A MODIFICATION OF TUY'S ALGORITHM FOR CANONICAL DC PROGRAMMING PROBLEM

NGUYEN TRONG TOAN

Abstract: A version of outer - approximation method is presented for the Canonical DC optimization problem. Some computational experiments are described to compare it with other methods.

Keywords: Reverse Convex Programming Canonical DC Optimization. Outer Approximation. ε-Approximate Feasible Solution.

1. INTRODUCTION.

In this paper, we are concerned with the Canonical DC optimization problem (CDC), also referred to as the reverse convex programming problem [3 - 7]:

Minimize $f(x)$

subject to:

$x \in D \cap \text{int}G.$

Where $D = \{x: h(x) \leq 0\}$ and $G = \{x: g(x) \geq 0\}$; $h(x)$ and $g(x)$ are bounded convex functions in $\mathbb{R}^n$; $f(x) = \langle c, x \rangle$, $c, x \in \mathbb{R}^n$. Assume that $D$ is bounded. It has been proved that any DC optimization problem can be reduced to the CDC (1).

CDC problem is a mathematical model for many practical applied problems. Besides, it plays an important role in the global optimization theory. Therefore, is has received much attention in recent years (see (1) and its references). The main difficulty for solving the problem is due to the presence of the reverse convex constraint $g(x) \geq 0$, which destroys the convexity and even the adjacency of the feasible set of the problem. Up to now, there were many different methods for solving CDC. However, several of them have not yet been interested sufficiently in their convergence, efficiency of computational test.

This paper includes 4 sections. After the introduction, the second section describes a typical outer-approximation algorithm for CDC, which presented by II. Tuy (see [1]). The third one presents our modification of Tuy's algorithm and its theoretical background. The last one presents some computational experiences of the algorithms.

2. TUY'S OUTER- APPROXIMATION ALGORITHM

To solve the problem (1), it often takes us a very great amount of calculation. Besides, to meet the application necessary of the problem we can be completely satisfied with an approximate optimal solution as follow:

1 Department of Applied Mathematics Hanoi University of Technology
**Definition:** Given a sufficiently small positive number \( \varepsilon \), vector \( x_0 \in \mathbb{R}^n \) is called \( \varepsilon \)-feasible solution of CDC if:

\[
h(x_0) \leq 0, \quad g(x_0) \leq 0.
\]

And it is called \( \varepsilon \)-approximate optimal solution if:

\[
h(x_0) \leq 0, \quad g(x_0) \leq 0, \quad f(x_0) - f^* \leq 0;
\]

where \( f^* \) is the optimal value of CDC.

It is clearly, when \( \varepsilon_k \downarrow 0 \), all cluster points of sequence \( (x_{ek}) \) (\( \varepsilon_k \)-approximate optimal solutions of CDC) are exact optimal solutions of CDC. Furthermore, if an optimal solution \( w \) of the convex program Min \( \{ f(x): x \in D \} \) satisfies inequality \( g(w) \leq 0 \), then it must be an optimal solution for CDC (1) as well. Therefore, the condition \( g(w) > 0 \) is always assumed. By translating the origine if necessary, we can always suppose that:

\[
0 \in \text{int } D \cap \text{int } G \tag{2}
\]

**ALGORITHM 1 (see [1])**

**Initialization.**

Let \( \gamma = \langle c, x^* \rangle \), where \( x^* \) is the current best solution (if there is on such solution then let \( x^* = 0 \) and \( \gamma = +\infty \)). Let \( k = 1 \).

Build a polytope \( P_1 \) and its vertex set \( V_1 \), such that:

\[
\{ x \in D: \langle c, x \rangle \leq \gamma - \varepsilon \subset P_1 \subset \{ x: \langle c, x \rangle \leq \gamma - \varepsilon \}. \]

**Step \( k = 1, 2, \ldots \)**

- Compute \( x^k \in \text{arg min}\{ g(x): x \in V_k \} \). If \( g(x^k) > 0 \) then terminate.
  
  a. If \( x^k < +\infty \) \( x^* \) is an \( \varepsilon \)-approximate optimal solution of CDC.
  
  b. If \( x^k = +\infty \) then the problem has no feasible solution.

- Select \( w^k \in V_k \) that \( \langle c, w^k \rangle \leq \min \{ \langle c, x \rangle: x \in V_k + \varepsilon \}. \) If \( h(w^k) \leq \varepsilon \) and \( g(w^k) \leq 0 \) then terminate: \( w^k \) is an \( \varepsilon \)-approximate optimal solution of CDC.

- If \( h(u^k) \geq \varepsilon/2 \) then
  
  a. Let \( x^k + 1 = x^k, x^k + 1 = \gamma \); .
  
  b. Let \( p^k \in \partial h(w^k) \) (such a \( p^k \) exists because \( h(.) \) is convex, so \( \partial h(w^k) \neq \emptyset \).
     \[
     k(x) = \langle p^k, x - w^k \rangle + h(w^k) \tag{3}
     \]
  
  c. Compute the vertex set \( V_{k+1} \) of the polytope \( P_{k+1} = P_k \cap \{ x: l_k(x) \leq 0 \} \).
  
  d. Go to step \( k + 1 \).

- Select \( \gamma^k \in \{ w^k, x^k \} \) so that \( g(\gamma^k) = \varepsilon \) (\( \gamma^k \) exists since \( g(x^k) \leq 0 \) and \( g(w^k) > 0 \)). If \( h(\gamma^k) > \varepsilon \) then:
  
  a. Let \( x^k + 1 = x^k, x^k + 1 = \gamma \).
  
  b. Select \( u^k \in \{ w^k, x^k \} \) such that \( h(u^k) = \varepsilon \) (\( \gamma^k \) exists since \( h(w^k) \leq \varepsilon/2 \) and \( h(u^k) > \varepsilon \));
  
  Let \( p^k \in \partial h(u^k) \), and:
\[ l_k(x) = \langle p_k, x - u^k \rangle \]  

(c) Compute the vertex \( V_{k+1} \) of the polytope \( P_{k+1} = P_k \cap \{ x: l_k(x) \leq 0 \} \);

d. Go to step \( k + 1 \).

- If \( h(\gamma) \leq \varepsilon \) then let \( x_{k+1}^* = x_k^*, \gamma_{k+1} = \langle c, \gamma_k \rangle \).

a. If \( \langle c, w_k - \gamma_k \rangle \leq 0 \) then terminate: \( x_{k+1}^* \) is an \( \varepsilon \)-approximate optimal solution of CDC;

b. Otherwise, let

\[ l_k(x) = \langle c, x - \gamma_k \rangle + \varepsilon \]  

c. Compute the vertex set \( V_{k+1} \) of the polytope \( P_{k+1} = P_k \cap \{ x: l_k(x) \leq 0 \} \)

d. Go to step \( k + 1 \).

The finiteness of the algorithm is guaranteed by the following theorem:

**Theorem 1** ([1]). The algorithm 1 terminates after a finitely many steps by an \( \varepsilon \)-approximate optimal solution or by the evidence that the problem has no feasible solution.

**Remark**

- Algorithm 1 uses a large number of cut-hyperplanes of different types in the solution process. Therefore, the total number of vertices \( P_k \) may quickly become quite large, and it makes increasing the computational cost and amount of memory for its storage. It causes a certain difficulty in using this algorithm.

- Algorithm 1 pays a great attention in solving be convex programming problem \( \min \{ \langle c, x \rangle: x \in D \} \). Only if it had found such \( w^k \) that \( h(w^k) < \varepsilon/2 \) (it means \( w^k \) is the \( \varepsilon \)-approximate optimal solution of the above convex program) and \( g(w^k) > 0 \), then \( \gamma^k \) or \( u^k \) in turn are calculated and their cuts are built. That maybe unreasonable if the solution \( w \) did not satisfy the reverse - convex constraint.

### 3. THE MODIFICATION ALGORITHM

The following algorithm is a modification of the above algorithm. In order to prevent the number of vertices of the approximate polytopes \( P_k \) from increasing, too quickly, in each step, after finding \( x^k \) and \( w^k \) as the above algorithm we solve equation \( g(x) = 0 \) on \( \left| w^k, x^k \right| \) to find vector \( u^k \) (or let \( u^k = w^k \) if \( g(w^k) \leq 0 \)) and to cut it from \( P_{k+1} \) if \( h(u^k) > \varepsilon \). The selection of such \( u^k \) bases on the following property of CDC.

**Theorem 2.** ([1]). If convex programming problem \( \min \{ f(x): x \in D \} \) has an optimal solution \( w \), satisfies \( g(w) > 0 \), and the CDC problem (1) has a feasible solution, then there exists such an optimal solution \( x^* \) of CDC that:

\[ g(x^*) = 0 \]  

Furthermore, since \( h(x) \) is convex, so \( h(u^k) \leq \max \{ h(w^k), h(x^k) \} \); and if \( h(u^k) > \varepsilon \), then either \( x^k \) or \( w_k \) is cut from \( P_{k+1} \) as \( u^k \). That is also the reason we attempts to find an \( \varepsilon \)-approximate optimal solution satisfies the equality (6).
ALGORITHM 2

Initialization.

Build a polytope $P_1 \supset D$ and its vertex set $V_1$. Select $\varepsilon > 0$.

Let $w^1 = \arg \min \{ \langle c, x \rangle : x \in V_1 \}$.

$x^1 = \arg \min \{ g(x) : x \in V_1 \}$.

$\gamma_1 \geq \max \{ \langle c, x \rangle : x \in D \} + \varepsilon$

Step $k = 1, 2, ...$

If $g(x^k) > 0$ then terminate

If $g(w^k) \leq 0$ let $u^k = w^k$; otherwise compute $u^k \in |w^k, x^k|$ such that $g(u^k) = 0$

(u$^k$ exists because $g(w^k) > 0$ and $g(x^k) \leq 0$). There are two cases:

a. If $h(u^k) \leq \varepsilon$, let $\gamma_{k+1} = \langle c, u^k \rangle$, $x^k = u^k$, $P_{k+1} = w^k$, $x^{k+1} = \arg \min \{ g(x) : x \in P_{k+1} \}$

and go to step $k + 1$.

b. If $h(u^k) > \varepsilon$, let $p_k \in \partial h(u^k)$ (since $h(.)$ is convex, $\partial h(u^k) \neq \emptyset$),

$$l_k(x) = \langle p_k, x - u^k \rangle + h(u^k)$$

Compute the vertex set $V_{k+1}$ of the polytope

$$P_{k+1} = P_k \cap \{ x : l_k(x) \leq 0 \}$$

If $l_k(x^k) \leq 0$ then let $x^{k+1} = x^k$, otherwise compute:

$$x^{k+1} = \arg \min \{ g(x) : x \in P_{k+1}, \langle c, x \rangle \leq \gamma_{k+1} - \varepsilon \}$$

If $l_k(w^k) \leq 0$ then let $w^{k+1} = w^k$, otherwise compute:

$$w^{k+1} = \arg \min \{ \langle c, x \rangle : x \in V_{k+1} \}$$

Then go to step $k + 1$.

Theorem 3. The algorithm 3 terminates after a finite number of step and yields either $\varepsilon$-approximate optimal solution or an evidence that the problems has no feasible solution.

Proof. Suppose that the algorithm is infinite. Clearly, $P_1 \supset P_2 \supset ... \supset P_k \supset D$.

From (7) and (9), it implies $\langle c, w^k \rangle \leq \gamma_k + 1 = \langle c, u^k \rangle \leq \langle c, x^k \rangle \leq \gamma_k - \varepsilon$. It is easy to see that the case a never occurs more than $\| y^k - \langle x, w^k \rangle \| \varepsilon + 1$ time. So that, the case b must occur infinitely many times. Because $P_1$ is bounded, there exists a convergent subsequence of the sequence $\{ u^k \}$. It means that there exist two sufficiently large number $k$ and $s (s \geq k + 1)$ such that $\langle p_k, u^s - u^k \rangle > -\varepsilon$. But $h(u^k) \cdot \varepsilon$, it conflicts with (8):

$$u^s \in P_{k+1}, l_k(u^s) = \langle p_k, u^s - u^k \rangle + h(u^k) \leq 0$$

By the above contradiction, it is evident that the algorithm must be finite.
Suppose now that the algorithm terminates at step $k$. Since (7) and (9), $P_k$ has no such feasible solution $x$ that $g(x) \leq 0$ and $<c, x> \leq \gamma_k^k - \epsilon$. If the case $a$ has ever occurred then the recent $x^*$ is an $\epsilon$-approximate optimal solution. Otherwise, it means that $\gamma_k^k = \gamma^k$ since $P_k \supseteq \mathbb{D}$, it shows that the problem has no feasible solution.

The theorem has been completely proved.

4. COMPUTATIONAL EXPERIENCE

The algorithms were coded in PASCAL and run on a personal computer AT 386DX to test 12 different problems. The result is described in the following table:

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Algorithm 1</th>
<th>Algorithm 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N</td>
<td>M1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

Table: Computational result of the algorithm 1 & 2

Where:
- N: Number of variables;
- M1: Number of linear constrains, sign constrains not include;
- M2: Number of convex constrains;
- STEP: Number of iterations;
- VER 1: Maximal number of vertices of polytope $P_k$;
- VER 2: Sum of generated vertices;
- CUT 1: Number of cuts (3) and (4);
- CUT 2: Number of cuts by levels of the function $f(x)$ (in (5), (7), and (9)).
- TIME: CPU time in % of second; I/O time not includes.
From the above table we note that: because the cuts by levels of the objective function \( f(x) \) (CUT 2) are not used to make new polytopes algorithms 2, this leads to lower VER 1 and TIME. In the case when \( f(x) \) is convex, the problem may be formulated as: Minimize \( t \), \( h_1(x) = f(x) \leq 1 \) and the old constraints. Two among the tested problems cited in the above table are of this form:

**Problem 9.**

\[
\begin{align*}
\text{Minimize} & \quad \text{CUT 2} \\
\text{subject to} & \quad x_1 + x_2 \leq 30, \quad x_1 \geq 0, \quad x_2 \geq 0 \\
h(x) & = -x_1 + 18x^2/484 - 10 \leq 0 \\
g(x) & = -x^2_1 - x^2_2 + 484 \leq 0
\end{align*}
\]

The optimal solution \( x^* = (6.48079, 21.02378) \), \( f^* = 89.21686 \) with \( \varepsilon = 0.001 \).

**Problem 10.**

\[
\begin{align*}
\text{Minimize} & \quad \text{CUT 2} \\
\text{subject to} & \quad x_1 + x_2 \leq 5, \quad x_1 \geq 0, \quad x_2 \geq 0 \\
h(x) & = x^2_1 - 4.4x^1 + x^2_2 - 2.4x_2 + 4.03 \leq 0 \\
g(x) & = 4x_1 - x^2_1 - 0.36x^2_2 - 2.56 \leq 0
\end{align*}
\]

The optimal solution \( x^* = (2.77534, 1.526646) \), \( f^* = 0.87743 \) with \( \varepsilon = 0.0001 \).

**REFERENCES**


Địa chỉ liên hệ:

1. Nguyễn Trọng Toan, Khoa Cơ Bản Học Viên Khổng Quân
   54c Dương Trường Chinh Tel: 8.535 143

2. Nguyễn Đức Nghia, Khoa Công Nghệ Thông Tin ĐH BK Hà nội
   Tel (Home) 8.520 674