z(x) \text{ is non-dominated and } x \in S\}$ .

**Corollary 2** (Technique's convergence). *If  $N$  is limited, the computation technique, presented in Section 4, converges to the searched non-dominated solution, with an error  $\varepsilon > 0$ , in a finite number of iterations.*

**Proof.** The non-dominated region of the problem is successively subdivided according to Step 4 (by subdividing the admissible region by the 'middle' of the pay-off table). The resulting sub-regions are either

discarded or subdivided until the range of their pay-off tables is lower or equal to a pre-specified error,  $\varepsilon > 0$ . Those lasting sub-regions are analyzed in order to pick the searched non-dominated solution (second part of Step 3). Considering the sets  $N^+ = N \cap \{z : z_k(x) \geq \frac{1}{2}y, x \in S, \text{ for some } y \in R, \text{ for some } 1 \leq k \leq p\}$  and  $N^- = N \cap \{z : z_k(x) \leq \frac{1}{2}y, x \in S, \text{ for some } y \in R, \text{ for some } 1 \leq k \leq p\}$  it is clear that  $N = N^+ \cup N^-$ . So, the only discarded regions are the ones to which, according to Corollary 1, the searched non-dominated solution does not belong.

A limited set can always be successively subdivided into smaller portions, in a finite number of iterations, until the dimensions of those portions being lower than a pre-specified number, if that number is bigger than zero. This concludes the proof.  $\square$

Let  $Z(\bar{z}, \lambda^1)$  be the set of ideal points of all the regions that were discarded (Step 6) on computing the non-dominated solution  $\bar{z}$  associated with  $\lambda^1 \in A$ , that is

$$Z(\bar{z}, \lambda^1) = \left\{ z^* : z^* \text{ is the ideal point of } A \subset S, \lambda^1 \in A \right. \\ \left. \text{and } \sum_{k=1}^p \lambda_k^1 z_k^* < \sum_{k=1}^p \lambda_k^1 \bar{z}_k \right\}.$$

**Corollary 3.** *If  $\bar{z}$  is a non-dominated solution that maximizes  $\sum_{k=1}^p \lambda_k^1 z_k(x), x \in S, \lambda^1 \in A$ , then the set  $A(\bar{z}, \lambda^1) = \{\lambda \in A : \sum_{k=1}^p (z_k^* - \bar{z}_k) \lambda_k < 0 \text{ for each } z^* \in Z(\bar{z}, \lambda^1)\}$  is part of the weight indifference region associated with  $\bar{z}$ .*

**Proof.** The proof of this corollary is a direct consequence of the theorem, Corollary 1 and the following.

The algorithm presented in the previous section builds a search tree of feasible regions. The regions that fulfill the condition of the theorem are discarded, the others are sub-divided until their ideal point is lower enough to fulfill the condition or until the range of their pay-off table is lower or equal to a pre-specified error,  $\varepsilon$ .  $\bar{z}$  belongs to at least one of these last regions, which were neither discarded nor sub-divided.

Without loss of generality, consider  $z^1$  the ideal point of one of those regions neither discarded nor

subdivided, and  $z^1$  a non-dominated solution of that region. So

$$\sum_{k=1}^p z_k^1 \lambda_k^1 \leq \sum_{k=1}^p \bar{z}_k \lambda_k^1 \leq \sum_{k=1}^p z_k' \lambda_k^1,$$

because  $\bar{z}$  is the non-dominated solution that maximizes the weighted-sum of the objective functions and the region was not discarded.

Besides that,  $\sum_{k=1}^p z_k' \lambda_k^1 \leq \sum_{k=1}^p (z_k^1 + \varepsilon) \lambda_k^1$ , because the range of variation of the pay-off table is lower or equal to  $\varepsilon$ .

Concluding:

$$\sum_{k=1}^p z_k^1 \lambda_k^1 \leq \sum_{k=1}^p \bar{z}_k \lambda_k^1 \leq \sum_{k=1}^p (z_k^1 + \varepsilon) \lambda_k^1$$

$$0 \leq \sum_{k=1}^p \bar{z}_k \lambda_k^1 - \sum_{k=1}^p z_k^1 \lambda_k^1 \leq \varepsilon.$$

That is, the weighted-sum of the objective functions for the non-dominated solutions of the regions that were neither discarded nor sub-divided only differ among themselves by an amount that is lower or equal to the pre-specified error, and so must be considered indifferent among themselves. This concludes the proof.  $\square$

**Remark.** The proof of this corollary implies that the second part of the technique's Step 3 can be simplified. It is not necessary to examine all the sub-regions the index of which belong to set  $Q$ , but only to guarantee that the incumbent solution  $z^1$  is not weakly non-dominated.

### 6. Illustrative example

In this section we will illustrate the computation of indifference regions through the example (Kornbluth and Steuer, 1981a):

$$\max \left\{ z_1 = \frac{x_1 - 4}{-x_2 + 3} \right\}$$

$$\max \left\{ z_2 = \frac{-x_1 + 4}{x_2 + 1} \right\}$$

$$\max \{ z_3 = -x_1 + x_2 \}$$

s.t.  $-x_1 + 4x_2 \leq 0,$   
 $x_1 - 1/2x_2 \leq 4,$   
 $x_1, x_2 \geq 0.$

The used software was developed by the author and other (João Lourenço) and can be obtained for free from the author of this paper.

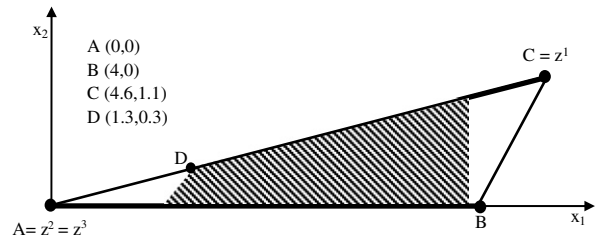


Fig. 1. Feasible and non-dominated region.

$S$  is the feasible region defined by the constraints of the problem. Fig. 1 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.  $z^1$ ,  $z^2$  and  $z^3$  are the non-dominated solutions that maximize each objective function, respectively.

Computing (through the technique presented in Section 4 and using an error  $\varepsilon = 0.1$ ) the non-dominated solution associated with the weight vector  $\lambda = (0.01, 0.8, 0.19)$ , we obtain the solution  $x_1 = 0.0, x_2 = 0.0, z_1 = -1.3, z_2 = 4.0, z_3 = 0.0$ . The resulting regions organized in a search tree are presented in Fig. 2. Note that Fig. 2, as Figs. 3–5 are screen shots from the software. The  $X$  stands for  $S$  in this paper notation.

Fig. 3 presents the pay-off table solutions, the ideal point and their weighted sum for region  $X[1]$  of Fig. 2. This region corresponds to the initial feasible region, and so this table is also the pay-off table of the original problem.

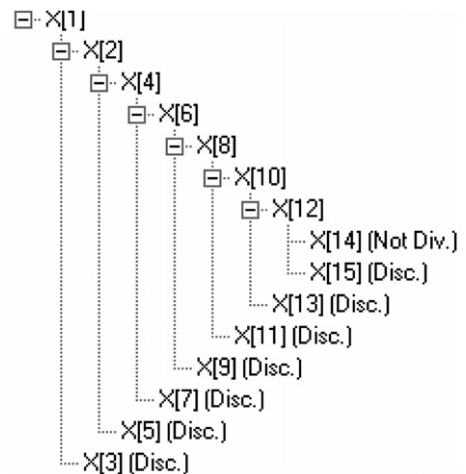


Fig. 2. Search tree of regions.  $\lambda = (0.01, 0.8, 0.19)$ . ‘Not Div.’ means ‘not divided’. ‘Disc.’ means ‘discarded’.

	z1	z2	z3	Weighted Su
Sol. 1 (z1)	0,308	-0,267	-3,429	-0,862
Sol. 2 (z2)	-1,333	4,000	0,000	3,187
Sol. 3 (z3)	-1,333	4,000	0,000	3,187
Id. Sol.	0,308	4,000	0,000	3,203

Fig. 3. Pay-off table and ideal point of region X[1]. ‘Su’ means ‘sum’. ‘Id. Sol.’ means ‘ideal solution or point’.

	z1	z2	z3	Weighted Su
Sol. 1 (z1)	-0,622	1,867	-2,133	1,082
Sol. 2 (z2)	-1,333	4,000	0,000	3,187
Sol. 3 (z3)	-1,333	4,000	0,000	3,187
Id. Sol.	-0,622	4,000	0,000	3,194

Fig. 4. Pay-off table and ideal point of region X[2].

	z1	z2	z3	Weighted Su
Sol. 1 (z1)	0,308	-0,267	-3,429	-0,862
Sol. 2 (z2)	-0,966	1,867	-1,091	1,276
Sol. 3 (z3)	-0,966	1,867	-1,091	1,276
Id. Sol.	0,308	1,867	-1,091	1,289

Fig. 5. Pay-off table and ideal point of region X[3].

As formally presented in Section 4, the regions X[2] and X[3] were generated through the division of region X[1]. The constraint that originates region X[2] is  $z_2 \geq \frac{1}{2}(4.000 - (-0.267)) + (-0.267) \iff (z_2 \geq 1.867)$ , and the constraint originating X[3] is  $z_2 \leq 1.867$ . The new regions were created by constraining the objective function two because it has the biggest range in the pay-off table of region X[1].

Figs. 4 and 5 present the pay-off table solutions, the ideal points and their weighted sums, of regions X[2] and X[3], respectively. The weighted sum value of the ideal point of region X[3] is lower than the weighted sum value of at least one of the solutions already computed (from regions X[1] and X[2]), as one can see by comparing the values in Figs. 3–5. This was the reason for discarding region X[3].

Table 1  
Ideal points of the discarded regions of Fig. 2

Region	z1	z2	z3
X[3]	0.3	1.9	-1.1
X[5]	-0.6	2.9	-1.1
X[7]	-1.0	3.0	-0.5
X[9]	-1.0	3.5	-0.5
X[11]	-1.2	3.7	-0.3
X[13]	-1.2	3.9	-0.1
X[15]	-1.3	3.9	-0.1

Table 2  
Constraints defining  $A(\bar{z}, \lambda)$

X[3]	$-2.7\lambda_1$	$+1.0\lambda_2$	$\geq -1.1$
X[5]	$-1.8\lambda_1$		$\geq -1.1$
X[7]	$-0.8\lambda_1$	$+0.6\lambda_2$	$\geq -0.5$
X[9]	$-0.9\lambda_1$		$\geq -0.5$
X[11]	$-0.4\lambda_1$		$\geq -0.3$
X[13]	$-0.1\lambda_1$	$+0.1\lambda_2$	$\geq -0.1$
X[15]	$-0.1\lambda_1$		$\geq -0.1$

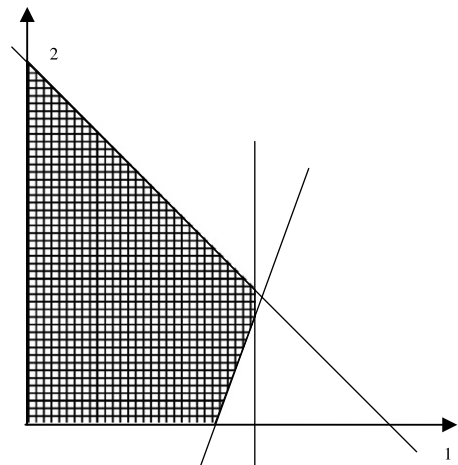


Fig. 6. Representation of  $A(\bar{z}, \lambda)$ .

The ideal points of each discarded region are listed in Table 1. The set of constraints defining  $A(\bar{z}, \lambda)$  and computed according to the Corollary 3 is presented in Table 2 and depicted in Fig. 6. Note that we consider  $\lambda_3 = 1 - \lambda_1 - \lambda_2$  and  $\lambda \geq 0$  and that a lot of constraints are redundant.

It is only simple to represent the weight indifference regions for problems up to three objective functions; nevertheless the information gathered in Table 2 is very interesting and can be used in several ways in order to help the decision maker on exploring the non-dominated region.



**7. Computational results of the technique**

We will report some computational results from tests performed on randomly generated problems. The technique was coded in Delphi Pascal 5.0 for Microsoft Windows and a simplex code for solving linear problems was obtained through URL <http://www.netcologne.de/~nc-weidenma/readme.htm>, and adapted to the present case. Three versions have been implemented by the author of this paper differing on the allowed number of regions: 10,000, 50,000 and 100,000 regions. Note that these differences imply to allocate different amounts of memory to the application and consequently the application will run at different speeds – as less memory is reserved the fastest it will run, as long as the allocated memory is enough. The application versions were used depending on the problems dimensions. The three versions were limited to 200 ( $m$ ) constraints and 140 ( $n$ ) decision vari-

ables. There was no limit on the number of objective functions ( $p$ ). The used data structures were dynamic arrays. All the data generated by the technique is kept in the data structures.

The tests used randomly generated problems according to Kuno (2002): data  $c_{kj}, d_{kj} \in [0.0, 0.5]$  and  $a_{ij} \in [0.0, 1.0]$  were uniformly random numbers;  $b$  was set to constant and equals to one. In Kuno (2002) all constant terms of denominators and numerators were the same number, which ranged from 2.0 and 100.0. Instead, in our tests  $\alpha_k, \beta_k \in [2.0, 100]$  were also uniformly random numbers. All the reported tests were carried out for what we found to be the worst case computation, that is: when all the weights are equal to one another. Each measure was obtained through the average of 20 runs, ignoring the two worst and two best values.

We used a Pentium 830, 3 MHz, with 4 Gbytes of RAM memory, under the Windows XP, Pack 2, operating system. The tests were run with the appli-

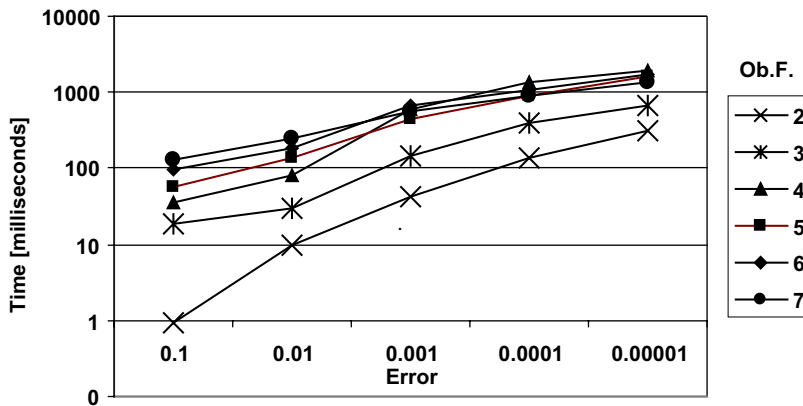


Fig. 7. Performance of the technique according to time (milliseconds). Both axes have a logarithmic scale.

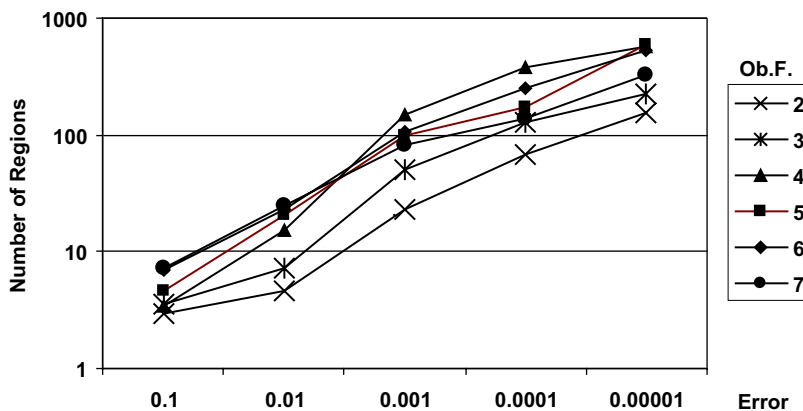


Fig. 8. Performance of the technique according to the number of regions. Both axes have a logarithmic scale.

cation priority set to high and the pagination was managed by the operating system.

Fig. 7 presents the performance of the technique according to the elapsed time in milliseconds. The problems were generated with 10 decision variables and 10 constraints. Fig. 8 presents the number of generated regions for the same problems. The standard deviation of the measures (averages) depicted in both figures are approximately equal to, but lower than the averages.

Fig. 9 shows the performance according to the elapsed time in milliseconds, with an error of 0.001. The problems were once again generated with 10 decision variables and 10 constraints. The standard deviations followed the same pattern of the results presented in Figs. 7 and 8.

Finally, Fig. 10 presents the performance of the technique according to the elapsed time, in millisec-

onds, for several different numbers of constraints and variables. The problems were generated with three objective functions and the error was kept to 0.001.

The biggest problem we were able to solve in less than an hour (3,600,000 milliseconds) had 20 objective functions, 200 constraints and 140 decision variables. The error was 0.001. The maximum number of regions (in the 20 runs of the problem) was bigger than 99,000, that is, it was near the limits of the software implementation. The Windows XP allocated more than 600 Mbytes of RAM just for the application. In order to reach higher problem dimensions it will be necessary to re-code the technique: to use data structures that do not save everything. This will add an extra complexity to the code while saving time (fewer exchanges between the RAM and the hard disk) and memory usage.

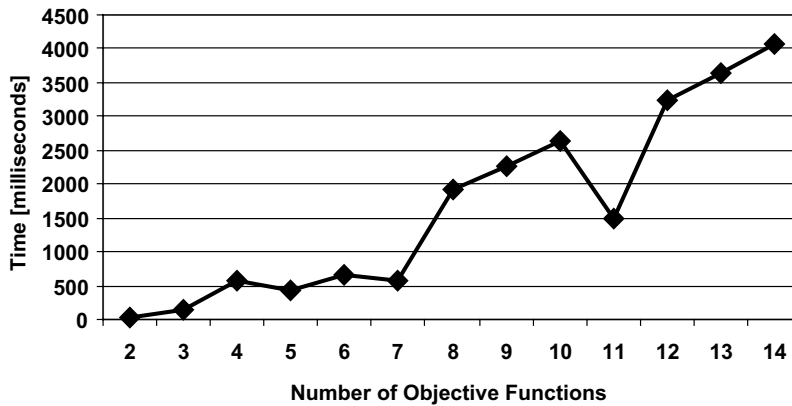


Fig. 9. Performance of the technique according to time (milliseconds). Error = 0.001 for all measures.

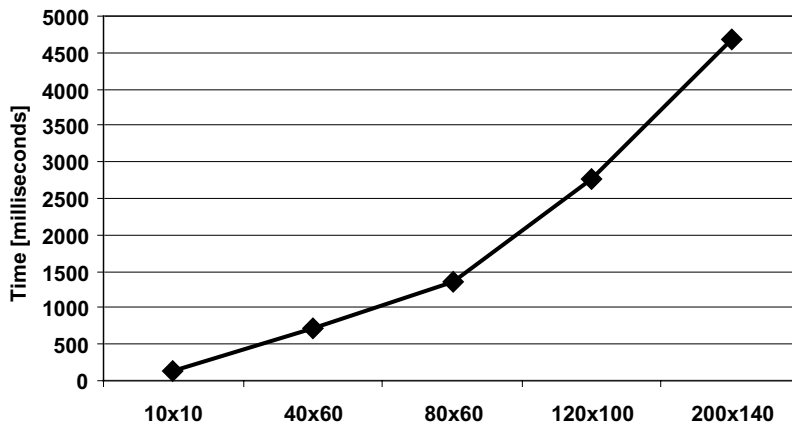


Fig. 10. Time in milliseconds against no. of constraints x no. of variables. Error = 0.001, three objective functions.

It is not possible to make a fair comparison with other techniques without coding and test them with the same problems and in the same computational environment. Nevertheless, we believe that the technique presented in this paper allows for bigger problems in less time than the one from Kuno (2002). Roughly speaking the difference is to go from seconds to milliseconds. It is worth to note that Schable and Shi (2003), Dai et al. (2005) acknowledge that the technique of Kuno (2002) works very well. The break through of the technique presented in this paper was achieved by the explicit consideration of a sum of linear ratios as a multiobjective problem: probably there is much to gain in the design of a new hybrid technique incorporating concepts from both techniques.

Phuong and Tuy (2003) reported computational experiments with a set of problems not randomly generated, but taken from the literature. We also tested the technique with those problems. The aim of the tests was not for performance comparison, but to guarantee the rightness of our computer implementation – we obtained the same results with higher precision. There is no point on comparing the performance of both techniques because the one of Phuong and Tuy (2003), being slower, also deals with other kinds of problems.

## 8. Conclusions

In this paper we presented a new technique to optimize a weighted sum of the linear fractional objective functions, i.e., to compute the non-dominated solution of the MOLFP problem associated with a given weight vector. The performance of the technique proved to be very good in the computational tests.

An issue, not considered in the presented technique but under research, is to use the primal–dual simplex to compute the pay-off tables. Probably the speed of the technique would drastically improve if we start the computation of a solution of a region's pay-off table from the optimal base of the corresponding solution of the pay-off table of the precedent region.

We also presented the conditions for building a set of constraints defining the weight variation ranges that necessarily leads to the same non-dominated solution. It is possible that weight vectors not fulfilling the set of constraints also lead to the same

non-dominated solution, but at least we defined part of the indifference region. This information is very useful in multiobjective, because in most applications the decision makers will not be satisfied with just one run of the technique. They will want to know what happens if they change the weight vector. Knowing part of the weight indifference region will ease the computation burden.

## Acknowledgements

The work presented here was partially supported by FCT and FEDER, under research project POCTI/EGE/58828/2004.

## References

- Charnes, A., Cooper, W., 1962. Programming with linear fractional functions. *Naval Research Logistics Quarterly* 9, 181–186.
- Costa, J., 2005. An interactive method for multiple objective linear fractional programming problems. *OR Spectrum* 27, 633–652.
- Costa, J., Lourenço, J., 2001. Computing the weighted sum in multiple objective linear fractional programming. In: *Proceedings of the 6th International Conference of the Decision Science Institute, Chihuahua, México*.
- Dai, Y., Shi, J., Wang, S., 2005. Conical partition algorithm for maximizing the sum of dc ratios. *Journal of Global Optimization* 31, 253–270.
- Falk, J., Palocsay, S., 1992. Optimizing the sum of linear fractional functions. In: Floudas, C., Pardalos, P. (Eds.), *Recent Advances in Global Optimization*. Princeton Univ. Press, Princeton, NJ.
- Freund, R., Jarre, F., 2001. Solving the sum-of-ratios problem by an interior-point method. *Journal of Global Optimization* 19, 83–102.
- Iserman, H., Steuer, R., 1987. Computational experience concerning payoff tables and minimum criteria values over the efficient set. *European Journal of Operational Research* 33, 91–97.
- Konno, H., Abe, N., 1999. Minimization of the sum of three linear fractional functions. *Journal of Global Optimization* 15, 419–432.
- Konno, H., Fukaiishi, K., 2000. A branch and bound algorithm for solving low rank linear multiplicative and fractional programming problems. *Journal of Global Optimization* 18, 283–299.
- 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, 1024–1039.
- Kornbluth, J., Steuer, R., 1981b. Goal programming with linear fractional criteria. *European Journal of Operational Research* 8, 58–65.

- Kuno, T., 2002. A branch-and-bound algorithm for maximizing the sum of several linear ratios. *Journal of Global Optimization* 22, 155–174.
- Phuong, N., Tuy, H., 2003. A unified monotonic approach to generalized linear fractional programming. *Journal of Global Optimization* 26, 229–259.
- Reeves, G., Reid, R., 1988. Minimum values of the efficient set in multiple objective decision making. *European Journal of Operational Research* 36, 334–338.
- Schaible, S., Shi, J., 2003. Fractional programming: The sum-of-ratios case. *Optimization Methods and Software* 18, 219–229.
- Stancu-Minasian, I., 1997. *Fractional Programming: Theory, Methods and Applications*. Kluwer Academic Publishers, Dordrecht, The Netherlands.
- Steuer, R., 1986. *Multiple Criteria Optimization: Theory, Computation and Application*. Wiley, New York.