Tạp chí Tin học và Điều khiển học, T. 17, S. 4 (2001), 28–36

VỀ MỘT THUẬT TOÁN XẤP XỈ NGOÀI CHO BÀI TOÁN QUI HOẠCH DC DẠNG CHÍNH TẮC

NGUYỄN TRỌNG TOÀN, NGUYỄN VĂN TUẤN

Abstract. In this paper, a new outer approximation algorithm for solving canonical DC progamming problem is proposed. A table of computational experiments is also presented to compare it with some other methods.

Tóm tắt. Bài báo trình bày một thuật toán mới dạng xấp xỉ ngoài cho bài toán qui hoạch DC dạng chính tắc. Bài báo cũng đưa ra một bảng thống kê các thử nghiệm tính toán để so sánh hiệu quả của thuật toán mới so với một số thuật toán được nghiên cứu trước đó.

1. GIỚI THIỆU

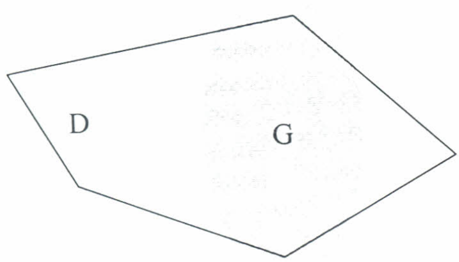
Bài toán qui hoạch DC dạng chính tắc (CDC) là bài toán tối ưu hóa sau:

$$\text{Tìm} \quad \text{Min}\,\{f(x) : x \in \Omega = D \setminus \text{int}G\}. \tag{1}$$

trong đó D và G là các tập lồi đóng, thường được viết dưới dạng $D = \{x : h(x) \le 0\}$ và $G = \{x : g(x) \ge 0\}$ với $h(x)$ là hàm lồi hữu hạn và $g(x)$ là hàm lõm trên không gian R^n; hàm mục tiêu là một hàm tuyến tính có dạng $f(x) = \langle c, x\rangle$, $c \in R^n$. Không làm mất tính tổng quát, có thể giả thiết tập D là giới nội.

Bài toán qui hoạch CDC là mô hình toán học cho nhiều bài toán ứng dụng thực tế, mặt khác nó giữ vai trò quan trọng trong việc phát triển lý thuyết tối ưu toàn cục. Người ta đã chứng minh được rằng hầu hết các bài toán tối ưu liên tục đều có thể qui dẫn về bài toán CDC. Do đó nó đã thu hút được sự quan tâm của nhiều nhà nghiên cứu (xem [1–12] và các thư mục trong đó).

Bài toán Min $\{f(x) : x \in D\}$ là bài toán qui hoạch lồi. Bài toán này đã được các nhà nghiên cứu xây dựng các thuật toán giải khá hữu hiệu. Vì vậy khó khăn chủ yếu trong việc giải bài toán CDC là sự có mặt bổ sung của ràng buộc lồi đảo $g(x) \le 0$. Nó làm cho miền chấp nhận được của bài toán trở nên không lồi, thậm chí không liên thông (xem hình 1).



Hình 1

Hiện nay đã có rất nhiều thuật toán khác nhau được đề nghị để giải bài toán trên. Tuy nhiên, việc nghiên cứu tập trung chủ yếu vào việc giải bài toán ở mức độ lí thuyết. Các thử nghiệm, phân tích, đánh giá và so sánh hiệu quả tính toán của các thuật toán đã được đề nghị là rất khó và chưa được quan tâm đúng múc. Rất ít những thí dụ đưa ra để minh họa cho các thuật toán, mà đó thường chỉ là những bài toán khá đơn giản với kích thước rất nhỏ. Nguyên nhân chính của vấn đề này là khi tăng kích thước bài toán thử nghiệm, thời gian tính toán và dung lượng bộ nhớ cần thiết của máy tính điện tử MTĐT dành cho thuật toán cũng tăng lên rất nhanh. Các thử nghiệm trên thế giới cho thấy, ngay với máy tính cỡ lớn cũng chỉ giải được bài toán này một cách hiệu quả khi kích thước bài toán nhỏ ($n \le 10$).

Bài báo này nhằm trình bày một thuật toán dạng xấp xỉ ngoài để giải bài toán trên. Trong đó cũng trình bày các thuật toán xấp xỉ ngoài của một số tác giả khác cho bài toán CDC. Mặt khác các thuật toán đã được lập trình trên PASCAL và chạy trên máy tính PC Pentium 550 MHz để thử nghiệm và so sánh hiệu quả.

2. MỘT VÀI THUẬT TOÁN XẤP XỈ NGOÀI CHO BÀI TOÁN CDC

Việc tìm lời giải chính xác cho bài toán CDC thông thường đòi hỏi khối lượng tính toán và bộ nhớ MTĐT rất lớn. Do đó, trong ứng dụng thực tế người ta có thể thỏa mãn với một lời giải xấp xỉ của bài toán theo nghĩa sau đây.

Định nghĩa. Cho trước một số ε dương và đủ bé, một vectơ $x_\varepsilon \in R^n$ được gọi là lời giải xấp xỉ ε-xấp xỉ tối ưu của bài toán CDC nếu nó thỏa mãn các điều kiện sau:

$$h(x_\varepsilon) \le \varepsilon, \quad g(x_\varepsilon) \le \varepsilon, \quad f(x_\varepsilon) - f^* \le \varepsilon, \tag{2}$$

trong đó f^* là giá trị tối ưu của bài toán CDC.

Rõ ràng là khi cho $\varepsilon \to 0$, mọi điểm tụ (điểm hội tụ của một dãy con hội tụ) của dãy $\{x_\varepsilon\}$ các lời giải ε-xấp xỉ của bài toán CDC đều là lời giải tối ưu chính xác của bài toán CDC. Vì vậy mỗi bài toán ứng dụng cụ thể, có thể chọn được một độ chính xác cần thiết.

Nếu lời giải tối ưu w của bài toán qui hoạch lồi Min $\{f(x) : x \in D\}$ thỏa mãn điều kiện $g(w) \le 0$ ($w \in \Omega$), thì đương nhiên w cũng là lời giải tối ưu của bài toán CDC. Vì vậy, không làm mất tính tổng quát luôn luôn có thể giả thiết $g(w) > 0$. Lớp các bài toán qui hoạch lồi đã có những thuật toán giải khá hiệu quả, vì vậy cũng có thể giải bài toán qui hoạch lồi trước để khẳng định giả thiết này.

Các thuật toán xấp xỉ ngoài thường dựa trên tính chất cơ bản của qui hoạch lõm là: lời giải của bài toán Min $\{g(x) : x \in D\}$ đạt được tại ít nhất một đỉnh của đa diện lồi D. Vì vậy các thuật toán xấp xỉ ngoài đầu tiên đã được xây dựng cho bài toán qui hoạch lõm (xem [12]), về sau chúng được các nhà nghiên cứu cải tiến để giải các bài toán tối ưu không lồi khác.

Thuật toán 1. (H. Tuy, xem [1, 5])

Bước khởi tạo

Đặt $\gamma_1 = \langle c, x^{*1}\rangle$, ở đây x^{*1} là lời giải tốt nhất hiện có (nếu chưa tìm được x^{*1} như vậy thì đặt $\gamma_1 = +\infty$).

Đặt $k = 1$. Xây dựng đa diện P_1 cùng với tập đỉnh V_1 của nó, sao cho:

$$\{x \in D : \langle c, x\rangle \le \gamma_1 - \varepsilon\} \subset P_1 \subset \{x : \langle c, x\rangle \le \gamma_1 - \varepsilon\}.$$

Bước $k = 1, 2, \ldots$

- Tính $x^k \in \arg\min\{g(x) : x \in V_k\}$. Nếu $g(x^k) > 0$ thì dừng:
  a. Nếu $\gamma_k < +\infty$ thì x^{*k} là lời giải ε-xấp xỉ tối ưu của bài toán CDC.
  b. Nếu $\gamma_k = +\infty$ thì bài toán CDC không có lời giải.
- Chọn $w^k \in V_k$ sao cho $\langle c, w^k\rangle \le \min\{\langle c, x\rangle : x \in V_k\} + \varepsilon$. Nếu $h(w^k) \le \varepsilon$, $g(w^k) \le \varepsilon$ thì dừng: w^k là một lời giải ε-xấp xỉ tối ưu.

- Nếu $h(w^k) \geq \varepsilon/2$ thì:
  a. Đặt $x^{*k+1} = x^{*k}$, $\gamma_{k+1} = \gamma_k$;
  b. Chọn $p^k \in \partial h(w^k)$ và xây dựng lát cắt: $l_k(x) = \langle p^k, x - w^k\rangle + h(w^k)$;
  c. Tính tập đỉnh V_{k+1} của đa diện $P_{k+1} = P_k \cap \{x : l_k(x) \leq 0\}$;
  d. Chuyển sang bước $k+1$.
- Chọn $y^k \in [w^k; x^k]$ sao cho $g(y^k) = \varepsilon$ (tồn tại y^k như vậy, vì $g(x^k) \leq 0$ và $g(w^k) > \varepsilon$). Nếu $h(y^k) > \varepsilon$ thì:
  a. Đặt $x^{*k+1} = x^{*k}$, $\gamma_{k+1} = \gamma_k$;
  b. Chọn $u^k \in [w^k; y^k]$ sao cho $h(u^k) = \varepsilon$ (tồn tại u^k như vậy vì $h(w^h) \leq \varepsilon/2$ và $h(y^k) > \varepsilon$). Chọn $p^k \in \partial h(u^k)$ và xây dựng lát cắt: $l_k(x) = \langle p^k, x - u^k\rangle$;
  c. Tính tập đỉnh V_{k+1} của đa diện $P_{k+1} \cap \{x : l_k(x) \leq 0\}$;
  d. Chuyển sang bước $k+1$.
- Nếu $h(y^k) \leq \varepsilon$ thì đặt $x^{*k+1} = x^{*k}$, $\gamma_{k+1} = \langle c, y^k\rangle$.
  a. Nếu $\langle c, w^k - y^k\rangle \geq 0$ thì dừng x^{*k+1} là một lời giải ε-xấp xỉ tối ưu.
  b. Ngược lại, xây dựng lát cắt: $l_k(x) = \langle c, x - y^k\rangle + \varepsilon$;
  c. Tính tập đỉnh V_{k+1} của đa diện $P_{k+1} = P_k \cap \{x : l_k(x) \leq 0\}$;
  d. Chuyển sang bước $k+1$.

Từ những kết quả của việc lập trình để thử nghiệm hiệu quả của thuật toán trên cho thấy:

- Thuật toán 1 sử dụng nhiều loại lát cắt trong các tình huống khác nhau và trong quá trình tính toán số lượng lát cắt được sử dụng thường khá lớn. Vì thế số đỉnh của các đa diện P_k tăng khá nhanh, dẫn đến thời gian tính toán cũng tăng và yêu cầu về bộ nhớ để lưu trữ các đỉnh cũng trở nên một trở ngại cho việc thực hiện thuật toán.

- Thuật toán 1 ưu tiên tìm lời giải ε-xấp xỉ của bài toán qui hoạch lồi Min $\{\langle c, x\rangle : x \in D\}$ trước. Tại mỗi bước k, nếu $h(w^k) \geq \varepsilon/2$ thì lát cắt theo w^k được sử dụng và chỉ khi $h(w^k) < \varepsilon/2$ và $g(x^k) \leq 0$ (tức w^k là lời giải ε-xấp xỉ cho bài toán vừa nhắc) thì vấn đề tìm y^k hay u^k mới được đặt ra và lúc đó các lát cắt theo chúng mới được sử dụng. Thứ tự ưu tiên này có lẽ sẽ là chưa hợp lý nếu như phương án w^k tìm được không thỏa mãn ràng buộc lồi đảo.

Để khắc phục các nhược điểm trên, Thuật toán 2 sau đây được nghiên cứu dựa trên nguyên tắc xây dựng các đa diện xấp xỉ ngoài và những lát cắt xấp xỉ tương tự như Thuật toán 1 và có chú ý đến những ưu điểm của thuật toán chia đôi của các tác giả N. Đ. Nghĩa và N. Đ. Hiếu [4, 6] để giảm bớt tốc độ tăng số đỉnh của các đa diện xấp xỉ P_k. Giống như Thuật toán 1, tại mỗi bước khi đã xác định được các vectơ x^k và w^k như trên, ta sẽ tìm vectơ $u^k \in [x^k; w^k]$ thỏa mãn điều kiện $g(u^k) = \varepsilon$ hoặc đặt $u^k = w^k$ nếu $g(w^k) \leq \varepsilon$, sau đó cắt nó khỏi P_{k+1} nếu $h(u^k) > \varepsilon$. Việc chọn u^k như vậy để xem xét được dựa trên cơ sở tính chất quan trọng sau đây của bài toán CDC:

Định lý 1. (xem [1]) *Nếu lời giải w của bài toán qui hoạch lồi* Min $\{\langle c, x\rangle : x \in D\}$ *thỏa mãn bất đẳng thức $g(w) > 0$ và bài toán CDC có lời giải thì tồn tại ít nhất một lời giải x^* của bài toán CDC sao cho $g(x^*) = 0$.*

Mặt khác, do hàm $h(\cdot)$ lồi nên $h(u^k) \leq \max\{h(w^k), h(x^k)\}$, vì vậy nếu $h(u^k) > \varepsilon$ thì hoặc x^k hoặc w^k sẽ bị cắt khỏi P_{k+1} cùng với u^k bởi lát cắt được xây dựng đối với u^k. Còn khi $h(u^k) \leq \varepsilon$, vì $g(u^k) \leq \varepsilon$, nên u^k là một lời giải ε-xấp xỉ chấp nhận được của bài toán CDC. Như vậy không cần thiết phải xây dựng các lát cắt riêng cho x^k và w^k như trong Thuật toán 1. Tính hội tụ của Thuật toán 2 sau đây có thể được chứng minh hoàn toàn tương tự trong Thuật toán 1. Kết quả thử nghiệm cho thấy Thuật toán 2 có nhiều ưu điểm về tốc độ tính toán và bộ nhớ MTĐT so với Thuật toán 1 và thuật toán chia đôi đã nói ở trên (xem [8–10]).

Thuật toán 2

Bước khởi tạo

Xây dựng đa diện $P_1 \supset D$ với tập đỉnh V_1 của nó. Chọn $\varepsilon > 0$. Đặt

$$w^1 = \arg\min\{\langle c, x\rangle : x \in V_1\}, \quad x^1 = \arg\min\{g(x) : x \in V_1\}.$$

Chọn $\gamma_1 \geq \max\{\langle c, x\rangle : x \in V_1\}$.

Bước $k = 1, 2, \ldots$

- Nếu $g(x^k) > \varepsilon$ hoặc không tồn tại thì dừng. Có 2 trường hợp xảy ra:
  a. Nếu đã có một lời giải chấp nhận được x^*, thì x^* là lời giải ε-xấp xỉ tối ưu.
  b. Ngược lại, bài toán không có lời giải chấp nhận được.
- Nếu $g(w^k) \leq \varepsilon$ thì chọn $u^k = w^k$; ngược lại tìm $u^k \in [w^k, x^k]$ thỏa mãn $g(u^k) = \varepsilon$ (phương trình có nghiệm vì $g(w^k) > \varepsilon$ và $g(x^k) \leq \varepsilon$). Có 2 trường hợp xảy ra:
  a. Nếu $h(u^k) \leq \varepsilon$, thì u^k là một lời giải chấp nhận được. Đặt $x^* = u^k$, $\gamma_{k+1} = \langle c, u^k\rangle$, $P_{k+1} = P_k$, $w^{k+1} = w^k$, $x^{k+1} = arg\min\{g(x) : x \in P_{k+1}, \langle c, x\rangle \leq \gamma_{k+1} - \varepsilon\}$, rồi chuyển sang bước $k+1$.
  b. Nếu $h(u^k) > \varepsilon$. Lấy $p^k \in \partial h(u^k)$ (do $h(\cdot)$ là hàm lồi nên $\partial h(x^k) \neq \emptyset$). xây dựng đa diện P_{k+1} bằng cách bổ sung vào P_k ràng buộc cắt:

$$l_k(x) = \langle p^k, x - u^k\rangle + h(u^k) \leq 0.$$

  Tính tập đỉnh V_{k+1} của đa diện P_{k+1}. Đặt $\gamma_{k+1} = \gamma_k$.
  Nếu $l_k(x^k) \leq 0$ thì đặt $x^{k+1} = x^k$, ngược lại tính:

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

  Nếu $l_k(w^k) \leq 0$ thì đặt $w^{k+1} = w^k$, ngược lại tính

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

  Chuyển sang bước $k+1$.

3. THUẬT TOÁN CẢI TIẾN VÀ KẾT QUẢ THỬ NGHIỆM TRÊN MTĐT

Việc xây dựng thuật toán mới được dựa trên một tính chất quan trọng sau đây của lời giải tối ưu của bài toán CDC:

Định lý 2. *Giả sử lời giải tối ưu cực biên w của bài toán qui hoạch lồi $Min\{\langle c, x\rangle : x \in D\}$ thỏa mãn bất đẳng thức $g(w) > 0$ và bài toán CDC có lời giải thì tồn tại ít nhất một lời giải tối ưu x^* của bài toán CDC sao cho $g(x^*) = 0$ và $h(x^*) = 0$.*

Chứng minh. Bằng phản chứng: Giả dử không tồn tại lời giải tối ưu x^* như vậy. Trước hết cần khẳng định tập $\{x : g(x) = 0, h(x) = 0\} \neq \emptyset$. Bởi vì nếu xảy ra trường hợp ngược lại thì do D và G là hai tập lồi cùng chứa w nên ta chỉ cần xét 2 khả năng D nằm hoàn toàn trong G hoặc G nằm hoàn toàn trong D.

- Trường hợp 1: D nằm hoàn toàn trong G. Khi đó $\Omega = D \setminus \mathrm{int}G = \emptyset$. Điều này mâu thuẫn với giả thiết bài toán CDC có lời giải.

- Trường hợp 2: G nằm hoàn toàn trong D. Điều này cũng mâu thuẫn với giải thiết w lời giải tối ưu cực biên của bài toán Min $\{\langle c, x\rangle : x \in D\}$ và $g(w) > 0$.

Do đó, theo Định lý 1 tồn tại lời giải tối ưu x^1 của bài toán CDC thỏa mãn $g(x^1) = 0$ và $h(x^1) < 0$. Giả sử x^* là một lời giải tối ưu của bài toán Min $\{f(x) : g(x) = 0, h(x) = 0\}$. Theo giả thiết phản chứng thì $f(x^1) < f(x^*)$.

Xét một vectơ $x^2 \in R^n$ thỏa mãn $x^1 = \lambda x^2 + (1-\lambda)x^*$, với $0 < \lambda < 1$. Vì $g(\cdot)$ là hàm lõm nên $g(x^2) \leq 0$. Vì $h(x^1) < 0$, nếu chọn λ khá gần 1 thì x^2 khá gần x^1 và do đó $h(x^2) < 0$, nên $x^2 \in \Omega$. Hơn nữa do hàm $f(x)$ là đơn điệu và $f(x^1) < f(x^*)$ nên $f(x^2) < f(x^1)$. Điều này mâu thuẫn với giả thiết x^1 là lời giải tối ưu của bài toán CDC, chứng tỏ giả thiết phản chứng là không đúng. Vì vậy phải tồn tại ít nhất một lời giải tối ưu x^* của bài toán CDC sao cho $g(x^*) = 0$ và $h(x^*) = 0$.

Định lý 2 rõ ràng mạnh hơn Định lý 1 vì có thêm kết luận $h(x^*) = 0$. Hơn nữa, từ đó dễ thấy: nếu D là một đa diện thì x^* hoặc là một đỉnh của D hoặc là giao điểm của một cạnh của D với mặt cong $g(x) = 0$. Vì vậy có thể chỉ cần tìm lời giải của bài toán CDC tại các điểm như vậy.

Thuật toán 3

Bước khởi tạo

Xây dựng đa diện $P_1 \supset D$ với tập đỉnh V_1. Chọn $\varepsilon > 0$.
Đặt $w^1 = \arg\min\{\langle c, x\rangle : x \in V_1\}$.

Bước $k = 1, 2, \ldots$

Có 2 trường hợp xảy ra:

1. Nếu $g(w^k) \le \varepsilon$. Có 2 trường hợp xảy ra:
   a. Nếu $h(w^k) \le \varepsilon$. Dừng thuật toán và $x^* = w^k$ là lời giải ε-xấp xỉ tối ưu.
   b. Nếu $h(w^k) > \varepsilon$. Lấy $p^k \in \partial h(w^k)$ (do $h(\cdot)$ là hàm lồi nên $\partial h(w^k) \ne \emptyset$). Xây dựng P_{k+1} bằng cách bổ sung vào P_k ràng buộc cắt:
   $$l_k(x) = \langle p^k, x - w^k\rangle + h(w^k) \le 0.$$
   Tính tập đỉnh V_{k+1} của đa diện P_{k+1}. Tính $w^{k+1} = \arg\min\{\langle c, x\rangle : x \in V_{k+1}\}$ và chuyển sang bước $k+1$.
2. Nếu $g(w^k) > \varepsilon$. Tính:
   $$u^k = \arg\min\{\langle c, x\rangle : x \in E_k \text{ (tập các điểm trên cạnh của } P_k), \ g(x) \le \varepsilon\}. \tag{3}$$
   Có 3 trường hợp xảy ra:
   a. Nếu u^k không tồn tại: Dừng thuật toán, bài toán không có lời giải.
   b. Nếu $h(u^k) \le \varepsilon$: Dừng thuật toán $x^* = u^k$ là lời giải ε-xấp xỉ tối ưu.
   c. Nếu $h(u^k) > \varepsilon$: Lấy $p^k \in \partial h(u^k)$ (do $h(.)$ là hàm lồi nên $\partial h(u^k) \ne \emptyset$). Xây dựng P_{k+1} bằng cách bổ sung vào P_k ràng buộc cắt:
   $$l_k(x) = \langle p^k, x - u^k\rangle + h(u^k) \le 0.$$
   Tính tập đỉnh V_{k+1} của đa diện P_{k+1}. Tính $w^{k+1} = \arg\min\{\langle c, x\rangle : x \in V_{k+1}\{$ và chuyển sang bước $k+1$.

Trong các thuật toán xấp xỉ ngoài đã nêu, để tính tập đỉnh mới V_{k+1} của đa diện P_{k+1} từ tập đỉnh V_k của đa diện P_k khi bổ sung một ràng buộc cắt $l_k(x)$ đã sử dụng kỹ thuật của các tác giả T. V. Thiệu, B. T. Tâm và V. T. Bản trình bày trong [12].

Một điều cần chú ý trong mỗi bước lặp của Thuật toán 3, để tìm phương án u^k của bài toán (3), có thể giải rất nhiều phương trình $g(x) = \varepsilon$ trên các cạnh của đa diện P_k và so sánh giá trị của hàm mục tiêu trên các nghiệm đó. Mỗi lần giải phương trình có thể sẽ làm thay đổi phương án tốt nhất hiện có và tạo ra cận dưới mới cho giá trị hàm mục tiêu. Tuy nhiên, trong thực hành lập trình chúng tôi sử dụng phương pháp dây cung để giải lặp các phương trình đó. Do hàm $g(x)$ lõm nên sau mỗi bước lặp hàm $g(x)$ giảm dần. Ta chỉ cần giải phương trình trên các cạnh có hàm mục tiêu tăng dần. Vì vậy có rất nhiều phương trình $g(x) = \varepsilon$ không cần phải giải hoặc không cần giải đến cùng nếu lời giải xấp xỉ hiện thời làm cho hàm mục tiêu lớn hơn cận dưới đã có. Chính điều này làm giảm đáng kể khối lượng tính toán của thuật toán.

Để nghiên cứu hiệu quả của thuật toán mới, chúng tôi đã tiến hành lập trình trên PASCAL đối với các thuật toán 1, 2, 3 và thuật toán chia đôi của các tác giả N. Đ. Nghĩa và N. Đ. Hiếu [4, 6] và thử nghiệm gần 100 bài toán mẫu trong công trình [5] với 7 kiểu hàm lồi đảo $g(x)$ khác nhau. Kết quả thử nghiệm và so sánh Thuật toán 2 với các Thuật toán 1 và thuật toán chia đôi cho thấy

Thuật toán 2 có nhiều ưu điểm (xem [8–10]). Kết quả thử nghiệm của hai thuật toán 2 và 3 được thống kê trong bảng dưới đây. Các tham số trong bảng có ý nghĩa như sau:

- N : Số biến của bài toán;
- M : Số ràng buộc tuyến tính, không kể các ràng buộc về dấu;
- Mh : Số ràng buộc phi tuyến lồi;
- Vmax : Số đỉnh lớn nhất của các đa diện P_k;
- Cut : Số lát cắt được xây dựng theo các ràng buộc lồi;
- Time : Thời gian tính toán trên CPU, không kể thời gian nhập liệu, đơn vị đo là giây.

Kết quả được thống kê trong bảng cho thấy hiệu quả của thuật toán mới đề nghị nói chung tốt hơn Thuật toán 2 cả về thời gian tính trên MTĐT (Time) lẫn dung lượng bộ nhớ cần thiết (Vmax) của từng bài toán trong đa số các bài toán được thử nghiệm. Đặc biệt sự chênh lệch về Time và Vmax của hai thuật toán trong nhiêu bài toán là rất lớn. Hãy xem trong bảng số liệu về các bài toán ht7, ht8, ht15, ht18, ht20, ng2, ng6, ng9, ng29, tt7, tt9, vd20...

Bài toán	Kích thước			Thuật toán 2			Thuật toán 3		
	N	M	Mh	Vmax	Cut	Time	Vmax	Cut	Time
ht1	5	10	2	100	22	1,20	100	22	1,32
ht2	5	12	2	69	7	0,88	46	5	0,06
ht3	8	6	1	709	38	97,05	709	38	97,16
ht4	6	12	2	187	29	9,01	223	26	10,10
ht5	7	12	2	314	31	28,13	468	30	61,13
ht6	6	12	2	180	17	3,07	222	13	0,72
ht7	7	10	0	254	9	8,18	88	4	0,11
ht8	7	10	1	612	19	79,75	158	7	0,66
ht9	8	12	0	132	5	1,15	62	3	0,05
ht10	9	12	0	596	9	8,68	528	8	7,09
ht11	10	8	0	304	5	13,84	304	5	2,53
ht12	8	9	2	854	19	41,91	854	19	42,40
ht13	7	10	0	102	5	0,16	70	4	0,11
ht14	10	12	0	892	9	23,67	330	5	2,63
ht15	8	10	1	749	23	111,28	651	21	61,79
ht17	8	12	1	781	32	96,45	982	31	126,05
ht18	8	6	2	414	24	40,86	240	19	9,28
ht19	8	10	0	241	8	36,25	173	7	1,49
ht20	8	6	1	937	28	156,26	752	24	88,05
ht21	8	10	0	221	8	1,92	184	7	0,76
ng1	6	8	1	284	49	24,17	1292	54	40,87
ng2	8	6	1	884	20	100,62	408	14	13,24
ng3	8	6	1	911	33	110,07	402	26	17,03
ng4	9	10	0	210	5	0,93	210	5	0,99
ng5	10	8	1	453	5	2,25	143	3	0,11
ng6	6	10	2	301	26	21,53	97	15	0,66

Bài toán	Kích thước			Thuật toán 2			Thuật toán 3		
	N	M	Mh	Vmax	Cut	Time	Vmax	Cut	Time
ng7	10	12	0	490	7	5,93	490	7	5,27
ng8	6	10	2	209	23	7,14	102	15	0,55
ng9	7	15	2	864	56	456,93	594	40	99,85
ng10	7	12	2	404	29	23,07	272	23	9,66
ng11	4	15	3	55	19	0,22	33	14	0,05
ng12	5	15	2	24	9	0,00	24	9	0,00
ng14	5	20	3	160	38	7,31	264	43	13,95
ng15	6	12	2	234	27	8,73	220	26	10,11
ng16	7	8	2	224	30	14,94	156	24	2,53
ng17	8	10	0	388	8	12,80	333	7	3,02
ng18	9	10	0	322	5	1,48	208	4	0,49
ng22	5	10	2	150	23	4,06	81	10	0,44
ng24	5	15	3	70	24	0,72	84	24	1,21
ng25	5	10	1	96	25	1,04	134	25	1,87
ng26	6	9	2	503	35	58,50	156	20	5,99
ng27	5	15	3	102	29	1,70	146	35	6,04
ng28	6	14	2	119	21	1,87	179	25	7,86
ng29	6	8	2	388	31	17,36	76	13	0,55
ng30	6	12	2	295	24	19,34	722	30	135,01
tt1	4	3	1	8	2	0,00	5	0	0,00
tt2	4	5	1	5	1	0,06	5	1	0,05
tt3	5	7	2	144	18	1,60	82	11	0,61
tt4	5	6	1	48	5	0,11	20	2	0,00
tt5	5	7	4	32	18	0,22	62	16	0,33
tt6	6	7	3	255	42	15,93	246	37	18,35
tt7	7	8	3	982	30	200,64	794	22	74,86
tt8	8	7	1	206	5	3,46	206	5	0,83
tt9	8	9	5	470	12	27,52	247	8	1,98
tt10	6	7	1	72	18	0,94	85	18	0,93
tt11	8	9	5	1002	19	190,21	925	18	99,63
tt12	5	7	3	128	32	3,52	146	29	3,57
tt13	6	10	3	92	7	0,44	91	8	0,82
tt14	12	8	0	449	4	59,27	449	4	1,87
tt15	14	8	0	693	4	3,73	693	4	3,73
tt16	15	5	0	600	3	1,37	70	1	0,00
tt18	5	6	3	618	68	141,71	562	61	199,21
tt19	5	6	3	114	16	1,92	84	12	0,38
tt20	6	8	1	283	27	17,08	71	15	0,71
vd1	2	3	0	3	1	0,00	3	1	0,00

Bài toán	Kích thước			Thuật toán 2			Thuật toán 3		
	N	M	Mh	Vmax	Cut	Time	Vmax	Cut	Time
vd2	8	6	0	24	1	0,00	24	1	0,00
vd3	2	4	0	3	0	0,00	3	0	0,00
vd4	2	5	0	3	0	0,00	3	0	0,00
vd5	3	8	0	7	4	0,00	6	3	0,00
vd6	2	5	0	5	3	0,00	4	2	0,00
vd7	2	1	2	6	6	0,00	6	6	0,00
vd8	2	4	0	4	2	0,00	4	2	0,00
vd9	3	1	2	24	13	0,06	28	20	0,05
vd10	3	1	2	28	15	0,11	32	22	0,17
vd11	3	1	2	40	20	0,16	48	27	0,50
vd12	3	3	1	4	1	0,00	4	1	0,00
vd13	5	6	1	24	7	0,06	22	5	0,06
vd14	2	5	0	5	3	0,00	4	2	0,00
vd15	5	8	0	18	3	0,06	18	3	0,00
vd16	2	1	1	7	7	0,00	6	5	0,00
vd17	8	12	0	139	5	0,55	139	5	0,55
vd18	7	9	0	36	2	0,06	36	2	0,06
vd19	5	10	1	116	20	3,51	74	13	0,66
vd20	9	8	1	724	17	60,09	188	9	0,76
vd21	9	6	0	120	3	0,16	120	3	0,11
vd22	8	8	0	51	2	0,00	51	2	0,05
vd23	8	10	0	116	4	0,22	121	4	0,16
vd24	8	12	0	289	7	2,14	202	6	0,77
vd25	8	15	0	151	5	0,44	151	5	0,44
vd26	8	15	0	380	7	3,78	393	6	2,09
vd27	8	15	0	142	4	1,70	96	3	0,33
vd28	8	15	0	401	9	5,33	256	7	1,43
vd29	8	16	0	197	6	1,43	145	5	0,55
vd30	8	9	0	237	6	1,26	237	6	1,32

TÀI LIỆU THAM KHẢO

[1] H. Tuy, Canonical DC programming problem: Outer approxiamtion methods revisited, *Operation Research Letters* **18** (1995) 99–106.

[2] H. Tuy, Convex program with an additional reverse convex constraint, *J. Optim. Theory Appl.* **52** (1987) 463–485.

[3] L. D. Muu, A convergent algorithm for solving linear programming with an additional convex constraint, *Kybernetika* **21** (1985) 428–435.

[4] N. D. Nghia and N. D. Hieu, A method for solving reverse convex programming problems, *Acta Math. Vietnam.* **2** (1986) 241–252.

[5] N. Đ. Nghĩa, Xây dựng chương trình giải qui hoạch dạng chính tắc bằng thuật toán Hoàng Tụy, Báo cáo kết quả thực hiện đề tài "Bộ chương trình tối ưu toàn cục", mã số 1.4.3, chủ nhiệm đề tài Hoàng Tụy, Hà Nội, 1996.

[6] N. Đ. Nghia, N. Đ. Hiếu, Về thuật toán Hoàng Tụy giải qui hoạch lồi với một ràng buộc lồi đảo bổ sung và một số kết quả thử nghiệm trên máy tính, *Tạp chí Toán học* **XV** (2) (1987) 3–8.

[7] N. Đ. Nghĩa, N. Đ. Hiếu, Thuật toán giải bài toán qui hoạch tuyến tính với một ràng buộc lồi đảo, *Tuyển tập các công trình nghiên cứu khoa học - Toán*, ĐHBK Hà Nội, 1984.

[8] N. T. Toan, A modification of Tuy's algorithm for canonical DC programming problem, *J. Computer Science and Cybernetics* **1** (1998) 34–39.

[9] N. T. Toàn, "Xây dựng thuật toán hữu hiệu giải một số bài toán tối ưu với cấu trúc đặc biệt", Luận án Tiến sĩ, Hà Nội, 1998.

[10] N. T. Toàn, N. Đ. Nghĩa, Thử nghiệm, so sánh và cải biên một số thuật toán giải bài toán qui hoạch lồi đảo dạng chính tắc, *Tuyển tập các báo cáo khoa học tại Hội thảo khoa học toàn quốc lần 1 về "Tối ưu và Điều khiển"*, Qui Nhơn, 1996, 155–163.

[11] R. Horst and T. Tuy, *Global Optimization* (deterministic approaches), 1st ed. 1990) 2nd ed., Springer, Berlin, 1993.

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

Nhận bài ngày 15 - 3 - 2001.
Nhận lại sau khi sửa ngày 25 - 6 - 2001.

Bộ môn Tin học, Học viện Phòng không - Không quân.