ACTA MATHEMATICA VIETNAMICA
Volume 27, Number 2, 2002, pp. 119-139

BILINEAR PROGRAMMING APPROACH TO OPTIMIZATION OVER THE EFFICIENT SET OF A VECTOR AFFINE FRACTIONAL PROBLEM

LE DUNG MUU AND HOANG QUANG TUYEN

Abstract. We formulate the problem of optimizing a linear function over the weakly efficient set of a multicriteria affine fractional program as a special bilinear problem. For solving the latter problem, we propose a decomposition branch-and-bound algorithm taking into account the affine fractionality of the criterion function. The bounding uses only linear subprograms, the branching takes place over a simplex in the criteria space.

1. Introduction

Consider the multicriteria mathematical programming problem

$$(VP) \qquad V\min\{F(x) = (f_1(x), ..., f_p(x) \mid x \in X\}$$

where $X \subset \mathbb{R}^n$ is a bounded polyhedral convex set (polytope).

We recall that a vector $x \in X$ is called *efficient* (resp. *weakly efficient*) point for Problem (VP) if there does not exist $y \in X$ such that $F(y) \leq F(x),\ F(y) \neq F(x)$ (resp. $F(y) < F(x)$). By $E(F, X)$ (resp. $WE(F, X)$) we will denote the set of all efficient (resp. weakly efficient) points. Here and subsequently for two vectors $a = (a_1, ..., a_p)$ and $b = (b_1, ..., b_p)$ the notion $a < b$ (resp. $a \leq b$) mean that $a_i < b_i$ (resp. $a_i \leq b_i$) for all $i = 1, ..., p$. The inner product of two vectors a and b is written as $\langle a, b\rangle$ or a^Tb, b^Ta.

Recently the problems of optimizing a real valued function over the weakly efficient and the efficient sets of (VP) have attracted much attention because of their important applications in decision making. A number of solution methods have been developed for solving these problems (see e.g.[2, 3, 5, 6, 7, 12, 16, 19, 23] and the references therein). Most existing methods have been obtained assuming that Problem (VP) is linear. These methods are based upon one or more of the following properties of efficiency:

- Both the efficient and weakly efficient sets are closed.

Received November 22, 2000; in revised form March 12, 2001.

Key words and phrases. Affine fractional, efficiency, optimization over the weakly efficient and efficient sets, bilinear programming, decomposition.

This paper is supported in part by the Basic Research Program in Natural Sciences.

- Both these two sets are connected and consist of facets of the constrained polyhedron X.
- They can be characterized as equalities defined by d.c. or saddle functions.

Unfortunately these properties in general are no longer valid when (VP) is an affine fractional vector optimization problem, i.e., each function F_i is affine fractional (see [8, 18]). Namely, for affine fractional case the efficient set may be neither closed nor open. Both the efficient and the weakly efficient sets do not necessarily consist of facets of constrained polyhedron. Up to now the question of how to formulate these sets as the solution-sets of inequalities or equalities defined by d.c. or saddle functions is, to our knowledge, still open. This explains why the developed methods for linear case could not be applied directly to affine fractional case.

Affine fractional functions are widely used as performance measures in some management situations, production planning and the analysis of financial enterprises. Thus the multicriteria programming problems with affine fractional criterion functions are important and have wide applications.

In this paper we consider the optimization problems over the efficient and weakly efficient sets of a multicriteria affine fractional program. These problems are given respectively as

$$(P) \qquad \min\{d^T x \mid x \in E(F, X)\}$$

$$(WP) \qquad \min\{d^T x \mid x \in WE(F, X)\}$$

where $E(F, X)$ and $WE(F, X)$ are the efficient and weakly efficient sets of Problem (VP) with F being a vector affine fractional function.

The main difficulty of these problems arises from the fact that both $E(F, X)$ and $WE(F, X)$, are in general neither convex nor given as a constrained set of an ordinary mathematical programming problem.

These problems have been studied in [9, 22]. In [9] Choo and Akin proposed a parametric algorithm for solving (P) with $E(F, X)$ being the efficient set of a bicriteria affine fractional program. This parametric algorithm seems to work well for bicriteria case, but it cannot be extended for three or more criteria. Recently in [13], Malivert shows that Problems (P) and (WP) can be proceeded by solving a sequence of linearly constrained penalized problems of the form

$$\min\{f(x) + t_k p_w(x) \mid x \in X\}$$

where $t_k > 0$ and p_w is a certain penalty function representing the efficient and weakly efficient sets, which are in general neither convex nor differentiable. So the penalized problems remain difficult global optimization ones. No solution method for solving the penalized problems has been discussed in [13].

In our recent paper [22] we used the necessary and sufficient condition for efficiency established by Malivert in [13] to characterize the weakly efficient set by an equality defined by a biconvex function. We proposed there an algorithm for approximating a minimal point of a convex function over the weakly efficient set $WE(F,X)$. From a computational point of view, the proposed algorithm of [22] has the disadvantage that it requires solving minimax subproblems rather than linear ones for computing lower bounds.

In this paper we continue our work by refining the algorithm of [22] for approximating a global optimal solution of a linear function over the efficient and weakly efficient sets of the multiple objective affine fractional Problem (VP).

We equivalently formulate Problems (P) and (WP) as special cases of linear programs with additional bilinear constraints. We then propose a decomposition branch-and-bound method for solving the latter problems. The main difference between this algorithm and the algorithm of [22] is that here the subprograms needed to solve are linear whereas in [22] they are minimax problems.

The content of this article is as follows. In Section 2, we first illustrate our work by an example showing that both Problems (P) and (WP), unlike the linear case, do not necessarily attain their optimal solutions among the vertices of the constrained polyhedron X. Next we show how to formulate (P) and (WP) as linear problems with additional bilinear constraints. Section 3 describes an algorithm solving Problem (WP) when all vertices of X are known, particularly when X is a simplex. In Section 4 we describe a relaxation algorithm for solving Problem (WP) without requiring that all vertices of X are known in advance. In Section 5 we show how to approximate a solution of Problem (P) by solving problems of types (WP). Finally, in Section 6, we illustrate the first algorithm by a small example and report some computational results.

2. Preliminaries

To be precise throughout the paper we suppose that the criterion function in the multicriteria programming problem (VP) is given by

$$F(x) = \Big(\frac{A_1x+s_1}{B_1x+t_1}, ..., \frac{A_px+s_p}{B_px+t_p}\Big),$$

where A_i, B_i are n-dimensional vectors; s_i, t_i are real numbers for all $i = 1, .., p$. As usual we assume that $B_ix + t_i > 0$ for all $x \in X$ and all $i = 1, ..., p$. Thus F is continuous on X. By definition, the efficient and weakly efficient sets of (VP) can be given by

$$\begin{aligned} E(F,X) &= \{x \in X \mid \ \nexists y \in X : F(y) \le F(x), F(y) \ne F(x)\}, \\ WE(F,X) &= \{x \in X \mid \ \nexists y \in X : F(y) < F(x)\}. \end{aligned}$$

Since X is compact, the weakly efficient set $WE(F,X)$ is compact too [8] whereas the efficient set $E(F,X)$ in general neither closed nor open. Since $WE(F,X)$ is compact, Problem (WP) has always a global optimal solution. Throughout the paper we assume that (P) has also a global optimal solution.

It is worth pointing out that, in contrast to the linear case, both Problems (P) and (WP) do not necessarily attain their optimal solutions among the vertices of X. To see this let us consider the follwing example.

$$\min F(x) = (f_1(x), f_2(x)) = \Big(\frac{-x_1}{x_1+x_2}, \frac{3x_1-2x_2}{x_1-x_2+3}\Big)$$

subject to

$$(x_1, x_2) \in X = \left\{ x \;\middle|\; \begin{array}{c} x_1 - 2x_2 \leq 2 \\ -x_1 - 2x_2 \leq -2 \\ -x_1 + x_2 \leq 1 \\ x_1 \leq 6 \\ x_1 \geq 0,\ x_2 \geq 0 \end{array} \right\}$$

$$\min\big\{-x_1 - x_2 \;\big|\; (x_1, x_2) \in WE(F, X)\big\}$$

Since the affine fractional function $f_i(x)$ $(i = 1, 2)$ is monotone along the line segment joining any two points y^1 and y^2 in X, it follows that if

$$f_i(y^1) < f_i(y^2) \quad (i = 1, 2)$$

then

$$f_i(y^1) < f_i(\lambda y^1 + (1-\lambda)y^2) < f_i(y^2) \quad (i = 1, 2)$$

for every $\lambda \in (0, 1)$.

Take six points (Figure 1)

$$y^1 = (0, 1);\ y^2 = (2, 0);\ y^3 = (6, 2),$$
$$y^4 = (6, 7);\ y^5 = (2, 3);\ y^6 = \big(\frac{1}{2}, \frac{3}{4}\big).$$

A simple calculation shows that

$$F(y^1) = (0, -1),\ F(y^2) = \big(-1, \frac{6}{5}\big),\ F(y^3) = \big(-\frac{3}{4}, 2\big),$$
$$F(y^4) = \big(-\frac{6}{13}, 2\big),\ F(y^5) = \big(-\frac{2}{5}, 0\big),\ F(y^6) = \big(-\frac{2}{5}, 0\big).$$

Using the monotonicity of $f_i(x)$ we can see that the weakly efficient set $WE(F, X)$ for this example consists of three segments $[y^1, y^5]$, $[y^5, y^6]$ and $[y^6, y^2]$. This can also be verified by using Corollary 1 in [13]

Thus

$$\min\{f(x) := d^T x = -x_1 - x_2 \mid (x_1, x_2) \in WE(F, X)\}$$
$$= f(y^5) = -2 - 3 = -5 < f(y^i), \quad (i = 1, 2, 3, 4).$$

Clearly, $y^5 \notin V(X) = \{y^1, y^2, y^3, y^4\}$.

The following theorem due to Malivert [13] will be useful for our purpose.

Theorem 2.1. [13] *A vector $x \in X$ is efficient (resp. weakly efficient) if and only if there exist real numbers $\lambda_i > 0$ (resp. $\lambda_i \geq 0$ not all zero) for all $i = 1, \dots, p$ such that*

$$\sum_{i=1}^{p} \lambda_i[(B_i x + t_i)A_i - (A_i x + s_i)B_i](x - y) \le 0 \quad \forall y \in X.$$

Figure 1

By dividing by $\sum\limits_{i=1}^{p} \lambda_i > 0$ we can assume that $\sum\limits_{i=1}^{p} \lambda_i = 1$. So, if

$$\Lambda_0 := \Big\{\lambda = (\lambda_i, ..., \lambda_p) \mid \lambda > 0,\ \sum_{i=1}^{p} \lambda_i = 1\Big\},$$

$$\Lambda := \Big\{\lambda = (\lambda_i, ..., \lambda_p) \mid \lambda \ge 0,\ \sum_{i=1}^{p} \lambda_i = 1\Big\}$$

then

$$E(F, X) = \Big\{x \in X \mid \exists \lambda \in \Lambda_0,$$
$$\sum_{i=1}^{p} \lambda_i[(B_i x + t_i)A_i - (A_i x + s_i)B_i](x - y) \le 0 \quad \forall y \in X\Big\}.$$

$$WE(F, X) = \Big\{x \in X \mid \exists \lambda \in \Lambda,$$
$$\sum_{i=1}^{p} \lambda_i[(B_i x + t_i)A_i - (A_i x + s_i)B_i](x - y) \le 0 \quad \forall y \in X\Big\}.$$

Thus both Problems (P) and (WP) can take the form

$$(IP) \qquad \begin{cases} \min\{f(x) = d^T x\} \\ \text{s.t.} \\ x \in X,\ \lambda \in \bar{\Lambda}, \\ \sum\limits_{i=1}^{p} \lambda_i[(B_i x + t_i)A_i - (A_i x + s_i)B_i](x - y) \leq 0 \quad \forall y \in X \end{cases}$$

where $\bar{\Lambda} = \Lambda_0$ for (P) and $\bar{\Lambda} = \Lambda$ for (WP).

Let $v^1, ...v^q$ denote the vertices of X. Problem (IP) then can be further reduced due to the following proposition.

Proposition 2.1. *We have*

$$\sum_{i=1}^{p} \lambda_i[(B_i x + t_i)A_i - (A_i x + s_i)B_i](x - y) \leq 0 \quad \forall y \in X$$

if and only if

$$\sum_{i=1}^{p} \lambda_i[(B_i x + t_i)A_i - (A_i x + s_i)B_i](x - v^k) \leq 0 \quad \forall k = 1, ..., q.$$

Proof. Since $v^1, ..., v^q \in X$, we only need to prove the "only if" part. Since every $y \in X$ can be expressed as

$$y = \sum_{k=1}^{q} \gamma_k v^k,\ 0 \leq \gamma_k \leq 1 \quad \text{and} \quad \sum_{k=1}^{q} \gamma_k = 1,$$

we have

$$\begin{aligned} &\sum_{k=1}^{q} \gamma_k \sum_{i=1}^{p} \lambda_i[(B_i x + t_i)A_i - (A_i x + s_i)B_i](x - v^k) \leq 0 \quad \forall k = 1, ..., q \\ \Leftrightarrow\ &\sum_{i=1}^{p} \lambda_i[(B_i x + t_i)A_i - (A_i x + s_i)B_i]\Big(\Big(\sum_{k=1}^{q} \gamma_k\Big)x - \sum_{k=1}^{q} \gamma_k v^k\Big) \leq 0 \\ \Leftrightarrow\ &\sum_{i=1}^{p} \lambda_i[(B_i x + t_i)A_i - (A_i x + s_i)B_i](x - y) \leq 0 \quad \forall y \in X. \end{aligned}$$

□

For simplicity we define

$$\begin{aligned} M(\lambda, x, y) &:= \sum_{i=1}^{p} \lambda_i[(B_i x + t_i)A_i - (A_i x + s_i)B_i](x - y), \\ M_k(\lambda, x) &:= M(\lambda, x, v^k) \\ &= \sum_{i=1}^{p} \lambda_i[(B_i x + t_i)A_i - (A_i x + s_i)B_i](x - v^k), \quad k = 1, ..., q. \end{aligned}$$

By Theorem 2.1, if $(\lambda, x) \in \Lambda \times X$ (resp. $(\Lambda_0 \times X)$ then x is weakly efficient (resp. efficient) if and only if

$$M(\lambda, x, y) \leq 0 \quad \forall y \in X$$

or

$$M_k(\lambda, x) \leq 0 \quad \forall k = 1, ..., q.$$

Clearly,

(i) $M(\lambda, x, .)$ is affine on X for each fixed (λ, x);

(ii) For each k, the function $M_k(\lambda, x)$ is bilinear on $\Lambda \times X$.

For each v^k we define

$$G_k(\lambda) = \sum_{i=1}^{p} \lambda_i [(t_i + B_i v^k) A_i - (s_i + A_i v^k) B_i],$$

$$b_k(\lambda) = \sum_{i=1}^{p} \lambda_i [t_i A_i - s_i B_i] v^k.$$

Let $G(\lambda)$ denote the $(q \times n)$-matrix whose kth row is $G_k(\lambda)$ $(k = 1, ..., q)$, and $b(\lambda)$ denote the q-dimensional vector whose kth entry is $b_k(\lambda)$. Suppose that

$$X = \{x \geq 0 \mid Gx \leq b\}.$$

Then by using Proposition 2.1 and the definitions of $G(\lambda)$ and $b(\lambda)$ we can rewrite Problems (IP) as

$$(P\bar{\Lambda}) \qquad \begin{cases} \min\{f(x) := d^T x\} \\ \text{s.t.} \\ Gx - b \leq 0, x \geq 0, \\ G(\lambda)x - b(\lambda) \leq 0, \lambda \in \bar{\Lambda}. \end{cases}$$

Remark 2.1. To determine Problem $(P\bar{\Lambda})$ we require that all vertices of the polyhedron X are known in advance. So this formulation is recommended to use when the vertices of X can be easily calculated. A special case appeared already in some applications (see e.g. [17, 18]) where each component x_i of the decision variable x represents the ratio of ith quantity to be defined. In this case X is a simplex given by

$$X := \Big\{x = (x_1, ..., x_n) \mid \sum_{i=1}^{n} x_i = 1, \ x_i \geq 0 \quad \forall i = 1, ..., n\Big\}.$$

Clearly X has exactly n vertices which are the unit vectors of R^n.

3. Solution method

From the preceeding section we see that the problems of optimizing a function f over the efficient and weakly efficient sets of a multicriteria affine fractional program (VP) can be formulated as bilinearly constrained problems of the form $(P\bar{\Lambda})$ with $\bar{\Lambda} = \Lambda_0$ and $\bar{\Lambda} = \Lambda$, respectively. The main difference between these two problems is that Λ is a unit simplex where $\Lambda_0 = int\Lambda$. Problem $(P\Lambda)$ is thus easier to be handled.

Bilinear programming is an important topic of mathematical programming. A lot of methods have been developed for solving bilinear programming problems (see e.g. [1, 11, 14, 15, 20, 21] and the references therein). Most of the developed methods have been obtained by assuming that the bilinear term appears in the objective function.

In this section we first describe a decomposition method for solving Problem $(P\bar{\Lambda})$ with $\bar{\Lambda} = \Lambda$ by using its specific structure. Then we will use the fact that Λ_0 is the interior of Λ to approximate a solution of $(P\Lambda_0)$.

As usual, for a given $\varepsilon \geq 0$, we call a point x an ε-optimal solution to Problem (P) if x is feasible and $f(x) - f_* \leq \varepsilon(|f(x)| + 1)$, where f_* denotes the optimal value of (P).

The proposed algorithm is a branch-and-bound procedure using Lagrangian bounding operation and a simplicial subdivision. Unlike the algorithm in [22], this algorithm uses only linear programming for computing lower and upper bounds.

Let

$$H(\lambda) := \begin{bmatrix} G \\ G(\lambda) \end{bmatrix}, \quad h(\lambda) = \begin{bmatrix} b \\ b(\lambda) \end{bmatrix}$$

Then Problem $(P\bar{\Lambda})$ can be rewritten as follows

$$(B\bar{\Lambda}) \qquad \begin{cases} \min\{f(x) = d^T x\} \\ \text{s.t.} \\ H(\lambda)x - h(\lambda) \leq 0, \\ x \geq 0, \ \lambda \in \bar{\Lambda}. \end{cases}$$

Lagrangian Bounding Operation

Define the function $\phi : \Lambda \to R$ by setting

$$(P_\lambda) \qquad \phi(\lambda) = \min\{d^T x \mid H(\lambda)x - h(\lambda) \leq 0, \ x \geq 0\}.$$

Then the problem

$$(MP) \qquad \min\{\phi(\lambda) \mid \lambda \in \Lambda\}$$

is equivalent to Problems $(B\Lambda)$ and (WP) in the sense of the following proposition whose proof is obvious from the results of the preceeding section. Let f_* and w_* denote the optimal values of Problem (P) and (WP) respectively.

Proposition 3.1. *A point (λ^*, x^*) is optimal to Problem $(B\Lambda)$ if and only if x^* is optimal to (WP) and λ^* is optimal to (MP) and $w_* = f(x^*) = \phi(\lambda^*)$.*

Note that a feasible point of Problem $(B\Lambda)$ can be obtained by solving a standard linear program. In fact, if $\lambda \in \Lambda$ fixed and x^λ is an optimal solution of the linear problem (P_λ) then (λ, x^λ) is feasible for $(B\Lambda)$. Hence x^λ is feasible for (WP). So upper bounds for w_* are computed by solving a standard linear program. As the algorithm executes more and more feasible points can be found, and thereby upper bounds for w_* can be iteratively improved.

We now compute lower bound for w_* by using Lagrangian duality. Let S be a fully dimensional subsimplex of Λ. Let $V(S)$ denote the vertex set of S. Consider Problem $(B\Lambda)$ restricted on S, i.e.,

$$(BS) \qquad \begin{cases} w_*(S) := \min d^T x \\ \text{s.t.} \\ H(\lambda)x - h(\lambda) \leq 0, \\ x \geq 0,\ \lambda \in S. \end{cases}$$

Let $L(u, \lambda, x)$ be the Lagrangian function with respected to the constraint $H(\lambda)x-b(\lambda)$ of this problem. That is

$$(1) \qquad L(u, \lambda, x) = d^T x + u^T (H(\lambda)x - h(\lambda)).$$

Define the function $m(u, \lambda)$ as

$$m(u, \lambda) = \min_{x \geq 0} \{d^T x + u^T (H(\lambda)x - h(\lambda))\}.$$

From the Lagrangian duality theorem we have

$$(2) \qquad m(u, \lambda) \leq \phi(\lambda) \quad \forall u \geq 0,\ \forall \lambda \in S.$$

Since for each fixed λ the function $H(\lambda)x - h(\lambda)$ is affine, we have

$$(3) \qquad \sup_{u \geq 0} m(u, \lambda) = \phi(\lambda).$$

Let

$$(4) \qquad \mu_S(u) = \min_{\lambda \in S} m(u, \lambda).$$

From (2) it follows that

$$\mu_S(u) = \min_{\lambda \in S} m(u, \lambda) \leq \min_{\lambda \in S} \phi(\lambda) = w_*(S) \quad \forall u \geq 0.$$

Hence

$$\sup_{u \geq 0} \mu_S(u) \leq w_*(S).$$

Therefore, taking

$$(5) \qquad \beta(S) = \sup_{u \geq 0} \mu_S(u)$$

we obtain a lower bound $\beta(S)$ for $w_*(S)$. The following lemma states that this lower bound can be computed by solving linear programs one for each vertex of S.

Lemma 3.1. *Let λ^i ($i = 1, ..., p$) be the vertices of S. Then*

$$(LS) \qquad \beta(S) := \min_{\lambda \in V(S)} \begin{cases} \sup -h^T(\lambda)u, \ \text{s.t.} \\ H^T(\lambda^i)u + d \geq 0, \quad u \geq 0,\ i = 1, ..., p \end{cases}$$

Proof. From (1) and (5) it follows that

$$\beta(S) = \sup_{u\geq 0} \mu_S(u) = \sup_{u\geq 0} \min_{\lambda\in S} m(u,\lambda).$$

Hence

$$\beta(S) = \sup_{u\geq 0} \min_{\lambda\in S} \min_{x\geq 0}\{d^T x + u^T(H(\lambda)x - h(\lambda))\}.$$

Then

$$\beta(S) = \sup_{u\geq 0} \min_{\lambda\in S} \big\{ \min_{x\geq 0}(d^T + u^T H(\lambda))x - u^T h(\lambda))\big\}. \tag{6}$$

If $\lambda \in S$ is such that $d^T + u^T H(\lambda) \not\geq 0$ for all $u \geq 0$, then

$$\min_{x\geq 0}(d^T + u^T H(\lambda))x = -\infty.$$

So, the supremum in (6) can be taken over all $u \geq 0$ satisfying

$$H^T(\lambda)u + d^T \geq 0 \quad \forall \lambda \in S \tag{7}$$

which implies

$$\min_{x\geq 0}(u^T H(\lambda) + d^T)x = 0.$$

Since

$$\big(H^T(\lambda)u + d \geq 0 \quad \forall\lambda \in S\big) \Longleftrightarrow \big(H^T(\lambda)u + d \geq 0 \quad \forall\lambda\in V(S)\big),$$

it follows from (6) and (7) that

$$\beta(S) = \sup_{u\geq 0} \min_{\lambda\in S} \Big\{ - u^T h(\lambda) \bigm| H^T(\lambda)u + d^T \geq 0 \; i = 1,\dots,p\Big\} \tag{8}$$

with $\lambda^i \in V(S)$, $i = 1,\dots,p$. By the minimax theorem, we can interchange the supremum and the infimum in (8) to obtain

$$\beta(S) := \min_{\lambda\in V(S)} \begin{cases} \sup -h^T(\lambda)u, \text{ s.t.} \\ H^T(\lambda^i)u + d \geq 0, \; i = 1,\dots p, \quad u \geq 0, \end{cases}$$

which proves the lemma. □

Simplicial Bisection

At each iteration k of the algorithm to be described below, a subsimplex of the simplex Λ will be bisected into subsimplices such a way so that as the algorithm executes the obtained lower and upper bounds tend to the same limit. This can be done by using an exhaustive simplicial bisection that is commonly known in global optimization [11]. This simplicial bisection can be described as follows. Let S_k be a subsimplex of full dimension of Λ that we want to bisect at iteration k. Let v^k, w^k be two vertices of S_k such that the edge joining these vertices is longest. Let $u^k = t_k v^k + (1 - t_k)w^k$ with $0 < t_k < 1$. Bisect S_k into two subsimplices S_{k1} and S_{k2}, where S_{k1} and S_{k2} are obtained from S_k by replacing v^k and w^k respectively by u^k. It is well known [11] that $S_k = S_{k1} \cup S_{k2}$, and that if $\{S_k\}$ is an infinite sequence of nested simplices generated by this simplicial bisection process such that $0 < \delta_0 < t_k < \delta_1 < 1$ for every k, then the sequence $\{S_k\}$ shrinks to a singleton.

Now we are in a position to describe the algorithm for solving $(P\bar{\Lambda})$ with $\bar{\Lambda} = \Lambda$.

Algorithm LB

Initialization. Choose a tolerance $\varepsilon \geq 0$ and set $S_0 := \Lambda$. For each $v \in V(S_0)$ solve the linear program

$$(L_v) \qquad \beta(v) := \begin{cases} \max -h^T(v)u, \text{ s.t.} \\ H^T(\lambda^i)u + d \geq 0 \quad u \geq 0, \ \forall \lambda^i \in V(S_0). \end{cases}$$

Let $\beta(S_0) := \min_{v \in V(S_0)} \beta(v)$ and $\lambda^0 \in S_0$, $u^0 \geq 0$ such that $\beta(S_0) = -h^T(\lambda^0)u^0$.

Solve the linear program

$$\begin{cases} \min\{f(x) := d^T x\}, \text{s.t.} \\ H(\lambda^0)x - h(\lambda^0) \leq 0, \ x \geq 0 \end{cases}$$

to obtain x^0. Let $\alpha_0 := \phi(\lambda^0) := d^T x^0$. Set $\beta_0 := \beta(S_0)$ and

$$\Gamma_0 := \begin{cases} \{S_0\} \text{ if } \alpha_0 - \beta_0 > \varepsilon(|\alpha_0| + 1), \\ \emptyset \quad \text{otherwise} \end{cases}$$

Let $k \leftarrow 0$ and go to Iteration k.

Iteration k $(k = 0, 1, ...)$

*Step k*1 (selection):

a) If $\Gamma_k = \emptyset$, then terminate: x^k is an ε-optimal solution and α_k is the ε-optimal value to Problem (WP).

b) If $\Gamma_k \neq \emptyset$ then select $S_k \in \Gamma_k$ such that

$$\beta_k = \beta(S_k) = \min\{\beta(S) \mid S \in \Gamma_k\}.$$

*Step k*2 (bisection): Bisect S_k into two simplices S_{k1} and S_{k2} by the exhaustive simplicial bisection described above.

*Step k*3 (bounding): Compute, for each $v \in V(S_{kj})$,

$$(LS_{kj}) \qquad \beta(v) := \begin{cases} \max -h^T(v)u, \text{ s.t.} \\ H^T(\lambda^i)u + d \geq 0 \quad u \geq 0, \ \forall \lambda^i \in V(S_{kj}) \end{cases}$$

Set $\beta(S_{kj}) = \min\limits_{v \in V(S_{kj})} \beta(v)$. Let u^{kj} be the obtained optimal solution and $\lambda^{kj} \in V(S_{kj})$ such that

$$\beta(S_{kj}) = -h^T(\lambda^{kj})u^{kj}, \ (j = 1, 2)$$

*Step k*4 (updating): Solving the linear programs, one for each λ^{kj}

$$\begin{cases} \min\{f(x) := d^T x\}, \text{ s.t.} \\ H(\lambda^{kj})x - h(\lambda^{kj}) \leq 0, \ x \geq 0 \end{cases}$$

we obtain new feasible points. Using these feasible points to update the upper bound. Let x^{k+1} be the best feasible point among x^k and the newly generated feasible points. Set $\alpha_{k+1} := d^T x^{k+1}$ and

$$\Gamma_{k+1} \leftarrow \{S \in (\Gamma_k \backslash \{S_k\}) \cup \{S_{k1}, S_{k2}\} \bigm| \alpha_{k+1} - \beta(S) > \varepsilon(|\alpha_{k+1}| + 1)\}$$

Increase k by 1 and go to iteration k.

Theorem 3.1. (i) *If Algorithm LB terminates at iteration k, then x^k is an ε-global optimal solution to Problem (WP).*

(ii) *If the algorithm does not terminate then $\beta_k \nearrow w_*$, $\alpha_k \searrow w_*$ as $k \to +\infty$, and any cluster point of the sequence $\{x^k\}$ is a globally solution to Problem (WP).*

Proof. (i) If the algorithm LB terminates at iteration k then $\Gamma_k = \emptyset$. This implies that $\alpha_k - \beta_k \leq \varepsilon(|\alpha_k| + 1)$. Since $\beta_k \leq w_*$, and $\alpha_k = f(x^k)$ it follows that $f(x^k) - w_* \leq \varepsilon(|f(x^k)| + 1)$. Hence x^k is an ε-optimal solution.

(ii) Since $S_k = S_{k1} \cup S_{k2}$, by the rule for computing lower bound $\beta(S_k)$ we have

$$\beta_k = \beta(S_k) \leq \beta(S_{k+1}) = \beta_{k+1} \ \forall k.$$

Also, since α_{k+1} is the currently smallest upper bound determined at Step $k4$ we have $\alpha_{k+1} \leq \alpha_k \ \forall k$. Thus, both $\beta_* = \lim \beta_k$ and $\alpha_* = \lim \alpha_k$ exist and satisfy

$$\beta_* \leq w_* \leq \alpha_* \tag{9}$$

Suppose that the algorithm does not terminate. Then it generates an infinite sequence of nested simplices that, for simplicity of notation, we also denote by $\{S_k\}$. Since the subdivision is exhaustive, this sequence strinks to a singleton, say $\lambda^* \in \Lambda$. By the rule for computing lower bound β_k we have

$$\beta_k = \sup_{u \geq 0} \min_{\lambda \in S_k} m(u, \lambda) \geq \min_{\lambda \in S_k} m(u, \lambda) \ \forall u \geq 0.$$

Since the sequence $\{S_k\}$ tends to λ^* as $k \to +\infty$ we obtain

$$\beta_* = \lim \beta_k \geq m(u, \lambda^*) \quad \forall u \geq 0. \tag{10}$$

Since $\phi(\lambda^k)$ is an upper bound determined at Step $k1$ and α_{k+1} is the currently smallest upper bound obtained at Step $k4$ we have

$$\alpha_{k+1} \leq \phi(\lambda^k) \quad \forall k.$$

Since $\lambda^k \to \lambda^*$, it follows from the continuity of ϕ [4] that

$$\alpha_* = \lim \alpha_k = \lim \alpha_{k+1} \leq \lim \phi(\lambda^k) = \phi(\lambda^*). \tag{11}$$

On the other hand, by Lagrangian duality theorem for the linear program determining $\phi(\lambda^*)$ we have

$$\sup_{u \geq 0} m(u, \lambda^*) = \phi(\lambda^*).$$

Then from (10) and (11) it follows that

$$\alpha_* \leq \phi(\lambda^*) \leq \beta_*$$

which together with (9) implies

$$\beta_* = w_* = \alpha_* = \phi(\lambda^*).$$

Let x^* be any cluster point of the sequence $\{x^k\}$. By the definition we have $\alpha_k = f(x^k)$. Since $\alpha_k \searrow w_*$, it follows from continuity of f that $w_* = f(x^*)$. Since $x^k \in WE(F,X)$ for all k and $WE(F,X)$ is closed [8], $x^* \in WE(F,X)$. Hence x^* is a globally optimal solution to Problem (WP). □

Remark 3.1. (i) When $\varepsilon > 0$ the algorithm must terminate affer a finite number of iterations. Indeed, if the algorithm does not terminate at iteration k, then $\alpha_k - \beta_k > \varepsilon(|\alpha_k| + 1)$. Since $\alpha_k - \beta_k \to 0$ as $k \to +\infty$, it follows that when $\varepsilon > 0$ the inequality $\alpha_k - \beta_k > \varepsilon(|\alpha_k| + 1)$ can not happen indefinitely.

(ii) The subdivision takes place on the simplex Λ whose dimension is just equal to the number of the criteria of Problem (VP) that is just equel to the number of linear programs needed to solve for computing lower bounds. So the algorithm is expected to be efficient only when the number of the criteria in (VP) is small.

4. Relaxation approach

The algorithm described in the previous section requires that all vertices of the constrained polyhedron X are known in advance. So the algorithm can be used only when the vertices of X are easy to compute. In the case where computing all vertices of X is expensive, we propose a relaxation algorithm that computes vertices of X iteratively in a branch-and-bound procedure. It may expect that the algorithm finds a globally optimal solution without computing all of vertices of X. To this end, we suppose that some vertices $v^1, ..., v^r$ of X are known. As before, for each v^k we define

$$G_k(\lambda) = \sum_{i=1}^{p} \lambda_i[(t_i + B_i v^k)A_i - (s_i + A_i v^k)B_i],$$

$$b_k(\lambda) = \sum_{i=1}^{p} \lambda_i[t_i A_i - s_i B_i]v^k.$$

Denote by $G^r(\lambda)$ the $(r \times p)$ matrix whose kth row is $G_k(\lambda)$ $(k = 1, ..., r)$, and by $b^r(\lambda)$ the r-dimensional vector whose kth entry is $b_k(\lambda), (k = 1, \ldots, r)$ and

$$H^r(\lambda) := \begin{bmatrix} G \\ G^r(\lambda) \end{bmatrix}, \quad h^r(\lambda) = \begin{bmatrix} b \\ b^r(\lambda) \end{bmatrix}.$$

Consider the following problem

$$(B_r\Lambda) \qquad \begin{cases} \min\{f(x) := d^T x\}, \text{ s.t.} \\ H^r(\lambda)x - h^r(\lambda) \leq 0, \ x \geq 0, \ \lambda \in \Lambda. \end{cases}$$

Let (λ^r, x^r) be an optimal solution of this problem, If x^r is weakly efficient, then x^r is an optimal solution to (WP). Otherwise, the optimal value of $(B_r\Lambda)$ is a lower bound for that of $(B\Lambda)$, because the feasible set of $(B\Lambda)$ is contained in the feasible domain of $(B_r\Lambda)$. Then one can increase lower bound by adding a new vertex of X to obtain Problem $(B_{r+1}\Lambda)$ and so on. Since the number of the vertices of X is finite, this procedure must terminate yielding an optimal solution to $(B\Lambda)$.

This procedure can be described in details as follows. Let the tolerance $\epsilon > 0$ be given.

Algorithm RLB

Step 0. Choose one or more distinct vertices $v^1, ..., v^r$ of X.

Step 1. Solve $(B_r\Lambda)$ by using Algorithm LB. Let (λ^r, x^r) be an ϵ-optimal solution to $(B_r\Lambda)$.

Step 2. Solve the linear program

$$(L_r) \qquad \max\big\{M(\lambda^r, x^r, y) \mid y \in X\big\}$$

to obtain an optimal solution $v^{r+1} \in V(X)$.

a) If $M(\lambda^r, x^r, v^{r+1}) \le 0$, then terminate: (λ^r, x^r) is an ϵ-optimal solution to $(B\Lambda)$ (hence x^r is weakly efficient and therefore it is an ϵ-optimal solution to (WP)).

b) If $M(\lambda^r, x^r, v^{r+1}) > 0$, then use v^{r+1} to define Problem $(B_{r+1}\Lambda)$. Increase r by one and go back to Step 1.

Theorem 4.1. *Algorithm RLB terminates after a finite number of Step 1 yielding an ϵ-optimal solution to Problem $(B\Lambda)$.*

Proof. . By the definition of $M(\lambda, x, y)$ we see that Problem $(P\Lambda)$ is equivalent to the problem

$$\begin{cases} \min\{f(x) := d^T x\}, \text{ s.t.} \\ M(\lambda, x, v^i) \le 0, \ \forall\ v^i \in V(X) \\ \lambda \in \Lambda, \ x \in X, \end{cases}$$

and the relaxed problem $(B_r\Lambda)$ is equivalent to

$$\begin{cases} \min\{f(x) := d^T x\}, \text{ s.t.} \\ M(\lambda, x, v^i) \le 0, \ i = 1, ..., r \\ \lambda \in \Lambda, \ x \in X. \end{cases}$$

Since (λ^r, x^r) is feasible for $(B_r\Lambda)$, we have $M(\lambda^r, x^r, v^i) \le 0$ for every $i = 1, ..., r$. Since

$$M(\lambda^r, x^r, v^{r+1}) = \max_{x \in X} M(\lambda^r, x^r, x),$$

it follows from Theorem 1.2 that if $M(\lambda^r, x^r, v^{r+1}) \le 0$ then x^r is weakly efficient, and therefore (λ^r, x^r) solves $(B\Lambda)$. If $M(\lambda^r, x^r, v^{r+1}) > 0$, then $v^{r+1} \ne v^j$ for every $j = 1, ..., r$, because $M(\lambda^r, x^r, v^j) \le 0$ for every $j = 1, ..., r$. Thus if the algorithm does not terminate, it generates, at each Step 2, a new vertex of X

that distinct from all previous ones. Hence the algorithm must terminate after a finite number of Step 1, since the number of the vertices of X is finite. □

5. The efficient set case

Because of nonclosedness of Λ_0, Problem $(P\Lambda_0)$ is more difficult to handle than $(P\Lambda)$. To avoid nonclosedness of Λ_0, for $\delta > 0$ we define

$$\Lambda_\delta := \Big\{\lambda = (\lambda_1, ..., \lambda_p) | \lambda_j \geq \delta > 0,\ \forall j,\ \sum_{j=1}^{p} \lambda_j = 1\Big\}.$$

Clearly, for every $\delta > 0$ sufficiently small, $\Lambda_\delta \neq \emptyset$ and

$$\Lambda_\delta \subset \Lambda_0 \subset \Lambda.$$

Let $\delta > 0$ be fixed such that $\Lambda_\delta \neq \emptyset$. Consider the problem

$$(P\Lambda_\delta) \qquad \begin{cases} \min\big\{f(x) = d^T x\big\}, \text{ s.t.} \\ H(\lambda)x - h(\lambda) \leq 0,\ x \geq 0,\ \lambda \in \Lambda_\delta. \end{cases}$$

Since the feasible domain of this problem is compact, it admits an optimal solution. Corresponding to $(P\Lambda_\delta)$ we define the problem

$$(MP_\delta) \qquad \min\{\phi(\lambda) \mid \lambda \in \Lambda_\delta\}.$$

Since Λ_δ is compact, Problems $(P\Lambda_\delta)$ and (MP_δ) are equivalent in the sense that if (λ^*, x^*) is optimal for $(P\Lambda_\delta)$ then λ^* is optimal for (MP_δ). Conversely, if λ^* is optimal for (MP_δ) and x^* optimal solution of problem defining $\phi(\lambda^*)$, then (λ^*, x^*) solves $(P\Lambda_\delta)$.

Clearly, if (λ^*, x^*) is an optimal solution to $(P\Lambda)$ and $\lambda^* \in \Lambda_0$ then (λ^*, x^*) is optimal solution to $(P\Lambda_0)$ as well. Otherwise an optimal solution of $(P\Lambda_0)$ can be approximated by solving Problem $(P\Lambda_\delta)$ with $\delta > 0$ sufficiently small, as shown in the following proposition.

Proposition 5.1. *Suppose that Problem (P) has an optimal solution. Let $\{\delta_k\}$ be a decreasing sequence of positive numbers tending to zero. Suppose that δ_k is small enough such that $\Lambda_{\delta_k} \neq \emptyset$ for every k. Let λ^k be an optimal solution of Problem (MP_{δ_k}) and x^k be a solution of the problem defining $\phi(\lambda^k)$, i.e., $\phi(\lambda^k) = f(x^k)$. Then $f(x^k) \to f_*$ as $k \to \infty$, and $x^k \in E(F, X)$ for all k.*

Proof. Since $\delta_k > \delta_{k+1} > 0$, we have $\Lambda_{\delta_k} \subset \Lambda_{\delta_{k+1}}$. Let $f_k := \phi(\lambda^k)$. Then

$$\begin{aligned} f_k &= \min\{\phi(\lambda) | \lambda \in \Lambda_{\delta_k}\} \\ &\geq \min\{\phi(\lambda) | \lambda \in \Lambda_{\delta_{k+1}}\} \\ &= \phi(\lambda^{k+1}) \\ &= f_{k+1} \geq \min_{\lambda \in \Lambda_0} \phi(\lambda) = f_*. \end{aligned}$$

Since $\{f_k\}$ is a decreasing sequence bounded from below, there exists f_0 such that

$$f_0 = \lim_{\delta_k \to 0} f_k.$$

As $f_* \leq f_k$ for all k, it follows that $f_* \leq f_0$.

Since $x^* \in E(X,F)$, there exists $\lambda^* \in \Lambda_0$ such that $\phi(\lambda^*) = f_*$. Thus $\phi(\lambda^*) \leq f_0$. We show that $\phi(\lambda^*) = f_0$. Indeed, otherwise if $\phi(\lambda^*) < f_0$, then $\lambda^* \notin \Lambda_{\delta_k}$ for all k sufficiently large. However, since $\lambda^* \in \Lambda_0$, and $\delta_k \searrow 0$, one can easy to see that $\lambda^* \in \Lambda_{\delta_k}$ for all $\delta_k < \min\{\lambda_1^*, ..., \lambda_p^*\}$. We arrive at a contradiction. Thus $f_0 = \phi(\lambda^*) = f_*$, which implies that $f(x^k) \to f_*$.

Note that

$$\lambda^k \in \Lambda_{\delta_k} \subset \Lambda_0 \ \ \forall\ k.$$

Thus, since x^k is an optimal solution of Problem (P_{λ^k}) defining $\phi(\lambda^k)$, we have $x^k \in E(F,X)$ for every k. □

Remark 5.1. Since the set $E(F,X)$ may be not closed, a cluster point of the sequence $\{x^k\}$ may not be efficient, but it must be weakly efficient because $E(F,X) \subset WE(F,X)$ and $WE(F,X)$ is closed.

6. Example and computational results

We now illustrate the LB algorithm by the following example which has been introduced in Section 2.

$$\text{(VP)} \qquad V\min_{x\in X} F(x) = (f_1(x), f_2(x)) = \Big(\frac{-x_1}{x_1+x_2}, \frac{3x_1-2x_2}{x_1-x_2+3}\Big)$$

where

$$X = \{x | Gx \leq b, x \geq 0\},$$

$$G = \begin{bmatrix} 1 & -2 \\ -1 & -2 \\ -1 & 1 \\ 1 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ -2 \\ 1 \\ 6 \end{bmatrix}.$$

Let $\Lambda = \{\lambda = (\lambda_1, \lambda_2) | \lambda_1 \geq 0, \lambda_2 \geq 0, \lambda_1 + \lambda_2 = 1\}$ and $d = [-1, -1]$. The problem to be solved is

$$\min\{d^T x = -x_1 - x_2 | x \in WE(F,X)\}.$$

Compute

$$G(\lambda) = \begin{bmatrix} -\lambda_1 + 8\lambda_2 & -6\lambda_2 \\ 9\lambda_2 & -2\lambda_1 - 4\lambda_2 \\ -2\lambda_1 + 7\lambda_2 & 6\lambda_1 \\ -7\lambda_1 + 2\lambda_2 & 6\lambda_1 \end{bmatrix}, \quad b(\lambda) = \begin{bmatrix} -6\lambda_2 \\ 18\lambda_2 \\ 42\lambda_2 \\ 12\lambda_2 \end{bmatrix}.$$

Hence

$$H(\lambda) = \begin{bmatrix} 1 & -2 \\ -1 & -2 \\ -1 & 1 \\ 1 & 0 \\ \lambda_1 + 8\lambda_2 & -6\lambda_2 \\ 9\lambda_2 & -2\lambda_1 - 4\lambda_2 \\ -2\lambda_1 + 7\lambda_2 & 6\lambda_1 \\ -7\lambda_1 + 2\lambda_2 & 6\lambda_1 \end{bmatrix}, \quad h(\lambda) = \begin{bmatrix} 2 \\ -2 \\ 1 \\ 6 \\ -6\lambda_2 \\ 18\lambda_2 \\ 42\lambda_2 \\ 12\lambda_2 \end{bmatrix},$$

$H^T(\lambda^1) =$

$$\begin{bmatrix} +1 & -1 & -1 & -1 & (-\lambda_1^1 + 8\lambda_2^1) & 9\lambda_2^1 & (-2\lambda_1^1 + 7\lambda_2^1) & (-7\lambda_1^1 + 2\lambda_2^1) \\ -2 & -2 & +1 & 0 & -6\lambda_2^1 & (2\lambda_1^1 - 4\lambda_2^1) & 6\lambda_1^1 & 6\lambda_1^1 \end{bmatrix},$$

$H^T(\lambda^2) =$

$$\begin{bmatrix} +1 & -1 & -1 & +1 & (-\lambda_1^2 + 8\lambda_2^2) & 9\lambda_2^2 & (-2\lambda_1^2 + 7\lambda_2^2) & (-7\lambda_1^2 + 2\lambda_2^2) \\ -2 & -2 & +1 & 0 & -6\lambda_2^2 & (2\lambda_1^2 - 4\lambda_2^2) & 6\lambda_1^2 & 6\lambda_1^2 \end{bmatrix}.$$

For a fixed $\lambda = (\lambda_1, \lambda_2) \in V(S_k)$, the problems of determining upper and lower bounds are

$$\alpha = \begin{cases} \min\{-x_1 - x_2\}, \ \text{s.t.} \\ \begin{bmatrix} 1 & -2 \\ -1 & -2 \\ -1 & 1 \\ 1 & 0 \\ -\lambda_1 + 8\lambda_2 & -6\lambda_2 \\ 9\lambda_2 & +2\lambda_1 - 4\lambda_2 \\ -2\lambda_1 + 7\lambda_2 & 6\lambda_1 \\ -7\lambda_1 + 2\lambda_2 & 6\lambda_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \leq \quad \begin{bmatrix} 2 \\ -2 \\ 1 \\ 6 \\ -6\lambda_2 \\ 18\lambda_2 \\ 42\lambda_2 \\ 12\lambda_2 \end{bmatrix} \end{cases},$$

$\beta(S_k) =$

$$\min_{\beta(\lambda^1), \beta(\lambda^2)} \begin{cases} \beta(\lambda^1) := \max -[4u_1 - 2u_2 + u_3 + 6u_4 - 6\lambda_2^1 u_5 + 18\lambda_2^1 u_6 + 42\lambda_2^1 u_7 + 12\lambda_2^1 u_8] \\ \beta(\lambda^2) := \max -[4u_1 - 2u_2 + u_3 + 6u_4 - 6\lambda_2^2 u_5 + 18\lambda_2^2 u_6 + 42\lambda_2^2 u_7 + 12\lambda_2^2 u_8] \\ \text{subject to} \\ 2u_1 - u_2 - u_3 + u_4 + (-\lambda_1^1 + 8\lambda_2^1)u_5 + 9\lambda_2^1 u_6 + \\ (-2\lambda_1^1 + 7\lambda_2^1)u_7 + (-7\lambda_1^1 + 2\lambda_2^1)u_8 - 1 \geq 0 \\ -4u_1 - 2u_2 + u_3 + 0u_4 - 6\lambda_2^1 u_5 + (2\lambda_1^1 - 4\lambda_2^1)u_6 + 6\lambda_1^1 u_7 + 6\lambda_1^1 u_8 - 1 \geq 0 \\ 2u_1 - u_2 - u_3 + u_4 + (-\lambda_1^2 + 8\lambda_2^2)u_5 + 9\lambda_2^2 u_6 + \\ (-2\lambda_1^2 + 7\lambda_2^2)u_7 + (-7\lambda_1^2 + 2\lambda_2^2)u_8 - 1 \geq 0 \\ -4u_1 - 2u_2 + u_3 + 0u_4 - 6\lambda_2^2 u_5 + (2\lambda_1^2 - 4\lambda_2^2)u_6 + 6\lambda_1^2 u_7 + 6\lambda_1^2 u_8 - 1 \geq 0 \\ u_1 \geq 0, u_2 \geq 0, u_3 \geq 0, u_4 \geq 0, \quad u_5 \geq 0, u_6 \geq 0, u_7 \geq 0, u_8 \geq 0. \end{cases}$$

Intialization. Choose $\varepsilon = 0.05$ and set $S_0 = [(0,1),(1,0)]$. Then

$$\beta(S_0) = \min\big\{\beta((0,1)), \beta((1,0))\big\} = \min\{-13,-13\} = -13.$$

Take $\lambda = (1,0) \in S_0$, $\alpha_0 = [-1,-1][2,0]^T = -2, \beta_0 = \beta(S_0) = -13$. Since $\alpha_0 - \beta_0 = -2+13 = 11 \geq 0.05(|-2|+1) = 0.15$, we have $\Gamma_0 = \{S_0\}$. Set $k := 0$ and go to Iteration k.

Iteration k. The following table presents the result attained by the maple V release 4 package after 27 Iterations.

In order to obtain a preliminary evaluation of the performance of Algorithm RLB, we have written a computer code by Maple V that implements this algorithm. We have used the code to solve thirty one randomly generated problems on a Pentium II personal computer. The computed results are reported in Table 2. In this table, m_1 stands for the number of newly added constraints.

Table 1

Iteration k	β_k	Incum -bent $\alpha k+1$	x^{k+1}	Selected partition set	Number of stored partition sets
0	-13	-2	(2, 0)	$[(0,1),(1,0)]$	2
1	-13	”	”	$[(\frac{1}{2},\frac{1}{2}),(1,0)]$	3
2	-13	-3.75	$(\frac{3}{2},\frac{9}{4})$	$[(0,1),(\frac{1}{2},\frac{1}{2})]$	4
3	-13	”	”	$[(0,1),(\frac{1}{4},\frac{3}{4})]$	4
4	-13	”	”	$[(\frac{1}{4},\frac{3}{4}),(\frac{1}{2},\frac{1}{2})]$	3
5	-12.5	”	”	$[(\frac{1}{2},\frac{1}{2}),(\frac{3}{4},\frac{1}{4})]$	4
6	-11	”	”	$[(\frac{3}{4},\frac{1}{4}),(1,0)]$	5
7	-10.45	”	”	$[(\frac{3}{8},\frac{5}{8}),(\frac{1}{2},\frac{1}{2})]$	6
8	-9.8	”	”	$[(\frac{1}{4},\frac{3}{4}),(\frac{3}{8},\frac{5}{8})]$	5
9	-9.26	”	”	$[(\frac{1}{2},\frac{1}{2}),(\frac{5}{8},\frac{3}{8})]$	4
10	-7.8	”	”	$[(\frac{1}{2},\frac{1}{2}),(\frac{9}{16},\frac{7}{16})]$	5

Table 1 (continue)

Iteration k	β_k	Incumbent $\alpha k + 1$	x^{k+1}	Selected partition set	Number of stored partition sets
11	-7.4	”	”	$[(\frac{5}{8},\frac{3}{8}),(\frac{3}{4},\frac{1}{4})]$	4
12	-7.4	-4.69	$(\frac{15}{8},\frac{45}{16})$	$[(\frac{7}{8},\frac{1}{8}),(1,0)]$	5
13	-7.18	”	”	$[(\frac{9}{16},\frac{7}{16}),(\frac{5}{8},\frac{3}{8})]$	4
14	-6.27	”	”	$[(\frac{19}{32},\frac{13}{32}),(\frac{5}{8},\frac{3}{8})]$	5
15	-6.23	”	”	$[(\frac{9}{16},\frac{7}{16}),(\frac{19}{32},\frac{13}{32})]$	5
16	-5.91	”	”	$[(\frac{5}{8},\frac{3}{8}),(\frac{21}{32},\frac{11}{32})]$	3
17	-5.68	”	”	$[(\frac{5}{8},\frac{3}{8}),(\frac{41}{64},\frac{23}{64})]$	3
18	-5.68	”	”	$[(\frac{39}{64},\frac{25}{64}),(\frac{5}{8},\frac{3}{8})]$	4
19	-5.59	”	”	$[(\frac{3}{4},\frac{1}{4}),(\frac{7}{8},\frac{1}{8})]$	3
20	-5.55	”	”	$[(\frac{11}{16},\frac{5}{16}),(\frac{3}{4},\frac{1}{4})]$	3
21	-5.46	”	”	$[(\frac{21}{32},\frac{11}{32}),(\frac{11}{16},\frac{5}{16})]$	4
22	-5.33	”	”	$[(\frac{5}{8},\frac{3}{8}),(\frac{11}{16},\frac{5}{16})]$	2
23	-5.31	-4.92	$(\frac{63}{32},\frac{189}{64})$	$[(\frac{83}{128},\frac{45}{128}),(\frac{21}{32},\frac{11}{32})]$	3
24	-5.30	”	”	$[(\frac{41}{64},\frac{23}{64}),(\frac{83}{128},\frac{45}{128})]$	2
25	-5.30	”	”	$[(\frac{21}{32},\frac{11}{32}),(\frac{43}{64},\frac{21}{64})]$	1
26	-5.22	”	”	$[(\frac{21}{32},\frac{11}{32}),(\frac{85}{128},\frac{43}{128})]$	1
27	-5.17	”	”	$[(\frac{21}{32},\frac{11}{32}),(\frac{169}{256},\frac{87}{256})]$	0

An ε-global optimal solution is $x = (63/32, 189/64)$.

The ε-optimal value is $f_{*\varepsilon} = -4.92$.

Table 2

Pron -lems	p	n	m	ε	f_*	m_1	Iter.	CPU time(s)
1	2	2	6	0.005	-4.995	4	101	189
2	”	9	0.01	-99.5	3	8	56.5	
3	”	”	”	”	-38.8	6	10	54
4	3	6	7	0.01	-120	1	1	2
5	”	”	”	”	-5.4	2	2	16
6	”	”	”	”	-35.2	5	12	90
7	3	9	10	0.01	-179	4	4	12
8	”	”	”	”	-110	4	45	390
9	”	”	”	”	-47	2	6	32.5
10	”	”	”	”	-90	1	1	2
11	”	”	”	”	-180	2	2	5
12	3	12	13	0.01	-59.3	5	84	753
13	”	”	”	”	-186	4	6	49
14	”	”	”	”	-192	2	2	4
15	”	”	”	”	0	5	38	362
16	”	”	”	”	-189	3	50	371
17	4	9	10	0.01	-144	6	164	2001
18	”	”	”	”	-141	6	76	862
19	”	”	”	”	-92	2	2	6
20	”	”	”	”	-144	3	9	90
21	”	”	”	”	-22.5	5	10	667
22	4	12	13	0.01	-66	5	25	416
23	”	”	”	”	0	5	763	11078
24	”	”	”	”	-4026	5	134	1997
25	”	”	”	”	-62	3	3	17
26	”	”	”	”	-61	5	5	47
27	4	16	17	0.01	-246	4	6	116
28	”	”	”	”	-255	4	20	359
29	”	”	”	”	-249	2	2	9
30	”	”	”	”	-246	1	1	4
31	”	”	”	”	-32	2	2	10

References

[1] F. A. Al-Khayyal and J. E. Falk, *Jointly constrained biconvex programming*, Math. Oper. Res. **8** (1983), 273-286.

[2] L. T. H. An, D. T. Pham, and L. D. Muu, *Numerical solution for optimization over the efficient set by d.c. optimization algorithms*, Oper. Res. Lett. **19** (1996), 117-128.

[3] L. T. H. An, D. T. Pham, and L. D. Muu, *An d.c. optimization approach to optimizing over the efficient and weakly efficient sets*, To appear in J. Optimization Theory Appl.

[4] C. Berge, *Topological Spaces*, Macmillan, New York 1968.

[5] H. Benson, *An algorithm for optimizing over the weakly-efficient set*, Euro. J. Oper. Res. **25** (1986), 192-199.

[6] S. Bolintineanu, *Optimality conditions for minimization over the (weakly or properly) efficient set*, J. Math. Anal. Appl. **173** (1993), 523-541.

[7] S. Bolintineanu and M. E. Maghri, *Penalisation dans l'optimisation sur l'ensemble faiblement efficient*, Recherche Operationnelle **31** (1997), 295-312.

[8] E. U. Choo and D. R. Atkins, *Connectedness in multiple linear fractional programming*, Management Sci. **29** (1983), 250-255.

[9] E. U. Choo and D. R. Atkins, *Bicriteria Linear Fractional Programming*, J. Optimization Theory Appl. **36** (1982), 203-220.

[10] M. Dür and R. Horst, *Lagrange duality and partitions techniques in nonconvex global optimization*. J. Optimization Theory Appl. **95** (1990), 347-369.

[11] R. Horst and H. Tuy, *Global Optimization (Deterministic Approaches)*, third edition, Springer, Berlin 1996.

[12] L. T. Luc and L. D. Muu, *Global optimization approach to optimization over the efficient set*, Proceeding of 8th French-German Conference on Optimization, 213-221, Springer Verlag, Berlin 1997.

[13] C. Malivert, *Multicriteria fractional optimization*. In: Proceedings of the 2nd Catalans Days on Appl. Math., Presses Universitaires de Perpignan, pp. 189-198, 1995.

[14] L. D. Muu and W. Oettli, *An algorithm for indefinite quadratic programming with convex constraints*, Oper. Res. Lett. **10** (1989), 323-327.

[15] L. D. Muu and W. Oettli, *A combimed branch-and-bound and cutting plane method for solving a certain class of nonconvex optimization problems*, J. Global Optimization **3** (1993), 377-391.

[16] J. Philip J, *Algorithms for the vector maximization problem*, Math. Program. **2** (1972), 207-228.

[17] S. Schaible, *Fractional programming: applications and algorithms*, Euro. Oper. Res. **7** (1981), 111-120.

[18] R. E. Steuer, *Multiple Criteria Optimization: Theory, Computation, and Application*, John Willey and Sons, New York, 1986.

[19] P. T. Thach, H. Konno, and D. Yokota, *Dual approach to optimization on the set of Pareto-optimal solutions*. J. Optimization Theory Appl. **88** (1996), 689-707.

[20] T. V. Thieu, *Relationship between bilinear programming and concave programming*, Acta Math. Vietnam. **2** (1980), 106-113.

[21] T. V. Thieu, *A note on the solution of bilinear programming problem by reduction to concave minimization*, Math. Program. **41** (1988), 249-260.

[22] H. Q. Tuyen and L. D. Muu, *Biconvex programming approach to optimization over the weakly efficient set of a multiple objective affine fractional problem*, Oper. Res. Lett. **28** (2001), 81-92.

[23] S. Yamada, T. Tanino, and M. Inuiguchi, *An inner approximation method for optimization over the weakly efficient set*. To appear in J. Global Optimization.

Hanoi Institute of Mathematics,
P.O. Box 631 Boho, 10000 Hanoi, Vietnam