ACTA MATHEMATICA VIETNAM
Vol. II No 2 (1986)

A METHOD FOR SOLVING REVERSE CONVEX PROGRAMMING PROBLEMS

NGUYEN DUC NGHIA and NGUYEN DUC HIEU

1. INTRODUCTION

In this paper we shall be concerned with the following problem:

$$\text{(P)} \qquad \text{Minimize} \qquad f(x) = \langle c, x \rangle \tag{1}$$

subject to

$$x \in D = \{x : h_i(x) \leqslant 0,\ i = 1, 2, \ldots, m\} \tag{2}$$

$$g(x) \leqslant 0, \tag{3}$$

where $h_i(x)$ $(i = 1, 2, \ldots m)$ and $-g(x)$ are real-valued convex functions defined throughout R^n, c and x are n-dimensional vectors. We shall assume that D is compact and has a nonempty interior.

This problem, often called the reverse convex programming problem, has in recent years attracted an increasing attention from researchers (see [2], [3], [5], [7], and the bibliography given in [8]). For the origin, the practical applications, and the theoretical interest of this problem, we refer the reader to [8] and [13].

In general, finding an exact optimal solution to problem (P) is computationally very expensive. Therefore, in this paper we present a finite algorithm for finding a vector $x(\varepsilon, \theta)$ satisfying

$$x(\varepsilon, \theta) \in D,\ g(x(\varepsilon, \theta)) \leqslant \theta,\ f(x(\varepsilon, \theta)) - f^* \leqslant \varepsilon,$$

where f^* denotes the optimal value of the problem (1) — (3). Such a vector will be called (ε, θ)-solution of problem (P). While in practice it is usually sufficient to have an (ε, θ)-solution with reasonably small $\varepsilon, \theta > 0$, the cost for finding it may often be much less than finding an exact optimal solution. The theoretical justification of the concept of (ε, θ)-solution is given by the following

THEOREM 1. *If $\bar{x}(\varepsilon, \theta)$ is an (ε, θ)-solution of problem (P) and $\bar{x}(\varepsilon, \theta) \to \bar{x}$ as $\varepsilon, \theta \to 0$ then $\bar{x}$ is an optimal solution.*

Proof. By hypothesis

$$\overline{x}(\varepsilon, \theta) \in D,\ g(\overline{x}(\varepsilon, \theta)) \leqslant 0,\ f(\overline{x}(\varepsilon, \theta)) - f^* \leqslant \varepsilon \tag{4}$$

Letting $\varepsilon, \theta \to 0$ we obtain from the continuity of $g(x)$ and $f(x)$: $g(\overline{x}) \leqslant 0$, $f(\overline{x}) - f^* \leqslant 0$. Since D is compact we also have $\overline{x} \in D$, and consequently, $f(\overline{x}) - f^* \geqslant 0$. Therefore, $f(\overline{x}) = f^*$, and so $\overline{x}$ is an optimal solution of (P). □

The paper is divided into several sections. After the Introduction (Section 1), a general conceptual scheme of the method is described in Section 2. By this scheme problem (P) is reduced to a sequence of concave minimization problems for the solution of which a convergent outer approximation method is given in Section 3. An implementable realization of the scheme is described in Section 4. In Section 5 the method is specialized to the case where D is a polytope. In Section 6 a two-dimensional example is presented to illustrate how the algorithm works in practice. Finally, in Section 7 some preliminary computational experience is reported.

2. GENERAL SCHEME.

Let us begin with a conceptual scheme for solving problem (P) which has been presented in [7]. This scheme can be considered as a variation of the general framework developed in [8] and reduces the problem of finding an ε-optimal solution of problem (P) to a finite number of concave minimization problems (by an ε-optimal solution we mean a vector x satisfying the constraints (2) and (3) and such that $f(x) \leqslant f^* + \varepsilon$).

Let:

$$x^0 \in \arg\min \{g(x) : x \in D\}$$

$$x^{-1} \in \arg\min \{ < c, x > : x \in D\}.$$

It is natural to assume that

$$g(x^0) \leqslant 0 < g(x^{-1}).$$

Indeed, if $g(x^0) > 0$ then the set of feasible solutions of (P) is empty, while $g(x^{-1}) \leqslant 0$ implies that x^{-1} is an optimal solution.

General scheme.

Initialization. Solve the convex problem

$$\min \{< c, x > : x \in D\}$$

obtaining x^{-1} ($g(x^{-1}) > 0$). Set $\gamma_0 = < c, x^{-1} >$ (γ_0 is a lower bound for f^*).

Solve the concave minimization problem

$$\min \{g(x) : x \in D\}$$

obtaining x^0 ($g(x^0) \leqslant 0$). Set $\beta_0 = < c, x^0 >$ (β_0 is an upper bound for f^*).

Select $\varepsilon > 0$.

Iteration t $(t = 0, 1, 2, \ldots)$

At the begining of this iteration we already have β_t, γ_t (upper and lower bounds for f^*) and x^t (current best solution), with $<c, x^t> = \beta_t$.

If $\beta_t - \gamma_t \leqslant \varepsilon$ then x^t is an ε-optimal solution.

Otherwise, solve the relaxed problem:

(P_t) $\quad \min \{g(x) : x \in D, <c, x> \leqslant \alpha_t\}$,

where $\alpha_t = \frac{1}{2}(\beta_t + \gamma_t)$. Let $\bar{x}^t$ be an optimal solution to this problem. Two cases can occur;

(i) If $g(\bar{x}^t) > 0$ then set $\beta_{t+1} = \beta_t$, $\gamma_{t+1} = \alpha_t$, $x^{t+1} = x^t$ and go to iteration $t+1$.

(ii) If $g(\bar{x}^t) \leqslant 0$ then set $\beta_{t+1} = <c, \bar{x}^t>$, $\gamma_{t+1} = \gamma_t$, $x^{t+1} = \bar{x}^t$ and go to iteration $t+1$.

THEOREM 2. *The above scheme finds an ε-optimal solution of problem (P) after at most*

$$t_\varepsilon = \max\{0, [\log_2 (M/\varepsilon)] + 1\}$$

iterations, where $M = <c, x^0> - <c, x^{-1}>$.

To prove this Theorem we need the following lemma.

LEMMA 1. *If* $\eta = \min\{g(x) : x \in D, <c, x> \leqslant \alpha\} > 0$ *then* $<c, x^*> > \alpha$, *where x^* denotes an optimal solution of (P).*

Proof. Assume the contrary, that $<c, x^*> \leqslant \alpha$. From the definition of η we have $g(x^*) \geqslant \eta > 0$. This contradicts the feasibility of x^*. □

Proof of Theorem 2. We first note that at every iteration $t = 0, 1, 2, \ldots$ β_t and γ_t are actually an upper and a lower bound for f^*. Indeed, if $g(\bar{x}^t) > 0$ then by Lemma 1, $\alpha_t < f^*$. If $g(\bar{x}^t) \leqslant 0$ then $\bar{x}^t$ satisfies the constraints (2)—(3) and hence $<c, \bar{x}^t> \geqslant f^*$. Therefore, if β_t, γ_t are upper and lower bounds for f^*, then this must be true also for β_{t+1}, γ_{t+1}. Since this is true for $t = 0$, this must be true for all $t = 0, 1, 2, \ldots$

Let us now show that:

$$<c, x^t> - \gamma_t \leqslant M/2^t \quad (t = 0, 1, 2, \ldots). \tag{5}$$

The inequality being obvious for $t=0$, let us assume that it holds for $t \leqslant k$ and prove it for $t = k+1$. Two cases can occur:

(i) If $g(\overline{x}^k) > 0$ then $< c, x^{k+1} > - \gamma_{k+1} = < c, x^k > - \alpha_k =$

$< c, x^k > - \frac{1}{2}(\beta_k + \gamma_k) = \frac{1}{2}(< c, x^k > - \gamma_k) \leqslant M/2^{k+1}$.

(ii) If $g(\overline{x}^k) \leqslant 0$ then $< c, x^{k+1} > - \gamma_{k+1} = < c, \overline{x}^k > - \gamma_k$

$\leqslant \alpha_k - \gamma_k = \frac{1}{2}(\beta_k + \gamma_k) - \gamma_k = \frac{1}{2}(\beta_k - \gamma_k) =$

$= (< c, x^k > - \gamma_k)/2 \leqslant M/2^{k+1}$.

Thus inequality (5) holds for all $t = 0, 1, 2, \ldots$ Since $< c, x^* > > \gamma_t$, we have

$$0 \leqslant < c, x^t > - < c, x^* > \leqslant < c, x^t > - \gamma_t \leqslant M/2^t.$$

Therefore, after t_ε iterations we must obtain a feasible solution x^{t_ε}, which satisfies

$$0 \leqslant < c, x^{t_\varepsilon} > - < c, x^* > \leqslant M/2^{t_\varepsilon} \leqslant \varepsilon . \square$$

The above described scheme requires solving at each iteration t the relaxed concave minimization problem:

$$(\mathrm{P}_t) \text{ Minimize } g(x), \text{ subject to} \tag{6}$$

$$x \in D \tag{7}$$

$$< c, x > \leqslant \alpha_t . \tag{8}$$

At the present time, several algorithms are available for solving this problem (see e.g. [4], [6], [9], [10], [11], [12]). However, except when D is a polyhedron, these algorithms are infinite, though convergent. In other words, if D is not a polyhedron, then by a finite procedure we are able to find only an approximate solution to (P_t). Therefore, the incorporation of concave minimization algorithms into the above scheme so as to make it implementable is not a trivial matter. Before discussing this incorporation, we shall describe in the next Section an outer approximation procedure for concave minimization.

3. AN OUTER APPROXIMATION PROCEDURE FOR SOLVING THE RELAXED PROBLEMS

Each relaxed problem (P_t) is a concave minimization problem. Obviously, (P_{t+1}) differs from (P_t) only by the additional constraint (8). By this fact, it is advisable to choose for solving the relaxed problems an algorithm which would permit the use of the information obtained in solving (P_t) for the solution of (P_{l+1}), in such a way to guarantee the convergence of the whole scheme. The outer approximation algorithms (see [4], [9], [12]) meet these requirements.

In this Section we shall describe a procedure for solving (P_l) which is a concrete realization of the general outer approximation method proposed in [9]. Given any $\varepsilon > 0$ this algorithm provides an ε -optimals solution to problem (6) – (8) after a finite number of iterations.

For brevity of presentation we shall denote the set of all x satisfying (7) (8) by C, so that the problem concerned is

$$\min \{g(x) : x \in C\} \tag{9}$$

Recall that we assume that the function $g(x)$ is concave on R^n and C is compact, int $C \neq \phi$.

PROCEDURE φ.

Initialization. Set $u^0 = +\infty$. Construct a polytope $S_1 \supset C$ with vertex set V_1. Select $\varepsilon > 0$.

Step $k = 1, 2, \ldots$

i) Find $v^k \in \arg\min \{g(v) : v \in V_k\}$. Clearly $g(v^k)$ is a lower bound for $\min\{g(x) : x \in C\}$.

ii) Solve the convex programming problem:

$$(Q_k) \qquad \min \{\|v^k - z\|^2 : z \in C\}$$

obtaining z^k. Set $u^k = \min\{u^{k-1}, g(z^k)\}$, and let x^k be the corresponding solution. Obviously u^k is an upper bound for $\min\{g(x) : x \in C\}$.

There are two possible cases:

If $u^k - g(v^k) \leqslant \varepsilon$ then x^k is an ε -optimal solution of problem (9) and the algorithm stops.

Otherwise, we have $u^k - g(v^k) > \varepsilon$. Set

$$l_k(x) = \langle v^k - z^k, x - z^k \rangle.$$

Form the new polytope

$$S_{k+1} = S_k \cap \{x : l_k(x) \leqslant 0\}.$$

Find the set V_{k+1} of all vertices of S_{k+1} and go to step $k+1$.

Ng. V. Thoai established in [11] the finiteness of the above algorithm on the basis of the «cutting plane convergence principle» [9]. We give here a different proof which will also be useful subsequently. First observe the following

LEMMA 2. *If* $\| v^k - z^k \| \leqslant \delta = \varepsilon / L$, *where L is the Lipschitz constant of the function* $g(x)$ *on* S_1, *then* z^k *is an* ε *- optimal solution of problem (9).*

Proof. Let $\overline{x}^*$ be an optimal solution of the problem (9). We have

$$g(z^k) - g(\overline{x}^*) \leqslant g(z^k) - g(v^k) \leqslant L \| v^k - z^k \| \leqslant \varepsilon.$$

Hence z^k is an ε - optimal solution. □

LEMMA 3. *For large enough k we must have*

$$\| v^k - z^k \| \leqslant \delta.$$

Proof. Suppose the contrary, that

$$\| v^k - z^k \| > \delta \ \forall k.$$

Denote by L_k the hyperplane $l_k(x) = 0$, and by N_k the convex polyhedral cone generated by all the rays emanating from v^k and passing through points $x \in L_k \cap S_k$. Let

$$T' = N_k \cap \{x : l_k(x) \geqslant 0\},$$

$$T'' = N_k \cap \{x : l_k(x) \geqslant \eta_k\},$$

where $\eta_k = \min \{l_k(x) : x \in C\}$. Clearly $T' \subset S_k \setminus S_{k+1}$ while $T'' \supset C$. If V', V'' are the volumes of T', T'' and h', h'' are the distances from v^k to the hyperplanes L_k and $H_k = \{x : l_k(x) = \eta_k\}$ (respectively), then

$$V'/V'' = (h'/h'')^n .$$

But $h'' \leqslant \Delta$ = diameter of S_1, while $h' = \| v^k - z^k \| > \delta$. Consequently, $V'/V'' \geqslant (\delta/\Delta)^n$, hence $V' \geqslant (\delta/\Delta)^n V'' \geqslant (\delta/\Delta)^n V$, where V is the volume of C. Since $\text{vol}(S_{k+1}) \leqslant \text{vol}(S_k) - V' \leqslant \text{vol}(S_k) - (\delta/\Delta)^n V$, we get $\text{vol}(S_{k+1}) \leqslant \text{vol}(S_1) - k(\delta/\Delta)^n V < 0$ for large enough k, which is absurd. □

From the two previous Lemmas follows immediately

THEOREM 3. *Procedure* φ *terminates after a finite number of iterations and yields an* ε*-optimal solution of problem (9).*

Remark 1. For the construction of the polytope $S_1 \supset C$ and the computation of the set V_{k+1} from knowledge of V_k we can use any method from [4] and [12].

Remark 2. From the results of [6] it is not necessary to find the optimal solution of problem (Q_k). It suffices to obtain a solution z^k such that the δ-expansion of S_{k+1}, as defined in [6], contains C.

4. FINITE ALGORITHM FOR FINDING AN (ε, θ) - SOLUTION OF PROBLEM (P).

Incorporating the above procedure φ into the scheme presented in Section 2, we can formulate the following algorithm for solving problem (P).

MAIN ALGORITHM.

Initialization. Construct a polytope $S_0 \supset D$ with a simple vertex set V_0. Select $\varepsilon > 0$ and $\theta > 0$. Compute

$$\beta_0 = \max \{ < c, x > : v \in V_0 \}, \gamma_0 = \min \{ < c, x > : x \in D \}.$$

Iteration $k = 0, 1, 2, \ldots$

$$\text{If } \beta_k - \gamma_k \leqslant \varepsilon, \qquad \text{stop.}$$

Otherwise, solve the relaxed problem

$$(\widetilde{P}_k) \qquad \min \{ g(x) : x \in S_k, < c, x > \leqslant \alpha_k \}. \tag{11}$$

where $\alpha_k = \frac{1}{2}(\beta_k + \gamma_k)$. Let x^k be an optimal solution of (11). There are 3 possible cases;

A. $g(x^k) > 0$: Set $S_{k+1} = S_k$, $\beta_{k+1} = \beta_k$, $\gamma_{k+1} = \alpha_k$, and go to iteration $k + 1$.

B. $g(x^k) \leqslant 0$ and $x^k \in D$: Set $S_{k+1} = S_k$, $\beta_{k+1} = < c, x^k >$, $\gamma_{k+1} = \gamma_k$, $x^\varepsilon = x^k$ and go to iteration $k + 1$.

C. $g(x^k) \leqslant 0$ and $x^k \notin D$: Solve the convex program

$$(\widetilde{Q}_k) \quad \min \{ \| z - x^k \|^2 : z \in D \}$$

obtaining an optimal solution z^k. Set $l_k(x) = < x^k - z^k, x - z^k >$,

$$S_{k+1} = S_k \cap \{ x : l_k(x) \leqslant 0 \}.$$

Compute the vertex set V_{k+1} of S_{k+1}.

Two subcases may occur:

C1. $g(z^k) \leqslant 0$; Set $\beta_{k+1} = \min \{\beta_k, < c, z^k >\}$ and let x^ε be the corresponding solution. Set $\gamma_{k+1} = \gamma_k$ and go to iteration $k + 1$.

C2. $g(z^k) > 0$: Set $\beta_{k+1} = \beta_k$, $\gamma_{k+1} = \gamma_k$ and go to iteration $k + 1$.

THEOREM 4. *The Main Algorithm terminates after a finite number of iteration nd yields an (ε, θ)-solution of problem (P).*

Proof. Since cases A or B imply $\beta_{k+1} - \gamma_{k+1} \leqslant \frac{1}{2} (\beta_k - \gamma_k)$, the Algorithm may be infinite only if for all large enough k (e.g. $k \geqslant k_0$) only case C occurs. First assume that for $k \geqslant k_0$ only case $C2$ occurs. Then for $k \geqslant k_0$ $\beta_k - \gamma_k$ remains unchanged from iteration to iteration, i.e. we are actually solving the problem

$$\min \{g(x) : x \in D, < c, x > \leqslant \alpha_{k_0}\}.$$

Therefore, by Lemma 3, for large enough k we shall have $\| x^k - z^k \| \leqslant \delta = \theta / L$, where L is the Lipschitz constant of $g(x)$ on S_0. Since $g(x^k) \leqslant 0$ and $g(z^k) - g(x^k) \leqslant L \| x^k - z^k \| \leqslant \theta$, it follows that $g(z^k) \leqslant \theta$. This contradiction shows that if the Algorithm is infinite, case $C1$ must occur infinitely often, e. g. in iterations k_ν, $\nu = 1, 2, \ldots$ Then it is easily seen that there exists $k_\mu \geqslant k_0$ such that $\beta_{k_\nu+1} > \alpha_{k_\nu}$ and $\beta_{k_\nu} - \gamma_{k_\nu} > \varepsilon$ for all $k_\nu \geqslant k_\mu$. Clearly, $< c, z^k > \geqslant \beta_{k+1}$ and $< c, x^k > \leqslant \alpha_k$ for all k. Consequently,

$$0 < \beta_{k_\nu+1} - \alpha_{k_\nu} \leqslant < c, z^{k_\nu} - x^{k_\nu} > \leqslant \| c \| \; \| z^{k_\nu} - x^{k_\nu} \|$$

for all $k_j \geqslant k_\mu$. By construction of the sequence $\{\beta_k\}$ we have $\beta_{k+1} \leqslant \beta_k$ and so this sequence is monotone nonincreasing. Since it is bounded below, it must converge. Therefore, $\beta_{k_\nu} - \beta_{k_\nu+1} \leqslant \varepsilon / 4$ for all large enough k_ν. Now, by an argument analogous to that used for the proof of Lemma 3, we can show that

$$\| z^{k_\nu} - x^{k_\nu} \| \leqslant \varepsilon / (4 \; \| c \|)$$

for all large enough k_ν. Hence

$$\varepsilon / 2 < \frac{1}{2} (\beta_{k_\nu} - \gamma_{k_\nu}) = (\beta_{k_\nu} - \beta_{k_\nu+1}) + (\beta_{k_\nu+1} - \alpha_{k_\nu})$$

$$\leqslant \varepsilon/4 + \| c \| \; \| z^{k_\nu} - x^{k_\nu} \| \leqslant \varepsilon/4 + \varepsilon/4 = \varepsilon/2$$

for large enough $k_\nu \leqslant k_\mu$, which is absurd. Therefore, the Algorithm cannot be infinite, and must terminate at some iteration k. Then $x^\varepsilon \in D$, $g(x^\varepsilon) \leqslant \theta$ and $< c, x^\varepsilon > - f^* \leqslant \beta_k - \gamma_k \leqslant \varepsilon$, as was to be proved. □

Remark 3. Problem (11) is a concave program. Since the set V_k of vertices of S_k is known, we can apply the method described in [12].

5. CASE OF LINEAR CONSTRAINTS

In this Section we shall discuss the case where D is a polytope defined by the system of linear inequalities

$$< A_i , x > \leqslant b_i , \qquad i = 1,2, \ldots, m$$

$$x_j \geqslant 0 , \qquad j = 1,2, \ldots, n$$

where $A_j \in R^n$, $b_i \in R$, $i = 1,2,\ldots,m$. Then the relaxed problem (P_l) is a linearly constrained concave minimization problem of the form

$$\text{Minimize } g(x), \text{ subject to}$$

$$x \in D = \{x : <A_i, x> \leqslant b_i,\ i = \overline{1,m};\ x_j \geqslant 0,\ j = \overline{1,n}\} <c, x> \leqslant \alpha_l .$$

For solving this problem several finite algorithms can be used (see e.g. [1], [12], [11]). Incorporating the finite algorithm of Thieu-Tam-Ban [12] into the general scheme we obtain the following

ALGORITHM 2.

Initialization. Compute

$$N = \max \{ \sum_{j=1}^{n} x_j : x \in D \}.$$

Set $S_o = \{x :\ x_j \geqslant 0,\ j = \overline{1,n} :\ \sum_{j=1}^{n} x_j \leqslant N\}$. Find the vertex set V_o of the simplex S_o. Compute

$$\beta_o = \max \{ <c, v> : v \in V_o \} \text{ and } \gamma_o = \min\{ <c, v> : v \in V_o \}.$$

Iteration $k = 0, 1, 2, \ldots$

$$\text{If } \beta_k - \gamma_k \leqslant \varepsilon, \text{ stop.}$$

Otherwise, solve the relaxed problem

$$\min \{ g(x) : x \in S_k,\ <c, x> \leqslant \alpha_k \},$$

where $\alpha_k = \frac{1}{2}(\beta_k + \gamma_k)$. Let x^k be an optimal solution of this problem. There are 3 possible cases.

A. $g(x^k) > 0$: Set $S_{k+1} = S_k$, $\beta_{k+1} = \beta_k$, $\gamma_{k+1} = \alpha_k$ and go to iteration $k+1$.

B. $g(x^k) \leqslant 0$ and $x^k \in D$: Set $S_{k+1} = S_k$, $x^\varepsilon = x^k$, $\beta_{k+1} = <c, x^k>$, $\gamma_{k+1} = \gamma_k$ and go to iteration $k+1$.

C. $g(x^k) \leqslant 0$ and $x^k \notin D$: Select the index

$$i_k = \text{agr max} \{ <A_i, x^k> - b_i,\ i = 1, \ldots, m \}.$$

Find the vertex set V_{k+1} of the polytope

$$S_{k+1} = S_k \cap \{ x : <A_{i_k}, x> - b_{i_k} \leqslant 0 \}.$$

Set $\beta_{k+1} = \beta_k$, $\gamma_{k+1} = \gamma_k$ and go to iteration $k+1$.

From the results of [12] and Theorem 2 we get

THEOREM 5. *Algorithm 2 terminates after a finite number of iterations and yields an* ε *-optimal solution of problem (1)—(3).*

6. ILLUSTRATIVE EXAMPLE

To illustrate the Main Algorithm we give a two-dimensional example:

Minimize $f(x) = -3x_1 - x_2$, subject to

$$h_1(x) = -x_1 + x_2 - 1 \leqslant 0$$

$$h_2(x) = (x_1 - 2)^2 + (x_2 - 2)^2 - 4 \leqslant 0$$

$$h_3(x) = (x_1 - 2)^2 - x_2 + 1 \leqslant 0$$

$$g(x) = -(x_1 - 3)^2 - (x_2 - 2.5)^2 + 1.25 \leqslant 0.$$

Initialization.

$$S_o = \{(x_1, x_2) : 0 \leqslant x_1,\ 0 \leqslant x_2, x_1 + x_2 \leqslant 8\}.$$

$$V_o = \{(0,0), (0,8), (8,0)\}.$$

$\beta_0 = 0$; $\gamma_0 = -13.853$. Let $\varepsilon = 0.5$ and $\theta = 0.01$.

Iteration 0. $\beta_0 - \gamma_0 = 13.853 > \varepsilon$. Solving $(\widetilde{P}_0)$, yields $x^0 = (0,8)$. Since $g(x^0) = -38 < 0$ and $x^0 \notin D$ we are in case C . Solving $(\widehat{Q}_o)$ yields $z^0 = (2.825, 3.825)$. The cutting plane is $l_0(x) = -2.825x_1 + 4.175x_2 - 8$, and the vertex set of S_1 is

$$V_1 = \{(0, 1.915), (3.63, 4.37), (0, 0), (8, 0)\}.$$

Since $g(z^0) < \theta$ we set $\beta_1 = -12.3$, $\gamma_1 = -13.853$, $x^\varepsilon = z^0$.

Iteration 1. $\beta_1 - \gamma_1 = 1.553 > \varepsilon$. Solving $(\widetilde{P}_1)$, yields $x^1 = (8, 0)$. Since $g(x^1) = -30$ and $x^1 \notin D$ we are in case C. Solving $(\widehat{Q}_1)$ yields $z^1 = (3.134, 2.825)$. The cutting plane is $l_1(x) = 4.87x_1 - 2.28x_2 - 9.9$, and the vertex set of S_2 is

$$V_2 = \{(2.05, 0), (4.06, 3.94), (0,0), (3.63, 4.37), (0, 1.915)\}.$$

Since $g(z^1) > \theta$ we set $\beta_2 = -12.3$, $\gamma_2 = -13.853$.

Iteration 2. $\beta_2 - \gamma_2 = 1.553 > \varepsilon$. Solving $(\widetilde{P}_2)$, yields $x^2 = (3.63, 4.37)$.

Since $g(x^2) < 0$ and $x^2 \notin D$ we are in case C. Solving (Q_2) yields $z^2 = (3.134, 3.641)$. The cutting plane is $l_2(x) = 0.496x_1 + 0.73x_2 - 4.225$, and the vertex set of S_3 is

$V_3 = \{(2.9, 3.85), (3.7, 3.35), (0, 0), (2.05, 0), (0, 1.915)\}$.
Since $g(z^2) < 0$ we set $\beta_3 = -13.043$, $\gamma_3 = -13.983$, $x^{\varepsilon} = (3.134, 3.641)$.

Iteration 3. $\beta_3 - \gamma_3 = 0.840 > \varepsilon$. Solving $(\widetilde{P}_3)$, yields $x^{\varepsilon} = (3.7, 3.35)$. Since g $(x^3) > 0$ we are in case *A*. Set $S_4 = S_3$, $\beta_4 = -13.853$, $\gamma_4 = -13.448$.

Iteration 4. $\beta_4 - \gamma_4 = 0.405 < \varepsilon$. The Algorithm terminates and gives an (0.5, 0.01)-solution $x^{\varepsilon} = (3.134, 3.641)$.

7. COMPUTATIONAL EXPERIENCE.

The Algorithm 2 was coded in BASIC and run on a micro computer APPLE II. The results on 8 test problems are summarized in the following tableau.

Problem	Size of A	Number of iterations	Number of cuts	Maximal number of generated vertices
1	3.2	6	2	5
2	6.2	6	3	5
3	4.3	7	3	6
4	6.3	10	4	8
5	7.4	7	3	12
6	6.4	6	2	8
7	7.5	14	4	24
8	6.8	8	3	48

Acknowledgement. We should like to thank Prof. Hoang Tuy for his encouragement and his remarks which have helped to substantially improve a first draft of the paper. We are also grateful to Prof. Tran Vu Thieu and Dr. Nguyen Van Thoai for valuable comments.

REFERENCES

[1] V.T. Ban, *A finite algorithm for minimizing a concave function under linear constraints and its applications*, Proceedings of IFIP Working Conference on Recent Advances on System Modelling and Optimization, Hanoi, January 1983.

[2] R.J. Hillestad and S.E. Jacobsen, *Reverse convex programming*, Appl. Math. Optim. 6(1980), 63–78.

[3] R.J. Hillestad and S.E. Jacobsen, *Linear programs with an additional reverse convex constraint*, Appl. Math. Optim., 6(1980), 257–269.

[4] K.R. Hoffman, *A method for globally minimizing concave function over convex sets*, Math. Programming, 20(1981), 22–32.

[5] L.D. Muu, *A convergent algorithm for solving linear programming with an additional reverse convex constraint*, Preprint. Institute of Mathematics. Hanoi. 1984.

[6] B.M. Mukhamediev, *An approximation method for solving concave minimization problems* Zh. Vychisl. Math. i Mat. Fiz., (22) 1982, 727—731.

[7] N.D. Nghia and N.D Hieu, *An algorithm for solving linear programming problems with an additional reverse convex constraint*, Proceedings of Math. Conf., Polytechnical Institute, Hanoi 1984, 135—142.

[8] H. Tuy, *Convex programs with an additional reverse convex constraint*, Journal of Optimization Theory and Applications, (to appear).

[9] H. Tuy, *On outer approximation algorithms for concave minimization problems*, Acta Mathematica Vietnamica, 8 (1983), 3 — 34.

[10] Ng. V. Thoai and H. Tuy, *Convergent Algorithms for minimizing a concave function*, Math. Oper. Res., 5 (1980), 556 — 566.

[11] Ng. V. Thoai, *Verfahren zur Losung Konkaver Optimierungsaufgaben*, Seminarberichte № 63 (1984), Humboldt-Universitat, Berlin.

[12] T. V. Thieu, B. T. Tam and V. T. Ban, *An outer approximation method for globally minimizing a concave function over a compact convex set*, Acta Math. Vietnam, 8 (1983), 21 — 40.

[13] H. Tuy, *Global minimization of a concave function subject to mixed linear and reverse convex constraints*, Proceedings of IFIP Working Conference on Recent Advances on System Modelling and Optimization, Hanoi, January 1983.

Received July 15, 1985

POLYTECHNICAL INSTITUTE, HANOI, VIETNAM.