*Vietnam Journal of Mathematics* **28**:3 (2000) 217-226

Vietnam Journal of MATHEMATICS

© Springer-Verlag 2000

# Bisection Search Algorithm for Optimizing Over the Efficient Set

**Thai Quynh Phong**[1] **and Hoang Quang Tuyen**[2]

[1] *Teachers' College, University of Da Nang, Da Nang, Vietnam*

[2] *Institute of Mathematics, P. O. Box 631, Bo Ho, Hanoi, Vietnam*

Received May 30, 1999

Revised February 24, 2000

**Abstract.** We are concerned with the problem of maximizing a linear function over the efficient set related to a multiple-objective linear programming problem. Mathematically, this problem is classified as a global optimization problem. We propose an implementable algorithm, which is hoped to be efficient when the number of criteria is small relative to the number of variables.

## 1. Introduction

The following multiple-objective linear programming (MOLP for short) model is very important in decision-making:

$$\text{(MOLP)} \qquad \text{VMAX } Cx \text{ subject to } x \in X,$$

where $C$ is a $p \times n$-matrix and

$$X = \{x \in \mathbb{R}^n : Ax \leq b, x \geq 0\}$$

with $A$ being an $m \times n$-matrix and $b \in \mathbb{R}^m$. We assume that $p \geq 2$.

We recall that an efficient solution for problem (MOLP) is a point $x^0 \in X$ such that whenever $Cx \geq Cx^0$ for some $x$, then $Cx = Cx^0$. By $X_E$ we denote the set of all efficient solutions for (MOLP). The problem of main concern in this paper is to find $x^*$ which maximizes a quantity $\langle d, x\rangle$ over $X_E$, i.e.,

$$\text{(P)} \qquad \max\{\langle d, x\rangle : x \in X_E\}$$

for some $d \in \mathbb{R}^n$. This problem has received in recent years increasing attention from researchers since it has many applications in multiple decision-making (see, e.g., [2-4, 8, 14,...]). The interested reader is referred to [2] for discussions on this subject.

Since $X_E$ is non-convex, problem (P) is classified as a global optimization problem. However, by contrast with other global optimization problems having non-convex feasible domain, a feasible point of problem (P) can be easily found by solving a linear program. More precisely, for sufficiently large $M$, there exists a simplex

$$\Lambda = \{\lambda \in \mathbb{R}^p : \lambda \geq e,\ \langle e, \lambda\rangle \leq M\} \tag{1}$$

($e = (1, ..., 1) \in \mathbb{R}^p$) such that $x^0 \in X_E$ if and only if $x^0$ is an optimal solution to the problem

$$(\mathrm{P}_\lambda) \qquad \max\{\lambda^T Cx \colon x \in X\}$$

for some $\lambda \in \Lambda$. Using this fact, Philip in [11] briefly outlined a cutting plane procedure for solving (P). This procedure requires finding all efficient extreme points which lie on this cutting plane in the newly created polyhedron. However, it is not clear how this main step can be implemented.

Recently, two implementable methods were developed by Benson in [2, 3], which are based on the fact that problem (P) is equivalent to the following infinitely constrained non-convex optimization problem, denoted by (Q):

$$\max\{\langle d, x\rangle : \lambda^T Cx \geq \lambda^T Cy,\ \forall y \in X, x \in X, \lambda \in \Lambda\}.$$

The only drawback is that in both algorithms, at each step we have to deal with a bilinear programming problem. Alternatively, Muu in [9] proposed to formulate problem (Q) by a convex-concave programming problem

$$(\widetilde{\mathrm{Q}}) \qquad \max\{\langle d, x\rangle : g(\lambda) - \lambda^T Cx \leq 0,\ x \in X,\ \lambda \in \Lambda\},$$

where

$$g(\lambda) = \max\{\lambda^T Cy : y \in X\} \tag{2}$$

is a piecewise-linear convex function. A branch-and-bound algorithm was then developed, where the simplicial subdivision process is performed in $\mathbb{R}^p$, so Muu's method is hoped to be efficient if the number of criteria is relatively small with regard to the number of variables (i.e., $n \gg p$). Some cutting plane methods were proposed in [6] and [7] for solving (P) in the case when $X$ may be unbounded and when the objective function can be concave, respectively. Interesting dimension reduction algorithms were described in [10] that involve only $k$ variables, where $k$ is the rank of the matrix $C$. Recently, another approach using so-called *d.c. optimization algorithms* were studied in [1]. Despite the efficiency of reported numerical experiences, it is not clear that a global optimal solution must be found by these methods.

The purpose of this paper is to develop an algorithm for solving (P) which is hoped to be efficient in the case $n \gg p$. The idea underlying the algorithm is to use *bisection search scheme* for locating the optimal value of the considered problem (cf. [12]). More precisely, starting from an interval $[\gamma_0, \beta_0]$ containing the optimal value $d^*$ of $(P)$ we shall reduce this interval each time by half by solving the following problem:

$$(\mathrm{P}_k) \qquad \text{Find } x \in X_E \text{ such that } \langle d, x\rangle \geq \alpha_k,$$

where $\alpha_k = (\gamma_k + \beta_k)/2$. This procedure repeats until $\beta_k - \gamma_k \leq \varepsilon$. It is easily seen that this procedure is finite.

Bisection search has been successfully applied by the authors to elaborate algorithms for solving the problems whose feasible set is defined by a system of d.c. inequalities [12], in particular by a system of quadratic functions. The idea of bisection search for problem (P) is not new. The interested reader is referred to two algorithms developed by Benson in [4] and [5] for optimizing over the *weakly efficient* set and for the case where $d$ is linearly dependent on the rows of $C$ respectively. It should be noted that in both mentioned methods, the main computational burden involves concave minimization (or the same, convex maximization) problems whose objective functions are defined implicitly.

In our method, solving $(\mathrm{P}_k)$ at each step is reduced to check the feasibility of the set

$$X_E^{\alpha_k} = \{x \in X_E : \langle d, x\rangle \geq \alpha_k\}.$$

Our main contribution is to develop an efficient outer approximation procedure for this problem, which can be easily incorporated into the bisection scheme. As a result we obtain a bisection search algorithm which does not require solving completely $(\mathrm{P}_k)$ at each step. We first show that $X_E^{\alpha_k}$ can be identified with a subset of $\Lambda$ which can be viewed as the projection of a difference of two convex sets (i.e., d.c. set). Thus, solving $(\mathrm{P}_k)$ amounts to finding a point of this d.c. set and the outer approximation algorithm for solving d.c. feasibility problem, developed in our early works (cf. e.g. [12]) can be applied. The advantage of this method is that it works in the $p$-dimensional space so it is hoped that it will be efficient if the number of criteria is small relative to the number of variables.

In the next section we shall describe the outer approximation procedure for solving subproplems $(\mathrm{P}_k)$. The bisection search algorithm for solving (P) is discussed in Sec. 3. Finally in Sec. 4, we illustrate the algorithm by a small example.

## 2. Outer Approximation Procedure for Solving $(\mathrm{P}_k)$

Recall that $x \in X_E$ if there is a $\lambda \in \Lambda$ such that

$$\lambda^T Cx = \max\{\lambda^T Cy : y \in X\}$$

so, for a given $\alpha$, we have

$$\begin{aligned} X_E^{\alpha} &= \{x \in X_E : \langle d, x\rangle \geq \alpha\} \\ &= \{x \in X : \langle d, x\rangle \geq \alpha,\ \lambda \in \Lambda,\ \lambda^T Cx \geq \lambda^T Cy,\ \forall y \in X\}. \end{aligned}$$

This section is devoted to the problem of finding a point $x \in X_E^\alpha$. Using function $g(\cdot)$ defined via (2), we can rewrite $X_E^\alpha$ as

$$X_E^\alpha = \{x \in X : \langle d, x\rangle \geq \alpha,\, g(\lambda) \leq \lambda^T Cx,\, \lambda \in \Lambda\}. \tag{3}$$

Let us introduce the function

$$h_\alpha(\lambda) = \max\{\lambda^T Cx : x \in X,\, \langle d, x\rangle \geq \alpha\}. \tag{4}$$

It is easy to see the following properties of the function $h_\alpha(\lambda)$.

**Proposition 1.** *The function $h_\alpha(\lambda)$ is a piecewise linear convex function in $\mathbb{R}^p$ which is decreasing in $\alpha$, i.e.,*

$$\alpha < \alpha' \Rightarrow h_\alpha(\lambda) \geq h_{\alpha'}(\lambda),\, \forall \lambda \in \mathbb{R}^p.$$

*Moreover, if $d_{\min}$ and $d_{\max}$ are the minimum and maximum of $\langle d, x\rangle$ on $X$, respectively, then we have*
- *$h_\alpha(\lambda) = g(\lambda)$ for all $\alpha < d_{\min}$,*
- *$h_\alpha(\lambda) \leq g(\lambda)$ if $d_{\min} \leq \alpha \leq d_{\max}$.*

By setting

$$\Omega = \{(\lambda, t) \in \mathbb{R}^p \times \mathbb{R} : \lambda \in \Lambda,\, g(\lambda) - t \leq 0\}, \tag{5}$$

$$\Omega_\alpha = \{(\lambda, t) \in \mathbb{R}^p \times \mathbb{R} : t - h_\alpha(\lambda) > 0\}, \tag{6}$$

we obtain two convex sets $\Omega$, $\Omega_\alpha$ such that $\Omega \subset \overline{\Omega}_\alpha$ and the following:

**Proposition 2.** $X_E^\alpha \neq \emptyset \Leftrightarrow \Omega \setminus \Omega_\alpha \neq \emptyset$.

*Proof.* Suppose that $x \in X_E^\alpha$. Then there exists a $\lambda \in \Lambda$ such that $g(\lambda) \leq \lambda^T Cx$. By taking $t = g(\lambda)$ we have

$$t \leq \lambda^T Cx,\, x \in X,\, \langle d, x\rangle \geq \alpha$$

that implies

$$t \leq \max\{\lambda^T Cx : x \in X,\, \langle d, x\rangle \geq \alpha\} = h_\alpha(\lambda).$$

Thus $(\lambda, t) \in \Omega \setminus \Omega_\alpha$, i.e., $\Omega \setminus \Omega_\alpha \neq \emptyset$.

Conversely, assume that $(\lambda, t) \in \Omega \setminus \Omega_\alpha$. By definition we have

$$g(\lambda) \leq t,\, \lambda \in \Lambda \text{ and } t \leq h_\alpha(\lambda). \tag{7}$$

Denoting by $x$ the maximizer of the problem (4), we have

$$t \leq h_\alpha(\lambda) = \lambda^T Cx,\, \langle d, x\rangle \geq \alpha,\, x \in X,$$

and in view of (7),

$$g(\lambda) \leq \lambda^T Cx,\, \lambda \in \Lambda,\, \langle d, x\rangle \geq \alpha,\, x \in X,$$

which means $(x, \lambda) \in X_E^\alpha$, i.e., $X_E^\alpha \neq \emptyset$.

Thus, solving $(\mathrm{P}_k)$ is reduced to finding an element of the d.c. set $\Omega \setminus \Omega_\alpha$. The idea underlying our method is to construct a sequence of polyhedral convex sets $S_k$ approximating $\Omega$ more and more closely from outside in such a way that $u = (0,1)$ is a unique direction of recession of $S_k$, i.e., the recession cone of $S_k$ has the form

$$O^+ S_k = \{\theta u : u = (0,1),\ \theta \geq 0\}.$$

Let $V_k$ be the vertex set of $S_k$ and define

$$W_k = \{(\lambda, t) \in V_k : t - h_\alpha(\lambda) \leq 0\}.$$

**Lemma 1.** *If $W_k = \emptyset$, then $\Omega \setminus \Omega_\alpha = \emptyset$.*

*Proof.* Since $S_k$ has only one recession direction along axis $t$, from [13], one has that

$$\min\{t - h_\alpha(\lambda) : (\lambda, t) \in S_k\} = \min\{t - h_\alpha(\lambda) : (\lambda, t) \in V_k\}.$$

Therefore, $W_k = 0$ implies that

$$\min\{t - h_\alpha(\lambda) : (\lambda, t) \in V_k\} > 0$$

or

$$\min\{t - h_\alpha(\lambda) : (\lambda, t) \in S_k\} > 0.$$

This means, for all $(\lambda, t) \in \Omega$, we have $t - h_\alpha(\lambda) > 0$ or $\Omega \setminus \Omega_\alpha = \emptyset$.

Thus, if $W_k = \emptyset$, then $X_E^\alpha = \emptyset$. Otherwise we can choose a $(\lambda^k, t_k) \in W_k$.

- If $(\lambda^k, t_k) \in \Omega$, i.e., $\lambda^k \in \Lambda$ and $g(\lambda^k) \leq t_k$, then $(\lambda^k, t_k) \in \Omega \setminus \Omega_\alpha$ and an optimal solution $x^k$ of the linear program
$$\max\{(\lambda^k)^T Cx : x \in X, \langle d, x\rangle \leq \alpha\}$$
will belong to $X_E^\alpha$.
- Otherwise denote by $y^k$ an optimal solution of the linear program
$$\max\{(\lambda^k)^T Cy : y \in X\}.$$
Then the hyperplane $\{(\lambda, t) : \lambda^T Cy^k - t = 0\}$ will separate strictly $(\lambda^k, t^k)$ from $\Omega$. We define $S_{k+1}$ as a subset of $S_k$ defined by this hyperplane.

By repeating the above procedure we generate a sequence of polyhedral sets

$$\Omega \subset \cdots \subset S_k \subset \cdots \subset S_1 \subset S_0$$

along with a sequence of points $(x^0, t_0)$, $(x^1, t_1)$,..., $(x^k, t_k)$,... such that $(x^k, t_k) \in S_k$.

The choice of $(\lambda^k, t_k)$ can be done at least by one of the following methods:

(1) For each $\lambda \in \Lambda$, denote by $x_\lambda$ an optimal solution of the problem $(P_\lambda)$. Then we can take

$$(\lambda^k, t_k) \in \arg\max\{\langle d, x_\lambda\rangle : (\lambda, t) \in W_k\}.$$

This choice may be useful because we wish to maximize $\langle d, x\rangle$ over $X_E$.

(2) $(\lambda^k, t_k)$ solves the following problem:

$$\theta = \min\{t - h_\alpha(\lambda) : (\lambda, t) \in V_k\}.$$

It should be noted that if $\theta > 0$, then $W_k = \emptyset$.

The above discussion leads to the following algorithm for solving $(\mathrm{P}_k)$.

**OA Procedure**

*Initialization.* Take $v \in X : Cv \neq 0$. Set

$$S_0 = \{(\lambda, t) : \lambda \in \Lambda,\ \lambda^T Cv \le t\}$$

and denote by $V_0$ the vertex set of $S_0$. Set $k = 0$.

*Iteration* $k = 0, 1, \ldots$

(1) Solve the problem

$$\theta = \min\{t - h_\alpha(\lambda) : (\lambda, t) \in V_k\}$$

for $\theta, \lambda^k, t_k$ where $h_\alpha(\lambda)$ is defined via (4). If $\theta > 0$, then stop: $X_E^\alpha = \emptyset$ or $(P_k)$ has no solution. Otherwise go to 2.

(2) Solve the linear program

$$\max\{(\lambda^k)^T Cy : y \in X\}$$

to obtain an optimal solution $y^k$ and the optimal value $\eta$. If $\eta \le t_k$, then stop: $(\lambda^k, t_k) \in \Omega \setminus \Omega_\alpha$ and $X_E^\alpha \neq \emptyset$. Otherwise go to 3.

3) Set

$$S_{k+1} = S_k \cap \{(\lambda, t) : \lambda^T Cy^k \le t\}.$$

Compute the vertex set $V_k$ of $S_k$ and go to 1.

Since $g(\lambda)$ is a piecewise linear convex function, $\Omega$ is a polyhedral convex set in $\mathbb{R}^{p+1}$. By construction, $S_k$ is a polyhedral convex set approximating $\Omega$ from outside such that $S_{k+1}$ is obtained from $S_k$ by adding a cutting plane $t = \lambda^T Cy^k$. Note that $y^k$ is a vertex of the polytope $X$ whose number is finite, so Step 3 in the above algorithm cannot repeat infinitely many times. We have the following

**Proposition 3.** *OA procedure terminates after a finite number of steps either yielding a solution to $(P_k)$ or showing that it has no solution.*

*Comment.*

1. Denote by $x^k$ a solution of problem (4). Then if $\eta \le t_k$ in step 2, we have $x^k \in X_E^\alpha$. Note that, in general, $x^k$ needs not be a vertex of $X$. The point $y^k \in X_E \cap X_{ex}$ but need not belong to $X_E^\alpha$.
2. It is not difficult to see that the above procedure is in fact an outer approximation algorithm for minimizing a concave function $t - h_\alpha(\lambda)$ over a convex set $\{(\lambda, t) : \lambda \in \Lambda,\ g(\lambda) - t \le 0\}$ which is equivalent to the following problem:

$$\min\{g(\lambda) - h_\alpha(\lambda) : \lambda \in \Lambda\}.$$

The advantage of the above procedure is that it can be easily incorporated into the bisection search scheme. As a result we shall obtain an algorithm for solving problem (P) without requiring solving completely a subproblem ($\mathrm{P}_k$) at each iteration.

## 3. Bisection Search Algorithm

Now we are in position to describe the algorithm for solving (P).

### Algorithm

*Initialization.* Find any point $x^0 \in X_E \cap X_{ex}$. Set $\gamma_0 = \langle d, x^0\rangle$ and

$$\beta = d_{\max} = \max\{\langle d, x\rangle : x \in X\}.$$

Define

$$S_0 = \{(\lambda, t) : \lambda \in \Lambda,\ \lambda^T C\overline{x}^0 \le t\}$$

and compute the vertex set $V_0$ of $S_0$. Chose $\varepsilon > 0$ and set $x^{\mathrm{opt}} = x^0$, $k = 0$.

*Iteration* $k = 0, 1, \ldots$

<u>k1</u> If $\beta_k - \gamma_k \le \varepsilon$, then stop. Otherwise go to Step $k2$.
<u>k2</u> Let $\alpha_k = 1/2(\beta_k + \gamma_k)$ and solve the problem

$$\theta = \min\{t - h_{\alpha_k}(\lambda) : (\lambda, t) \in V_k\}$$

to obtain $\theta, \lambda^k, t_k$.
<u>k3</u> If $\theta > 0$, then set

$$S_{k+1} = S_k,\ V_{k+1} = V_k,\ \beta_{k+1} = \alpha_k,\ \gamma_{k+1} = \gamma_k$$

and $k = k + 1$. Go to Step $k1$.
<u>k4</u> Otherwise, find and optimal solution $y^k$ and the optimal value $\eta$ to the linear program

$$\max\{(\lambda^k)^T Cy : y \in X\}.$$

If $\eta \le t_k$, find the best efficient point $x^{\mathrm{opt}}$ and set

$$S_{k+1} = S_k,\ V_{k+1} = V_k,\ \beta_{k+1} = \alpha_k,\ \gamma_{k+1} = \langle d, x^{\mathrm{opt}}\rangle.$$

Set $k = k + 1$ and go to Step $k1$.
<u>k5</u> If $\eta > t_k$ then set

$$\beta_{k+1} = \beta_k,\ \gamma_{k+1} = \gamma_k,\ S_{k+1} = S_k \cap \{(\lambda, t) : \lambda^T Cy^k \le t\}.$$

Compute the vertex set $V_k$ of $S_k$. Set $k = k + 1$ and go to Step $k2$.

**Theorem 1.** *The Bisection Search Algorithm finds an approximately optimal solution for problem (P) after a finite number of iterations. If at each iteration we choose $x^{\mathrm{opt}}$ to be a vertex of $X$, then, for $\varepsilon$ sufficiently small, the obtained solution is an exact optimal solution.*

*Proof.* The finiteness of the Algorithm follows from the finiteness of the bisection scheme and the OA procedure for solving ($P_k$).

Suppose that $x^{\text{opt}}$ is chosen to be a vertex and $d^*$ is the optimal value. We want to prove that, for $\varepsilon$ sufficiently small, $x^{\text{opt}}$ is exact in the sense that $\langle d, x^{\text{opt}}\rangle = d^*$.

First we observe that if we denote by

$$V^* = \{x \in X_E \cap X_{ex} : \langle d, x\rangle = d^*\}, \tag{8}$$

then for all $y \in X_E \cap X_{ex}$ which do not belong $V^*$ we must have $d^* - \langle d, y\rangle \geq \delta$ for some $\delta > 0$. In fact, from (8) we have

$$d^* > \overline{d} = \max\{\langle d, x\rangle : x \in \operatorname{conv}[(X_E \cap X_{ex}) \setminus V^*]\}$$

that means for all $y \in (X_E \cap X_{ex}) \setminus V^*$ we have

$$d^* - \langle d, y\rangle \geq d^* - \overline{d} = \delta > 0.$$

It is easy to see that, for $\varepsilon < \delta$, if the algorithm terminates at Step $k1$, then there does not exist any $y \in (X_E \cap X_{ex}) \setminus V^*$ such that $\langle d, y\rangle \in [\gamma_k, \beta_k]$. Thus we must have $\gamma_k = x^{\text{opt}} = d^*$ that completes the proof. ■

*Comment.* Usually in $k4$ we take $x^{\text{opt}}$ to be an optimal solution $x^k$ of the linear program

$$\max\{(\lambda^k)^T Cx : x \in X,\ \langle d, x\rangle \geq \alpha_k\}$$

which may not be a vertex of $X$. We can get an efficient point better than $x^k$ that is a vertex by using Local Optimum Search Procedure proposed by Benson in [4]. This procedure is based in the fact that $X_E$ is simply connected, i.e., if an efficient point $x$ belongs to some *efficient* edge $e$ of $X$, then $e \subset X_E$. If $x^k$ is not a vertex, then we can find an extreme point of the edge containing $x^k$ which is better than $x^k$, i.e., a point $y \in X_E \cap X_{ex}$ such that $\langle d, y\rangle \geq \langle d, x^k\rangle$. Further, for each efficient edge emanating from $y$, if there is an extreme point $y'$ such that $\langle d, y'\rangle > \langle d, y\rangle$, then we set $y = y'$. This process is continued until $y$ cannot be improved.

## 4. Example

Consider a small example. We want to maximize $x_1 - x_2 + x_3$ over the efficient set of the vector optimization problem

$$\max\{x_1 - x_3,\ x_2\}$$

subject to

$$x_1 + x_2 \leq 3,\ 0 \leq x_1 \leq 2,\ 0 \leq x_2 \leq 2,\ 0 \leq x_3 \leq 2.$$

Let us define

$$d = (1, -1, 1),\quad C = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

and

$$D = \{(x_1, x_2, x_3) : x_1 + x_2 \leq 3,\ 0 \leq x_1 \leq 2,\ 0 \leq x_2 \leq 2, 0 \leq x_3 \leq 2\}.$$

*Initialization.* We take

$$\Lambda = \{(\lambda_1, \lambda_2) : \lambda_1 + \lambda_2 \leq 10,\ \lambda_1, \lambda_2 \geq 0\}.$$

By solving

$$\max\{\langle d, x\rangle : x \in D\},$$

we get an upper bound $\beta_0 = 4$. For $\lambda^0 = (10, 10)$, we solve

$$\max\{(\lambda^0)^T Cx : x \in D\}$$

to obtain an efficient point $x^0 = (1, 2, 0)$. We set $\gamma_0 = \langle d, x^0\rangle = -1$, $x^{\text{opt}} = (1, 2, 0)$.

We define

$$S_0 = \{(\lambda, t) : \lambda \in \Lambda,\ \lambda_1 + \lambda_2 - t \leq 0\}$$

and compute the vertex set

$$V_0 = \{v^1 = (1, 1, 3),\ v^2 = (1, 9, 19),\ v^3 = (9, 1, 11)\}.$$

*Iteration 1.* Let $\alpha_0 = 1.5$. For $v^1 = (\lambda^1, t_1) = (1, 1, 3)$ we solve the linear program

$$\max\{\lambda^1 Cx : x \in D,\ \langle d, x\rangle \geq 1.5\}$$

to get $h_{\alpha 0}(\lambda^0) = 2.5$ so $t_1 - h_{\alpha 0}(\lambda^1) = 3 - 2.5 = 0.5$. Analogously for other vertices we compute $t_2 - h_{\alpha_0}(\lambda^2) = 4$ and $t_3 - h_{\alpha_0}(\lambda^3) = -7.5$. Since

$$\min\{t - h_{\alpha_0}(\lambda) : v = (\lambda, t) \in V_0 = -7.5 < 0\}$$

at $\lambda^3 = (9,\ 1)$, we solve

$$\max\{(\lambda^3)^T Cx : x \in D\}$$

to obtain $y^1 = (2, 1, 0)$ and construct a new constraint

$$l_1(\lambda, t) = 2\lambda_1 + \lambda_2 - t \leq 0.$$

Set $S_1 = S_0 \cap \{(\lambda, t) : l_1(\lambda, t) \leq 0\}$ and compute

$$V_1 = \{v^1 = (1,\ 1,\ 3),\ v^2 = (1, 9, 19),\ v^3 = (9,\ 1,\ 19),\ v^4 = (5, 5, 15)\}.$$

We set $\beta_1 = 4$, $\gamma_1 = -1$.

*Iteration 2.* Let $\alpha_1 = 1.5$. Analogously, we compute

$$\min\{t - h_{\alpha_1}(\lambda) : v = (\lambda, t) \in V_1\} = 0.5 > 0.$$

Since $0.5 > 0$, we set $S_2 = S_1$, $V_2 = V_1$, and $\beta_2 = 1.5$, $\gamma_2 = -1$.

With $\varepsilon = 0.1$, the algorithm terminates after 11 iterations proving that $x^{\text{opt}} = (1,\ 2,\ 0)$ is in fact the optimal solution of the considered problem.

*Acknowledgement.* The authors would like to thank the referees for their insightful comments and suggestions which have substantially improved the presentation of the revised paper.

## References

1. Le T. H. An, P. D. Tao, and L. D. Muu, Numerical solution for optimization over the efficient set by D.C. optimization algorithms, *OR Letters* **19** (1996) 117-128.
2. H. P. Benson, An all-linear programming relaxation algorithm for optimizing over the efficient set, *J. of Global Optimization* **1** (1991) 83-104.
3. H. P. Benson, A finite, non-adjacent extreme point search algorithm for optimization over the efficient set, *J. of Optimization Theory and Applications* **73** (1992) 64-73.
4. H. P. Benson, A bisection-extreme point search algorithm for optimization over the efficient set, *J. of Global Optimization* **3** (1993) 95-111.
5. H. P. Benson, Optimization over the efficient set, *J. Math. Anal. Appl.* **73** (1992) 73-84.
6. S. Bolintieanu, Minimization of a quasiconcave function over an efficient set, *Research Paper University of Melbourne, Australia* **15** (1990).
7. J. Fulop, A cutting plane algorithm for linear optimization over the efficient set, Computer and Automation Institute, Hungarian Academy of Sciences, 15, 1992 (preprint).
8. L. T. Luc and L. D. Muu, Global optimization approach to optimization over the efficient set, in: *Proceeding of 8th French–Germany Symposium on Optimization*, Springer-Verlag, 1997, pp. 213-221.
9. L. D. Muu, Methods for optimizing a linear function over the efficient set, Institute of Mathematics, Hanoi, 1991 (preprint).
10. L. D. Muu, Dimension reduction algorithm for optimizing over the efficient set, Institute of Mathematics, Hanoi, 1993 (preprint).
11. J. Phillip, Algorithms for the vector maximizing problem, *Math. Program.* **2** (1972) 207-209.
12. T. Q. Phong, Pham D. Tao, and T. L. H. An, A method for solving D.C. programming problem. Application to fuel mixture nonconvex optimization problem, *J. of Global Optimization* **6** (1995) 87-105.
13. R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, 1970.
14. P. T. Thach, H. Kono, and D. Yokota, Dual approach to optimization on the set of Pareto-optimal solutions, *J. of Optimization Theory and Applications* **88** (1996) 680-707.