

Lê Dung Muu

A convergent algorithm for solving linear programs with an additional reverse convex constraint

*Kybernetika*, Vol. 21 (1985), No. 6, 428--435

Persistent URL: <http://dml.cz/dmlcz/125889>

## Terms of use:

© Institute of Information Theory and Automation AS CR, 1985

Institute of Mathematics of the Academy of Sciences of the Czech Republic provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This paper has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library*  
<http://project.dml.cz>

## A CONVERGENT ALGORITHM FOR SOLVING LINEAR PROGRAMS WITH AN ADDITIONAL REVERSE CONVEX CONSTRAINT

LÊ DUNG MUU

An inequality  $g(x) \geq 0$  is often said to be a reverse convex constraint if the function  $g$  is continuous and convex. The feasible regions for linear program with an additional reverse convex constraint are generally non-convex and disconnected.

In this paper a convergent algorithm for solving such a linear problem is proposed. The method is based upon a combination of the branch and bound procedure with the linearization of the reverse convex constraint by using the cutting-plane technique.

### 1. INTRODUCTION

The problem, denoted by (R), to be considered is

$$\begin{aligned} (1.1) \quad & \text{Min } f(x) \\ (1.2) \quad & \text{s.t. } Ax \leq b \\ & g(x) \geq 0 \end{aligned}$$

where  $f: R^n \rightarrow R^1$  is a linear function,  $g: R^n \rightarrow R^1$  is a convex function and  $A$  is an  $m \times n$  matrix. We shall assume that the constraints (1.1) define a polytope, i.e., a bounded polyhedral convex set. Since the constraint (1.2) would be convex if the inequality were reversed, it is called a reverse convex constraint. So Problem (R) differs from an ordinary linear program only by an additional reverse convex constraint. However, it is this additional constraint which causes the main difficulty of the problem, because the feasible set (1.1), (1.2) is no longer convex, nor is it necessarily connected.

A number of problems in engineering design, control theory, operation research (e.g. resource allocation problems where economies of scale are prevalent), complementary geometric programming can be given in the form of Problem (R). The more details about the nature of practical problems which lead to the problem of this rather special type can be found in [1]–[4] and [7], [8].

Note that any concave program of the form:

$$(C) \quad \begin{array}{l} \text{Min } G(x) \\ \text{s.t. } Ax \leq b \end{array}$$

(where  $G : R^n \rightarrow R^1$  is a concave function) can be converted into a Problem (R) by writing

$$\begin{array}{l} \text{Min } z \\ \text{s.t. } Ax \leq b \\ z - G(x) \geq 0 \end{array}$$

Thus, there is a very close connection between concave programming and linear programming with an additional reverse convex constraint. This suggests that ideas and methods of concave programming could be useful for the study of Problem (R). In fact, Tuy cuts (see [10]) originally developed for solving Problem (C) have been applied by some authors (see e.g. Hillestad and Jacobsen [6]) to optimization problem with reverse convex constraints. Although the convergence of cutting methods of this kind is still an open question, Hillestad and Jacobsen [6] have reported successful applications of Tuy cuts to problems that would be otherwise unapproachable.

In this paper we shall present a branch and bound method for solving Problem (R) which borrows essential ideas from the cone splitting method first proposed by H. Tuy [10] and further developed by Ng. V. Thoai and H. Tuy [9] for solving the concave minimization problem. Our method proceeds according to the same scheme as that of Thoai and Tuy's. Namely, starting from an initial cone containing the feasible set, we split it into smaller and smaller subcones. Then subcones are fathomed by bounding operations based on Tuy cuts. If some step occurs where all the generated subcones have been fathomed, the incumbent (i.e. the current best known feasible solution) gives an optimal solution. Otherwise, the algorithm generates a sequence having a limit which is an optimal solution of the problem. The method is thus guaranteed to converge, without any condition imposed upon the data, apart from the boundness of the set defined by the linear constraints (1.1).

## 2. DESCRIPTION OF THE METHOD

Denote by  $S$  the polyhedral convex set defined by the linear constraints (1.1) and assume that this set is nondegenerate.

The algorithm we now propose for the solution of the Problem (R) is a procedure of the branch and bound type. The outline of this method had been used in [9] for minimizing a concave function over a polytope. At each step of the algorithm a certain collection of cones must be examined and three basic operations must be performed:

- 1) Compute for each new cone  $C$  a suitable lower bound  $\mu(C)$  of the objective function  $f$  over  $F \cap C$  ( $F$  denotes the feasible region of the Problem (R)).
- 2) Split a cone into two smaller cones.
- 3) Investigate some criterions for deletion of a cone.

We shall specify later the precise rules for these operations. Assume for the moment that these rules have been defined. Then the algorithm can be described as follows:

*Initialization.* Solve the following linear program by the simplex method:

$$(L) \quad \text{Min } \{f(x) : Ax \leq b\}.$$

Let  $s^0$  be the obtained solution of this problem which is a vertex of the set  $S$ . If  $s^0 \in F$ , stop:  $s^0$  is an optimal solution of (R).

Otherwise, denote by  $C_0$  the convex polyhedral cone generated by  $n$  rays emanating from  $s^0$  and passing through  $n$  extreme points  $s^1, \dots, s^n$  of  $S$  adjacent to  $s^0$ . Let  $\mathcal{C}_0 = \{C_0\}$  and compute a suitable lower bound of  $f$  over  $F \cap C_0$ .

If a point of  $F$  is available, let  $x^0$  be the best one currently known and let  $\gamma_0 = f(x^0)$ . In the contrary case let  $\gamma_0 = \infty$ . The feasible points of the Problem (R) may be generated when we solve the linear program (L) and when we compute the lower bound of  $f$ .

*Step  $k$  ( $k = 0, 1 \dots$ ).* Delete all  $C \in \mathcal{C}_k$  if either there is no point of  $F$  in  $C$  or the best point of  $F$  in  $C$  is already known.

Denote by  $\mathcal{R}_k$  the set of remaining cones. If  $\mathcal{R}_k = \emptyset$ , stop:  $x^k$  is an optimal solution if  $\gamma_k < \infty$ ,  $F = \emptyset$  if  $\gamma_k = \infty$ .

Otherwise, if  $\mathcal{R}_k \neq \emptyset$  then select  $C_k \in \mathcal{R}_k$  with the smallest  $\mu(C_k)$ . Divide this cone into two smaller subcones  $C_{k_1}$  and  $C_{k_2}$  such that  $C_k = C_{k_1} \cup C_{k_2}$ .

For each  $i = 1, 2$  compute a lower bound  $\mu(C_{k_i})$  of  $f$  on  $F \cap C_{k_i}$ . These operations may generate some new feasible points. Let  $x^{k+1}$  be the best among  $x^k$  and all newly generated feasible solutions and let  $\gamma_{k+1} = f(x^{k+1})$  be the new current best value. Form the new set  $\mathcal{C}_{k+1}$  by substituting  $C_{k_1}$  and  $C_{k_2}$  for  $C_k$  in  $\mathcal{R}_k$ , then go to step  $k + 1$ .

Understandably, the convergence as well as efficiency of the above algorithm depends upon the concrete rules for the bound estimation, the bisection and for the deletion. We proceed to describe these rules in the next part of this section.

Firstly, let us compute a suitable lower bound of  $f$  on the intersection of a cone  $C$  (vertex at  $s^0$ ) with  $F$ .

Denote by  $T$  the simplex defined by the extreme points  $s^1, \dots, s^n$  of  $S$  adjacent to  $s^0$ . Let  $v^j$  be the intersection of  $T$  with the  $j$ th edge of  $C$ . Let

$$(2.1) \quad t_j = \max \{t : g(s^0 + t(v^j - s^0)) \leq 0\}$$

Since  $g(s^0) < 0$  and  $g$  is convex on  $R^n$ ,  $t_j > 0$  for each  $j \in \{1, \dots, n\}$ .

Let  $J_C = \{j : 1 \leq j \leq n, t_j < \infty\}$  and

$$(2.2) \quad y^j = s^0 + t_j(v^j - s^0) \quad j \in J_C$$

and denote by  $\alpha_C$  the largest lower bound of  $f$  on  $C \cap F$ , which is available for us. Since  $f(s^0) = \text{Min} \{f(x) : x \in S\}$ ,  $\alpha_C \geq f(s^0)$ .

Now, a lower bound of  $f$  on  $C \cap F$  may be computed as follows:

$$(2.3) \quad \mu(C) = \begin{cases} \infty & \text{if } J_C = \emptyset \\ \text{Max}_{j \in J_C} \{\alpha_C, \min f(y^j)\} & \text{if } f(S^0) \neq f(v^i) \quad \forall i \in J_C \\ \alpha_C & \text{otherwise.} \end{cases}$$

**Lemma 2.1.** The just defined  $\mu(C)$  is a lower bound of  $f$  on  $C \cap F$ .

**Proof.** If  $J_C = \emptyset$  then by the convexity of  $g$  we have  $g(x) < 0$  for all  $x \in C$ . Hence  $F \cap C = \emptyset$ .

Now, by virtue of the choice of  $\alpha_C$ , it will suffice to prove the lemma when  $\mu(C) = \text{Min} \{f(y^j) : j \in J_C\}$  and  $\mu(C) > \alpha_C$ . In this case using the linearity of  $f$  we can choose for each  $i \notin J_C$  a point  $u^i$  on the edge passing through  $v^i$  ( $i \notin J_C$ ) so that  $f(u^i) = \mu(C)$ .

Denote by  $K_C$  the hyperplane defined by  $y^j$  ( $j \in J_C$ ) and  $u^i$  ( $i \notin J_C$ ) and by  $K_C^-$  the open halfspace determined by  $K_C$  containing  $s^0$  and by  $K_C^+$  its complement. Then

$$C \cap S \cap K_C^- \subset \text{co} \{s^0, y^j, u^i : j \in J_C, i \notin J_C\}$$

where  $\text{co } A$  stands for the convex hull of  $A$ .

Let  $x$  be any point of  $C \cap S \cap K_C^-$  then

$$x = \alpha_0 s^0 + \sum_{j \in J_C} \alpha_j y^j + \sum_{i \notin J_C} \alpha_i u^i$$

with  $\alpha_k \geq 0$  for every  $k$  and  $\alpha_0 + \sum_{j \in J_C} \alpha_j + \sum_{i \notin J_C} \alpha_i = 1$ .

From this and the convexity of  $g$  and from  $x \in K_C^-$ , it follows  $g(x) < 0$ . This implies  $F \cap C \subset K_C^+$ . Hence, if  $x$  is any point of  $C \cap F$ , the segment  $(s^0, x]$  must meet  $K_C$  at a point  $v \in (s^0, x]$ .

Since  $f$  is linear and  $f(x) \geq f(s^0)$  we have

$$(2.4) \quad f(v) \leq f(x).$$

On the other hand

$$f(v) \geq \text{Min} \{f(x) : x \in \text{co} \{y^j, u^i : j \in J_C, i \notin J_C\}\} = \mu(C).$$

This and (2.4) imply  $f(x) \geq \mu(C)$ . □

There are several ways how to compute a lower bound of  $f$  over  $C \cap F$  besides (2.3). The following lower bound seems to be suitable:

$$(2.5) \quad \mu(C) = \text{Min} \{f(x) : x \in S \cap H_C^+ \cap C\}$$

where  $H_C^+$  is the closed halfspace not containing  $s^0$  and defined by the hyperplane through  $y^j$  ( $j \in J_C$ ), parallel, to  $v^i - s^0$  ( $i \notin J_C$ ).

Here, as usual, let  $\text{Min} \{f(x) : x \in \emptyset\} = \infty$ .

The problem (2.5) is a linear program. Since  $H_C^+ \cap S \cap C_1$  may be empty, it is expedient to solve (2.5) by the dual-simplex method.

*The bisection process*

We shall use the same bisection as was used in [9]. More precisely, let  $v^j$  be the point where the  $j$ th edge of the cone  $C$  meets  $T = [s^1, \dots, s^n]$ . Choose the longest side of the simplex  $[v^1, \dots, v^n]$ , say  $[v^{j_1}, v^{j_2}]$ , and let  $u$  be the midpoint of this side. Then for each  $k = 1, 2$  take  $C_k$  to be the cone whose set of edges obtains from that of  $C$  by substituting the halfline from  $s^0$  through  $u$  for the edge passing through  $v^{j_k}$  ( $k = 1, 2$ ). It is immediate that  $C = C_1 \cup C_2$ .

This bisection has the following property which was proved in [9].

**Lemma 2.2.** Any infinite decreasing sequence of cones  $C_j$  generated by the above bisection process tends to a halfline emanating from  $s^0$ .

(By a decreasing sequence  $C_j$  we mean a sequence such that  $C_j \subset C_{j+1}$  for every  $j$ ).

*Deletion rules*

At the  $(k + 1)$ th step of the algorithm a cone  $C$  is deleted if either of the following conditions holds:

- (i)  $\mu(C) \geq \gamma_k$
- (ii)  $C \cap S \subset H_C^-$

$H_C^-$  being the open halfspace containing  $s^0$  and determined by the hyperplane  $H_C$  through  $y^j$  ( $j \in J_C$ ) and parallel to  $v^j - s^0$  ( $j \notin J_C$ ).

**Lemma 2.3.** The above rules (i), (ii) delete no feasible point whose value (with respect to  $f$ ) is less than the value of the best available feasible point.

*Proof.* Obvious since  $\mu(C)$  is a lower bound of  $f$  on  $C \cap F$ ,  $\gamma_k$  is the best currently known value and since  $C \cap F \subset H_C^+$ . □

To verify the rule (ii) solve the following linear problem:

$$(L_1) \quad \text{Max} \left( \sum_{j \in J_C} \lambda_j / t_j \right) \quad \text{subject to} \quad A(s^0 + B\lambda) \leq b, \quad \lambda \geq 0$$

where  $B$  is the  $(n \times n)$ -matrix with columns  $v^i - s^0$  ( $i = 1, \dots, n$ ) and  $\lambda \in R^n$ .

Let  $\lambda^*$  be an optimal solution of  $(L_1)$ . Then it has been shown in [11] that (ii) holds iff  $\sum_{j \in J_C} (\lambda^* / t_j) < 1$ .

### 3. CONVERGENCE OF THE ALGORITHM

Suppose that the above rules for bound estimation, and splitting and for deletion are applied in the algorithm. Then we can state the following convergent theorem.

**Theorem 1.** The algorithm described in Section 2 either terminates at some step  $k$ , or it generates an infinite sequence  $z^k$  having at last one cluster point which solves the problem (R).

*Proof.* If the procedure terminates at some step  $k$  then by Lemma 2.3,  $x^k$  is a solution if  $\gamma_k < \infty$ , otherwise, i.e., if  $\gamma_k = \infty$ ,  $F = \emptyset$ .

Suppose now that the procedure never terminates. Then, by Lemma 2.2, there exists a decreasing subsequence of cones  $C_{k_j}$  tending to a ray  $L$  emanating from  $s^0$ , i.e.  $\bigcap_{j=1}^{\infty} C_{k_j} = L$ .

First, we claim that there exists at least one point  $x$  on  $L$  such that  $g(x) \geq 0$ . Indeed, in the contrary case we can take a point  $w \in L$  so that there is a ball  $U$  around  $w$  satisfying  $U \cap S = \emptyset$  and  $g(u) < 0$  for every  $u \in U$ . Then, for all  $j$  large enough  $C_{k_j}$  lies inside the cone generated by  $s^0$  and  $U$ . This implies that  $C_{k_j}$  was already deleted. Consequently, since  $g$  is convex and  $g(s^0) < 0$ , there is an  $\bar{x} \in L$  such that  $g(\bar{x}) = 0$ .

Next we show that  $\bar{x} \in S$ . Indeed, if  $\bar{x} \notin S$  then since  $S$  is closed there is an open ball  $W$  around  $\bar{x}$  satisfying  $W \cap S = \emptyset$ . Using again the fact that  $g$  is convex,  $g(\bar{x}) = 0$ ,  $g(s^0) < 0$  a point  $w^1 \in W \cap [s^0, \bar{x}]$  can be chosen that satisfies  $g(w^1) < 0$ . Let  $W_1$  be a ball centred at  $w^1$  so that  $W_1 \subset W$  and  $g(x) < 0$  for each  $x \in W_1$ . Since  $\bigcap_{j=1}^{\infty} C_{k_j} = L$ , each edge of  $C_{k_j}$  meets  $W_1$  provided  $j$  is large enough. Denote by  $u^{kj,i}$  a point in the intersection of the  $i$ th edge of  $C_{k_j}$  with  $W_1$ , then  $C_{k_j} \cap S \subset \text{Co} \{s^0, u^{kj,1}, \dots, u^{kj,n}\}$ . From this and  $g(s^0) < 0$ ,  $g(u^{kj,i}) < 0$  for all  $i = 1, \dots, n$  it follows that either  $C_{k_j} \cap S \subset \bar{H}_{C_{k_j}}$  or  $g(x) < 0$  for every  $x$  laying on the edges of  $C_{k_j}$ . Hence, by the rule of deletion  $C_{k_j}$  was deleted. This contradiction shows that  $\bar{x} \in S$ , which together with  $g(\bar{x}) = 0$  implies  $\bar{x} \in F$ .

In the same way can show that  $\bar{x} \in \partial S$  (the boundary of  $S$ ).

Now observe that since  $g$  is convex,  $g(s^0) < 0$ ,  $g(\bar{x}) = 0$ , for all  $j$  large enough the index-set  $J_{C_{k_j}} = \{1, \dots, n\}$ .

Let us denote by  $z^{kj}$  the point satisfying

$$f(z^{kj}) = \text{Min} \{f(y^j) : j \in J_{C_{k_j}}\}$$

(the points  $y^j$  are defined by (2.2)). It is clear that  $z^{kj} \rightarrow \bar{x}$  as  $j \rightarrow \infty$ .

Using the definition of  $z^{kj}$  and the rule of selection of a cone for bisection we have

$$f(z^{kj}) = \mu(C_{k_j}) \leq \text{Min} \{f(x) : x \in F\}.$$

This and  $z^{kj} \rightarrow \bar{x} \in F$  imply

$$f(\bar{x}) = \text{Min} \{f(x) : x \in F\}.$$

□

Fig. 1 illustrates the algorithm with the following two dimensional example:

$$\begin{aligned} & \text{Min } -x_2 \\ & \text{subject to } 2x_1 + x_2 \leq 8 \\ & \quad 3x_1 - x_2 \leq 3 \\ & \quad x_1 \geq 0, \quad x_2 \geq 0, \quad x_2 \leq 6 \\ & \quad x_1^2 - x_2 \geq 0 \end{aligned}$$

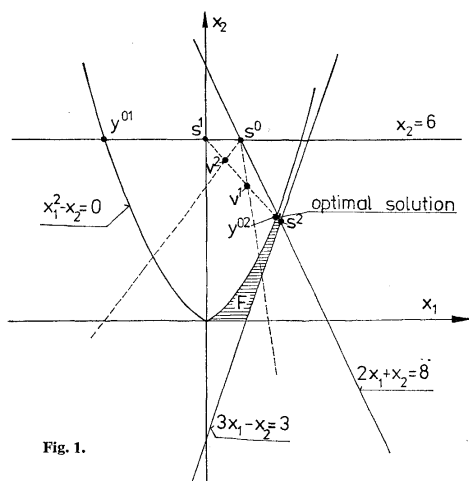


Fig. 1.

The algorithm starts from the vertex  $s^0 = (1, 6)$  of  $S$ , which is also a minimal point of the object function  $f(x) = -x_2$  subject to  $x \in S$ . Two adjacent vertices of  $s^0$  are  $s^1 = (0, 6)$  and  $s^2 = (2.2, 3.6)$ .

At the step 0 the collection  $\mathcal{C}_0$  of cones consists of the sole cone  $C_0 = \text{cone}\{s^0, s^1, s^2\}$ . Two points on the edges of  $C_0$  satisfying the equation  $g(x) = x_1^2 - x_2 = 0$  are  $y^{01} = (-\sqrt{6}, 6)$ ,  $y^{02} = (2, 4)$ . The lower bound of  $f$  on  $C_0$  computed by (2.3) is  $\mu(C_0) = f(y^{01}) = -6$  and the best currently known feasible point is  $x^1 = y^{02} = (2, 4)$  which yields an upper bound  $\gamma_1 = f(x^1) = -4$ . The cone  $C_0$  is split into two cones  $C_{01} = \text{cone}\{s^0, v^1, s^1\}$  and  $C_{02} = \text{cone}\{s^0, v^1, s^2\}$  with  $v^1 = (s^1 + s^2)/2 = (1.1, 4.8)$ .

At the step 1,  $\mathcal{C}_1 = \{C_{01}, C_{02}\}$  with  $\mu(C_{01}) = f(y^{01}) = -6$ ,  $\mu(C_{02}) = f(y^{02}) = -4$  and  $x^2 = x^1 = y^{02} = (2, 4)$ ,  $\gamma_2 = f(x^2) = -4$ .

Thus,  $C_{02}$  is deleted (since  $\gamma_1 = \mu(C_{02}) = -4$ ). The set of the remaining cones



is  $\mathcal{R}_1 = \{C_{01}\}$ . The cone  $C_{01}$  is split into two cones  $C_{11} = \text{cone}\{s^0, v^2, s^1\}$  and  $C_{12} = \text{cone}\{s^0, v^2, v^1\}$  with  $v^2 = (0.55, 5.4)$ .

At the step 2,  $\mathcal{C}_2 = \{C_{11}, C_{12}\}$ . Since  $C_{11} \cap S \subset H_{C_{11}}^-$ , the cone  $C_{11}$  is deleted ( $H_{C_{11}}^-$  denotes the open halfspace containing  $s^0$  and defined by the hyperplane passing through  $y^{01} = (-\sqrt{6}, 6)$  and  $y^{12} = (-1.6, 2.6)$ ).

For the cone  $C_{12}$  we have  $\mu(C_{12}) = f(y^{12}) = -2.6$ . Since  $\mu(C_{12}) = f(y^{12}) > \gamma_2 = -4$ , this cone is deleted too.

Hence,  $\mathcal{R}_2 = \emptyset$  and therefore  $x^2 = (2, 4)$  is an optimal solution with the minimum value  $f(x^2) = -4$ .

**Remark.** The above example shows that the lower bound computed by (2.3) in general cannot be improved.

#### ACKNOWLEDGEMENT

The author is grateful to Prof. Hoang Tuy for introducing him into the subject.

(Received December 12, 1983.)

#### REFERENCES

- [1] M. Avriel and A. C. Williams: Complementary geometric programming. *SIAM J. Appl. Math.* 19 (1970), 125–141.
- [2] M. Avriel and A. C. Williams: An extension of geometric programming with applications in engineering optimization. *J. Engng. Math.* 5 (1971), 187–194.
- [3] P. P. Bansal and S. E. Jacobsen: Characterization of local solution for a class of nonconvex programs. *J. Optim. Theory Appl.* 15 (1975), 127–131.
- [4] R. J. Hillestad: Optimization problems subject to a budget constraint with economies of scale. *Oper. Res.* 23 (1975), 1091–1098.
- [5] R. J. Hillestad and S. E. Jacobsen: Linear programs with an additional reverse convex constraint. *Appl. Math. Optim.* 6 (1980), 257–269.
- [6] R. J. Hillestad and S. E. Jacobsen: Reverse convex programming. *Appl. Math. Optim.* 6 (1980) 63–78.
- [7] R. Meyer: The validity of a family of optimization methods. *SIAM J. Control* 8 (1970), 41–54.
- [8] J. B. Rosen: Iterative solution of nonlinear optimal control problems. *SIAM J. Control* 4 (1966), 223–244.
- [9] N. V. Thoai and H. Tuy: Convergent algorithms for minimizing a concave function. *Math. Oper. Res.* 4 (1980), 556–565.
- [10] H. Tuy: Concave programming under linear constraints. *Dokl. Akad. Nauk SSSR* 159 (1964), 32–35.
- [11] H. Tuy: Conical algorithm for solving a class of complementarity problems. Preprint series 18 (1981), Hanoi.
- [12] U. Ueing: A combinatorical method to compute a global solution of certain nonconvex optimization problems. In: *Numerical Methods for Non-Linear Optimization* (F. A. Lootsma ed.), pp. 223–230, Academic Press, New York 1972.

*Dr. Lê Dung Muu, Institute of Mathematics, Vien Toan Hoc, Box 631 Bo Ho, Hanoi, Vietnam.*