GHI CHÚ BÀI TẬP 1
QUY HOẠCH TUYẾN TÍNH 2025

Lê Hoàng Vũ - 22120461

Mục lục

CHƯƠNG

I

Bài tập tuần 1

Bảng 1. Nội dung của CHƯƠNG I

SECTION 1

Yêu cầu bài toán

Sau đây là nội dung cụ thể của bài toán lấy điểm cộng của tuần 1, cũng là một phần của bài toán Quy hoạch tuyến tính:

Bài toán 1 Viết `function` để nhập/xuất dữ liệu từ bàn phím: số nguyên dương M và danh sách M điều kiện ($M \leq 10$, các số thuộc miền $[-50; 50]$) dạng

$$a[i] \cdot x_1 + b[i] \cdot x_2 \leq d[i]$$

và `minimize/maximize` hàm mục tiêu

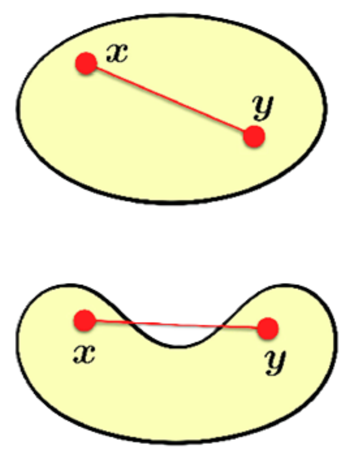
$$F = c_1 \cdot x_1 + c_2 \cdot x_2$$

(mặc định $x_1, x_2 \geq 0$); trả lời các câu hỏi sau:

1. Danh sách các điểm cực biên.
2. Miền ràng buộc có bị chặn hay không?
3. GTNN và GTLN tìm được (có thể không có).

Hiện nay, có nhiều thư viện hỗ trợ giải bài toán này một cách tự động mà không cần thực hiện các bước thủ công, chẳng hạn như `scipy.optimize.linprog`, `cvxpy`, `pulp`, hay `gurobi`. Những thư viện này cung cấp các công cụ mạnh mẽ để giải quyết bài toán tối ưu hóa tuyến tính một cách nhanh chóng và chính xác.

Tuy nhiên, đối với bài tập này, em chọn tiếp cận theo hướng sử dụng các kiến thức cơ sở từ môn Quy hoạch tuyến tính để giải bài toán mà không sử dụng các thư viện trên. Cách tiếp cận này giúp hiểu rõ bản chất của phương pháp và rèn luyện kỹ năng xây dựng và giải quyết bài toán tối ưu từ đầu.



Hình 1. Minh họa tập lồi (ở trên) và không phải tập lồi (ở dưới).

SECTION 2

Cơ sở lý thuyết

Định nghĩa 1 (Tập lồi) Một tập $\Gamma \subset \mathbb{R}^n$ là **tập lồi** nếu Γ chứa hai điểm nào đó thì Γ cũng chứa cả đoạn thẳng nối hai điểm ấy. Nói cách khác, Γ là lồi nếu:

$$\begin{cases} x_1, x_2 \in \Gamma \\ \lambda \in \mathbb{R}, 0 \leq \lambda \leq 1 \end{cases} \Rightarrow (1-\lambda)x_1 + \lambda x_2 \in \Gamma.$$

Đầu tiên là lý thuyết nền tảng mở ra môn học Quy hoạch tuyến tính này - Tập lồi (`Convex set`). Ở cấp 2 chúng ta đã được học về đa giác lồi và lõm, và được giải thích nôm na là một đa giác là lồi nếu khi chúng ta lấy một cạnh của đa giác thì cả đa giác sẽ nằm về đúng một bên của cạnh đó. Định nghĩa về tập lồi này cũng tương tự, vì nếu ta để ý kĩ, $(1-\lambda)x_1 + \lambda x_2$ chính là đoạn thẳng nối hai điểm x_1, x_2.

Định nghĩa 2 (Tổ hợp lồi) Cho các điểm $x^1, x^2, \ldots, x^k \in \mathbb{R}^n$, điểm $x \in \mathbb{R}^n$ có dạng

$$x = \sum_{i=1}^{k} \lambda_i x^i \quad \text{với } \lambda_i \geq 0 \text{ và } \sum_{i=1}^{k} \lambda_i = 1,$$

là tổ hợp lồi của các điểm $x^1, x^2, \ldots, x^k$.

Định nghĩa 3 (Điểm cực biên) Cho tập lồi $S \subset \mathbb{R}^n$. Một điểm $x \in M$ được gọi là **điểm cực biên** (`extreme point`) của M nếu x không thể biểu diễn được dưới dạng tổ hợp lồi chặt của hai điểm phân biệt bất kì nào của M, nghĩa là

$$\nexists y, z \in M, y \neq z \text{ sao cho } x = \lambda y + (1-\lambda) z \text{ với } 0 < \lambda < 1.$$

Định nghĩa 4 (Siêu phẳng - `Hyperplane`) Trong không gian $\mathbb{R}^n$, một **siêu phẳng** H được định nghĩa là tập hợp các điểm thỏa mãn phương trình:

$$H = \{\mathbf{x} \in \mathbb{R}^n \mid \langle \boldsymbol{\alpha}, \mathbf{x} \rangle = b\}$$

trong đó $\boldsymbol{\alpha} \in \mathbb{R}^n$ là vector pháp tuyến của siêu phẳng còn $b \in \mathbb{R}$ là một hằng số.

Từ định nghĩa (4), ta nhận thấy siêu phẳng trong không gian $\mathbb{R}^2$ chính là một đường thẳng, có phương trình:

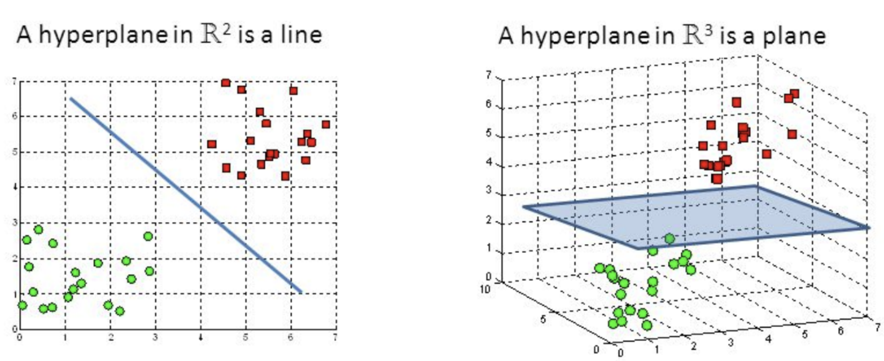
$$a_1 x_1 + a_2 x_2 = b.$$

Ở đây, $\boldsymbol{\alpha} = (a_1, a_2)^\top$ là **vector pháp tuyến**, vuông góc với đường thẳng. Đường thẳng này chia mặt phẳng thành hai nửa không gian.

Mặt khác, siêu phẳng trong không gian $\mathbb{R}^3$ lại chính là một mặt phẳng, có phương trình:

$$a_1 x_1 + a_2 x_2 + a_3 x_3 = b.$$

Tương tự như trên, $\boldsymbol{\alpha} = (a_1, a_2, a_3)^\top$ là **vector pháp tuyến**, xác định hướng của mặt phẳng. Mặt phẳng này chia không gian $\mathbb{R}^3$ thành hai nửa không gian.



Định nghĩa 5 (Nửa không gian đóng - mở) Gọi siêu phẳng H với $\mathbf{a} \in \mathbb{R}^n$ là vector pháp tuyến của siêu phẳng còn $b \in \mathbb{R}$ là một hằng số.

Ta định nghĩa tập $H = \{\mathbf{x} \in \mathbb{R}^n \mid \langle \mathbf{a}, \mathbf{x} \rangle \leqslant b\}$ (hoặc $\langle \mathbf{a}, \mathbf{x} \rangle \geqslant b$ là nửa không gian đóng. Nếu không có dấu $=$ thì là nửa không gian mở.

Định nghĩa 6 Tập lồi đa diện $P \subset \mathbb{R}^n$ là giao của một số hữu hạn các nửa không gian đóng. Nghĩa là nó là tập nghiệm của một hệ hữu hạn các bất đẳng thức tuyến tính

$$\langle a^i, x \rangle \geq b_i, \quad i = 1, 2, \ldots, k.$$

Mỗi bất đẳng thức trong hệ trên được gọi là *một ràng buộc*.

Định lý 1 (Đặc trưng của điểm cực biên) Cho tập đa diện $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ và $\bar{x} \in P$. Khi đó, các điều kiện sau là tương đương:

(i) $\bar{x}$ là một điểm cực biên của P.

(ii) Tồn tại $c \in \mathbb{R}^n$ sao cho $\bar{x}$ là nghiệm tối ưu duy nhất của bài toán quy hoạch tuyến tính, chẳng hạn như $\max\{c^T x : x \in P\}$.

Proof **Chiều thuận $(i) \Rightarrow (ii)$**: Giả sử $\bar{x}$ là điểm cực biên của P. Khi đó, theo định nghĩa, không tồn tại hai điểm phân biệt $x, y \in P$ và hệ số $0 < \lambda < 1$ sao cho $\bar{x} = \lambda x + (1 - \lambda)y$.

Do đó, nếu chọn c là một véc-tơ sao cho $c^T x$ đạt cực đại tại $\bar{x}$, thì $\bar{x}$ phải là nghiệm tối ưu duy nhất của bài toán $\max\{c^T x : x \in P\}$.

Chiều đảo $(ii) \Rightarrow (i)$: Giả sử $\bar{x}$ là nghiệm tối ưu duy nhất của bài toán quy hoạch tuyến tính $\max\{c^T x : x \in P\}$. Giả sử ngược lại rằng $\bar{x}$ có thể được biểu diễn dưới dạng tổ hợp lồi chặt của hai điểm phân biệt $x, y \in P$, tức là

$$\bar{x} = \lambda x + (1 - \lambda)y, \quad 0 < \lambda < 1.$$

Do tính chất tuyến tính của hàm mục tiêu, ta có:

$$c^T \bar{x} = \lambda c^T x + (1 - \lambda)c^T y \leq \lambda c^T \bar{x} + (1 - \lambda)c^T \bar{x} = c^T \bar{x}.$$

Suy ra $c^T x = c^T \bar{x}$ và $c^T y = c^T \bar{x}$, nghĩa là x và y cũng là nghiệm tối ưu của bài toán, mâu thuẫn với giả thiết rằng $\bar{x}$ là nghiệm tối ưu duy nhất. Do đó, $\bar{x}$ phải là điểm cực biên của P. $\square$ $\square$

Hệ quả 1 Gọi D là tập các phương án của bài toán `QHTT` và là tập lồi đa diện. Khi đó, hàm mục tiêu của bài toán quy hoạch tuyến tính sẽ đạt `min/max` tại một điểm cực biên của tập D. Nếu hàm mục tiêu không chỉ nhận `min/max` tại một điểm cực biên của tập lồi D mà tại nhiều điểm thì nó sẽ đạt giá trị `min/max` tại những điểm là tổ hợp tuyến tính lồi của các điểm đó.

Định nghĩa 7 Cho bài toán quy hoạch tuyến tính tổng quát: Tìm $\min$ hoặc $\max f(\vec{x}) = c^T\vec{x}$ thoả mãn $A\vec{x} \geq \vec{b}, \quad \vec{x} \geq 0.$

Khi đó một vector $\vec{x} \in \mathbb{R}^n$ được gọi là một **phương án chấp nhận được/ khả thi** (`feasible solution`) nếu nó thoả mãn tất cả các ràng buộc của bài toán, tức là:

$$A\vec{x} \geq \vec{b}, \quad \vec{x} \geq 0.$$

Tập hợp tất cả các phương án chấp nhận được gọi là **tập chấp nhận được/ miền khả thi** (`feasible region`).

Trong khi đó, một phương án chấp nhận được $\vec{x}^*$ được gọi là *phương án tối ưu* nếu nó cho giá trị nhỏ nhất/ lớn nhất (tùy theo yêu cầu bài toán) của hàm mục tiêu $f(\vec{x})$ trong toàn bộ tập chấp nhận được, tức là:

$$f(\vec{x}^*) \leq f(\vec{x}), \quad \forall \vec{x} \in \text{tập chấp nhận được}.$$

Định lý 2 (Tìm điểm cực biên) Cho tập lồi đa diện $M = \{u \in \mathbb{R}^n \mid Au \geq b\}$, trong đó A là ma trận cấp $m \times n$. Một điểm $v \in M$ là điểm cực biên của M khi và chỉ khi tồn tại một tập hợp con $\{i_1, i_2, \ldots, i_n\} \subset \{1, 2, \ldots, m\}$ sao cho các véc-tơ hàng tương ứng $\{a_{i_1}, a_{i_2}, \ldots, a_{i_n}\}$ của ma trận A là độc lập tuyến tính, đồng thời v là nghiệm duy nhất của hệ phương trình:

$$\langle a_{i_k}, u\rangle = b_{i_k}, k = 1, 2, \ldots, n.$$

Xét bài toán quy hoạch tuyến tính (QHTT) tổng quát:

$$(I) \quad \begin{cases} f(u) = \langle c, u\rangle \to \min \\ Au \geq b, \quad u \in \mathbb{R}^n \end{cases} \tag{2.1}$$

trong đó A là ma trận kích thước $m \times n$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$. Và:

- M là tập các phương án chấp nhận được của (I).
- $ext(M)$ là tập các điểm cực biên của M.
- M^* là tập các phương án tối ưu của (I).

Nhận xét 1 Tập $M^* \neq \emptyset$ khi và chỉ khi $M \neq \emptyset$ và hàm mục tiêu $f(u)$ bị chặn dưới trên M (nếu $f(u) \to \max$ thì điều kiện bị chặn dưới của $f(u)$ được thay bằng điều kiện bị chặn trên)

Từ đó ta rút ra được một thuật toán đơn giản để giải bài toán `QHTT` bằng điểm cực biên như sau:

- **Bước 1**: Kiểm tra $M^* \neq \emptyset$.
- **Bước 2**: Tìm tập $ext(M)$.
- **Bước 3**: Xác định $\min\{f(u), u \in ext(M)\}$ hoặc $\max\{f(u), u \in ext(M)\}$.

Định nghĩa 8 Cho S là một tập lồi đóng, khác rỗng trong $\mathbb{R}^n$. Một vectơ khác không $\vec{d} \in \mathbb{R}^n$ được gọi là một **hướng** (hay hướng suy biến) của S nếu với mọi $\vec{x} \in S$, ta có

$$\vec{x} + \lambda\vec{d} \in S, \quad \forall\lambda > 0.$$

Nói cách khác, nếu ta di chuyển từ một điểm bất kỳ trong S theo hướng $\vec{d}$ với một khoảng cách dương, ta vẫn ở trong S.

Hai hướng $\vec{d}_1$ và $\vec{d}_2$ được gọi là **phân biệt** nếu chúng không tỉ lệ dương với nhau, tức là

$$\vec{d}_1 \neq \alpha\vec{d}_2, \quad \forall\alpha > 0.$$

Một hướng $\vec{d}$ của S được gọi là **hướng cực biên** nếu nó không thể được biểu diễn như một tổ hợp tuyến tính dương của hai hướng phân biệt. Cụ thể, nếu tồn tại $\lambda_1, \lambda_2 > 0$ sao cho

$$\vec{d} = \lambda_1\vec{d}_1 + \lambda_2\vec{d}_2$$

thì khi đó phải có $\vec{d}_1 = \alpha\vec{d}_2$ với một số $\alpha > 0$, tức là hai hướng đó thực chất chỉ là một hướng được nhân với một hệ số dương.

Bổ đề 1 Một hướng $d \in D$ là một hướng cực biên của S khi và chỉ khi d là một điểm cực biên của D khi D được xem như một tập đa diện.

Proof **(⇒)** Giả sử rằng d là một điểm cực biên của D (xem như một tập đa diện) nhưng không phải là một hướng cực biên của S. Khi đó, tồn tại hai hướng d_1 và d_2 của S và hai hằng số λ_1 và λ_2 với $\lambda_1, \lambda_2 \geq 0$ sao cho:

$$d = \lambda_1 d_1 + \lambda_2 d_2.$$

Không mất tính tổng quát, ta có thể giả sử rằng d_1 và d_2 là các vector thỏa mãn:

$$e^T d_i = 1, \quad i = 1, 2.$$

Nếu không, ta có thể chuẩn hóa chúng sao cho tổng các thành phần bằng 1 và điều chỉnh λ_1, λ_2 tương ứng. Khi đó, ta có:

$$1 = e^T d = \lambda_1 e^T d_1 + \lambda_2 e^T d_2 = \lambda_1 + \lambda_2.$$

Hơn nữa, vì d_1 và d_2 là các hướng của S, chúng cũng thuộc D. Do đó, ta đã tìm thấy một tổ hợp lồi của các phần tử trong D bằng d, điều này mâu thuẫn với giả thiết rằng d là một điểm cực biên.

(⇐) Ngược lại, giả sử rằng d là một hướng cực biên của S có tổng các thành phần bằng 1 nhưng không phải là một điểm cực biên của D. Khi đó, d có thể được biểu diễn như một tổ hợp lồi của hai hướng khác $d_1, d_2 \in D$:

$$d = \lambda_1 d_1 + \lambda_2 d_2, \quad \text{với } \lambda_1 + \lambda_2 = 1 \text{ và } \lambda_1, \lambda_2 \in (0, 1).$$

Điều này rõ ràng là mâu thuẫn, vì mọi tổ hợp lồi nghiêm ngặt đều là một tổ hợp dương, và do đó giả định rằng d là một hướng cực biên là sai. Như vậy ta có điều phải chứng minh! □

Tính chất 1 Xét tập đa diện:

$$S = \{\vec{x} \in \mathbb{R}^n \mid A\vec{x} = \vec{b}, \vec{x} \geq 0\},$$

trong đó A là ma trận $m \times n$ và $\vec{b}$ là một vectơ có kích thước m.

Giả sử ma trận A có hạng m, nếu không, ta giả sử hệ phương trình $A\vec{x} = \vec{b}$ có nghiệm. Khi đó ta có những tính chất sau:

(i) Số lượng điểm cực biên của S là hữu hạn, và S luôn có ít nhất một điểm cực biên.

(ii) Số lượng hướng cực biên của S cũng là hữu hạn.

(iii) Mỗi điểm trong S có thể được biểu diễn dưới dạng một tổ hợp lồi của các điểm cực biên cộng với một tổ hợp tuyến tính không âm của các hướng cực biên. Cụ thể, $\vec{x} \in S$ khi và chỉ khi có thể viết dưới dạng:

$$\vec{x} = \sum_{j=1}^{k} \lambda_j \vec{x}_j + \sum_{j=1}^{\ell} \mu_j \vec{d}_j, \tag{2.2}$$

trong đó:

$$\sum_{j=1}^{k} \lambda_j = 1, \quad \lambda_j \geq 0 \quad \forall j = 1, \ldots, k,$$

$$\mu_j \geq 0 \quad \forall j = 1, \ldots, \ell.$$

Proof **(i)**. Giả sử $x \in S$. Nếu x là một điểm cực biên, thì ta có ngay điều phải chứng minh.

Giả sử x không phải là điểm cực biên của S. Mà ta lại có một điểm $x_0 \in S$ là một điểm cực biên của S khi và chỉ khi x_0 là giao điểm của đúng n siêu phẳng độc lập tuyến tính trong tập các siêu phẳng tạo nên S. Từ đó ta suy ra x nằm tại giao của $r < n$ mặt phẳng ràng buộc (trong đó r có thể bằng 0).

Việc x không phải là một điểm cực biên của P cũng đồng nghĩa rằng tồn tại $y_1, y_2 \in S$ và $\lambda > 0$ sao cho $x = \lambda y_1 + (1-\lambda)y_2$. Để điều này xảy ra, chúng ta cần các ràng buộc tại x cũng phải ràng buộc cả y_1 và y_2. Nói cách khác nếu một siêu phẳng ràng buộc điểm x, thì siêu phẳng đó cũng phải ràng buộc cả y_1 và y_2.

Đặt $d = y_2 - y_1$ là hướng từ y_1 đến y_2. Khi đó, ta có:

$$\begin{cases} y_1 = x - (1-\lambda)d \\ y_2 = x + \lambda d \end{cases} \tag{2.3}$$

Các giá trị $x + \gamma d$ và $x - \gamma d$ với $\gamma > 0$ tương ứng với sự di chuyển từ x theo hướng d. Từ phương trình (2.3), ta có thể di chuyển và mở rộng theo cả hai hướng dương và âm của d mà vẫn thuộc S. Gọi γ là giá trị lớn nhất sao cho cả $x + \gamma d$ và $x - \gamma d$ đều thuộc S. Rõ ràng, chúng ta không thể di chuyển vô hạn theo cả hai hướng vì $x \geq 0$, do đó γ là hữu hạn.

Không mất tính tổng quát, giả sử rằng γ được xác định bởi $x - \gamma d$. Khi đó, đặt $x_1 = x - \gamma d$. Vì có r siêu phẳng ràng buộc tại x (cũng như tại y_1 và y_2), nên rõ ràng các mặt phẳng này cũng ràng buộc tại x_1 và ít nhất một mặt phẳng khác cũng ràng buộc tại x_1. Như vậy, có ít nhất $r+1$ mặt phẳng ràng buộc tại x_1. Nếu $r+1=n$, ta đã xác định được một điểm cực biên. Không thì, ta có thể lặp lại quá trình này cho đến khi tìm được một điểm cực biên.

Để chứng minh số lượng điểm cực biên là hữu hạn, ta lưu ý rằng mọi điểm cực biên là giao điểm của n siêu phẳng ràng buộc độc lập tuyến tính tạo nên S. Có tổng cộng $n+m$ siêu phẳng ràng buộc tạo nên S, do đó số lượng điểm cực biên có thể có bị giới hạn bởi $\binom{n+m}{n}$. Điều này hoàn tất chứng minh của **(i)**. □

(ii). Từ tính chất **(i)** và bổ đề (1) ta có được kết quả này.

(iii). Giả sử $x \in P$, ta sẽ chứng minh rằng tồn tại $\lambda_1, \ldots, \lambda_k$ và $\mu_1, \ldots, \mu_l$ sao cho phương trình (2.2) đúng. Đầu tiên gọi:

$$\overline{P} = P \cap \{x \in \mathbb{R}^n : e^T x \leq M\} \tag{2.4}$$

trong đó M là một hằng số đủ lớn sao cho $e^T x_i < M$ với mọi $i = 1, \ldots, k$ và $e^T x < M$. Nghĩa là, M đủ lớn để tổng các thành phần của bất kỳ điểm cực biên nào nhỏ hơn M và tổng các thành phần của x cũng nhỏ hơn M.

Theo cách đặt ở (2.4), rõ ràng $\overline{P}$ bị chặn. Và thức tế là, nếu P bị chặn, thì $\overline{P} = P$. Hơn nữa, $\overline{P}$ là một tập đa diện lồi bị chặn trong P, do đó các điểm cực biên của $\overline{P}$ cũng là các điểm cực biên của P.

Gọi tập hợp các điểm cực biên của $\overline{P}$ là:

$$E_P = \{x_1, \ldots, x_k, \ldots, x_{k+u}\} \tag{2.5}$$

Theo tính chất **(ii)** của (1), ta biết rằng $0 \leq u < \infty$. Nếu $x \in E_P$, thì x có thể được biểu diễn như một tổ hợp lồi của các phần tử trong E_P. Do đó, ta giả sử $x \notin E_P$.

Bây giờ, giả sử hệ phương trình $Gx = g$ biểu diễn các siêu phẳng ràng buộc của $\overline{P}$ tại x. Rõ ràng, $\text{rank}(G) < n$ (nếu không thì x sẽ là một điểm cực biên của E_P).

Chọn $d \neq 0$ là nghiệm của bài toán $Gd = 0$ và đặt $\gamma = \max\{\gamma : x + \gamma d \in X\}$.

Vì X bị chặn và x không thuộc E_P, nên $0 < \gamma_1 < \infty$. Đặt $y_1 = x + \gamma_1 d$. Cũng từ tính chất **(ii)** của (1), tại y_1 ta có ít nhất một siêu phẳng ràng buộc độc lập mới của P. Nếu lúc này có n siêu phẳng ràng buộc, thì y là một điểm cực biên của P. Nếu không, ta lặp lại quá trình này để tìm được một điểm cực biên.

Rõ ràng, $Gy_1 = g$. Như ta mong đợi, $Gy_2 = g$ và có ít nhất một siêu phẳng ràng buộc mới tại y_2. Do đó, ta có thể viết:

$$x = \delta y_1 + (1-\delta) y_2$$

với $\delta \in (0, 1]$ và $\delta = \gamma_2/(1+\gamma_2)$. Tiếp tục quá trình trên, ta có thể biểu diễn x như một tổ hợp lồi của các điểm cực biên của E_P.

Từ đó, ta có thể viết:

$$x = \sum_{i=1}^{k+u} \delta_i x_i, \quad \sum_{i=1}^{k+u} \delta_i = 1.$$

Nếu $\delta_i = 0$ với mọi $i > k$, ta đã biểu diễn được x dưới dạng tổ hợp lồi của các điểm cực biên của P.

Giả sử tồn tại $v > k$ sao cho $\delta_v > 0$. Khi đó, điểm x_v là điểm cực biên của $\overline{P}$ nhưng không phải điểm cực biên của P. Do đó, có một hướng cực biên d_j sao cho:

$$x_v = x_{i(v)} + \theta_v d_{j(v)}$$

với $x_{i(v)}$ là một điểm cực biên của P và $\theta_v d_{j(v)}$ là một hướng cực biên của P.

Thay vào biểu thức của x:

$$x = \sum_{i=1}^{k} \lambda_i x_i + \sum_{j=1}^{l} \mu_j d_j.$$

Điều này hoàn tất chứng minh. □

Định lý 3 Giả sử S là một tập đa diện khác rỗng có dạng:

$$S = \{\vec{x} \in \mathbb{R}^n \mid A\vec{x} = \vec{b}, \vec{x} \geq 0\},$$

trong đó A là một ma trận $m \times n$ có hạng m. Khi đó, tập S có ít nhất một hướng cực biên khi và chỉ khi nó không bị chặn.

Proof **Chiều thuận:** Nếu S tồn tại một hướng cực biên, giả sử là $\vec{d}$. Theo định nghĩa 8, ta nhận thấy nếu di chuyển từ một điểm bất kỳ trong S theo hướng $\vec{d}$ với một khoảng cách dương, ta vẫn ở trong S. Điều này cũng có nghĩa là tập S không bị chặn.

Chiều đảo: Bây giờ, giả sử rằng S là không bị chặn nhưng lại không có hướng cực biên. Ta sẽ chứng minh phản chứng. Với mọi $\vec{x} \in S$, ta có:

$$\|\vec{x}\| = \left\| \sum_{j=1}^{k} \lambda_j \vec{x}_j \right\| \leq \sum_{j=1}^{k} \lambda_j \|\vec{x}_j\| \leq \sum_{j=1}^{k} \|\vec{x}_j\|$$

Điều này mâu thuẫn. Do đó chiều đảo được chứng minh! □

Kết luận 1 Như vậy ta đã chứng minh được một tập lồi đóng khác rỗng có ít nhất một hướng cực biên khi và chỉ khi nó không bị chặn. Và để ý thì tập hợp tất cả các phương án của bài toán quy hoạch tuyến tính lại là một tập lồi như vậy. Mà tập các phương án chấp nhận được này lại chính là miền khả thi của bài toán!

Nói tóm lại, mọi thứ đã liên kết lại với nhau để đưa chúng ta đến một kết luận như sau: Miền khả thi của bài toán quy hoạch tuyến tính bị chặn khi và chỉ khi nó có ít nhất một hướng cực biên.

Định lý 4 Một đa diện lồi $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ là bị chặn nếu và chỉ nếu tập hợp các vector pháp tuyến của nó (các hàng của ma trận A) tạo thành một nón lồi bao phủ toàn bộ không gian $\mathbb{R}^n$. Trong không gian 2D, điều này tương đương với việc các vector pháp tuyến phải "quét"qua toàn bộ vòng tròn $360°$ mà không để lại khoảng trống lớn hơn $180°$.

Ta sẽ chứng minh định lý trên trong không gian $\mathbb{R}^n$

Proof **Chiều thuận:** Giả sử P bị chặn, tức là không tồn tại hướng $d \in \mathbb{R}^n \setminus \{0\}$ sao cho:

$$x + \lambda d \in P, \quad \forall \lambda \geq 0, \forall x \in P.$$

Ta cần chứng minh rằng tập hợp các vector pháp tuyến (các hàng của A) tạo thành một nón lồi bao phủ toàn bộ $\mathbb{R}^n$, tức là:

$$\text{span}_+\{a_i\} = \mathbb{R}^n,$$

trong đó a_i là các hàng của A.

Giả sử ngược lại: tập hợp các vector pháp tuyến không tạo thành một nón lồi bao phủ toàn bộ $\mathbb{R}^n$. Khi đó, tồn tại một hướng $d \in \mathbb{R}^n \setminus \{0\}$ sao cho:

$$\langle a_i, d \rangle \leq 0, \quad \forall i.$$

Xét một điểm $x \in P$, tức là $Ax \leq b$. Xét đường thẳng:

$$x(\lambda) = x + \lambda d, \quad \lambda \geq 0.$$

Để kiểm tra $x(\lambda) \in P$, ta xét:

$$A(x + \lambda d) \leq b \Leftrightarrow Ax + \lambda Ad \leq b.$$

Do $Ax \leq b$, ta có:

$$\lambda Ad \leq b - Ax.$$

Vì $b - Ax \geq 0$ và theo giả thiết $\langle a_i, d \rangle \leq 0$, nên:

$$\lambda \langle a_i, d \rangle \leq 0, \quad \forall \lambda \geq 0.$$

Do đó, $x + \lambda d \in P$ với mọi $\lambda \geq 0$, mâu thuẫn với giả thiết rằng P bị chặn. Vậy tập hợp các vector pháp tuyến phải tạo thành một nón lồi bao phủ toàn bộ $\mathbb{R}^n$.

Chiều đảo: Giả sử tập hợp các vector pháp tuyến tạo thành một nón lồi bao phủ toàn bộ $\mathbb{R}^n$. Nghĩa là với mọi $d \in \mathbb{R}^n \setminus \{0\}$, tồn tại một a_i sao cho:

$$\langle a_i, d \rangle > 0.$$

Ta cần chứng minh rằng P bị chặn, tức là không tồn tại hướng $d \neq 0$ sao cho:

$$x + \lambda d \in P, \quad \forall \lambda \geq 0.$$

Giả sử ngược lại, tức là tồn tại hướng $d \neq 0$ và một điểm $x \in P$ sao cho $x + \lambda d \in P$

với mọi $\lambda \geq 0$. Khi đó:

$$A(x + \lambda d) \leq b \Leftrightarrow Ax + \lambda Ad \leq b.$$

Vì $b - Ax \geq 0$, ta cần:

$$\lambda Ad \leq b - Ax.$$

Do giả thiết về nón lồi, tồn tại a_i sao cho $\langle a_i, d\rangle > 0$. Khi λ đủ lớn, vế trái vượt quá $b_i - \langle a_i, x\rangle$, dẫn đến vi phạm ràng buộc. Do đó, không thể tồn tại hướng $d \neq 0$ sao cho $x + \lambda d \in P$ với mọi $\lambda \geq 0$, nghĩa là P bị chặn.

Ta có điều phải chứng minh!

□

Nhận xét 2 (Mối liên hệ với Định lý Farkas) Định lý `Farkas` là một kết quả nền tảng trong tối ưu tuyến tính và phân tích tính khả thi của hệ bất phương trình tuyến tính. Nó phát biểu rằng với một ma trận $A \in \mathbb{R}^{m\times n}$ và một vector $b \in \mathbb{R}^m$, chính xác một trong hai hệ sau có nghiệm:

(i) Hệ bất phương trình tuyến tính:

$$Ax \leq b, \quad x \in \mathbb{R}^n.$$

(ii) Hệ tồn tại một tổ hợp tuyến tính dương của các hàng của A tạo thành một vector $y \geq 0$ sao cho:

$$A^T y = 0, \quad b^T y < 0.$$

Mối liên hệ giữa Định lý 4 và Định lý `Farkas` có thể được hiểu qua cách chúng mô tả điều kiện về sự tồn tại của nghiệm trong các hệ bất đẳng thức tuyến tính. Cụ thể, Định lý Farkas`Farkas` phát biểu rằng với một ma trận A và một vector b, hệ bất phương trình

$$Ax \leq b$$

có nghiệm nếu và chỉ nếu không tồn tại một tổ hợp tuyến tính không âm của các hàng của A có thể tạo ra một vector nằm hoàn toàn trong nón đối cực của b. Điều này có liên hệ chặt chẽ với điều kiện về tính bị chặn của một đa diện lồi.

Theo Định lý 4, một đa diện lồi $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ là bị chặn khi và chỉ khi tập hợp các vector pháp tuyến (các hàng của A) tạo thành một nón lồi bao phủ toàn bộ không gian $\mathbb{R}^n$. Điều này có thể được diễn giải theo ngôn ngữ của Định lý `Farkas`: nếu tồn tại một hướng $d \in \mathbb{R}^n$ không bị chặn, thì có một siêu phẳng tách biệt được d với nón sinh bởi các hàng của A, tức là tồn tại một hệ số `Farkas` chứng minh rằng P không bị chặn.

Như vậy, Định lý `Farkas` có thể được xem là một công cụ mạnh để chứng minh hoặc diễn giải điều kiện của Định lý 4, giúp ta hiểu sâu hơn về cấu trúc hình học của tập nghiệm của hệ bất đẳng thức tuyến tính.

Section 3

Thuật toán đơn giản

Trở lại bài toán (1), với những kiến thức cơ sở từ `section` 2, ta sẽ bắt tay vào xây dựng ý tưởng cho những thuật toán đơn giản giải quyết từng phần của bài toán.

(1) Đầu tiên là về vấn đề **Tìm danh sách điểm cực biên**, lấy ý tưởng từ phương pháp hình học (`Geometry method`), ta có thuật toán sau:

1. Thêm điểm gốc tọa độ $(0, 0)$ vào danh sách nếu nó thỏa mãn tất cả các ràng buộc.
2. Xét giao điểm của mỗi ràng buộc với trục tọa độ (tức là khi $x = 0$ hoặc $y = 0$).
3. Xét giao điểm của hai ràng buộc bất kỳ (nếu chúng không song song).

Sau mỗi bước xét ra được điểm nào thì kiểm tra lại với các ràng buộc chưa xét để đảm bảo điểm đó thỏa mãn tất cả ràng buộc của bài toán. Đồng thời cũng loại bỏ các điểm trùng lặp.

Algorithm 1 Find Extreme Points of the Feasible Region

```
procedure FindExtremePoints(a, b, d)        ▷ a, b, d: Coefficients of constraint inequalities
    Input:
        a, b, d: Coefficients of M constraint inequalities
    Output: A list of extreme points
    extreme_points ← [ ]                     ▷ Initialize list of extreme points
    if all d[i] ≥ 0 then
        Add (0, 0) to extreme_points
    for i = 1 to M do
        if a[i] ≠ 0 then
            x1 ← d[i]/a[i]
            if x1 ≥ 0 and a[j] · x1 ≤ d[j], ∀j then
                Add (x1, 0) to extreme_points
        if b[i] ≠ 0 then
            x2 ← d[i]/b[i]
            if x2 ≥ 0 and b[j] · x2 ≤ d[j], ∀j then
                Add (0, x2) to extreme_points
    for each pair (i, j) in {1, ..., M} do
        det ← a[i] · b[j] − a[j] · b[i]
        if det ≠ 0 then                      ▷ Check if the two lines are parallel
            x1 ← (d[i] · b[j] − d[j] · b[i]) / det
            x2 ← (a[i] · d[j] − a[j] · d[i]) / det
            if x1 ≥ 0 and x2 ≥ 0 and a[k] · x1 + b[k] · x2 ≤ d[k], ∀k then
                Add (x1, x2) to extreme_points
    unique_points ← [ ]                      ▷ Initialize list of unique points
    for each p in extreme_points do
        Round p to 10 decimal places
        if p is not in unique_points then
            Add p to unique_points
    return unique_points
```

❒ **Độ phức tạp thuật toán:** Giả sử bài toán quy hoạch tuyến tính đang xét có M ràng buộc, ta phân tích các bước chính của thuật toán:

- Kiểm tra điểm gốc tọa độ: $O(M)$
- Tìm giao điểm với trục tọa độ: Lặp qua M ràng buộc, mỗi lần kiểm tra mất $O(M)$, tổng cộng $O(M^2)$.
- Tìm giao điểm giữa hai ràng buộc: Có $\binom{M}{2}$ cặp ràng buộc cần xét, tương đương độ phức tạp $O(M^2)$. Và mỗi lần kiểm tra mất $O(M)$, nên tổng cộng là $O(M^3)$.
- Loại bỏ điểm trùng lặp: Sử dụng danh sách hoặc tập hợp, có thể coi là $O(N \log N)$ với N là số điểm tìm được (vì không vượt quá số giao điểm đã xét nên cũng đồng nghĩa là không vượt quá $O(M^2)$).

Vậy độ phức tạp tổng thể của thuật toán này xấp xỉ $O(M^3)$ trong trường hợp xấu nhất.

Nhận xét 3 Thuật toán có ưu điểm là có thể tìm chính xác tất cả các điểm cực biên của miền ràng buộc. Do phương pháp tiếp cận trực tiếp bằng cách kiểm tra từng giao điểm có thể có lấy ý tưởng từ phương pháp hình học, thuật toán khá đơn giản và dễ cài đặt. Hơn nữa, thuật toán không yêu cầu sử dụng các công cụ tối ưu hóa hay thư viện nào phức tạp.

Tuy nhiên, thuật toán cũng có một số hạn chế. Độ phức tạp tính toán lên tới $O(M^3)$ trong trường hợp xấu nhất, điều này khiến thuật toán trở nên chậm khi số lượng ràng buộc M lớn. Ngoài ra, sau khi tìm ra giao điểm giữa các ràng buộc, thuật toán phải kiểm tra lại từng điểm để đảm bảo rằng nó thỏa mãn tất cả các ràng buộc, làm tăng đáng kể chi phí tính toán. Một vấn đề khác là sai số số thực khi xác định giao điểm, có thể dẫn đến sai lệch nhỏ trong kết quả nếu không được xử lý cẩn thận.

Nhận xét 4 (Giải pháp cải thiện bằng `Convex Hull`) Một cách tiếp cận hiệu quả hơn để tìm các điểm cực biên của miền ràng buộc là sử dụng thuật toán `Convex Hull`(bao lồi). Trong không gian hai chiều, tập hợp các điểm cực biên tạo thành một đa giác lồi, và việc tìm tập hợp các điểm này có thể được tối ưu hóa bằng các thuật toán xây dựng bao lồi như `Graham's Scan` hoặc `Andrew's Monotone Chain`, có độ phức tạp $O(N \log N)$, với N là số điểm cực biên.

Cách tiếp cận này hoạt động như sau: đầu tiên, tìm tất cả các giao điểm hợp lệ giữa các ràng buộc. Sau đó, thay vì kiểm tra từng điểm một cách riêng lẻ, ta sử dụng thuật toán `Convex Hull` để xác định tập hợp con các điểm thực sự nằm trên đường biên của miền ràng buộc. Điều này giúp loại bỏ các điểm không cần thiết và cải thiện đáng kể hiệu suất so với cách tiếp cận kiểm tra từng điểm một cách trực tiếp.

Sử dụng `Convex Hull` không chỉ giảm số lượng điểm cần xét mà còn giúp đảm bảo rằng thuật toán hoạt động ổn định ngay cả khi có nhiều ràng buộc. Ngoài ra, `Convex Hull` cũng giúp xử lý vấn đề sai số số thực tốt hơn, vì thuật toán bao lồi thường sử dụng các phép toán hình học chính xác để xác định biên của miền ràng buộc.

2 Tiếp theo ta sẽ kiểm tra vấn đề **Miền khả thi có bị chặn không?**

Quay lại phương pháp hình học, rõ ràng nếu ta có thể vẽ được miền khả thi của bài toán lên hệ trục tọa độ thì mọi thứ sẽ rất dễ dàng để biết miền này bị chặn hay không bị chặn. Tuy nhiên thuật toán ở mức đơn giản sẽ không thể nhìn vào hình rồi nói nó bị chặn hay không. Do đó ở trên, em đã sử dụng một số tính chất và định lý xoay quanh **Hướng cực biên** hay `Extreme point`.

Từ kết luận (1), ta nhận thấy việc cần làm ở đay đơn giản chỉ là kiểm tra xem miền khả thi của bài toán có tồn tại ít nhất một hướng cực biên hay không. Nếu có thì miền khả thi không bị chặn, ngược lại sẽ bị chặn.

Nói cách khác, thay vì kiểm tra nhiều điểm để xác định tính bị chặn của miền khả thi, ta có thể sử dụng ý tưởng **nhân vô hướng** với các hướng kiểm tra giới hạn. Cụ thể, chỉ cần xét một tập hợp hữu hạn các hướng đặc trưng, nếu tồn tại ít nhất một hướng $\vec{d} \neq (0,0)$ sao cho tất cả các ràng buộc a_j thỏa mãn điều kiện:

$$a_j \cdot d + b_j \leq 0$$

thì miền khả thi có thể mở rộng vô hạn theo hướng $\vec{d}$, tức là không bị chặn. Ngược lại, nếu với mọi hướng $\vec{d} \neq (0,0)$, tồn tại ít nhất một j sao cho:

$$a_j \cdot d + b_j > 0$$

thì miền khả thi bị chặn vì không có hướng nào cho phép mở rộng vô hạn.

Ý tưởng thì nghe có vẻ đơn giản, nhưng thực tế có vô số hướng để có thể thử theo cách trên. Hãy cùng xem xét một cách dưới xem ta có thể tối thiểu hóa số hướng kiểm tra này không?

Để tổng quát hóa, ta có thể kiểm tra theo bốn hướng cơ bản sau:

- $\mathbf{d} = (1,0)$: Kiểm tra xem x_1 có thể tăng vô hạn hay không (ví dụ: khi $x_2 = 0$).
- $\mathbf{d} = (0,1)$: Kiểm tra xem x_2 có thể tăng vô hạn hay không (ví dụ: khi $x_1 = 0$).
- $\mathbf{d} = (1,1)$: Kiểm tra hướng chéo, khi cả x_1 và x_2 cùng tăng.
- $\mathbf{d} = (-1,1)$: Kiểm tra khi x_1 giảm nhưng x_2 tăng (trong trường hợp $x_1 \geq 0$, hướng này có thể không mở rộng vô hạn nhưng vẫn cần kiểm tra để đảm bảo).

Do điều kiện miền khả thi theo bài toán ta đã có $x_1 \geq 0, x_2 \geq 0$, nên hướng $(-1,1)$ không cần kiểm tra vì nó chắc chắn vi phạm ràng buộc. Do đó, ta chỉ cần kiểm tra ba hướng sau là đủ:

$$\mathbf{d} = (1,0), \quad \mathbf{d} = (0,1), \quad \mathbf{d} = (1,1)$$

Từ cơ sở đó, ta có thể xây dựng một thuật toán xác định miền khả thi có bị chặn hay không một cách hiệu quả mà không cần kiểm tra vô số điểm một cách trực tiếp:

Algorithm 2 Check Whether the Feasible Region is Bounded -Naive Method

procedure CheckBoundedRegion(a, b) ▷ a, b: Coefficients of M constraint inequalities
  Input:
    a, b: Coefficients of M constraint inequalities
  Output: True if the region is bounded, **False** if the region is unbounded
  $directions \leftarrow [(1,0),(0,1),(1,1),(-1,1)]$ ▷ Directions to check
  for each (d_1, d_2) in $directions$ **do**
    $isBounded \leftarrow$ **True** ▷ Assume the region is bounded
    for $i = 1$ **to** M **do**
      if $a[i] \cdot d_1 + b[i] \cdot d_2 > 0$ **then**
        $isBounded \leftarrow$ **False** ▷ Found an unbounded expansion direction
        break
    if $isBounded =$ **False then**
      return False ▷ The region is unbounded
  return True ▷ No unbounded expansion direction found, the region is bounded

❒ **Độ phức tạp thuật toán:** Ta nhận thấy thuật toán kiểm tra miền khả thi có bị chặn không sẽ có độ phức tạp $\mathcal{O}(M)$, vì nó chỉ duyệt qua tất cả M bất phương trình và kiểm tra xem có hướng mở rộng vô hạn nào hay không. Cách tiếp cận này khá nhanh và trực tiếp, thay vì kiểm tra rất nhiều điểm thì chỉ cần kiểm tra vài hướng cụ thể.

Tuy nhiên cách này sẽ sai với một số những trường hợp. Ví dụ sau đây là một trong số đó:

Ví dụ

Ví dụ minh họa. Xét hai ràng buộc tuyến tính:

$$\begin{cases} -2x_1 + 3x_2 \leq 10 \\ 2x_1 - 4x_2 \leq 5 \end{cases} \quad \text{với } x_1 \geq 0, x_2 \geq 0.$$

Với mọi hướng như ở thuật toán trên ta thử thì sẽ đưa đến kết quả miền khả thi bị chặn. Nhưng thực tế thì không, điều này xảy ra vì hướng cực biên thỏa không nằm trong những trường hợp ta xét, mà là những hướng như $(2,1), (1, \frac{3}{4}), \ldots$

Như vậy dù độ phức tạp thấp, phương án ngây thơ (`Naive`) này dường như có thể gây ra kết quả sai vì tham lam muốn tối thiểu số hướng cực biên cần kiểm tra.

Ta sẽ phát triển một thuật toán khác, vẫn đi theo hướng cũ, nghĩa là theo kết luận một đa diện lồi trong không gian là bị chặn nếu và chỉ nếu nó không có bất kỳ hướng cực biên nào mà từ đó ta có thể mở rộng ra vô cực mà vẫn thỏa mãn tất cả các ràng buộc. Điều đó sẽ chủ yếu được dựa vào Định lý 4.

Cụ thể, thuật toán mới này sẽ kiểm tra tính bị chặn của miền khả thi dựa trên việc phân tích các `vector` pháp tuyến của tập bất đẳng thức. Trong không gian hai chiều, các `vector` pháp tuyến này tạo thành một tập hợp hướng trong mặt phẳng. Ta sắp xếp các góc tương ứng của chúng theo thứ tự tăng dần và tính các hiệu góc liên tiếp.

Nếu tồn tại một hiệu góc lớn hơn π, tức là có một khoảng trống lớn hơn $180°$, thì tồn tại một hướng không bị chặn, dẫn đến miền khả thi không bị chặn.

Ngược lại, nếu tất cả các hiệu góc đều không vượt quá π, nghĩa là các `vector` pháp tuyến bao phủ toàn bộ vòng tròn $360°$ mà không để lại khoảng trống lớn, thì miền khả thi bị chặn.

Algorithm 3 Check if the Feasible Region is Bounded - with Farkas Approach

procedure CheckBoundedRegion(a, b, d) ▷ Input: Coefficients (a_i, b_i, d_i) of M constraints ▷ Output: **True** if bounded, **False** otherwise
  $normals \leftarrow [(a_i, b_i)$ for $i = 1$ to $M]$ ▷ Collect normal vectors
  ▷ Ensure constraints for $x \geq 0$ and $y \geq 0$ are included
  if no $(nx, ny) = (-1, 0)$ in normals **then**
    Add $(-1, 0)$ to $normals$
  if no $(nx, ny) = (0, -1)$ in normals **then**
    Add $(0, -1)$ to $normals$
  $angles \leftarrow [\arctan 2(ny, nx)$ for (nx, ny) in $normals]$
  Sort $angles$ in ascending order
  ▷ Check for gaps larger than π (180°)
  for $i = 1$ **to** $len(angles)$ **do**
    $diff \leftarrow angles[(i+1) \mod len(angles)] - angles[i]$
    if $diff < 0$ **then**
      $diff \leftarrow diff + 2\pi$
    if $diff > \pi$ **then**
      return False
  return True ▷ Region is bounded

❐ **Độ phức tạp thuật toán:** Thuật toán kiểm tra tính bị chặn của miền khả thi thứ hai này đi theo hướng tiếp cận là định lý 4, chủ yếu tốn thời gian ở hai bước: tính toán góc của các `vector` pháp tuyến và sắp xếp chúng.

Việc duyệt qua các ràng buộc để thu thập vector pháp tuyến có độ phức tạp $O(M)$, với M là số ràng buộc của bài toán. Trong khi đó, bước sắp xếp danh sách góc chiếm $O(M \log M)$. Sau cùng, việc kiểm tra khoảng cách giữa các góc liên tiếp chỉ tốn $O(M)$. Do đó, tổng độ phức tạp của thuật toán là $O(M \log M)$.

Nhận xét 5 Độ phức tạp này không thua kém khi ta so với với việc ta sử dụng thuật toán `QuickHull` thuộc hàm `ConvexHul` của thư viện `scipy`. Cụ thể: Thuật toán `QuickHull` trong trường hợp trung bình có độ phức tạp là $O(N \log N)$, với N là số điểm cực biên, nhưng có thể lên đến $O(N^2)$ trong trường hợp xấu nhất khi tất cả các điểm nằm trên bao lồi.

Có thể thấy `QuickHull` trong `ConvexHull` hoạt động trên các điểm cực biên thay vì các ràng buộc, với số lượng điểm cực biên trung bình nhỏ hơn đáng kể so với tổng số ràng buộc. Vì vậy, trong thực tế, thuật toán này có thể chạy nhanh hơn nếu số điểm cực biên nhỏ.

(3) Cuối cùng ta sẽ **Tìm GTNN và GTLN của hàm mục tiêu nếu có.**

Cùng nhìn lại Định lý (1) và hệ quả (1), ta nhận thấy ý tưởng đơn giản nhất để tìm `GTLN` hoặc `GTNN` cho hàm mục tiêu: Gọi N là số điểm cực biên của miền khả thi và giả sử hàm mục tiêu có dạng $f(x, y) = c_1 \cdot x_1 + c_2 \cdot x_2$. Để tìm `GTLN` và `GTNN`, ta cần tính giá trị của $f(x_1, x_2)$ tại mỗi điểm cực biên và chọn ra giá trị phù hợp.

Tuy nhiên đây mới chỉ là một phần của vấn đề, vì điều đó chỉ đúng khi miền khả thi bị chặn. Nếu miền khả thi không bị chặn mà ta áp dụng phương pháp trên thì sẽ chỉ tìm được giá trị lớn nhất hàm mục tiêu có thể nhận từ tập điểm cực biên của nó. Do đó cần phải để ý đến tính bị chặn của miền khả thi khi tìm cực trị của hàm mục tiêu.

Ta có tính chất sau:

Tính chất 2 Tính bị chặn của miền khả thi đóng vai trò quyết định trong việc xác định sự tồn tại của GTNN và GTLN của hàm mục tiêu $f(x_1, x_2) = c_1 \cdot x_1 + c_2 \cdot x_2$ trong bài toán quy hoạch tuyến tính. Cụ thể:

1. Nếu miền khả thi bị chặn: Khi đó miền khả thi của bài toán là một tập đa diện lồi được tạo ra bởi bởi giao của các nửa mặt phẳng từ các ràng buộc tuyến tính $a_{i1}x_1 + a_{i2}x_2 \leq b_i$ (với $i = 1, 2, \ldots, m$) và các ràng buộc $x_1 \geq 0$, $x_2 \geq 0$. Một đa diện lồi có số lượng hữu hạn các điểm cực biên theo tính chất $(\mathbf{i})$ ở (1). Lại có S bị chặn, tập giá trị $\{f(x_1, x_2) \mid (x_1, x_2) \in S\}$ là một tập hữu hạn trên các điểm cực biên (do số điểm cực biên là hữu hạn). Hơn nữa, vì f là một hàm tuyến tính liên tục trên một tập đa diện lồi bị chặn, nên theo định lý Weierstrass, f đạt cực trị tại các điểm biên hoặc bên trong tập đa diện lồi của S, nhưng ta đã chứng minh được trong trường hợp tuyến tính, cực trị luôn nằm tại các điểm cực biên. Do đó trong trường hợp này bài toán luôn tồn tại cả GTNN và GTLN.

2. Nếu miền khả thi không bị chặn: tức là tồn tại một hướng (d_1, d_2) sao cho $(x_1 + t \cdot d_1, x_2 + t \cdot d_2) \in S$ với mọi $t \geq 0$ và một số t đủ lớn, thì sự tồn tại của GTNN và GTLN phụ thuộc vào hệ số c_1, c_2 và hướng không bị chặn:

- Nếu $c_1 \cdot d_1 + c_2 \cdot d_2 > 0$, thì $f(x_1 + t \cdot d_1, x_2 + t \cdot d_2) = f(x_1, x_2) + t \cdot (c_1 \cdot d_1 + c_2 \cdot d_2)$ có thể tăng vô hạn khi $t \to \infty$, dẫn đến GTLN không tồn tại.
- Nếu $c_1 \cdot d_1 + c_2 \cdot d_2 < 0$, thì f có thể giảm vô hạn khi $t \to \infty$, dẫn đến GTNN không tồn tại.
- Nếu $c_1 \cdot d_1 + c_2 \cdot d_2 = 0$, thì f không đổi dọc theo hướng đó, và ta cần kiểm tra thêm các điểm cực biên để xác định cực trị.

Do đó, trong trường hợp miền không bị chặn, chỉ có thể có một trong hai giá trị GTNN hoặc GTLN tồn tại (nếu f bị giới hạn ở một phía), hoặc cả hai đều không tồn tại (nếu f không bị chặn ở cả hai phía). Ví dụ, với miền $x_1 \geq 0, x_2 \geq 0, x_1 + x_2 \geq 1$ và $f = x_1 + x_2$, miền không bị chặn theo hướng $(1, 1)$, và f tăng vô hạn, nên không có GTLN, nhưng GTNN tồn tại tại.

3. Nếu miền khả thi rỗng: Cũng có thể hiểu là không tồn tại điểm cực biên nào thỏa mãn các ràng buộc của bài toán. Khi đó cả GTNN và GTLN đều không tồn tại, vì không có một phương án chấp nhận được nào, chứ chưa nói đến phương án tối ưu.

Algorithm 4 Find Minimum and Maximum Objective Values

procedure FindOptimalValues(V, c_1, c_2, is_bounded, directions)
    if V is empty **then**
        return None, None ▷ No feasible region
    $f_{\max} \leftarrow -\infty$, $f_{\min} \leftarrow \infty$
    for each $v \in V$ **do**
        $f \leftarrow c_1 \cdot v_1 + c_2 \cdot v_2$
        $f_{\max} \leftarrow \max(f_{\max}, f)$
        $f_{\min} \leftarrow \min(f_{\min}, f)$
    if is_bounded **then**
        return $f_{\max}, f_{\min}$
    for each $(d_1, d_2) \in$ directions **do**
        $d_{\text{coeff}} \leftarrow c_1 \cdot d_1 + c_2 \cdot d_2$
        if $d_{\text{coeff}} > 0$ **then**
            $f_{\max} \leftarrow$ None ▷ No max value
        else if $d_{\text{coeff}} < 0$ **then**
            $f_{\min} \leftarrow$ None ▷ No min value
    return $f_{\max}, f_{\min}$

❒ **Độ phức tạp thuật toán:** Độ phức tạp chủ yếu phụ thuộc vào số điểm cực biên. Để phân tích cụ thể thì nếu đặt số điểm cực biên là N, việc tính giá trị hàm mục tiêu tại mỗi điểm cực biên tốn $O(N)$, tìm giá trị lớn nhất và nhỏ nhất trong các giá trị này cũng tốn $O(N)$. Nếu miền khả thi không bị chặn, cần kiểm tra tất cả các hướng mở rộng có thể, điều này sẽ làm mất thêm $O(D)$, với D là số hướng mở rộng.

Do đó, tổng độ phức tạp của thuật toán là $O(N + D)$. Vì số điểm cực biên có thể lên đến $O(M^2)$, độ phức tạp tổng thể có thể đạt $O(M^2 + D)$.

Nhận xét 6 Phương pháp này đơn giản, dễ triển khai và không yêu cầu sử dụng các thuật toán tối ưu hóa phức tạp. Tuy nhiên, nếu số điểm cực biên N lớn, việc duyệt qua tất cả các điểm có thể trở nên tốn kém. Khi miền khả thi không bị chặn, việc kiểm tra tất cả các hướng mở rộng có thể làm tăng chi phí tính toán.

So với các thuật toán quy hoạch tuyến tính chuyên dụng như `Simplex` (trung bình $O(M)$), với M là số ràng buộc; hoặc `ellipsoid`, dù phương pháp này có thể kém hiệu quả trong các bài toán lớn.