ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC SÀI GÒN

BÁO CÁO ĐỀ CƯƠNG NGHIÊN CỨU KHOA HỌC
NGÀNH: TOÁN ỨNG DỤNG

Phương pháp giải bài toán Tối ưu tuyến tính nguyên

Thực hiện: Đỗ Ngọc Minh Thư & Nguyễn Chí Bằng

Hướng dẫn: PGS.TS. Tạ Quang Sơn

Ngày 13 tháng 5 năm 2024

Sinh viên lớp: DTU1221, Khóa: 22 @ Đại học Sài Gòn

NỘI DUNG BÁO CÁO

1. Giới thiệu và Đặt vấn đề
2. Phương pháp Gomory
3. Phương pháp Land-Doig
4. Kết luận và Hướng phát triển

Giới thiệu và Đặt vấn đề

Mục đích nghiên cứu

Tối ưu tuyến tính là một nội dung quan trọng trong chương trình đào tạo Cử nhân Toán ứng dụng. Lý thuyết về việc giải bài toán tối ưu tuyến tính đã được cung cấp cho sinh viên. Tuy vậy, có nhiều bài toán tối ưu cần được giải với nghiệm nguyên. Chẳng hạn như:

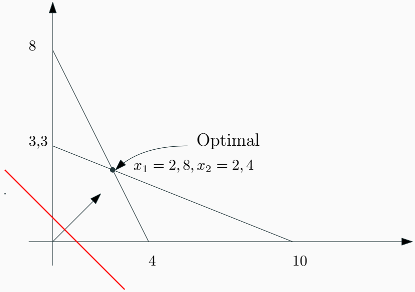
- Bài toán tối ưu nhân lực.
- Bài toán tối ưu vận chuyển hàng hóa.
- Bài toán tối ưu áp dụng trong tin học.

Có một lý thuyết riêng cho việc xử lý các bài toán Tối ưu tuyến tính và tìm nghiệm nguyên.

Mục đích của đề tài này là tìm hiểu một số phương pháp giải bài toán Tối ưu tuyến tính và tìm nghiệm nguyên cho bài toán.

Tại sao cần có một lý thuyết riêng cho bài toán Tối ưu tuyến tính nguyên

$$\text{Max} \quad f(x) = 2x_1 + 2x_2$$
$$\begin{cases} 2x_1 + x_2 \leq 8 \\ x_1 + 3x_2 \leq 10 \\ x_i \geq 0, \forall i = 1, 2. \end{cases}$$



Hình 1: Hình minh hoạ bài toán.

Nhận xét

- Nếu giải bài toán trên bằng phương pháp thông thường, ta nhận được nghiệm $x_1 = 2.8$, $x_2 = 2.4$.
- Nếu làm tròn nghiệm $x_1 \to 3$ và $x_2 \to 3$ thì điểm (x_1, x_2) không còn thuộc miền chấp nhận được.
- Nếu làm tròn nghiệm $x_1 \to 2$ và $x_2 \to 2$ thì điểm (x_1, x_2) chưa biết có phải nghiệm tối ưu hay không?

Nếu giải bài toán QHTT rồi sau đó làm tròn số thì có thể cho kết không như mong đợi.

Nhận xét

- Nếu giải bài toán trên bằng phương pháp thông thường, ta nhận được nghiệm $x_1 = 2.8$, $x_2 = 2.4$.

- Nếu làm tròn nghiệm $x_1 \to 3$ và $x_2 \to 3$ thì điểm (x_1, x_2) không còn thuộc miền chấp nhận được.

- Nếu làm tròn nghiệm $x_1 \to 2$ và $x_2 \to 2$ thì điểm (x_1, x_2) chưa biết có phải nghiệm tối ưu hay không?

Nếu giải bài toán QHTT rồi sau đó làm tròn số thì có thể cho kết không như mong đợi.

Nhận xét

- Nếu giải bài toán trên bằng phương pháp thông thường, ta nhận được nghiệm $x_1 = 2.8$, $x_2 = 2.4$.
- Nếu làm tròn nghiệm $x_1 \to 3$ và $x_2 \to 3$ thì điểm (x_1, x_2) không còn thuộc miền chấp nhận được.
- Nếu làm tròn nghiệm $x_1 \to 2$ và $x_2 \to 2$ thì điểm (x_1, x_2) chưa biết có phải nghiệm tối ưu hay không?

Nếu giải bài toán QHTT rồi sau đó làm tròn số thì có thể cho kết không như mong đợi.

Nhận xét

- Nếu giải bài toán trên bằng phương pháp thông thường, ta nhận được nghiệm $x_1 = 2.8$, $x_2 = 2.4$.
- Nếu làm tròn nghiệm $x_1 \to 3$ và $x_2 \to 3$ thì điểm (x_1, x_2) không còn thuộc miền chấp nhận được.
- Nếu làm tròn nghiệm $x_1 \to 2$ và $x_2 \to 2$ thì điểm (x_1, x_2) chưa biết có phải nghiệm tối ưu hay không?

Nếu giải bài toán QHTT rồi sau đó làm tròn số thì có thể cho kết không như mong đợi.

Tối ưu nguyên hoàn toàn (Pure integer linear program)

$$(H) \quad z_h = c^T x \quad \longrightarrow Max$$

$$\begin{cases} Ax \le b, \\ x \ge 0, \text{ nguyên} \end{cases} \tag{1}$$

- Trong đó $c^T = (c_1\ c_2\ \dots\ c_n)$, A là ma trận $m \times n$, $b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$, với $x \in Z^n$.
- Bài toán (H) gọi là bài toán **Tối ưu nguyên hoàn toàn.**
- Tập $S_h := \{x \in Z^n_+ : Ax \le b\}$ là tập nghiệm của bài toán Tối ưu nguyên hoàn toàn.

Tối ưu nguyên hoàn toàn (Pure integer linear program)

$$(H) \quad z_h = c^T x \quad \longrightarrow Max$$
$$\begin{cases} Ax \leq b, \\ x \geq 0, \text{ nguyên} \end{cases} \tag{1}$$

- Trong đó $c^T = (c_1 \; c_2 \; \dots \; c_n)$, A là ma trận $m \times n$, $b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$, với $x \in Z^n$.
- Bài toán (H) gọi là bài toán **Tối ưu nguyên hoàn toàn.**
- Tập $S_h := \{x \in Z_+^n : Ax \leq b\}$ là tập nghiệm của bài toán Tối ưu nguyên hoàn toàn.

Tối ưu nguyên hoàn toàn (Pure integer linear program)

$$(H) \quad z_h = c^T x \quad \longrightarrow Max$$
$$\begin{cases} Ax \leq b, \\ x \geq 0, \text{ nguyên} \end{cases} \tag{1}$$

- Trong đó $c^T = (c_1\ c_2\ \dots\ c_n)$, A là ma trận $m \times n$, $b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$, với $x \in Z^n$.
- Bài toán (H) gọi là bài toán **Tối ưu nguyên hoàn toàn.**
- Tập $S_h := \{x \in Z_+^n : Ax \leq b\}$ là tập nghiệm của bài toán Tối ưu nguyên hoàn toàn.

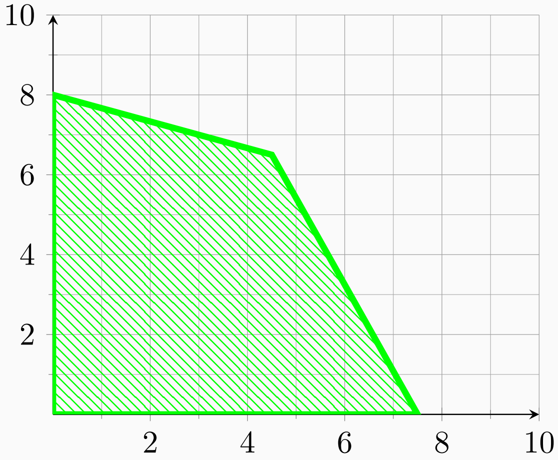
Tối ưu nguyên hoàn toàn (Pure integer linear program)

$$(H) \quad z_h = c^T x \quad \longrightarrow Max$$
$$\begin{cases} Ax \le b, \\ x \ge 0, \text{ nguyên} \end{cases} \qquad (1)$$

- Trong đó $c^T = (c_1\ c_2\ \ldots\ c_n)$, A là ma trận $m \times n$, $b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$, với $x \in Z^n$.
- Bài toán (H) gọi là bài toán **Tối ưu nguyên hoàn toàn.**
- Tập $S_h := \{x \in Z_+^n : Ax \le b\}$ là tập nghiệm của bài toán Tối ưu nguyên hoàn toàn.

Minh hoạ bài toán

$$
\begin{aligned}
&2x_1 + 2x_2 \quad \longrightarrow Max \\
&\begin{cases}
x_1 + 3x_2 \leq 24 \\
\dfrac{13}{3}x_1 + 2x_2 \leq 32.5 \\
x_1 \geq 0, \ \text{nguyên.} \\
x_2 \geq 0, \ \text{nguyên.}
\end{cases}
\end{aligned}
\tag{2}
$$



Hình 2: Tập chấp nhận được của bài toán Tối ưu nguyên hoàn toàn.

Tối ưu nguyên bộ phận (Mixed integer linear program)

$$(B) \quad z_b = c^T x + h^T y \quad \longrightarrow Max$$

$$\begin{cases} Ax + Gy \le b, \\ x \ge 0, \text{ nguyên} \\ y \ge 0. \end{cases} \tag{3}$$

- Trong đó $c^T = (c_1\ c_2\ \dots\ c_n)$, $h^T = (h_1\ h_2\ \dots\ h_p)$, A là ma trận $m \times n$, G là ma trận $m \times p$, $b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$, với $x \in Z^n$ và $y \in R^p$.
- Bài toán (B) gọi là bài toán **Tối ưu nguyên bộ phận.**
- Tập $S_b := \{(x, y) \in Z_+^n \times R_+^p : Ax + Gy \le b\}$ là tập nghiệm của bài toán Tối ưu nguyên bộ phận.

Tối ưu nguyên bộ phận (Mixed integer linear program)

$$(B) \quad z_b = c^T x + h^T y \quad \longrightarrow Max$$

$$\begin{cases} Ax + Gy \leq b, \\ x \geq 0, \text{ nguyên} \\ y \geq 0. \end{cases} \tag{3}$$

- Trong đó $c^T = (c_1\ c_2\ \dots\ c_n)$, $h^T = (h_1\ h_2\ \dots\ h_p)$, A là ma trận $m \times n$, G là ma trận $m \times p$, $b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$, với $x \in Z^n$ và $y \in R^p$.
- Bài toán (B) gọi là bài toán **Tối ưu nguyên bộ phận.**
- Tập $S_b := \{(x, y) \in Z_+^n \times R_+^p : Ax + Gy \leq b\}$ là tập nghiệm của bài toán Tối ưu nguyên bộ phận.

Tối ưu nguyên bộ phận (Mixed integer linear program)

$$(B) \quad z_b = c^T x + h^T y \quad \longrightarrow Max$$

$$\begin{cases} Ax + Gy \leq b, \\ x \geq 0, \text{ nguyên} \\ y \geq 0. \end{cases} \tag{3}$$

- Trong đó $c^T = (c_1\ c_2\ \ldots\ c_n)$, $h^T = (h_1\ h_2\ \ldots\ h_p)$, A là ma trận $m \times n$, G là ma trận $m \times p$, $b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$, với $x \in Z^n$ và $y \in R^p$.
- Bài toán (B) gọi là bài toán **Tối ưu nguyên bộ phận.**
- Tập $S_b := \{(x, y) \in Z_+^n \times R_+^p : Ax + Gy \leq b\}$ là tập nghiệm của bài toán Tối ưu nguyên bộ phận.

Tối ưu nguyên bộ phận (Mixed integer linear program)

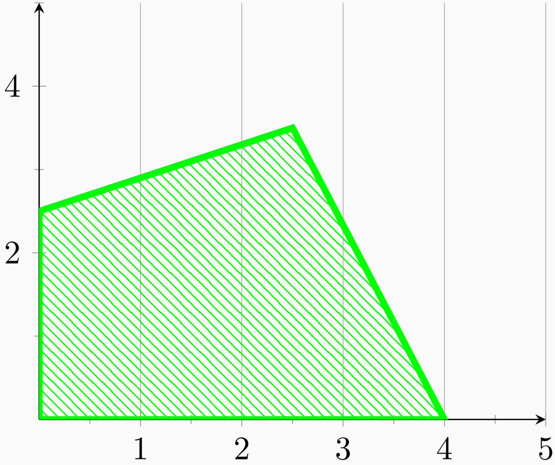
$$(B) \quad z_b = c^T x + h^T y \quad \longrightarrow Max$$
$$\begin{cases} Ax + Gy \leq b, \\ x \geq 0, \text{ nguyên} \\ y \geq 0. \end{cases} \tag{3}$$

- Trong đó $c^T = (c_1\ c_2\ \dots\ c_n)$, $h^T = (h_1\ h_2\ \dots\ h_p)$, A là ma trận $m \times n$, G là ma trận $m \times p$, $b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$, với $x \in Z^n$ và $y \in R^p$.
- Bài toán (B) gọi là bài toán **Tối ưu nguyên bộ phận.**
- Tập $S_b := \{(x, y) \in Z_+^n \times R_+^p : Ax + Gy \leq b\}$ là tập nghiệm của bài toán Tối ưu nguyên bộ phận.

Minh hoạ bài toán

$$x_1 + 2x_2 \quad \longrightarrow Max$$

$$\begin{cases} 5x_1 + \dfrac{15}{7}x_2 \leq 20 \\ -2.4x_1 + \dfrac{30}{7}x_2 \leq 15 \\ x_1 \geq 0, \text{ nguyên.} \\ x_2 \geq 0. \end{cases} \tag{4}$$



Hình 3: Tập chấp nhận được của bài toán Tối ưu nguyên bộ phận.

Phương pháp Gomory

Giới thiệu

Ta xét:

$$(P) \quad \text{Min} \quad \langle c, x \rangle$$
$$\text{s.t} \quad \begin{cases} Ax = b, \\ x_j \geq 0, j = 1, 2, ..., n. \end{cases} \tag{5}$$

Ta ký hiệu tập $F \subset \mathbb{R}^n$ là miền xác định của bài toán (P).

Giới thiệu

$$
\begin{aligned}
(P^N) \quad & \text{Min} \quad \langle c, x \rangle \\
\text{s.t} \quad & \begin{cases} Ax = b, \\ x_j \geq 0, j = 1, 2, ..., n. \\ x_j \text{ nguyên}, j = 1, 2.., n_1 \ (n_1 \leq n). \end{cases}
\end{aligned}
\tag{6}
$$

Ta gọi:

P^N là bài toán tối ưu nguyên.

F^N là miền xác định của bài toán.

Ý tưởng về phương pháp cắt

Khái niệm lát cắt đúng

Giả sử bài toán P^N là bài toán quy hoạch nguyên nào đó và phương án tối ưu của bài toán quy hoạch tuyến tính tương ứng x không thoả mãn điều kiện nguyên, tức là $x \notin F^N$.
Khi đó, bất đẳng thức:

$$\sum_j a_j x_j \leq \beta$$

được gọi là lát cắt đúng nếu thỏa mãn hai điều kiện.

1) Điều kiện cắt:

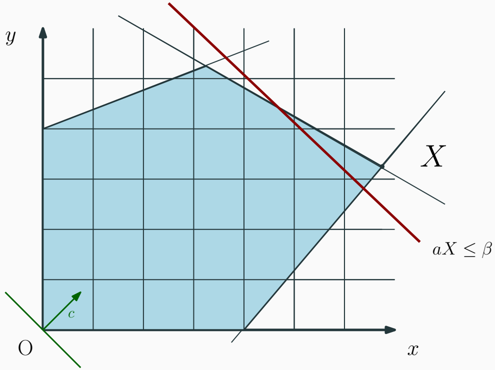
Phương án x không thỏa mãn điều kiện (25), tức là $Ax > \beta$.

2) Điều kiện đúng:

Nếu x là phương án của bài toán tối ưu nguyên thì x thỏa mãn điều kiện (25), tức là $F^N \subset \{x \mid ax \leq \beta\}$.

Khái niệm lát cắt đúng

Nói cách khác, lát cắt thêm vào sẽ không cắt đi một phương án nguyên nào của bài toán.



Ý tưởng phương pháp cắt của Danzig

Việc giải một bài toán P^N là một quá trình gồm nhiều bước:
a) Ở bước thứ r, giải bài toán bài toán quy hoạch tuyến tính phụ $P_r, r = 0, 1, \ldots$ với $F_0 = F$.
b) Tập các điểm nguyên của tất cả các đa diện lồi là như nhau:

$$F_0^N = F_1^N = F_2^N = \ldots = F_r^N = \ldots$$

Do đó, nếu phương án tối ưu x_r^* của bài toán P_r thoả mãn điều kiện nguyên thì nó cũng là phương án tối ưu x_0 của bài toán xuất phát P_0^N và quá trình kết thúc.

Ý tưởng phương pháp cắt của Danzig

c) Nếu x_r^* không thoả mãn điều kiện nguyên thì x_r^* không phải là phương án của bài toán P_{r+1}, tức là $x_r^* \notin F_{r+1}$.
Chuyển từ bước r sang bước $r+1$, tức là chuyển từ bài toán P_r sang P_{r+1} khi x_r^* không nguyên được thực hiện nhờ một lát cắt đúng
$a_r x \leq \beta_r$.
Việc bổ sung lát cắt này vào ràng buộc của bài toán P_r sẽ chuyển đa diện lồi F_r thành F_{r+1}.

Thuật toán Gomory

Cơ sở lý thuyết

Ta xét bài toán tối ưu nguyên hoàn toàn:

$$(P^N) \quad \text{Max}\langle c, x\rangle$$
$$\text{s.t} \begin{cases} Ax = b, \\ x_j \geq 0, j = 1, 2, ..., n. \\ x_j \text{nguyên}, j = 1, 2.., n. \end{cases} \tag{7}$$

Định nghĩa 4.1
Giả sử hệ véc-tơ $\{A^j, j \in J\}$ là cơ sở tương ứng với phương án cực biên ban đầu của bài toán P^N, các véc-tơ A^j và các biến x_j với $j \in J$ được gọi là các véc tơ cơ sở và biến cơ sở; còn các véc-tơ A^j và các biến x_j mà $j \notin J$ được gọi là các véc-tơ tự do và các biến tự do (biến phi cơ sở).

Cơ sở lý thuyết

Giả sử x là phương án tối ưu của bài toán P^N, từ đó ta có thể biểu diễn các biến cơ sở qua các biến phi cơ sở:

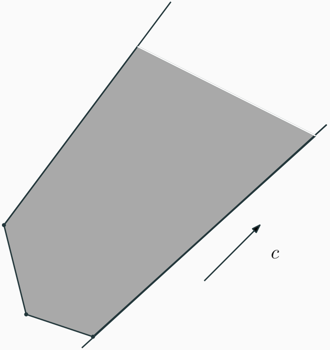
$$x_i = x_{i0} + \sum_{j \in N} x_{ij}(-x_j), i = \overline{0, m}. \tag{8}$$

Định lý 4.1
Giả sử x có x_{i0} không nguyên với $1 \leq i \leq n$ và:
1)
$$z_i(x) = -\{x_{i0}\} + \sum_{j \in \mathbb{N}} (-\{x_{ij}\})(-x_j), i = \overline{1, n}. \tag{9}$$
2) x là phương án của bài toán P^N.
Khi đó:
a) z_i nguyên.
b) $z_i \geq 0$.

Cơ sở lý thuyết

Hệ quả 4.1
Giả sử x không thoả mãn điều kiện nguyên, như vậy tồn tại x_{i0} $(1 \leq i \leq n)$ không nguyên . Khi đó các hệ thức (9) *và $z_i \geq 0$ xác định một lát cắt đúng.*

Dấu hiệu bài toán không có lời giải



Hình 4: $\overline{P}$ không có lời giải

Dấu hiệu bài toán không có lời giải

Về sau ta sẽ giả thiết:
1) Hàm mục tiêu $x_0 \equiv cx$ bị chặn trên F.
2) Nếu tập hợp các phương án tối ưu của P khác rỗng thì nó phải bị chặn, tức là nếu bài toán P giải được thì bài toán $\overline{P}$ cũng giải được.

Thuật toán Gomory

Bước 1: Giải bài toán $P \equiv P_0$ đã cho bằng phương pháp đơn hình đối ngẫu.
- Nếu P_0 không giải được thì P_0^N cũng không giải được.
- Nếu P_0 giải được và nghiệm của nó thỏa mãn điều kiện nguyên thì nó cũng là phương án tối ưu của P_0^N, còn nếu chưa thỏa điều kiện thì chuyển sang bước 2.

Thuật toán Gomory

Bước 2: Chọn dòng đầu tiên ứng với thành phần không nguyên:
$k = min\{i|i \in \{1, ..., n\}, x^r_{i0}$không nguyên$\}$ và xây dựng lát cắt đúng:

$$\begin{cases} x_{n+r+1} = -\{x^r_{k0}\} + \sum_{j \in N_r}(-\{x^r_{kj}\})(-x_j) \\ x_{n+r+1} \geq 0 \\ x_{n+r+1} \text{ nguyên} \end{cases}$$

Thêm lát cắt vào bảng đơn hình và tiếp tục giải bài toán P^N_{r+1}.
Bước 3: Sau khi tính toán với lát cắt nếu được phương án tối ưu thỏa mãn điều kiện nguyên thì thuật toán dừng lại. Nếu không thỏa mãn thì quay lại bước 2 cứ lần lượt như vậy thực hiện các bước lặp $r \geq 0$ cho đến khi thỏa mãn điều kiện.

Tính hữu hạn của thuật toán

Định lý 4.2
Giả sử có các điều kiện sau:
1) Tính nguyên của hàm mục tiêu $x_0 = cx$ được đảm bảo và x_0 được xét khi chọn dòng xây dựng lát cắt đúng.
2) Một trong các khẳng định sau là đúng:
i) Hàm mục tiêu x_0 bị chặn dưới trên F_0.
ii) Bài toán P_0^N có ít nhất một phương án x'.
Khi đó thuật toán Gomory thứ nhất kết thúc sau một số hữu hạn bước lặp lớn.

Phương pháp Land-Doig

Ý tưởng phương pháp Land-Doig

- **Phương pháp Land-Doig (Nhánh cận)**: Chia bài toán gốc thành các bài toán nhỏ và xử lý đến khi tìm ra kết quả tối ưu.
- **Cách hoạt động**: Thuật toán Land-Doig là một khung thuật toán chia bài toán thành **các bài toán con** và duyệt qua từng bài toán. Từ giải pháp ban đầu, thuật toán mở rộng các nhánh (tập hợp con các bài toán).
- **Ý tưởng cốt lõi**:
  - **Phân nhánh**: Mỗi nhánh đại diện cho một tập con các bài toán.
  - **Gọt**: Loại bỏ nhánh không thỏa điều kiện nghiệm, giúp tìm giải pháp tối ưu hiệu quả.

Bài toán quan tâm

$$
(P) \quad z_p = c^T x + h^T y \quad \longrightarrow Max
$$

$$
\begin{cases} Ax + Gy \leq b, \\ x, y \geq 0. \end{cases} \tag{10}
$$

- Trong đó (P) là bài toán (B) (hoặc (H)) với nghiệm thuộc tập số thực.
- Bài toán (P) là một bài toán **Tối ưu tuyến tính thông thường** hay gọi đơn giản là bài toán Tối ưu tuyến tính (Natural linear programming relaxation).
- Tập $S_p := \{(x, y) \in R^n_+ \times R^p_+ : Ax + Gy \leq b\}$ là tập nghiệm của bài toán Tối ưu tuyến tính.

Bài toán quan tâm

$$(P) \quad z_p = c^T x + h^T y \quad \longrightarrow Max$$
$$\begin{cases} Ax + Gy \leq b, \\ x, y \geq 0. \end{cases} \tag{10}$$

- Trong đó (P) là bài toán (B) (hoặc (H)) với nghiệm thuộc tập số thực.
- Bài toán (P) là một bài toán **Tối ưu tuyến tính thông thường** hay gọi đơn giản là bài toán Tối ưu tuyến tính (Natural linear programming relaxation).
- Tập $S_p := \{(x, y) \in R^n_+ \times R^p_+ : Ax + Gy \leq b\}$ là tập nghiệm của bài toán Tối ưu tuyến tính.

Bài toán quan tâm

$$(P) \quad z_p = c^T x + h^T y \quad \longrightarrow Max$$
$$\begin{cases} Ax + Gy \leq b, \\ x, y \geq 0. \end{cases} \tag{10}$$

- Trong đó (P) là bài toán (B) (hoặc (H)) với nghiệm thuộc tập số thực.
- Bài toán (P) là một bài toán **Tối ưu tuyến tính thông thường** hay gọi đơn giản là bài toán Tối ưu tuyến tính (Natural linear programming relaxation).
- Tập $S_p := \{(x, y) \in R_+^n \times R_+^p : Ax + Gy \leq b\}$ là tập nghiệm của bài toán Tối ưu tuyến tính.

Bài toán quan tâm

$$(P) \quad z_p = c^T x + h^T y \quad \longrightarrow Max$$
$$\begin{cases} Ax + Gy \leq b, \\ x, y \geq 0. \end{cases} \tag{10}$$

- Trong đó (P) là bài toán (B) (hoặc (H)) với nghiệm thuộc tập số thực.
- Bài toán (P) là một bài toán **Tối ưu tuyến tính thông thường** hay gọi đơn giản là bài toán Tối ưu tuyến tính (Natural linear programming relaxation).
- Tập $S_p := \{(x, y) \in R^n_+ \times R^p_+ : Ax + Gy \leq b\}$ là tập nghiệm của bài toán Tối ưu tuyến tính.

Mục tiêu

Giả sử ta nhận được tập phương án tối ưu của bài toán (B) sau hữu hạn lần giải, ký hiệu (x_b, y_b) và giá trị tối ưu là z_b thì ta có nhận xét sau:

Nhận xét 5.1

- *Nếu $S_b \subseteq S_p$ thì ta luôn nhận được $z_b \leq z_p$ và phương án có thể cải thiện.*
- *Nếu $S_b = S_p$ thì ta nhận được $z_b = z_p$ và bài toán được giải.*

Vì thế, ta chọn xử lý bài toán (B) (hoặc (H)) thông qua bài toán (P) bằng cách cải thiện phương án thu được từ bài toán (P) sao cho thoả điều kiện của bài toán (B) (hoặc (H)).

Mục tiêu

Giả sử ta nhận được tập phương án tối ưu của bài toán (B) sau hữu hạn lần giải, ký hiệu (x_b, y_b) và giá trị tối ưu là z_b thì ta có nhận xét sau:

Nhận xét 5.1

- *Nếu $S_b \subseteq S_p$ thì ta luôn nhận được $z_b \leq z_p$ và phương án có thể cải thiện.*
- *Nếu $S_b = S_p$ thì ta nhận được $z_b = z_p$ và bài toán được giải.*

Vì thế, ta chọn xử lý bài toán (B) (hoặc (H)) thông qua bài toán (P) bằng cách cải thiện phương án thu được từ bài toán (P) sao cho thoả điều kiện của bài toán (B) (hoặc (H)).

Mục tiêu

Giả sử ta nhận được tập phương án tối ưu của bài toán (B) sau hữu hạn lần giải, ký hiệu (x_b, y_b) và giá trị tối ưu là z_b thì ta có nhận xét sau:

Nhận xét 5.1

- *Nếu $S_b \subseteq S_p$ thì ta luôn nhận được $z_b \leq z_p$ và phương án có thể cải thiện.*
- *Nếu $S_b = S_p$ thì ta nhận được $z_b = z_p$ và bài toán được giải.*

Vì thế, ta chọn xử lý bài toán (B) (hoặc (H)) thông qua bài toán (P) bằng cách cải thiện phương án thu được từ bài toán (P) sao cho thoả điều kiện của bài toán (B) (hoặc (H)).

Ví dụ

$$(P) \quad 5.5x_1 + 2.1x_2 \longrightarrow Max$$

$$\begin{cases} -1x_1 & +x_2 & \leq & 2 \\ 8x_1 & +2x_2 & \leq & 17 \\ x_1 & & \geq & 0, \\ & x_2 & \geq & 0. \end{cases} \Longrightarrow \begin{cases} x_1 & = & 1.3 \\ x_2 & = & 3.3 \\ z & = & 14.08 \end{cases} \qquad (11)$$

Phương án có thể cải thiện.

Ví dụ

$$(P) \quad 3x_1 + 4x_2 \quad \longrightarrow Max$$

$$\left\{\begin{array}{cccc} 2.5x_1 & +\frac{15}{4}x_2 & \leq & 20 \\ x_1 & +\frac{5}{3}x_2 & \leq & \frac{50}{3} \\ x_1 & & \geq & 0, \\ & x_2 & \geq & 0. \end{array}\right. \Longrightarrow \left\{\begin{array}{ccc} x_1 & = & 5 \\ x_2 & = & 7 \\ z & = & 43 \end{array}\right. \tag{12}$$

Bài toán được giải.

Thuật toán Land-Doig

Phương pháp xác định cận

Ta gọi x_j với $1 \leq j \leq n$ là thành phần thứ j trong nghiệm thu được từ bài toán (P).

Định lý 6.1

Với mỗi $x_j \in \mathbb{R}$, tồn tại duy nhất số nguyên $k \in \mathbb{Z}$ sao cho $k \leq x_j < k + 1$.

Định nghĩa 6.1

- *Giá trị k khi đó ta gọi là phần nguyên nhỏ nhất của x_j, ký hiệu là $\lfloor x_j \rfloor$.*
- *Giá trị $k + 1$ gọi là phần nguyên lớn nhất của x_j, ký hiệu là $\lceil x_j \rceil$.*

Ví dụ 6.1
Ta có $x_1 = 3.3$, vậy khi đó phần nguyên nhỏ nhất của x_1 là $\lfloor x_1 \rfloor = 3$ và phần nguyên lớn nhất là $\lceil x_1 \rceil = 4$.

Phương pháp xử lý bài toán

- Từ bài toán minh hoạ (11) và (12), ta thấy rằng nếu $\exists x_j \notin \mathbb{Z}$, thì ta có thể tiếp tục **cải thiện phương án** cho đến khi $\forall x_j \in \mathbb{Z}$.

- Ta cải thiện như sau, nếu nghiệm thu được là $x_j \notin \mathbb{Z}$ ta thiết lập được 2 **bài toán con** từ bài toán (P) ban đầu, ký hiệu (P_1) và (P_2).

Phương pháp xử lý bài toán

- Từ bài toán minh hoạ (11) và (12), ta thấy rằng nếu $\exists x_j \notin \mathbb{Z}$, thì ta có thể tiếp tục **cải thiện phương án** cho đến khi $\forall x_j \in \mathbb{Z}$.
- Ta cải thiện như sau, nếu nghiệm thu được là $x_j \notin \mathbb{Z}$ ta thiết lập được 2 **bài toán con** từ bài toán (P) ban đầu, ký hiệu (P_1) và (P_2).

$$(P_1) \quad z_1 = c^T x + h^T y \quad \longrightarrow Max$$
$$\begin{cases} Ax + Gy \leq b \\ x_j \leq \lfloor x_j \rfloor, \\ x, y \geq 0. \end{cases} \tag{13}$$

- Tập $S_1 := S_p \cap \{(x, y) : x_j \leq \lfloor x_j \rfloor\}$ là tập nghiệm tối ưu của bài toán con (P_1).

$$
\begin{aligned}
(P_2) \quad & z_2 = c^T x + h^T y \quad \longrightarrow Max \\
& \begin{cases} Ax + Gy \leq b \\ x_j \geq \lceil x_j \rceil, \\ x, y \geq 0. \end{cases}
\end{aligned} \tag{14}
$$

- Tập $S_2 := S_p \cap \{(x, y) : x_j \geq \lceil x_j \rceil\}$ là tập nghiệm tối ưu của bài toán con (P_2).

Điều kiện nghiệm

- Nếu tồn tại (P_i) với $i = 1, 2$ không giải được $(S_i = \emptyset)$ thì bài toán **vô nghiệm**.

- Giả sử $x^{(i)}$ là nghiệm tối ưu của bài toán (P_i) và giá trị tối ưu là z_i với $i = 1, 2$.

Chú ý 6.1

Ta gọi $x_j^{(i)}$ là biến thứ j của bài toán thứ i.

Điều kiện nghiệm

- Nếu tồn tại (P_i) với $i = 1, 2$ không giải được $(S_i = \emptyset)$ thì bài toán **vô nghiệm**.
- Giả sử $x^{(i)}$ là nghiệm tối ưu của bài toán (P_i) và giá trị tối ưu là z_i với $i = 1, 2$.
  - Nếu $\forall x^{(i)} \in Z_+^n$, ta nói S_i là tập nghiệm thoả mãn bài toán tối ưu nguyên bộ phận, z_i^* là giá trị tối ưu và bài toán con (P_i) được giải **(gọt bởi nghiệm nguyên)**.
  - Nếu $\exists x^{(i)} \notin Z_+^n$ đồng thời $z_i \leq z_i^*$, ta dừng phân nhánh và bỏ qua bài toán **(gọt bởi cận)**.
  - Nếu $\exists x^{(i)} \notin Z_+^n$ đồng thời $z_i > z_i^*$, bài toán chưa tối ưu và có thể tiếp tục cải thiện.

Chú ý 6.1
Ta gọi $x_j^{(i)}$ là biến thứ j của bài toán thứ i.

Điều kiện nghiệm

- Nếu tồn tại (P_i) với $i = 1, 2$ không giải được $(S_i = \emptyset)$ thì bài toán **vô nghiệm**.
- Giả sử $x^{(i)}$ là nghiệm tối ưu của bài toán (P_i) và giá trị tối ưu là z_i với $i = 1, 2$.
    - Nếu $\forall x^{(i)} \in Z_+^n$, ta nói S_i là tập nghiệm thoả mãn bài toán tối ưu nguyên bộ phận, z_i^* là giá trị tối ưu và bài toán con (P_i) được giải **(gọt bởi nghiệm nguyên)**.
    - Nếu $\exists x^{(i)} \notin Z_+^n$ đồng thời $z_i \leq z_i^*$, ta dừng phân nhánh và bỏ qua bài toán **(gọt bởi cận)**.
    - Nếu $\exists x^{(i)} \notin Z_+^n$ đồng thời $z_i > z_i^*$, bài toán chưa tối ưu và có thể tiếp tục cải thiện.

Chú ý 6.1

Ta gọi $x_j^{(i)}$ là biến thứ j của bài toán thứ i.

Điều kiện nghiệm

- Nếu tồn tại (P_i) với $i = 1, 2$ không giải được ($S_i = \emptyset$) thì bài toán **vô nghiệm**.
- Giả sử $x^{(i)}$ là nghiệm tối ưu của bài toán (P_i) và giá trị tối ưu là z_i với $i = 1, 2$.
  - Nếu $\forall x^{(i)} \in Z_+^n$, ta nói S_i là tập nghiệm thoả mãn bài toán tối ưu nguyên bộ phận, z_i^* là giá trị tối ưu và bài toán con (P_i) được giải **(gọt bởi nghiệm nguyên)**.
  - Nếu $\exists x^{(i)} \notin Z_+^n$ đồng thời $z_i \leq z_i^*$, ta dừng phân nhánh và bỏ qua bài toán **(gọt bởi cận)**.
  - Nếu $\exists x^{(i)} \notin Z_+^n$ đồng thời $z_i > z_i^*$, bài toán chưa tối ưu và có thể tiếp tục cải thiện.

Chú ý 6.1

Ta gọi $x_j^{(i)}$ là biến thứ j của bài toán thứ i.

Điều kiện nghiệm

- Nếu tồn tại (P_i) với $i = 1, 2$ không giải được $(S_i = \emptyset)$ thì bài toán **vô nghiệm**.
- Giả sử $x^{(i)}$ là nghiệm tối ưu của bài toán (P_i) và giá trị tối ưu là z_i với $i = 1, 2$.
    - Nếu $\forall x^{(i)} \in Z_+^n$, ta nói S_i là tập nghiệm thoả mãn bài toán tối ưu nguyên bộ phận, z_i^* là giá trị tối ưu và bài toán con (P_i) được giải **(gọt bởi nghiệm nguyên)**.
    - Nếu $\exists x^{(i)} \notin Z_+^n$ đồng thời $z_i \leq z_i^*$, ta dừng phân nhánh và bỏ qua bài toán **(gọt bởi cận)**.
    - Nếu $\exists x^{(i)} \notin Z_+^n$ đồng thời $z_i > z_i^*$, bài toán chưa tối ưu và có thể tiếp tục cải thiện.

Chú ý 6.1

Ta gọi $x_j^{(i)}$ là biến thứ j của bài toán thứ i.

Điều kiện nghiệm

- Nếu tồn tại (P_i) với $i = 1, 2$ không giải được $(S_i = \emptyset)$ thì bài toán **vô nghiệm**.
- Giả sử $x^{(i)}$ là nghiệm tối ưu của bài toán (P_i) và giá trị tối ưu là z_i với $i = 1, 2$.
  - Nếu $\forall x^{(i)} \in Z_+^n$, ta nói S_i là tập nghiệm thoả mãn bài toán tối ưu nguyên bộ phận, z_i^* là giá trị tối ưu và bài toán con (P_i) được giải **(gọt bởi nghiệm nguyên)**.
  - Nếu $\exists x^{(i)} \notin Z_+^n$ đồng thời $z_i \leq z_i^*$, ta dừng phân nhánh và bỏ qua bài toán **(gọt bởi cận)**.
  - Nếu $\exists x^{(i)} \notin Z_+^n$ đồng thời $z_i > z_i^*$, bài toán chưa tối ưu và có thể tiếp tục cải thiện.

Chú ý 6.1
Ta gọi $x_j^{(i)}$ là biến thứ j của bài toán thứ i.

Ví dụ minh hoạ

$$
(H) \quad z_p = 5.5x_1 + 2.1x_2 \quad \longrightarrow Max
$$

$$
\begin{cases}
-x_1 & +x_2 & \leq & 2 \\
8x_1 & +2x_2 & \leq & 17 \\
x_1 & & \geq & 0, \text{ Nguyên} \\
 & x_2 & \geq & 0, \text{ Nguyên.}
\end{cases}
$$

Ví dụ minh hoạ

$$(P) \quad z_p = 5.5x_1 + 2.1x_2 \quad \longrightarrow Max$$

$$\begin{cases} -x_1 & +x_2 & \leq & 2 \\ 8x_1 & +2x_2 & \leq & 17 \\ x_1 & & \geq & 0, \\ & x_2 & \geq & 0. \end{cases}$$

Giải bài toán (P) bằng phương pháp đơn hình thông thường ta được nghiệm $x_1 = 1.3$, $x_2 = 3.3$ và $z_p = 14.08$.

Hình 5: Tập chấp nhận được của bài toán (P).

Chọn $x_1 = 1.3$ để cải thiện phương án, ta thu được 2 bài toán con sau:

$$(P_1) \quad z_1 = 5.5x_1^{(1)} + 2.1x_2^{(1)}$$

$$\begin{cases} -x_1^{(1)} & +x_2^{(1)} & \leq & 2 \\ 8x_1^{(1)} & +2x_2^{(1)} & \leq & 17 \\ x_1^{(1)} & & \leq & 1 \\ x_1^{(1)} & & \geq & 0 \\ & x_2^{(1)} & \geq & 0. \end{cases}$$

$$(P_2) \quad z_2 = 5.5x_1^{(2)} + 2.1x_2^{(2)}$$

$$\begin{cases} -x_1^{(2)} & +x_2^{(2)} & \leq & 2 \\ 8x_1^{(2)} & +2x_2^{(2)} & \leq & 17 \\ x_1^{(2)} & & \geq & 2 \\ x_1^{(2)} & & \geq & 0 \\ & x_2^{(2)} & \geq & 0. \end{cases}$$

Chọn $x_1 = 1.3$ để cải thiện phương án, ta thu được 2 bài toán con sau:

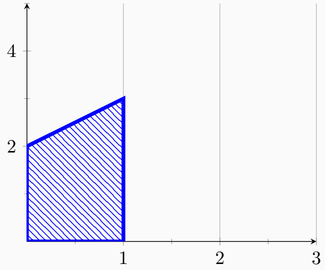
$$(P_1) \quad z_1 = 5.5x_1^{(1)} + 2.1x_2^{(1)}$$
$$\begin{cases} -x_1^{(1)} & +x_2^{(1)} & \leq & 2 \\ 8x_1^{(1)} & +2x_2^{(1)} & \leq & 17 \\ \color{blue}{x_1^{(1)}} & & \color{blue}{\leq} & \color{blue}{1} \\ x_1^{(1)} & & \geq & 0 \\ & x_2^{(1)} & \geq & 0. \end{cases}$$

$$(P_2) \quad z_2 = 5.5x_1^{(2)} + 2.1x_2^{(2)}$$
$$\begin{cases} -x_1^{(2)} & +x_2^{(2)} & \leq & 2 \\ 8x_1^{(2)} & +2x_2^{(2)} & \leq & 17 \\ \color{blue}{x_1^{(2)}} & & \color{blue}{\geq} & \color{blue}{2} \\ x_1^{(2)} & & \geq & 0 \\ & x_2^{(2)} & \geq & 0. \end{cases}$$

$$(P_1) \quad z_1 = 5.5x_1^{(1)} + 2.1x_2^{(1)}$$

$$\begin{cases} -x_1^{(1)} & +x_2^{(1)} & \leq & 2 \\ 8x_1^{(1)} & +2x_2^{(1)} & \leq & 17 \\ x_1^{(1)} & & \leq & 1 \\ x_1^{(1)} & & \geq & 0 \\ & x_2^{(1)} & \geq & 0. \end{cases}$$

Giải bài toán (P_1) ta được $x_1^{(1)} = 1, x_2^{(1)} = 3$ và $z_1 = 11.8$. Bài toán được giải **(gọt bởi nghiệm nguyên)**.



Hình 6: Tập chấp nhận được của bài toán (P_1).

- Tương tự bài toán (P_2) ta được $x_1^{(2)} = 2, x_2^{(2)} = 0.5$ và $z_2 = 12.05$.

- Chọn $x_2^{(2)} = 0.5$ để cải thiện phương án. Ta được 2 bài toán con (P_3) và (P_4):

$$(P_3) \quad z_3 = 5.5x_1^{(3)} + 2.1x_2^{(3)}$$

$$\begin{cases} -x_1^{(3)} & +x_2^{(3)} & \leq & 2 \\ 8x_1^{(3)} & +2x_2^{(3)} & \leq & 17 \\ x_1^{(3)} & & \geq & 2 \\ & x_2^{(3)} & \leq & 0 \\ x_1^{(3)} & & \geq & 0 \\ & x_2^{(3)} & \geq & 0. \end{cases}$$

$$(P_4) \quad z_4 = 5.5x_1^{(4)} + 2.1x_2^{(4)}$$

$$\begin{cases} -x_1^{(4)} & +x_2^{(4)} & \leq & 2 \\ 8x_1^{(4)} & +2x_2^{(4)} & \leq & 17 \\ x_1^{(4)} & & \geq & 2 \\ & x_2^{(4)} & \geq & 1 \\ x_1^{(4)} & & \geq & 0 \\ & x_2^{(4)} & \geq & 0. \end{cases}$$

- Tương tự bài toán (P_2) ta được $x_1^{(2)} = 2, x_2^{(2)} = 0.5$ và $z_2 = 12.05$.

- Chọn $x_2^{(2)} = 0.5$ để cải thiện phương án. Ta được 2 bài toán con (P_3) và (P_4):

$$(P_3) \quad z_3 = 5.5x_1^{(3)} + 2.1x_2^{(3)}$$
$$\begin{cases} -x_1^{(3)} & +x_2^{(3)} & \leq & 2 \\ 8x_1^{(3)} & +2x_2^{(3)} & \leq & 17 \\ x_1^{(3)} & & \geq & 2 \\ & x_2^{(3)} & \leq & 0 \\ x_1^{(3)} & & \geq & 0 \\ & x_2^{(3)} & \geq & 0. \end{cases}$$

$$(P_4) \quad z_4 = 5.5x_1^{(4)} + 2.1x_2^{(4)}$$
$$\begin{cases} -x_1^{(4)} & +x_2^{(4)} & \leq & 2 \\ 8x_1^{(4)} & +2x_2^{(4)} & \leq & 17 \\ x_1^{(4)} & & \geq & 2 \\ & x_2^{(4)} & \geq & 1 \\ x_1^{(4)} & & \geq & 0 \\ & x_2^{(4)} & \geq & 0. \end{cases}$$

- Giải bài toán (P_3) ta được $x_1^{(3)} = 2.125, x_2^{(3)} = 0$ và $z_3 = 11.6875 \Rightarrow$ không khả thi do $z_3 < z_1$ **(gọt bởi cận)**.
- Bài toán (P_4) **vô nghiệm**.
- Vậy phương án tối ưu của bài toán (H) là $x_1^{(1)} = 1, x_2^{(1)} = 3$ và $z = 11.8$.

- Giải bài toán (P_3) ta được $x_1^{(3)} = 2.125, x_2^{(3)} = 0$ và $z_3 = 11.6875 \Rightarrow$ không khả thi do $z_3 < z_1$ **(gọt bởi cận)**.
- Bài toán (P_4) **vô nghiệm**.
- Vậy phương án tối ưu của bài toán (H) là $x_1^{(1)} = 1, x_2^{(1)} = 3$ và $z = 11.8$.

Sơ đồ thuật toán

- Ta gọi bài toán (P) có nút ban đầu là N_0, tương ứng mỗi bài toán tối ưu tuyến tính thông thường (P_i) ứng với mỗi nút N_i trên sơ đồ nhánh và $\mathcal{L}$ là **danh sách chứa các nút** được lập thông qua phương pháp xác định cận và điều kiện nghiệm đã được nói trước đó.
- Ta đánh dấu giá trị tối ưu tốt nhất và nghiệm tối ưu tốt nhất của bài toán lần lượt là z^* và (x^*, y^*).

Sơ đồ thuật toán

Bước 1. Thiết lập

Đặt $\mathcal{L} := \{N_0\}$, $z^* = z_p$ và $(x^*, y^*) = (x, y)$.

Bước 2. Kiểm tra

Nếu $\mathcal{L} = \emptyset$ thì nghiệm tối ưu của bài toán là (x^*, y^*), giá trị tối ưu là z^* và bài toán được giải.

Nếu $\mathcal{L} \neq \emptyset$, chuyển sang bước 3.

Bước 3. Chọn nút

Chọn nút N_i từ danh sách $\mathcal{L}$ và xoá khỏi $\mathcal{L}$ sau đó chuyển sang bước 4.

Bước 4. Xác định cận

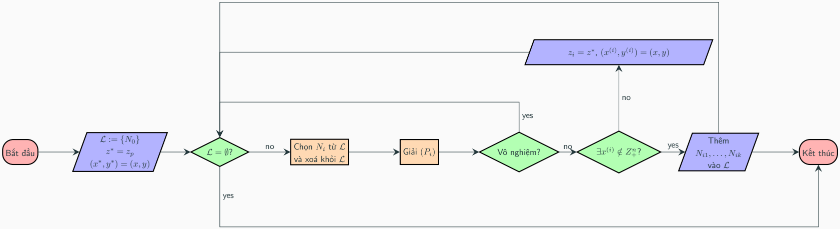
Giải bài toán (P_i), nếu bài toán vô nghiệm hoặc $z_i \leq z^*$, quay lại bước 2, nếu không, chuyển sang bước 5.

Bước 5. Gọt nghiệm

Nếu tồn tại $x^{(i)} \notin Z_+^n$, ta thêm nút $N_{i+1}, \ldots, N_k$ vào $\mathcal{L}$ và quay về bước 2.

Nếu không tồn tại $x^{(i)} \notin Z_+^n$, tức $\forall x^{(i)} \in Z_+^n$, ta đặt $z_i = z^*$, $(x^{(i)}, y^{(i)}) = (x^*, y^*)$ và quay lại bước 2.

Sơ đồ thuật toán



Hình 7: Lưu đồ giải thuật của thuật toán nhánh cận.

Kết luận và Hướng phát triển

Tóm tắt kết quả thu được

Đề tài nghiên cứu này đạt được các vấn đề sau:

- Giới thiệu về bài toán **Tối ưu tuyến tính cho ra nghiệm nguyên**.
- Trình bày về ý tưởng **Lát cắt và Thuật toán Gomory** để giải bài toán tối ưu nguyên.
- Tìm hiểu về **Phương pháp Land-Doig** hay còn gọi là **Phương pháp Nhánh cận**.

Tóm tắt kết quả thu được

Đề tài nghiên cứu này đạt được các vấn đề sau:

- Giới thiệu về bài toán **Tối ưu tuyến tính cho ra nghiệm nguyên**.
- Trình bày về ý tưởng **Lát cắt và Thuật toán Gomory** để giải bài toán tối ưu nguyên.
- Tìm hiểu về **Phương pháp Land-Doig** hay còn gọi là **Phương pháp Nhánh cận**.

Tóm tắt kết quả thu được

Đề tài nghiên cứu này đạt được các vấn đề sau đây:

- Giới thiệu về bài toán **Tối ưu tuyến tính cho ra nghiệm nguyên**.
- Trình bày về ý tưởng **Lát cắt và Thuật toán Gomory** để giải bài toán tối ưu nguyên.
- Tìm hiểu về **Phương pháp Land-Doig** hay còn gọi là **Phương pháp Nhánh cận**.

Tóm tắt kết quả thu được

Đề tài nghiên cứu này đạt được các vấn đề sau đây:

- Giới thiệu về bài toán **Tối ưu tuyến tính cho ra nghiệm nguyên**.
- Trình bày về ý tưởng **Lát cắt và Thuật toán Gomory** để giải bài toán tối ưu nguyên.
- Tìm hiểu về **Phương pháp Land-Doig** hay còn gọi là **Phương pháp Nhánh cận**.

Vấn đề và thách thức

Hạn chế của phương pháp Land-Doig

- **Thời gian tính toán lớn**: Việc phân nhánh tạo ra nhiều bài toán con, dẫn đến tăng chi phí tính toán.
- **Bộ nhớ tiêu tốn lớn**: Quá trình phân nhánh và lưu trữ các nhánh mở rộng tiêu tốn nhiều bộ nhớ, đặc biệt là với những bài toán có kích thước lớn.
- **Phụ thuộc vào chiến lược phân nhánh**: Hiệu quả của thuật toán phụ thuộc vào cách chọn nhánh để phân chia (cách chọn biến, cận).
- **Khó khăn trong đánh giá ràng buộc cận**: Việc xác định cận nào là tốt nhất còn hạn chế, dẫn đến khó xác định nhánh nào cần loại bỏ sớm.
- **Hiệu quả giảm dần với bài toán có nhiều ràng buộc**: Số lượng nhánh cần mở rộng có xu hướng tăng mạnh khi có nhiều ràng buộc, gây giảm hiệu quả.

Hạn chế của phương pháp Land-Doig

- **Thời gian tính toán lớn**: Việc phân nhánh tạo ra nhiều bài toán con, dẫn đến tăng chi phí tính toán.
- **Bộ nhớ tiêu tốn lớn**: Quá trình phân nhánh và lưu trữ các nhánh mở rộng tiêu tốn nhiều bộ nhớ, đặc biệt là với những bài toán có kích thước lớn.
- **Phụ thuộc vào chiến lược phân nhánh**: Hiệu quả của thuật toán phụ thuộc vào cách chọn nhánh để phân chia (cách chọn biến, cận).
- **Khó khăn trong đánh giá ràng buộc cận**: Việc xác định cận nào là tốt nhất còn hạn chế, dẫn đến khó xác định nhánh nào cần loại bỏ sớm.
- **Hiệu quả giảm dần với bài toán có nhiều ràng buộc**: Số lượng nhánh cần mở rộng có xu hướng tăng mạnh khi có nhiều ràng buộc, gây giảm hiệu quả.

Hạn chế của phương pháp Land-Doig

- **Thời gian tính toán lớn**: Việc phân nhánh tạo ra nhiều bài toán con, dẫn đến tăng chi phí tính toán.
- **Bộ nhớ tiêu tốn lớn**: Quá trình phân nhánh và lưu trữ các nhánh mở rộng tiêu tốn nhiều bộ nhớ, đặc biệt là với những bài toán có kích thước lớn.
- **Phụ thuộc vào chiến lược phân nhánh**: Hiệu quả của thuật toán phụ thuộc vào cách chọn nhánh để phân chia (cách chọn biến, cận).
- **Khó khăn trong đánh giá ràng buộc cận**: Việc xác định cận nào là tốt nhất còn hạn chế, dẫn đến khó xác định nhánh nào cần loại bỏ sớm.
- **Hiệu quả giảm dần với bài toán có nhiều ràng buộc**: Số lượng nhánh cần mở rộng có xu hướng tăng mạnh khi có nhiều ràng buộc, gây giảm hiệu quả.

Hạn chế của phương pháp Land-Doig

- **Thời gian tính toán lớn**: Việc phân nhánh tạo ra nhiều bài toán con, dẫn đến tăng chi phí tính toán.
- **Bộ nhớ tiêu tốn lớn**: Quá trình phân nhánh và lưu trữ các nhánh mở rộng tiêu tốn nhiều bộ nhớ, đặc biệt là với những bài toán có kích thước lớn.
- **Phụ thuộc vào chiến lược phân nhánh**: Hiệu quả của thuật toán phụ thuộc vào cách chọn nhánh để phân chia (cách chọn biến, cận).
- **Khó khăn trong đánh giá ràng buộc cận**: Việc xác định cận nào là tốt nhất còn hạn chế, dẫn đến khó xác định nhánh nào cần loại bỏ sớm.
- **Hiệu quả giảm dần với bài toán có nhiều ràng buộc**: Số lượng nhánh cần mở rộng có xu hướng tăng mạnh khi có nhiều ràng buộc, gây giảm hiệu quả.

Hạn chế của phương pháp Land-Doig

- **Thời gian tính toán lớn**: Việc phân nhánh tạo ra nhiều bài toán con, dẫn đến tăng chi phí tính toán.
- **Bộ nhớ tiêu tốn lớn**: Quá trình phân nhánh và lưu trữ các nhánh mở rộng tiêu tốn nhiều bộ nhớ, đặc biệt là với những bài toán có kích thước lớn.
- **Phụ thuộc vào chiến lược phân nhánh**: Hiệu quả của thuật toán phụ thuộc vào cách chọn nhánh để phân chia (cách chọn biến, cận).
- **Khó khăn trong đánh giá ràng buộc cận**: Việc xác định cận nào là tốt nhất còn hạn chế, dẫn đến khó xác định nhánh nào cần loại bỏ sớm.
- **Hiệu quả giảm dần với bài toán có nhiều ràng buộc**: Số lượng nhánh cần mở rộng có xu hướng tăng mạnh khi có nhiều ràng buộc, gây giảm hiệu quả.

Hướng phát triển trong tương lai

Học Máy (Machine learning)

Tối ưu hóa Thuật toán Land-Doig bằng phương pháp Học Máy

Trong quá trình giải quyết bài toán tối ưu tuyến tính nguyên, đặc biệt với các bài toán có **kích thước lớn**, việc quyết định lựa chọn phân nhánh nào cho một nút nhất định vẫn là một vấn đề nan giải và chưa có giải pháp hiệu quả.

Phương pháp Học máy (Machine learning) giúp mang đến một hướng tiếp cận tiềm năng thông qua việc cải thiện quy trình chọn nhánh bằng cách sử dụng dữ liệu để xác định các nhánh triển vọng, qua đó tăng tốc quá trình giải các bài toán tối ưu tuyến tính nguyên.[1]

[1] Arnaud Deza and Elias B. Khalil. **"Machine Learning for Cutting Planes in Integer Programming: A Survey".** In: *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence.* IJCAI-2023. International Joint Conferences on Artificial Intelligence Organization, Aug. 2023.

Tối ưu hóa Thuật toán Land-Doig bằng phương pháp Học Máy

Trong quá trình giải quyết bài toán tối ưu tuyến tính nguyên, đặc biệt với các bài toán có **kích thước lớn**, việc quyết định lựa chọn phân nhánh nào cho một nút nhất định vẫn là một vấn đề nan giải và chưa có giải pháp hiệu quả.

Phương pháp Học máy (Machine learning) giúp mang đến một hướng tiếp cận tiềm năng thông qua việc cải thiện quy trình chọn nhánh bằng cách sử dụng dữ liệu để xác định các nhánh triển vọng, qua đó tăng tốc quá trình giải các bài toán tối ưu tuyến tính nguyên.[1]

[1] Arnaud Deza and Elias B. Khalil. **"Machine Learning for Cutting Planes in Integer Programming: A Survey".** In: *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*. IJCAI-2023. International Joint Conferences on Artificial Intelligence Organization, Aug. 2023.

Tối ưu phân thức tuyến tính cho nghiệm nguyên

Tối ưu phân thức tuyến tính nguyên hoàn toàn

$$(H) \quad Q(x) = \frac{P(x)}{D(x)} \quad \longrightarrow Max$$
$$\begin{cases} Ax \leq b, \\ x \geq 0, \text{nguyên} \end{cases} \tag{15}$$

- Bài toán (H) gọi là bài toán **Tối ưu phân tuyến tính nguyên hoàn toàn.**
- Trong đó A là ma trận $m \times n$, $b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$, với $x \in \mathbb{Z}_+^n$. Tập

  $S_h := \{x \in \mathbb{Z}_+^n : Ax \leq b\}$ là tập nghiệm của bài toán Tối ưu phân tuyến tính nguyên hoàn toàn.
- $P(x) = p^T x + p_0$, với $p^T = (p_1\ p_2\ \dots\ p_n)$ và $D(x) = d^T x + d_0$, với $d^T = (d_1\ d_2\ \dots\ d_n)$ $(D(x) > 0, \forall x \in S_h)$.

Tối ưu phân thức tuyến tính nguyên bộ phận

$$(B) \quad Q(x,y) = \frac{P(x,y)}{D(x,y)} \quad \longrightarrow Max$$

$$\begin{cases} Ax + Gy \le b, \\ x \ge 0, \text{nguyên} \\ y \ge 0. \end{cases} \tag{16}$$

- Bài toán (B) gọi là bài toán **Tối ưu phân tuyến tính nguyên bộ phận.**

- Trong đó A là ma trận $m \times n$, G là ma trận $m \times t$, $b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$, với $x \in \mathbb{Z}^n_+$ và $y \in \mathbb{R}^t_+$. Tập $S_b := \{(x, y) \in \mathbb{Z}^n_+ \times \mathbb{R}^t_+ : Ax + Gy \leq b\}$ là tập nghiệm của bài toán Tối ưu phân tuyến tính nguyên bộ phận.

- $P(x, y) = p_1^T x + p_2^T y + p_0$, với $p_1^T = (p_1\ p_2\ \dots\ p_n)$ và $p_2^T = (p_1\ p_2\ \dots\ p_t)$.

- $D(x, y) = d_1^T x + d_2^T y + d_0$, với $d_1^T = (d_1\ d_2\ \dots\ d_n)$ và $d_2^T = (d_1\ d_2\ \dots\ d_t)$ $(D(x, y) > 0, \forall x \in S_b)$.

Có một lý thuyết riêng cho việc xử lý các bài toán Tối ưu phân thức tuyến tính và tìm nghiệm nguyên.[2]

[2] E.B. Bajalinov. **Linear-Fractional Programming Theory, Methods, Applications and Software.** Applied Optimization. Springer US, 2013.

Tài liệu

[1] E.B. Bajalinov. **Linear-Fractional Programming Theory, Methods, Applications and Software.** Applied Optimization. Springer US, 2013.

[2] Phan Hoàng Chơn. In: *Giáo trình Đại số tuyến tính* Đại học Sài Gòn (2022).

[3] M. Conforti, G. Cornuejols, and G. Zambelli. **Integer Programming.** Graduate Texts in Mathematics. Springer International Publishing, 2014. ISBN: 9783319110097. URL: https://books.google.com.vn/books?id=QdAMswEACAAJ.

[4] G.B. Dantzig and M.N. Thapa. **Linear Programming 2: 2: Theory and Extensions.** Linear Programming. Springer, 1997. ISBN: 9780387986135. URL: https://books.google.com.vn/books?id=qUvXMTOOPZwC.

[5] George B. Dantzig and Mukund Narain Thapa. **Linear Programming 1: Introduction.** Springer Series in Operations Research and Financial Engineering. New York: Springer, 1997. ISBN: 0387948333 9780387948331 0387986138 9780387986135.

[6] Arnaud Deza and Elias B. Khalil. **"Machine Learning for Cutting Planes in Integer Programming: A Survey".** In: *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*. IJCAI-2023. International Joint Conferences on Artificial Intelligence Organization, Aug. 2023.

[7] Nguyễn Hữu Điển. In: *Giáo trình Tối ưu tuyến tính và Ứng dụng* NXB ĐHQG Hà Nội ().

[8] H.A. Eiselt and C.L. Sandblom. **Linear Programming and its Applications.** Springer Berlin Heidelberg, 2007. ISBN: 9783540736714. URL: `https://books.google.com.ec/books?id=pCMCdycl2kMC`.

[9] Ralph Gomory. **"Outline of an Algorithm for Integer Solutions to Linear Programs and An Algorithm for the Mixed Integer Problem".** In: Jan. 2010, pp. 77–103. ISBN: 978-3-540-68274-5. DOI: 10.1007/978-3-540-68279-0_4.

[10] M. Jünger et al. **50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art.** Springer Berlin Heidelberg, 2009. ISBN: 9783540682790. URL: https://books.google.com.vn/books?id=bUJc_weiYfkC.

[11] Ailsa H. Land and Alison G. Doig. **"An Automatic Method of Solving Discrete Programming Problems".** In: *Econometrica* 28 (1960), p. 497. URL: https://api.semanticscholar.org/CorpusID:35442133.

[12] D.C. Lay. **Linear Algebra and Its Applications.** Pearson Education, 2003. ISBN: 9788177583335. URL: https://books.google.com.vn/books?id=v8Zls26Y0wkC.

[13] Đ.N. Nguyễn. **Tối ưu hoá: Quy hoạch tuyến tính và rời rạc.** Giáo dục, 1996. URL: https://books.google.com.vn/books?id=MNBrAQAACAAJ.

[14] S.S. Rao. **Engineering Optimization: Theory and Practice.** Wiley, 2009. ISBN: 9780470183526. URL: https://books.google.com.vn/books?id=YNt34dvnQLEC.

[15] Tạ Quang Sơn. In: *Bài giảng Quy hoạch tuyến tính* Đại học Sài Gòn (2023).

[16] Bùi Thế Tâm. In: *Quy hoạch rời rạc* NXB Hà Nội (2008).

Cảm ơn quý thầy cô và các anh chị đã quan tâm theo dõi!