BỘ GIÁO DỤC VÀ ĐÀO TẠO

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA

BÁO CÁO

BỘ MÔN ĐẠI SỐ TUYẾN TÍNH

Đề tài 13:

QUY HOẠCH TUYẾN TÍNH

MSSV	Họ và tên
2311070	Nguyễn Xuân Huy Hoàng
2310795	Phạm Anh Đức
2311694	Phạm Đăng Khôi
2311635	Phạm Đặng Đăng Khoa
2310761	Phạm Đình Được
2312186	Phạm Đình Phương Nam
2310686	Phạm Đỗ Thành Đạt

Giảng viên hướng dẫn: Ts.Nguyễn Thị Hoài Thương

Nhóm: ĐSTT-L07-13

Ngày 10 tháng 3 năm 2024

Mục lục

I. KHÁI NIỆM

1 Quy hoạch tuyến tính

Quy hoạch tuyến tính - linear programming (LP) là một thuật toán nhằm tìm ra phương án tối ưu (hoặc kế hoạch tối ưu) từ vô số các phương án quyết định. Phương án tối ưu là phương án thoả mãn được các mục tiêu đề ra của một hãng, phụ thuộc vào các hạn chế và các ràng buộc.

Thuật toán LP đề cập đến vấn đề phân bổ nguồn lực khan hiếm giữa các hoạt động cạnh tranh trong một phương án tối ưu để đạt được hiệu quả cao nhất, lãi gộp cao nhất hay doanh thu hoặc chi phí thấp nhất. Mô hình quy hoạch tuyến tính gồm *2 thành phần cơ bản*:

☐ **Hàm mục tiêu:** Biểu thị giá trị mà ta muốn tối ưu hóa (tối đa hóa hoặc tối thiểu hóa) để đạt được mục tiêu cụ thể.

☐ **Điều kiện ràng buộc:** Các ràng buộc dưới dạng các hạn chế về sự sẵn có của nguồn lực hay thoả mãn các yêu cầu tối thiểu.

↪ Do đó có thể hiểu đơn giản quy hoạch tuyến tính là lĩnh vực toán học nghiên cứu các bài toán tối ưu mà hàm mục tiêu (vấn đề được quan tâm) và các ràng buộc (điều kiện của bài toán) đều là hàm và các phương trình hoặc bất phương trình tuyến tính.

2 Một số khái niệm khác liên quan

☐ **Biến quyết định:** Các biến đại diện cho các hoạt động hoặc quyết định trong bài toán.

☐ **Biến slack và surplus:** Các biến được thêm vào một số ràng buộc để biến đổi chúng thành dạng chuẩn tắc.

☐ **Phương án:** Mỗi vector $X = (x_1, x_2, ..., x_n)$ trong R^n thỏa mãn tất cả các ràng buộc của một bài toán quy hoạch tuyến tính n biến được gọi là một phương án của bài toán đó.

☐ **Tập phương án (hay miền ràng buộc):** Tập hợp tất cả các phương án của một bài toán quy hoạch tuyến tính gọi là tập phương án hay miền ràng buộc của bài toán đó.

☐ **Phương án tối ưu:** Một phương án x^* của bài toán quy hoạch tuyến tính được gọi là nghiệm hay là phương án tối ưu nếu nó làm cho hàm mục tiêu đạt *min* (hoặc *max*) đúng như yêu cầu bài toán đó.

☐ **Phương án cực biên**

- **Đối với bài toán quy hoạch tuyến tính dạng tổng quát:** Xét một bài toán quy hoạch tuyến tính dạng tổng quát có n biến $x_1, x_2, ..., x_n$. Một phương án $x^* = (x_1^*, x_2^*, ..., x_n^*)$ của bài toán quy hoạch tuyến tính dạng tổng quát đang xét được gọi là **phương án cực biên** nếu nó thỏa mãn dấu "$=$" (còn gọi là thỏa mãn chặt) với ít nhất n ràng buộc trong đó có đúng n ràng buộc độc lập tuyến tính (tức là ma trận hệ số của n ràng buộc đó có hạng bằng n) trong hệ ràng buộc của bài toán quy hoạch tuyến tính dạng tổng quát.
- **Đối với bài toán quy hoạch tuyến tính dạng chính tắc hoặc chuẩn chính tắc:** Xét một bài toán quy hoạch tuyến tính dạng chính tắc có n biến $x_1, x_2, ..., x_n$. Một phương án $x^* = (x_1^*, x_2^*, ..., x_n^*)$ của bài toán quy hoạch tuyến tính dạng chính tắc đang xét là **phương án cực biên** nếu hệ các cột ma trận hệ số ứng với các $x_j^* > 0$ lập thành hệ độc lập tuyến tính. Ta gọi các ẩn dương là ẩn cơ sở, các ẩn triệt tiêu là

ẩn phi cơ sở. Các hệ số trong hàm mục tiêu ứng với các ẩn cơ sở (tương ứng, phi cơ sở) cũng được gọi là hệ số cơ sở (tương ứng, phi cơ sở).

□ **Khái niệm điểm cực trị:**

- Một điểm cực trị của một hàm mục tiêu trong bài toán quy hoạch là một điểm trong miền khả thi mà tại đó giá trị của hàm mục tiêu đạt cực đại hoặc cực tiểu.
- Các điểm cực trị thường nằm tại các giao điểm của các đường ràng buộc trong miền khả thi.

□ **Định lý về các điểm cực trị:**

- Định lý về các điểm cực trị khẳng định rằng một bài toán quy hoạch có nghiệm tối ưu (hoặc cực tiểu) khi và chỉ khi nó có một nghiệm cơ bản khả thi tối ưu (hoặc cực tiểu).
- Định lý này là nền tảng cho phương pháp Simplex, vì nó cho phép chúng ta tìm kiếm nghiệm tối ưu bằng cách khảo sát từ nghiệm khả thi này sang nghiệm khả thi khác.

□ **Bảng Simplex:** Một bảng được sử dụng để theo dõi tiến trình của quy trình Simplex và ghi lại thông tin về các biến, ràng buộc và giá trị hàm mục tiêu tại mỗi bước.

□ **Quy trình Simplex:**

- Quy trình Simplex bắt đầu với một nghiệm cơ bản khả thi bất kỳ.
- Tại mỗi bước, quy trình xác định một biến **vào** và một biến **ra** để di chuyển đến nghiệm cơ bản khả thi tiếp theo.
- Biến vào được chọn là biến có hệ số âm lớn nhất trong hàm mục tiêu tại nghiệm khả thi hiện tại.
- Biến ra được chọn là biến có tỷ lệ rẽ nhỏ nhất khi tính toán cho tất cả các ràng buộc.
- Quy trình tiếp tục cho đến khi tìm thấy nghiệm khả thi tối ưu hoặc xác định rằng bài toán quy hoạch không có nghiệm tối ưu.

3 Một số nhận xét quan trọng

□ Một bài toán quy hoạch tuyến tính có thể **không có phương án**, lúc đó nó **vô nghiệm**. Điều kiện cần và đủ để một bài toán quy hoạch tuyến tính có **phương án tối ưu** là nó có **phương án** và **hàm mục tiêu bị chặn**.

□ Nếu bài toán quy hoạch tuyến tính có **phương án tối ưu** và có **phương án cực biên** thì chắc chắn có **phương án cực biên tối ưu**. Khi đó, ta chỉ cần tìm nghiệm trong các **phương án cực biên**.

□ Bài toán quy hoạch tuyến tính dạng tổng quát luôn có **một số hữu hạn phương án cực biên**.

□ Đối với bài toán quy hoạch tuyến tính dạng chính tắc chuẩn, ta luôn dễ dàng tìm được một **phương án cực biên** bằng cách cho các ẩn ứng với các hệ số không thuộc ma trận con sơ cấp là ẩn phi cơ sở, tức là gán cho chúng giá trị bằng 0. Các ẩn ứng với các hệ số thuộc ma trận con sơ cấp là các ẩn cơ sở.

4 Bài toán thực tế:

4.1 Bài toán sản xuất

Bài toán: Từ m loại nguyên liệu hiện có người ta muốn sản xuất n loại sản phẩm nhằm thu được lợi nhuận cao nhất.

Ta tiến hành mô hình hoá bài toán:

Giả sử:

- a_{ij} là lượng nguyên liệu loại i dùng để sản xuất 1 sản phẩm loại j với $(i = 1, 2, ..., m)$ và $(j = 1, 2, ..., n)$
- b_i là số lượng nguyên liệu loại i hiện có.
- c_j là lợi nhuận thu được từ việc sản xuất một đơn vị sản phẩm loại j.

Vấn đề đặt ra là phải sản xuất mỗi loại sản phẩm là bao nhiêu sao cho tổng lợi nhuận thu được từ việc bán các sản phẩm lớn nhất trong điều kiện nguyên liệu hiện có.

Gọi $x_j \geq 0$ là số lượng sản phẩm thứ j sẽ sản xuất $(j = 1, 2, ..., n)$

Tổng lợi nhuận thu được từ việc sản xuất sản phẩm là:

$$f(x) = \sum_{j=1}^{n} c_j x_j = c_1x_1 + c_2x_2 + ... + c_nx_n$$

Vì yêu cầu lợi nhuận thu được là cao nhất nên ta cần có:

$$\max f(x) = \sum_{j=1}^{n} c_j x_j = c_1x_1 + c_2x_2 + ... + c_nx_n$$

Đây chính là hàm mục tiêu của bài toán!

- Lượng nguyên liệu thứ $i = 1 \longrightarrow m$ dùng để sản xuất sản phẩm thứ 1 là $a_{i1}x_1$
- Lượng nguyên liệu thứ $i = 1 \longrightarrow m$ dùng để sản xuất sản phẩm thứ 2 là $a_{i2}x_2$
- ...
- Lượng nguyên liệu thứ $i = 1 \longrightarrow m$ dùng để sản xuất sản phẩm thứ n là $a_{in}x_n$

Vậy lượng nguyên liệu thứ i dùng để sản xuất là các sản phẩm là:

$$a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n$$

Vì lượng nguyên liệu thứ $i = 1 \longrightarrow m$ dùng để sản xuất các loại sản phẩm không thể vượt quá lượng được cung cấp là b_i nên :

$$a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n \leq b_i (i = 1, 2, ..., m)$$

Vậy theo yêu cầu của bài toán ta có mô hình sau đây:

$$\max f(x) = \sum_{j=1}^{n} c_j x_j = c_1x_1 + c_2x_2 + ... + c_nx_n$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n \leq b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n \leq b_2 \\ ... \\ a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n \leq b_m \\ x_j \geq 0, (j = 1, 2, ..., n) \end{cases}$$

4.2 Bài toán vận tải:

Bài toán: Người ta cần vận chuyển hàng hoá từ m kho đến n cửa hàng bán lẻ. Lượng hàng hoá ở kho i là s_i $(i = 1, 2, ..., m)$ và nhu cầu hàng hoá của cửa hàng j là d_j $(j = 1, 2, ..., n)$. Cước vận chuyển một đơn vị hàng hoá từ kho i đến của hàng j là $c_{ij} \geq 0$ đồng.

Mô hình hoá bài toán:

- Nhằm tối ưu hóa việc di chuyển hàng hóa từ các nguồn cung đến các điểm tiêu thụ một cách hiệu quả và có thể làm giảm chi phí vận chuyển nên hàm mục tiêu của bài toán là:

$$\min f(x) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

- Để một phương án thực sự là chấp nhận được cho bài toán vận tải, các giá trị $x_{i,j}$ phải thỏa mãn các điều kiện ràng buộc
  - Theo điều kiện về khả năng cung cấp:

$$\sum_{j=1}^{n} x_{ij} = s_i \text{ với } (i = 1, ..., m)$$

  - Theo điều kiện về nhu cầu tiêu thụ:

$$\sum_{i=1}^{m} x_{ij} = d_j \text{ với } (j = 1, ..., n)$$

- Điều kiện biên $x_{ij} \geq 0$ với $(i = 1, ..., m)$ và $(j = 1, ..., n)$

Do đó ta có mô hình sau:

$$\min f(x) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$\begin{cases} \displaystyle\sum_{j=1}^{n} x_{ij} = s_i \text{ với } (i = 1, ..., m) \\ \displaystyle\sum_{i=1}^{m} x_{ij} = d_j \text{ với } (j = 1, ..., n) \\ x_{ij} \geq 0, (i = 1, ..., m); (j = 1, ..., n) \end{cases}$$

4.3 Bài toán khẩu phần ăn:

Một khẩu phần thức ăn có khối lượng P, có thể được làm từ n loại thức ăn. Giá mua một đơn vị thức ăn loại j là c_j. Để đảm bảo cơ thể phát triển bình thường thì khẩu phần cần m loại chất dinh dưỡng. Chất dinh dưỡng thứ i cần tối thiểu cho khẩu phần là b_i và có trong một đơn vị thức ăn loại j là a_{ij}.

Vấn đề đặt ra là nên cấu tạo một khẩu phần thức ăn như thế nào để ăn đủ no, đủ chất dinh dưỡng mà có giá thành rẻ nhất.

- Gọi $x_j \geq 0 (j = 1, 2, ..., n)$ là số lượng thức ăn thứ j cần mua.
- Tổng chi phí cho việc mua thức ăn là:

$$f(x) = \sum_{j=1}^{n} c_j x_j = c_1 x_1 + c_2 x_2 + ... + c_n x_n$$

- Vì chi phí bỏ ra để mua thức ăn phải là thấp nhất nên yêu cầu cần được thoả mãn là:

$$\min f(x) = \sum_{j=1}^{n} c_j x_j = c_1x_1 + c_2x_2 + ... + c_nx_n$$

- Lượng dinh dưỡng thứ $i = 1 \longrightarrow m$ thu được từ thức ăn thứ 1 là $a_{i1}x_1$
- Lượng dinh dưỡng thứ $i = 1 \longrightarrow m$ thu được từ thức ăn thứ 2 là $a_{i2}x_2$
- ...
- Lượng dinh dưỡng thứ $i = 1 \longrightarrow m$ thu được từ thức ăn thứ n là $a_{in}x_n$

Vậy lượng dinh dưỡng thứ i thu được từ các loại thức ăn là:

$$a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n$$

Vì lượng dinh dưỡng thứ $i = 1 \longrightarrow m$ thu được phải thoả mãn yêu cầu b_i về chất dinh dưỡng nên ta có ràng buộc sau:

$$a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n \geq b_i (i = 1, 2, ..., m)$$

Vậy theo yêu cầu của bài toán ta có mô hình sau đây:

$$\min f(x) = \sum_{j=1}^{n} c_j x_j = c_1x_1 + c_2x_2 + ... + c_nx_n$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n \geq b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n \geq b_2 \\ ... \\ a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n \geq b_m \\ x_j \geq 0, (j = 1, 2, ..., n) \end{cases}$$

II. CƠ SỞ LÝ THUYẾT

1 Bài toán quy hoạch tuyến tính

Bài toán có dạng $f(x_1, x_2, x_3, ..., x_k) \longrightarrow min, max$ với hệ các ràng buộc cho trước dưới dạng $\mathbf{Ax}(\leq, \geq, =)\mathbf{B}$ với $\mathbf{A}, \mathbf{B}$ là các ma trận mô tả hệ số của các biến trong ràng buộc.

Do xuất phát từ thức tế, các bài toán Quy hoạch tuyến tính thường xét các biến ≥ 0.

Để chuyển ràng buộc " $\leq$ " $\longrightarrow$ " $=$ " ta thực hiện thêm biến dạng:

$\hookrightarrow$ $a_1x_1 + a_2x_2 + ... + x_ka_k + \boxed{a_{k+1}} = c$ với $a_{k+1} \geq 0$ là biến thêm vào.

$\hookrightarrow$ *Tương tự*, nếu là " $\geq$ " $\longrightarrow$ " $=$ " thì $a_1x_1 + a_2x_2 + ... + x_ka_k - \boxed{a_{k+1}} = c$ với $a_{k+1} \geq 0$

Hay:

Tìm vector $X = (x_1, x_2, ..., x_n)$ làm cực tiểu (hoặc cực đại) hàm số $f(X)$, với các điều kiện $g_i(X) \leq 0, \forall i = 1, ..., m$, $x_j \geq 0, \forall j = 1, ..., k$; $k \leq n$

□ $minf(X)$ hoặc $maxf(X)$ (1.1)

□ Với điều kiện $\begin{cases} g_i(X) \leq k, i = \overline{1,k} & (1.2) \\ x_j(X) \geq 0, j = \overline{1,m} & (1.3) \end{cases}$

Hàm $f(X)$ gọi là **hàm mục tiêu**, các điều kiện (1.1), (1.2),(1.3) gọi là các **điều kiện buộc** của bài toán.

Mỗi vector $X = (x_j) \in R^n$ thỏa mãn hệ điều kiện buộc gọi là một **phương án**. Ta kí hiệu tập phương án là M.

Một phương án làm cực tiểu (hoặc cực đại) hàm mục tiêu gọi là **phương án tối ưu** (hoặc gọi là nghiệm của bài toán).

Khi $f(X)$ và $g_i(X)(i = 1, ...n)$ là các hàm tuyến tính, $M \subset R^n$ thì bài toán đã cho được gọi là **bài toán quy hoạch tuyến tính**.

2 Các dạng bài toán quy hoạch tuyến tính

2.1 Bài toán quy hoạch tuyến tính dạng tổng quát

Tổng quát

Đó là bài toán quy hoạch tuyến tính mà hệ ràng buộc chính có thể gồm các bất phương trình hay phương trình, các ẩn (biến) có thể chịu ràng buộc dấu không âm ≥ 0 , hoặc không dương ≤ 0 hoặc dấu bất kỳ. Hai bài toán xét trong mục tiêu trên đều là các bài toán quy hoạch tuyến tính dạng tổng quát.

□ Ví dụ:

$$f = x_1 + 2x_2 - 5x_3 + 3x_4 - 2x_5 + 5x_6 \longrightarrow min$$

$$\begin{cases} 2x_1 - x_2 + 4x_3 - 2x_4 - 3x_5 + 4x_6 = 4 \\ x_1 + 3x_2 - 2x_3 + 6x_4 - 4x_5 + 2x_6 \leq 6 \\ 3x_1 - x_2 - 4x_3 - 2x_4 - 3x_5 + 8x_6 \geq 1 \\ x_1 \geq 0, x_2 \geq 0, x_4 \leq 0, x_6 \geq 0 \end{cases}$$

2.2 Bài toán quy hoạch tuyến tính dạng chính tắc

Chính tắc

Đây là bài toán quy hoạch tuyến tính mà các hệ ràng buộc chính đều là các phương trình − nói cách khác, hệ ràng buộc chính là một hệ phương trình tuyến tính. Hơn nữa, mọi biến đều không âm − tức là mọi ràng buộc dấu có dạng $x_j \geq 0, \forall j$.

Cụ thể:

(1) $f(x) = c_1x_1 + c_2x_2 + ... + c_nx_n \longrightarrow max(min)$

(2) $a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n = b_i, i = 1, 2, ..., m$

(3) $x_j \geq 0, j = 1, 2, ..., n$

☐ Ví dụ:

$$f = x_1 + 2x_2 - 5x_3 + 3x_4 \longrightarrow max$$
$$\begin{cases} 2x_1 - x_2 + 4x_3 - 2x_4 = 3 \\ x_1 + 3x_2 - 2x_3 + 6x_4 = 8 \\ 3x_1 - x_2 - 4x_3 - 2x_4 = -4 \\ x_j \geq 0, j = 1, 2, 3, 4 \end{cases}$$

Nhận xét: Bài toán Quy hoạch tuyến tính dạng chính tắc là bài toán QHTT dạng tổng quát trong đó:

- Các ràng buộc chính đều là phương trình
- Các ẩn đều không âm.
- Trong trường hợp bài toán tổng quát có các ẩn chưa ràng buộc về dấu:
  - Ẩn $x_j \leq 0$ thì thay ẩn x_j bằng $-x'_j$.
  - Ẩn x_j có giá trị tuỳ ý thì thay $x_j = x'_j - x''_j$ với $x'_j \geq 0$ và $x''_j \geq 0$.

2.3 Bài toán quy hoạch tuyến tính dạng chính tắc chuẩn

Dạng chuẩn

Đó là bài toán quy hoạch tuyến tính dạng chính tắc đặc biệt trong đó hệ phương trình ràng buộc chính gồm m phương trình, n ẩn số với $m \leq n$, mỗi phương trình đều có vế phải không âm, đồng thời ma trận hệ số một ma trận con đơn vị hoặc chứa một ma trận con sơ cấp đơn giản cấp m (tức là ma trận nhận được từ ma trận đơn vị cấp m bằng cách đổi chỗ các dòng).

☐ Trong đó:

- Các hệ số tự do đều không âm.
- Trong ma trận hệ số tự do có đủ m vector cột đơn vị: $e_1, e_2, ..., e_m$

$$e_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, e_2 = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, ..., e_m = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$$

☐ Khi đó

- Các ẩn ứng với các vector cột đơn vị được gọi là các **ẩn cơ bản**. Cụ thể ẩn ứng với vector cột đơn vị e_k là ẩn cơ bản thứ k.
- Một phương án mà các ẩn cơ bản đều bằng 0 được gọi là **phương án cơ bản**.
- Một phương án cơ bản có đủ m thành phần dương được gọi là **không suy biến**. Ngược lại một phương án cơ bản có ít hơn m thành phần dương được gọi là **suy biến**.

☐ Ví dụ:

$$f = 2x_1 - 4x_2 - x_3 + 6x_4 \longrightarrow min$$

$$\begin{cases} x_1 + x_4 + x_5 = 12 \\ 12x_1 + x_3 + x_6 = 3 \\ x_1 + x_2 - x_3 - x_4 = 6 \\ x_j \geq 0, j = 1, 2, 3, 4, 5, 6 \end{cases}$$

Ta thấy bài toán trên có dạng chính tắc, hơn nữa các hệ số tự do đều không âm.

Ma trận hệ số ràng buộc A là: $\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 12 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & -1 & -1 & 0 & 0 \end{bmatrix}$

Có chứa đầy đủ 3 vector cột đơn vị e_1 (cột 5), e_2 (cột 6), e_3 (cột 2).

Do đó bài toán có dạng chuẩn, trong đó:

- Ẩn cơ bản thứ nhất là x_5
- Ẩn cơ bản thứ hai là x_6
- Ẩn cơ bản thứ ba là x_2

☐ **Nhận xét:** Trong bài toán trên, khi cho ẩn cơ bản thứ k bằng hệ số tự do thứ k, còn các ẩn không cơ bản bằng 0, nghĩa là cho $x_5 = 12, x_6 = 3, x_2 = 6, x_1 = 0, x_3 = 0, x_4 = 0$ ta được một phương án cơ bản của bài toán $x = (0, 6, 0, 0, 12, 3)$.

☐ **Chú ý:** Tổng quát, trong bài toán QHTT dạng chuẩn bất kì, khi cho ẩn cơ bản thứ k bằng hệ số tự do thứ k ($k = 1, 2, ..., m$), còn các ẩn không cơ bản bằng 0, ta được một phương án cơ bản của bài toán. Ta gọi đây là **phương án cơ bản** ban đầu của bài toán.

3 Thuật toán tìm phương pháp tối ưu tuyến tính

3.1 Phương pháp hình học

3.1.1 Thuật toán:

☐ Đối với bài toán QHTT 2 biến (với số ràng buộc tùy ý), ta có thể:

- **Bước 1:** Chuyển các ràng buộc về dạng đẳng thức, vẽ đường thẳng có phương trình tương ứng, lấy phần mặt phẳng ứng với dấu $\geq, \leq$ thích hợp tạo thành một đa giác lồi.
- **Bước 2:** Xác định các đỉnh của đa giác chính là các điểm cực biên.
- **Bước 3:** Để tìm lời giải tối ưu cho bài toán, ta thay tọa độ các điểm cực biên vào hàm mục tiêu và chọn ra giá trị lớn nhất/nhỏ nhất.

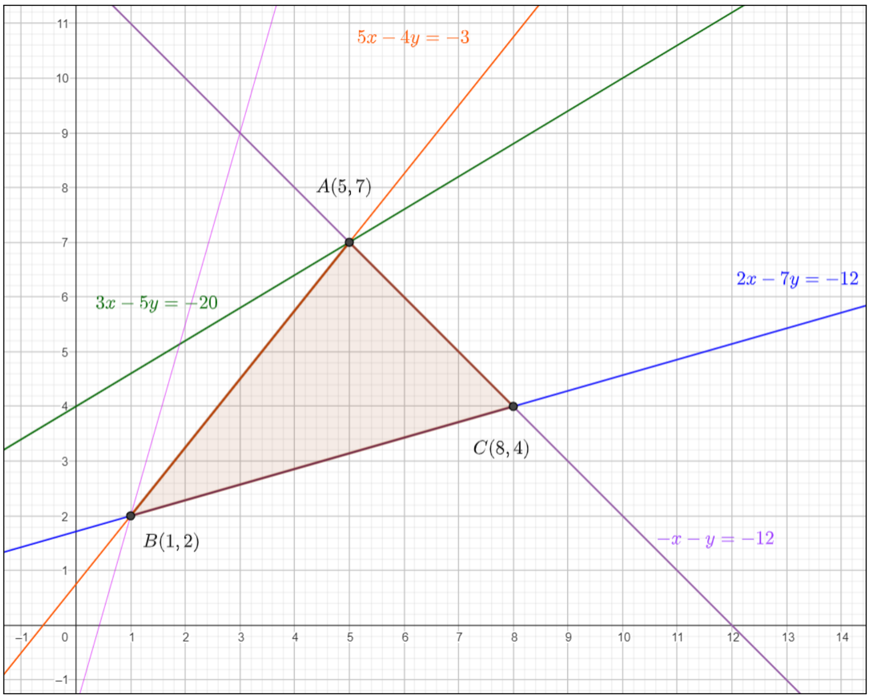
3.1.2 Ví dụ minh hoạ:

Ví dụ 1: Giải bài toán quy hoạch tuyến tính sau:

$$f = 7x - 2y \longrightarrow min$$

$$\begin{cases} 5x - 4y \geq -3 \\ 2x - 7y \leq -12 \\ -x - y \geq -12 \\ 3x - 5y \geq -20 \end{cases}$$

- **Bước 1:** Vẽ hình



- **Bước 2:** Xác định các điểm cực biên là $A(5,7); B(1,2); C(8,4)$.
- **Bước 3:** Thay toạ độ vào hàm mục tiêu: $f_A = 21$; $f_B = 4$; $f_C = 48$. Vậy phương án tối ưu duy nhất của bài toán là $(x,y) = (1,2); f_{min} = 3$.

3.1.3 Nhận xét:

☐ Ưu điểm

- Thuật toán đơn giản, dễ hiểu và dễ triển khai.
- Trong trường hợp bài toán có số biến và ràng buộc nhỏ, phương pháp hình học có thể đạt hiệu suất tốt và tìm ra giải pháp tối ưu trong thời gian ngắn do đó cũng dễ kiểm tra và hình dung.

☐ Nhược điểm

- Không hiệu quả với các bài toán lớn có số biến và ràng buộc lớn.
- Với số biến càng lớn, đòi hỏi không gian nhiều chiều khiến cho việc hình dung và vẽ hình trở nên khó khăn.

3.2 Thuật toán đơn hình - Simplex method

3.2.1 Thuật toán:

Xét bài toán quy hoạch tuyến tính có dạng như sau:

(1) $f(x) = c_1x_1 + c_2x_2 + ... + c_nx_n \longrightarrow max(min)$

(2) $a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n = b_i, i = 1, 2, ..., m$

(3) $x_j \geq 0, j = 1, 2, ..., n$

Để tìm phương án tối ưu *Danzig* đã đề xuất thuật toán như sau: xuất phát từ một phương án cực biên x^0. Kiểm tra xem x^0 có phải là lời giải tối ưu hay chưa. Nếu x_0 chưa phải là phương án tối ưu thì tìm cách cải tiến nó để được một phương án cực biên khác là x^1 tốt hơn x^0 theo nghĩa $Z(x^1) < Z(x^0)$ (max thì ngược lại). Quá trình được lặp lại nhiều lần tới khi tìm thấy cực biên tối ưu.

Thuật toán áp dụng cho bài toán mà các biến đều ≥ 0 và các ràng buộc ở dạng đẳng thức.

Trước hết, ta phải chọn ra phương án cực biên cơ sở (các hệ số của nó chúng tạo thành ma trận đơn vị). Nếu không có sẵn phương án đó, ta dùng phương pháp big M để tạo biến ảo để có ma trận đơn vị.

Hệ số	Ẩn cơ bản	Phương án	x_1 c_1	x_2 c_2	x_3 c_3	λ_i
c_1	x_1	b_1	a_{11}	a_{12}	a_{13}	
c_2	x_2	b_2	a_{21}	a_{22}	a_{23}	
	f_{min}	$\sum b_ic_i$	Δ_1	Δ_2	Δ_3	Δ_j

- Trong bài toán QHTT tìm *min*, thuật toán dừng lại khi tất cả các $\Delta \leq 0$. Nếu còn $\Delta > 0$, ta chọn ra số lớn nhất, và trên cột tương ứng, ta chọn phần tử xoay a_{ij} theo tiêu chí: nó là số dương, xét $\lambda_v = \min\limits_{a_{iv}>0} \dfrac{b_i}{a_{iv}}$ từ đó tìm $\max\{\Delta_v.\lambda_v | \forall \Delta_v > 0\}$
- Trong bài toán QHTT tìm *max*, thuật toán dừng lại khi tất cả các $\Delta \geq 0$. Nếu còn $\Delta < 0$, ta chọn ra số âm nhỏ nhất, và trên cột tương ứng, ta chọn phần tử xoay a_{ij} theo tiêu chí: nó là số dương, xét $\lambda_v = \min\limits_{a_{iv}>0} \dfrac{b_i}{a_{iv}}$ từ đó tìm $\max\{|\Delta_v|.\lambda_v | \forall \Delta_v < 0\}$

Sau khi chọn được phần tử xoay, ta sẽ có biến cơ bản vào/ra, và bảng đơn hình mới.

Tóm lại, thuật toán gồm các bước:

☐ **Bước 1:** Xác định Hàm mục tiêu và ràng buộc.

☐ **Bước 2:** Chuyển đổi Bất đẳng thức thành Phương trình ($\leq, \geq \longrightarrow =$)

☐ **Bước 3:** Xác định ma trận hệ số $A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$ và các ẩn cơ bản

☐ **Bước 4:** Xây dựng Bảng đơn hình hay Bảng Simplex ban đầu.

☐ **Bước 5:** Xác định các cột mục tiêu.

☐ **Bước 6:** Tính các tỷ số rời cơ sở $\lambda_v = \min\limits_{a_{iv}>0} \dfrac{b_i}{a_{iv}}$

☐ **Bước 7:** Thực hiện then chốt nhằm loại bỏ biến rời khỏi cơ sở và đưa biến vào cơ sở.

$$\text{Dòng có ẩn đưa vào} = \text{dòng chuẩn} = \frac{\text{dòng chủ yếu}}{\text{hệ số chủ yếu}}$$

Dòng thứ i = Dòng thứ i (cũ) - a_{iv}.dòng chuẩn. (a_{iv}: số nằm trên giao của dòng i và cột chủ yếu).

☐ **Bước 8:** Lặp lại các bước 5,6,7 trong trường hợp còn giá trị âm trong hàng hàm mục tiêu (hàng cuối cùng) đối với bài toán tìm *max* và giá trị dương đối với bài toán tìm *min*.

☐ **Bước 9:** Ghi nhận kết quả với các giá trị.

3.2.2 Ví dụ minh hoạ:

Ví dụ 1: Tìm giá trị tối đa của hàm $f(x) = 40x_1 + 30x_2$ (1)

$$\begin{cases} x_1 + x_2 \leq 12 \quad (2) \\ 2x_1 + x_2 \leq 16 \quad (3) \\ x_1 \geq 0, x_2 \geq 0 \quad (4) \end{cases}$$

☐ **Bước 1:** Xác định hàm mục tiêu và ràng buộc

- Bước này chỉ sử dụng cho các bài toán thực tế cần thiết lập hệ phương trình, hệ bất phương trình dựa trên yêu cầu đề bài.
- Vì bài toán đã xác định hàm mục tiêu và các ràng buộc nên ta qua bước thứ 2.

☐ **Bước 2:** Chuyển đổi Bất đẳng thức thành Phương trình.

Có thể thấy có 2 bất phương trình (2) và (3), ta lần lượt cộng biến thêm vào (biến slack) x_3 và x_4 cho 2 bất phương trình để chuyển về dạng phương trình.

$$\begin{cases} x_1 + x_2 + x_3 = 12 \quad (2) \\ 2x_1 + x_2 + x_4 = 16 \quad (3) \end{cases}$$

☐ **Bước 3:** Xác định ma trận hệ số và các ẩn cơ bản

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 2 & 1 & 0 & 1 \end{bmatrix}$$

Có chứa đầy đủ 2 vector cột đơn vị e_1 (cột 3), e_2 (cột 4)

Do đó bài toán có dạng chuẩn, trong đó:

– Ẩn cơ bản thứ nhất là x_3

– Ẩn cơ bản thứ hai là x_4

☐ **Bước 4:** Xây dựng Bảng Simplex ban đầu

Hệ số	Ẩn cơ bản	Phương án	x_1 40	x_2 30	x_3 0	x_4 0	λ_i
0	x_3	12	1	1	1	0	
0	x_4	16	2	1	0	1	
	0	0	-40	-30	0	0	Δ_j

☐ **Bước 5:** Xác định cột mục tiêu.

Tìm giá trị âm lớn nhất trong hàng hàm mục tiêu (dòng dưới cùng), dễ thấy cột 4 có giá trị -40 là nhỏ nhất nên cột mục tiêu là cột 4

☐ **Bước 6:** Tính tỷ số rời cơ sở

Ta lấy các giá trị phương án lần lượt ở dòng 1 và 2 chia cho phần tử ở cột chủ yếu hay cột 4 ở dòng 1 và 2 để tìm giá trị λ_i

Hệ số	Ẩn cơ bản	Phương án	x_1 40	x_2 30	x_3 0	x_4 0	λ_i
0	x_3	12	1	1	1	0	12
0	x_4	16	2	1	0	1	8
		0	-40	-30	0	0	Δ_j

Vì $\lambda_2 = 8$ nhỏ nhất nên dòng chủ yếu là dòng 3.

Giao của cột chủ yếu và dòng chủ yếu là hệ số chủ yếu có giá trị là 2.

Hệ số	Ẩn cơ bản	Phương án	x_1 40	x_2 30	x_3 0	x_4 0	λ_i
0	x_3	12	1	1	1	0	12
0	x_4	16	2*	1	0	1	8
		0	-40	-30	0	0	Δ_j

☐ **Bước 7:** Thực hiện then chốt. Tính bảng 2 theo công thức sau:

$$\text{Dòng có ẩn đưa vào} = \text{dòng chuẩn} = \frac{\text{dòng chủ yếu}}{\text{hệ số chủ yếu}}$$

Dòng thứ i = Dòng thứ i (cũ) - a_{iv}.dòng chuẩn. (a_{iv}: số nằm trên giao của dòng i và cột chủ yếu).

Hệ số	Ẩn cơ bản	Phương án	x_1 40	x_2 30	x_3 0	x_4 0	λ_i
0	x_3	4	0	1/2	1	$-1/2$	8
40	x_1	8	1*	1/2	0	1/2	16
		320	0	-10	0	0	Δ_j

☐ **Bước 8:** Lặp lại các bước 4,5,6, ta được bảng sau:

Hệ số	Ẩn cơ bản	Phương án	x_1 40	x_2 30	x_3 0	x_4 0	λ_i
30	x_2	8	0	1	2	-1	
40	x_1	4	1	0	-1	1	
		400	0	0	20	10	Δ_j

Không còn số âm trong hàng cuối cùng phương pháp đã tìm được phương án tối ưu, quy trình dừng lại.

☐ **Bước 9:** Ghi nhận kết quả

Kết luận cho bài toán phụ: PATƯ: $X^2 = (x_1, x_2, x_3, x_4) = (4, 8, 0, 0)$

GTTƯ: $f(X2) = 400$

Kết luận cho bài toán gốc: PATƯ: $X^2 = (x_1, x_2) = (4, 8)$

GTTƯ: $f(X2) = 400$

Ví dụ 2: Một công ty xây dựng cần phân phối nguồn lực cho ba dự án khác nhau: Dự án X, Y, và Z. Mỗi dự án cần một số lượng nhất định của ba loại nguồn lực: lao động, vật liệu, và máy móc. Công ty có một lượng hạn chế nguồn lực mỗi ngày và muốn phân phối chúng để tối đa hóa lợi nhuận tổng cộng từ ba dự án. Dưới đây là thông tin chi tiết:

- Mỗi dự án X, Y, và Z mang lại lợi nhuận lần lượt là 500, 300, và 400 đô la mỗi ngày.
- Mỗi dự án cần nguồn lực như sau:
  - Dự án X: 2 lao động, 3 vật liệu, 2 máy móc
  - Dự án Y: 4 lao động, 1 vật liệu, 1 máy móc
  - Dự án Z: 3 lao động, 2 vật liệu, 3 máy móc
- Công ty có tối đa 100 lao động, 90 vật liệu, và 40 máy móc mỗi ngày.

Hãy lập mô hình toán học của bài toán xác định số ngày làm việc cho từng dự án sao cho không bị động về nguồn lực mà lợi nhuận đạt về cao nhất

Lời giải Ở ví dụ này, ta sẽ làm nhanh bài toán. Gọi x_1, x_2, x_3 lần lượt là số ngày làm việc cho dự án X, Y và Z. Ta có các điều kiện $x_1, x_2, x_3 \geq 0$

Để không bị động về nguồn lực ta có các điều kiện sau:

$$\begin{array}{ccccccc} 2x_1 & + & 4x_2 & + & 3x_3 & \leq 100 \\ 3x_1 & + & x_2 & + & 2x_3 & \leq 90 \\ 2x_1 & + & x_2 & + & 3x_3 & \leq 40 \end{array}$$

Tổng doanh thu theo dự kiến là: $500x_1 + 300x_2 + 400x_3$

Để doanh thu đạt được cao nhất ta có điều kiện

$$500x_1 + 300x_2 + 400x_3 \longrightarrow max$$

Như vậy, mô hình toán học của bài toán là:

$$(1)\ 500x_1 + 300x_2 + 400x_3 \longrightarrow max$$

$$(2) \begin{cases} 2x_1 & + & 4x_2 & + & 3x_3 & \leq 100 \\ 3x_1 & + & x_2 & + & 2x_3 & \leq 90 \\ 2x_1 & + & x_2 & + & 3x_3 & \leq 40 \end{cases}$$

$$(3)\ x_1, x_2, x_3 \geq 0$$

Đưa bài toán về dạng chính tắc ta thêm các ẩn phụ $x_4, x_5, x_6 \geq 0$ sao cho:

$$\begin{array}{ccccccccc} 2x_1 & + & 4x_2 & + & 3x_3 & + & x_4 & = 100 \\ 3x_1 & + & x_2 & + & 2x_3 & + & x_5 & = 90 \\ 2x_1 & + & x_2 & + & 3x_3 & + & x_6 & = 40 \end{array}$$

Ma trận hệ số của ràng buộc: $A = \begin{bmatrix} 2 & 4 & 3 & 1 & 0 & 0 \\ 3 & 1 & 2 & 0 & 1 & 0 \\ 2 & 1 & 3 & 0 & 0 & 1 \end{bmatrix}$

Trong đó:

- Ẩn cơ bản (1): $x_4 = 100$
- Ẩn cơ bản (2): $x_5 = 90$
- Ẩn cơ bản (3): $x_6 = 40$

Ta có phương án xuất phát: $x = (x_1, x_2, x_3, x_4, x_5, x_6) = (0, 0, 0, 100, 90, 40)$

Ta xây dựng được bảng Simplex đầu tiên:

Hệ số	Ẩn cơ bản	Phương án	x_1 500	x_2 300	x_3 400	x_4 0	x_5 0	x_6 0	λ_i
0	x_4	100	2	4	3	1	0	0	50
0	x_5	90	3	1	2	0	1	0	30
0	x_6	40	2	1	3	0	0	1	20
		0	-500	-300	-400	0	0	0	Δ_j

Vì $\Delta_j \leq 0$ nên ta tiến hành tạo bảng thứ 2.

Ta lập được bảng sau:

Hệ số	Ẩn cơ bản	Phương án	x_1 500	x_2 300	x_3 400	x_4 0	x_5 0	x_6 0	λ_i
0	x_4	60	0	3	0	1	0	-1	20
0	x_5	30	0	$-1/2$	$-5/2$	0	1	$-3/2$	\\
500	x_1	20	1	$1/2$	$3/2$	0	0	$1/2$	40
		10000	0	-50	350	0	0	250	Δ_j

Vì $\Delta_j \leq 0$ nên ta tiếp tục chuyển sang bảng số 3

Hệ số	Ẩn cơ bản	Phương án	x_1 500	x_2 300	x_3 400	x_4 0	x_5 0	x_6 0	λ_i
300	x_2	20	0	1	0	$1/3$	0	$-1/3$	
0	x_5	40	0	0	$-5/2$	$1/6$	1	$-5/3$	
500	x_1	10	1	0	$3/2$	$-1/6$	0	$2/3$	
		11000	0	0	350	$50/3$	0	$700/3$	$\Delta_j \geq 0$

Kết luận cho bài toán phụ: PATƯ: $X^2 = (x_1, x_2, x_3, x_4, x_5, x_6) = (10, 20, 0, 0, 40, 0)$

GTTƯ: $f(X2) = 11000$

Kết luận cho bài toán gốc: PATƯ: $X^2 = (x_1, x_2, x_3) = (10, 20, 0)$

GTTƯ: $f(X2) = 11000$

Vậy để lợi nhuận đạt cao nhất:

- Số ngày làm việc cho dự án X: 20 ngày
- Số ngày làm việc cho dự án Y: 10 ngày
- Số ngày làm việc cho dự án Z: 0 ngày

3.2.3 Nhận xét:

☐ Ưu điểm

- Phương pháp đơn hình thường hiệu quả với các bài toán có số biến và ràng buộc lớn.
- Phương pháp đơn hình có thể giải quyết các bài toán với rất nhiều ràng buộc phức tạp và hàm mục tiêu không tuyến tính, miễn là chúng có thể được biểu diễn dưới dạng tuyến tính.
- Có thể dễ dàng tự động hóa phương pháp đơn hình, điều này có nghĩa là nó có thể được triển khai trong các phần mềm hoặc hệ thống thông tin để giải quyết các bài toán tối ưu.

☐ Nhược điểm

- Trong một số trường hợp, phương pháp đơn hình có thể rơi vào một vòng lặp vô hạn mà không thể tìm ra giải pháp tối ưu.
- Trong một số trường hợp, phương pháp đơn hình có thể yêu cầu một số lượng lớn các bước lặp để đạt được giải pháp tối ưu, đặc biệt là trên các bài toán có số lượng biến và ràng buộc lớn.

3.3 Phương pháp đối ngẫu:

3.3.1 Thuật toán:

Quy tắc chuyển đổi giữa bài toán $min - max$

☐ **TH1:** Bài toán gốc tìm min, bài toán đối ngẫu tìm max

Bài toán gốc (P)	Bài toán đối ngẫu (D)
Hệ số hàm mục tiêu	Vế phải của ràng buộc chính
Ràng buộc chính thứ i dấu $\begin{pmatrix} \geq \\ \leq \\ = \end{pmatrix}$	Biến thứ i dấu $\begin{pmatrix} \geq 0 \\ \leq 0 \\ \text{tuỳ ý} \end{pmatrix}$
Biến thứ j dấu $\begin{pmatrix} \geq 0 \\ \leq 0 \\ \text{tuỳ ý} \end{pmatrix}$	Ràng buộc chính thứ j dấu $\begin{pmatrix} \leq \\ \geq \\ = \end{pmatrix}$

☐ **TH2:** Bài toán gốc tìm max, bài toán đối ngẫu tìm min

Bài toán gốc (P)	Bài toán đối ngẫu (D)
Hệ số hàm mục tiêu	Vế phải của ràng buộc chính
Ràng buộc chính thứ i dấu $\begin{pmatrix} \geq \\ \leq \\ = \end{pmatrix}$	Biến thứ i dấu $\begin{pmatrix} \leq 0 \\ \geq 0 \\ \text{tuỳ ý} \end{pmatrix}$
Biến thứ j dấu $\begin{pmatrix} \geq 0 \\ \leq 0 \\ \text{tuỳ ý} \end{pmatrix}$	Ràng buộc chính thứ j dấu $\begin{pmatrix} \geq \\ \leq \\ = \end{pmatrix}$

☐ Bài toán đối ngẫu (D) giúp ta khảo sát tính chất của bài toán gốc (P) (mà không đi giải trực tiếp bài gốc), cụ thể là:

- Nếu (P) tìm max thì mỗi phương án của bài (D) sẽ cho ta chặn trên của (P).
- Nếu (P) tìm min thì mỗi phương án của bài (D) sẽ cho ta chặn dưới của (P).

☐ Sau khi chuyển thành bài toán đối ngẫu, ta có thể dùng phương pháp đơn hình để tìm phương án tối ưu.

☐ Đối với cặp bài toán đối ngẫu (P) và (D) chỉ xảy ra một trong ba trường hợp sau:

- Cả hai bài toán đều không có phương án tối ưu
- Cả hai bài toán đều có phương án, lúc đó chúng đều có phương án tối ưu và giá trị hàm mục tiêu đối với hai phương án tối ưu là bằng nhau.
- Một trong hai bài toán không có phương án, còn bài toán kia thì có phương án, khi đó bài toán có phương án không có phương án tối ưu.

3.3.2 Ví dụ minh hoạ:

Ví dụ: Lập bài toán đối ngẫu của bài toán QHTT sau:

$$(P) : f(x) = 3x_1 + 2x_2 + 1x_3 \longrightarrow max$$

$$\begin{cases} x_1 & + & 2x_2 & - & x_3 & \leq 4 \\ 2x_1 & - & x_2 & + & x_3 & = 8 \\ x_1 & - & x_2 & & & \leq 6 \end{cases}$$

$$x_1, x_2 \geq 0; x_3 \text{ tuỳ ý}$$

Áp dụng quy tắc chuyển đổi bài toán, ta có:

$$(D) : f(x) = 4y_1 + 8y_2 + 6y_3 \longrightarrow min$$

$$\begin{cases} y_1 & + & 2y_2 & + & y_3 & \geq 3 \\ 2y_1 & - & y_2 & - & y_3 & \geq 2 \\ -y_1 & + & y_2 & & & = 6 \end{cases}$$

$$y_1 \geq 0; y_2 \text{ tuỳ ý}; y_3 \geq 0$$

3.3.3 Nhận xét:

☐ Ưu điểm:

- Phương pháp đối ngẫu đưa ta một góc nhìn khác để giải bài toán, đôi khi việc giải bài toán gốc tốn nhiều công sức hơn bài toán đối ngẫu.
- Phương pháp đối ngẫu sẽ là một công cụ mạnh khi kết hợp với thuật toán đơn hình.

☐ Nhược điểm:

- Trong một vài trường hợp, bài toán đối ngẫu sẽ phức tạp hơn bài toán gốc.
- Phương pháp đối ngẫu có thể bị mắc kẹt ở một giải pháp không tối ưu và không thể tìm thấy giải pháp tốt hơn.

III. PHẦN MỀM TÌM PHƯƠNG ÁN TỐI ƯU CHO BÀI TOÁN VẬN TẢI

1 Bài toán vận tải:

Ta có mô hình bài toán vận tải như sau:

$$\min f(x) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

$$\begin{cases} \sum_{j=1}^{n} x_{ij} = s_i \text{ với } (i = 1, ..., m) \\ \sum_{i=1}^{m} x_{ij} = d_j \text{ với } (j = 1, ..., n) \\ x_{ij} \geq 0, (i = 1, ..., m); (j = 1, ..., n) \end{cases}$$

☐ Trong đó:

- m: Tổng số điểm cung cấp (điểm nguồn).
- n: Tổng số điểm tiêu thụ (điểm đích).
- s_i: Khả năng cung cấp của điểm nguồn thứ i $i = (1, 2, ..., m)$.
- d_j: Nhu cầu tiêu thụ của điểm đích thứ j $j = (1, 2, ..., n)$.
- c_{ij}: Cước phí vận chuyển từ điểm nguồn i tới điểm đích j.
- x_{ij}: Lượng hàng được vận chuyển từ điểm nguồn i tới điểm đích j.

Ta có tổng số hàng dự trữ ở m điểm phát (cung) là $\sum_{j=1}^{n} s_i$

Và tổng số nhu cầu của n điểm thu (cầu)là $\sum_{i=1}^{m} d_j$

Cân bằng cung cầu:

☐ Nếu $\sum_{j=1}^{n} s_i = \sum_{i=1}^{m} d_j$ thì ta nói bài toán cân bằng thu phát, do đó có thể dùng phương án cực biên để tìm cước phí nhỏ nhất.

☐ Ngược lại, bài toán không cân bằng thu phát khi:

- TH1: Cung nhiều hơn cầu: $\sum_{j=1}^{n} s_i > \sum_{i=1}^{m} d_j$ thì một số hàng hoá được để lại ở các điểm phát. Ta biểu diễn việc này bằng cách bổ sung một điểm thu giả d_{n+1} với cước phí $c_{i,n+1} = 0$ với mọi $i = 1, ..., n$.
- TH2: Cung ít hơn cầu: $\sum_{j=1}^{n} s_i < \sum_{i=1}^{m} d_j$ thì một số hàng hóa sẽ thiếu cho các điểm thu. Ta biểu diễn việc này bằng cách bổ sung một điểm phát giả s_{m+1} với cước phí $c_{m+1,j} = 0$ với mọi $j = 1, ..., m$.

☐ Do đó, **bài toán vận tải luôn có thể đưa về dạng cân bằng thu phát.**

☐ Tiêu chí chung để chọn lượng hàng và loại cột/hàng:

$$x_{ij} = min\{s_i; d_j\} = \begin{cases} s_i & \text{loại dòng } i, d_j = d_j - s_i \\ d_j & \text{loại cột } j, s_i = s_i - d_j \\ s_i = d_j & \text{loại dòng } i \text{ cột } j \end{cases}$$

Kho (i)	Điểm bán lẻ (j)				Tổng cung
	1	2	...	m	
1	c_{11}	c_{12}	...	c_{1m}	s_1
2	c_{21}	c_{22}	...	c_{2m}	s_2
...	...	...	...	...	...
n	c_{n1}	c_{n2}	...	c_{nm}	s_n
Tổng cầu	d_1	d_2	...	d_m	$\sum_{j=1}^{n} s_i$ hoặc $\sum_{i=1}^{m} d_j$

1.1 *Phương án cực biên cơ bản:*

☐ Phương án cực biên không suy biến

- Số ô được chọn bằng $m + n - 1$.

☐ Phương án cực biên suy biến

- Số ô được chọn nhỏ hơn $m + n - 1$.
- Ta sẽ chọn thêm một ô có giá trị nhỏ nhất theo thứ từ trên xuống, trái sang phải và không tạo thành chu trình.

1.2 *Phương pháp Tây Bắc:*

☐ Phương án dễ thực hiện nhưng kém hiệu quả:

- Chọn ô ở góc trái trên cùng (hay góc Tây Bắc) làm nơi xuất phát.
- Cung cấp tối đa từ khả năng của điểm nguồn cho các điểm đích theo thứ tự từ trái sang phải. Nếu nguồn i đã hết thì chuyển sang nguồn $i + 1$.
- Đáp ứng tối đa nhu cầu của 1 điểm đích tới các nguồn theo thứ tự ưu tiên từ trên xuống dưới. Nhu cầu j đã đủ thì chuyển sang như cầu $j + 1$.

☐ **VD:** Giả sử ta có bài toán cân bằng thu phát sau:

Kho (i)	Điểm bán lẻ (j)				Tổng cung
	1	2	3	4	
1	c_{11} 90	c_{12} 100	c_{13}	c_{14}	190
2	c_{21}	c_{22} 50	c_{23} 90	c_{24}	140
3	c_{31}	c_{32}	c_{33} 170	c_{34}	170
4	c_{41}	c_{42}	c_{43} 40	c_{44} 210	250
Tổng cầu	90	150	300	210	750

☐ Giải thích:

- Đầu tiên, chọn ô $(i, j) = (1, 1)$ do nguồn cung là $s_1 = 190 > 90 = d_1$ nên $x_{11} = 90$ và loại cột $j = 1$ theo tiêu chí chung. Khi này nguồn cung $s'_1 = s_1 - d_1 = 100$.
- Vì cột $j = 1$ đã bỏ, nên ta xét ô $(i, j) = (1, 2)$ và do $s'_1 = 100 < d_2$ nên $x_{12} = 100$. Khi nào nhu cầu $d'_2 = d_2 - s'_1 = 50$.
- Tiếp tục làm như vậy, ta được bảng trên.

1.3 Phương pháp cước phí nhỏ nhất:

☐ Phương án hiệu quả hơn phương pháp Tây Bắc:

- Ưu tiên chọn ô có cước phí nhỏ nhất từ trên xuống dưới để đáp ứng tối đa khả năng cũng như nhu cầu.
- Tiến hành bỏ qua những ô của điểm nguồn (bỏ hàng nhu cầu) hoặc điểm đích (bỏ cột cung cấp) đã hết khả năng cung cấp cũng như nhu cầu.
- Xác định lại ô có chi phí thấp nhất trong các ô còn lại và tiếp tục phân bổ giống như các bước trên.

☐ **<u>VD:</u>** Giả sử ta có bài toán cân bằng thu phát sau:

Kho (i)	Điểm bán lẻ (j)				Tổng cung
	1	2	3	4	
1	4 ☐	1 150	7 ☐	4 40	190
2	6 ☐	8 ☐	3 50	5 90	140
3	3 90	9 ☐	11 ☐	10 80	170
4	8 ☐	6 ☐	2 250	12 ☐	250
Tổng cầu	90	150	300	210	750

☐ Giải thích:

- Đầu tiên, chọn ô $(i, j) = (1, 2)$ do có cước phí nhỏ nhất với $c_{12} = 1$, vì $s_1 = 190 > 150 = d_2$ nên $x_{12} = 150$, tiến hành loại bỏ cột 2.
- Lặp lại việc tìm kiếm, chọn ô $(4, 3)$ với $c_{43} = 2$, ta có $x_{43} = 250$ loại bỏ hàng 4.
- Tiếp tục làm như vậy, ta được bảng trên.

1.4 Thuật toán thế vị:

Thuật toán thế vị chính là thuật toán đơn hình cho bài toán vận tải

☐ **Bước 1:** Xác định các thế vị $u_i, i = 1, ..., n$ và $v_j, j = 1, ..., m$ tương ứng với phương án cực biên x_0 bằng việc giải hệ phương trình:

$$u_i + v_j = c_{ij} \ \forall i, j$$

☐ **Bước 2:** Tính các ước lượng $\Delta_{ij} = u_i + v_j - c_{ij}$

☐ **Bước 3:** Nếu $\Delta_{ij} \le 0$ với mọi $(i = 1, ..., m)$ và $(j = 1, ..., n)$. Thuật toán dừng lại, ta có phương án tối ưu. Nếu không, chuyển sang *bước 4*.

☐ **Bước 4:**

1. Xác định ô điều chỉnh (i_s, j_s) với
$$\Delta_{i_s,j_s} = max\{\Delta_{ij} > 0 | (i = 1, ..., m); (j = 1, ..., n)\}$$
2. Xét chu trình chứa ô đó và các ô chọn ban đầu
3. Đánh dấu $+/-$ xen kẽ vào chu trình này. Với dấu $+$ được đánh cho ô chọn.
4. Xác định x_{ij} nhỏ nhất trong các ô được gán dấu $(-)$.
5. Bớt đi một lượng x_{ij} ở các ô được gán dấu $(-)$ và thêm một lượng x_{ij} ở các ô được gán dấu $(+)$. Quay lại *bước 2*.

☐ **Chú ý:** Cần thành lập phương án cực biên ban đầu (xuất phát) theo Nguyên lý phân bố tối đa với các ô chọn phân bổ bằng các phương pháp: góc Tây Bắc, cước phí thấp nhất,...

Kho (i)	Điểm bán lẻ (j)				Tổng cung	
	1	2	...	m		u_i
1	c_{11}	c_{12}	...	c_{1m}	s_1	u_1
2	c_{21}	c_{22}	...	c_{2m}	s_2	u_2
...	...	...	...	...	...	...
n	c_{n1}	c_{n2}	...	c_{nm}	s_n	u_n
Tổng cầu	d_1	d_2	...	d_m	$\sum_{j=1}^{n} s_i$ hoặc $\sum_{i=1}^{m} d_j$	
v_j	v_1	v_2	...	v_m		

☐ **Ví dụ 1.4:** Ta có bảng của bài toán vận tải như sau. Hãy tìm phương án tối ưu.

Kho (i)	Điểm bán lẻ (j)				Tổng cung	
	1	2	3	4		u_i
1	1	2	4	3	60	u_1
2	2	3	2	7	70	u_2
3	3	5	6	4	20	u_3
Tổng cầu	30	40	30	50	$\sum_{j=1}^{n} s_i$ hoặc $\sum_{i=1}^{m} d_j$	
v_j	v_1	v_2	v_3	v_m		

- Dùng phương pháp cước phí nhỏ nhất xác định nghiệm ban đầu của bài toán:
- Đây là phương án xuất phát cơ bản không suy biến, do có $m + n - 1 = 4 + 3 - 1 = 6$ ô chọn.
- **Bước 1:** Xác định các thế vị bằng cách giải hệ phương trình
  - $*$ $u_1 = 0$; $u_1 + v_1 = 1 \longrightarrow v_1 = 1$
  - $*$ $u_1 = 0$; $u_1 + v_2 = 2 \longrightarrow v_2 = 2$
  - $*$ $v_2 = 2$; $u_2 + v_2 = 3 \longrightarrow u_2 = 1$

Kho (i)	Điểm bán lẻ (j)				Tổng cung	u_i
	1	2	3	4		
1	^{1} 30	^{2} 30	□ ^{4}	□ ^{3}	60	u_1
2	□ ^{2}	^{3} 10	^{2} 30	^{7} 30	70	u_2
3	□ ^{3}	□ ^{5}	□ ^{6}	^{4} 20	20	u_3
Tổng cầu	30	40	30	50	$\sum_{j=1}^{n} s_i$ hoặc $\sum_{i=1}^{m} d_j$	
v_j	v_1	v_2	v_3	v_m		

$*$ $u_2 = 1$; $u_2 + v_3 = 2 \longrightarrow v_3 = 1$

$*$ $u_2 = 1$; $u_2 + v_4 = 7 \longrightarrow v_3 = 6$

$*$ $v_4 = 6$; $u_3 + v_4 = 4 \longrightarrow u_3 = -2$

Kho (i)	Điểm bán lẻ (j)				Tổng cung	u_i
	1	2	3	4		
1	^{1} 30	^{2} 30	□ ^{4}	□ ^{3}	60	0
2	□ ^{2}	^{3} 10	^{2} 30	^{7} 30	70	1
3	□ ^{3}	□ ^{5}	□ ^{6}	^{4} 20	20	-2
Tổng cầu	30	40	30	50	$\sum_{j=1}^{n} s_i$ hoặc $\sum_{i=1}^{m} d_j$	
v_j	1	2	1	6		

- **Bước 2:** Tính các ước lượng

Kho (i)	Điểm bán lẻ (j)				Tổng cung	u_i
	1	2	3	4		
1	^{1} 30	^{2} 30	□ ^{4} -3	□ ^{3} 3	60	0
2	□ ^{2} 0	^{3} 10	^{2} 30	^{7} 30	70	1
3	□ ^{3} -4	□ ^{5} -5	□ ^{6} -7	^{4} 20	20	-2
Tổng cầu	30	40	30	50	$\sum_{j=1}^{n} s_i$ hoặc $\sum_{i=1}^{m} d_j$	
v_j	1	2	1	6		

- **Bước 3:** Vì $\Delta_{14} = 3 > 0$ nên ta chuyển sang **Bước 4:**.

Kho (i)	Điểm bán lẻ (j)				Tổng cung	u_i
	1	2	3	4		
1	^{1} 30	$\boxed{-}$ ^{2} 30	□ ^{4} -3	$\boxed{+}$ ^{3} 3	60	0
2	□ ^{2} 0	$\boxed{+}$ ^{3} 10	^{2} 30	$\boxed{-}$ ^{7} 30	70	1
3	□ ^{3} -4	□ ^{5} -5	□ ^{6} -7	^{4} 20	20	-2
Tổng cầu	30	40	30	50	$\sum_{j=1}^{n} s_i$ hoặc $\sum_{i=1}^{m} d_j$	
v_j	1	2	1	6		

□ Tiếp tục **Bước 4:**

- Vì giá trị $x_{12} = x_{24} = 30$ nên ta xét đến cước phí $c_{12} = 2 < 7 = c_{24}$. Ta tiến hành loại bỏ ô $(i, j) = (2, 4)$. Các ô $(2, 2)$ và $(1, 4)$ tăng thêm 30 và ô $(1, 2)$ giảm về 0.

Kho (i)	Điểm bán lẻ (j)				Tổng cung	u_i
	1	2	3	4		
1	^{1} 30	^{2} 30	□ ^{4}	^{3} 30	60	0
2	□ ^{2}	^{3} 10	^{2} 30	/////////	70	1
3	□ ^{3}	□ ^{5}	□ ^{6}	^{4} 20	20	-2
Tổng cầu	30	40	30	50	$\sum_{j=1}^{n} s_i$ hoặc $\sum_{i=1}^{m} d_j$	
v_j	1	2	1	6		

- Quay lại **Bước 1:**, làm tương tự với **Bước 2:**, ta được

Kho (i)	Điểm bán lẻ (j)				Tổng cung	u_i
	1	2	3	4		
1	^{1} 30	^{2} 30	□ ^{4} -3	^{3} 30	60	0
2	□ ^{2} 0	^{3} 10	^{2} 30	/////////	70	1
3	□ ^{3} -1	□ ^{5} -2	□ ^{6} -4	^{4} 20	20	1
Tổng cầu	30	40	30	50	$\sum_{j=1}^{n} s_i$ hoặc $\sum_{i=1}^{m} d_j$	
v_j	1	2	1	3		

- Vì $\Delta_{ij} \leq 0$ với mọi $(i = 1, ..., m)$ và $(j = 1, ..., n)$. Thuật toán dừng lại, ta được một phương án tối ưu.

$$X^1 = \begin{bmatrix} 30 & 0 & 0 & 30 \\ 0 & 40 & 30 & 0 \\ 0 & 0 & 0 & 20 \end{bmatrix} \quad \text{với } f(X^1) = 380 = f_{min}$$

2 Đoạn Code hiện thực trên phần mềm Matlab cho bài toán vận tải

2.1 Code Matlab:

```
x = input('Nhap ma tran gia : ');
cung = input('Nhap ma tran cung : ');
cau = input('Nhap ma tran cau : ');
y=x;
cungg=cung;
cauu=cau;

% Phuong phap Tay Bac
tong_cung = sum(cung);
tong_cau = sum(cau);

if tong_cung == tong_cau
    disp('Can bang cung va cau');

    [m, n] = size(x);
    matrix = zeros(m, n);
    count = 1;

    for i = 1:m
        for j = count:n
            matrix(i, j) = min(cung(i), cau(j));

            % Cap nhat cung va cau
            cung(i) = cung(i) - matrix(i, j);
            cau(j) = cau(j) - matrix(i, j);

            % Neu cau dat 0, chuyen sang hang tiep theo
            if cau(j) == 0
                count = count + 1;
            end
        end
    end
    disp('Ma tran phan phoi:');
    disp(matrix);

    % Tinh tong chi phi toi uu
    minTB = sum(sum(matrix .* x));
    disp('Chi phi toi uu theo phuong phap Tay Bac:');
    disp(minTB);
else
    disp('Tong cung khong bang tong cau. Vui long kiem tra lai
   dau vao.');
end

% Phuong Phap cuoc phi nho nhat

x = y;
cung = cungg;
```

```
cau = cauu;
tong_cung = sum(cung);
tong_cau = sum(cau);

if tong_cung == tong_cau
    [row, col] = size(x);
    copy = x;
    newmatrix = zeros(length(cung), length(cau)); % Khoi tao newmatrix

    while any(copy(:) ~= -1) % Kiem tra neu van con phan tu khac -1 trong ma tran copy
        mink = min(copy(copy ~= -1)); % Tim gia tri nho nhat trong ma tran copy

        [min_row, min_col] = find(copy == mink, 1); % Tim vi tri cua gia tri nho nhat
        min_val = min(cung(min_row), cau(min_col)); % Tinh gia tri cuoc phi nho nhat
        newmatrix(min_row, min_col) = min_val; % Cap nhat ma tran newmatrix

        if cung(min_row) == newmatrix(min_row, min_col)
            copy(min_row, :) = -1; % Danh dau hang da duoc phuc vu het
            cau(min_col) = cau(min_col) - min_val; % Cap nhat nhu cau con lai
            cung(min_row) = cung(min_row) - min_val; % Cap nhat cung con lai
        else
            copy(:, min_col) = -1; % Danh dau cot da duoc phuc vu het
            cau(min_col) = cau(min_col) - min_val; % Cap nhat nhu cau con lai
            cung(min_row) = cung(min_row) - min_val; % Cap nhat cung con lai
        end
    end

    minCP = sum(sum(newmatrix .* x)); % Tinh cuoc phi nho nhat
    disp('Ma tran phan phoi:');
    disp(newmatrix);
    disp('Cuoc phi nho nhat tinh theo phuong phap Cuoc Phi Nho Nhat la: ');
    disp(minCP);
else
    disp('Tong cung khong bang tong cau. Vui long kiem tra lai dau vao.');
end


% So sanh 2 phuong phap:
```

```
disp('-----------------------------------------------');
if(minTB<minCP)
    disp('Phuong Phap Tay Bac toi uu hon Phuong Phap Cuoc Phi Nho
    Nhat');
else
    disp('Phuong Phap Cuoc Phi Nho Nhat toi uu hon Phuong Phap
   Tay Bac');
end

% Phuong phap the vi:

c=y;
row = -1;
col = -1;
n = length(cung);
m = length(cau);
while 1
    minn = 0;
    % Khai bao the vi u cua nguon cung
    u = -999.5 * ones(1, n);
    u(1)=0;
    % Khai bao the vi v cua nguon cap
    v = -999.5 * ones(1, m);
    delta = -9999.5 * ones(n, m);

    % vong lap duoi dung de xac dinh gia tri cua the vi u va v
    dk1 = 1;
    while 1
        for i = 1:n
            for j = 1:m
                if newmatrix(i,j) > 0
                    if u(i) ~= -999.5
                    v(j) = c(i,j) - u(i);
                    elseif v(j) ~= -999.5
                        u(i) = c(i,j) - v(j);
                    end
                end
            end
        end
        for i = 1:n
            for j = m:-1:1
                if newmatrix(i,j) > 0
                    if u(i) ~= -999.5
                        v(j) = c(i,j) - u(i);
                    elseif v(j) ~= -999.5
                        u(i) = c(i,j) - v(j);
                    end
                end
            end
        end
        for i = n:-1:1
            for j = 1:m
```

```
                if newmatrix(i,j) > 0
                    if u(i) ~= -999.5
                        v(j) = c(i,j) - u(i);
                    elseif v(j) ~= -999.5
                        u(i) = c(i,j) - v(j);
                    end
                end
            end
        end
        for i = n:-1:1
            for j = m:-1:1
                if newmatrix(i,j) > 0
                    if u(i) ~= -999.5
                        v(j) = c(i,j) - u(i);
                    elseif v(j) ~= -999.5
                        u(i) = c(i,j) - v(j);
                    end
                end
            end
        end
        for i = 1:n
            for j = 1:m
                if u(i) == -999.5 || v(j) == -999.5
                    dk1 = 0;
                end
            end
        end
        if (dk1)
            break;
        end
        dk1 = 1;
    end

    % Tinh he so uoc luong Delta va tim gia tri lon nhat trong cac he so
    for i = 1:n
        for j = 1:m
            if newmatrix(i,j) <= 0
                delta(i,j) = u(i) + v(j) - c(i,j);
                if minn < delta(i,j)
                    minn = delta(i,j);
                    row = i;
                    col = j;
                end
            end
        end
    end
    % Thuat toan dung lai khi gia tri lon nhat cua he so uoc luong la 0
    if minn <= 0
        break;
    end
```

```
  % Neu khong thuat toan tiep tuc thuc hien viec dieu chinh bang
  for i = 1:n
    for j = 1:m
      if i ~= row && j ~= col
      % Xac dinh chu trinh chua o co he so uoc luong duong lon nhat
        if newmatrix(i,j) > 0 && newmatrix(row,j) > 0 && newmatrix(i,col) > 0
          % Xac dinh o danh dau tru co gia tri nho nhat
          mincheo = min(newmatrix(row,j), newmatrix(i,col));
          % Thuc hien dieu chinh bang
          newmatrix(row,j) = newmatrix(row,j) - mincheo;
          newmatrix(i,col) = newmatrix(i,col) - mincheo;
          newmatrix(i,j) = newmatrix(i,j) + mincheo;
          newmatrix(row,col) = newmatrix(row,col) + mincheo;
          if newmatrix(row,j) == 0 && newmatrix(i,col) == 0
            % Buoc rut ngan thuat toan, so sanh cuoc phi
            if c(row,j) < c(i,col)
              newmatrix(i,col) = -999;
            else
              newmatrix(row,j) = -999;
            end
          end
          if newmatrix(row,j) == 0 && newmatrix(i,col) ~= 0
            newmatrix(row,j) = -999;
          end
          if newmatrix(row,j) ~= 0 && newmatrix(i,col) == 0
            newmatrix(i,col) = -999;
          end
        end
      end
    end
  end
  if minn <= 0
    break;
  end
end
sum = 0;
  for i = 1:n
    for j = 1:m
      if newmatrix(i,j) < 0
        newmatrix(i,j) = 0;
      end
```

```
            if newmatrix(i,j) > 0
                % Tinh tong cac gia tri
                sum = sum + newmatrix(i,j) * c(i,j);
            end
        end
    end
disp('Ma tran phan phoi');
disp(newmatrix);
disp('Chi phi toi uu hoa theo thuat toan the vi la');
disp(sum);
disp('Thuat toan the vi la thuat toan dua ra dap an toi uu nhat')
   ;
```

2.1.1 Giá trị đầu vào thứ nhất

Nhap ma tran gia: [2 1 5; 3 4 3; 4 6 6]

Nhap ma tran cung: [50 60 70]

Nhap ma tran cau: [40 85 55]

2.1.2 Kết quả thu được:

Kết quả thu được khi chạy chương trình

```
Nhap ma tran gia : [2 1 5; 3 4 3; 4 6 6]
Nhap ma tran cung : [50 60 70]
Nhap ma tran cau : [40 85 55]
Can bang cung va cau
Ma tran phan phoi:
    40    10     0
     0    60     0
     0    15    55

Chi phi toi uu theo phuong phap Tay Bac:
   750

Ma tran phan phoi:
     0    50     0
    40     0    20
     0    35    35

Cuoc phi nho nhat tinh theo phuong phap Cuoc Phi Nho Nhat la:
   650

---------------------------------------------------
Phuong Phap Cuoc Phi Nho Nhat toi uu hon Phuong Phap Tay Bac
Ma tran phan phoi
     0    50     0
     0     5    55
    40    30     0

Chi phi toi uu hoa theo thuat toan the vi la
   575

Thuat toan the vi la thuat toan dua ra dap an toi uu nhat
```

Hình I.1: Kết quả thu được khi chạy chương trình

☐ **Tiến hành kiểm tra lại kết quả.**

- Ta lập được bảng sau:

Kho (i)	Điểm bán lẻ (j)			Tổng cung
	1	2	3	
1	2	1	5	50
2	3	4	3	60
3	4	6	6	70
Tổng cầu	40	85	55	180

☐ Theo phương pháp Tây Bắc

- Chọn ô đầu tiên là ô $(i, j) = (1, 1)$ làm ô xuất phát. Vì $d_1 = 40 < 50 = s_1$. Nên $x_{11} = 40$. Khi đó $s_1 = 50 - 40 = 10$.
- Chuyển qua ô tiếp theo $(i, j) = (1, 2)$. Vì $s_1 = 10 < 85 = d_2$ nên $x_{12} = 10$. Khi đó $d_2 = 85 - 10 = 75$.
- Chuyển qua ô tiếp theo $(i, j) = (2, 2)$. Vì $s_2 = 60 < 75 = d_2$ nên $x_{22} = 60$. Khi đó $d_2 = 75 - 60 = 15$.
- Chuyển qua ô tiếp theo $(i, j) = (3, 2)$. Vì $d_2 = 15 < 70 = s_3$ nên $x_{32} = 15$. Khi đó $s_3 = 70 - 15 = 55$.
- Ta lập được bảng sau:

Kho (i)	Điểm bán lẻ (j)			Tổng cung
	1	2	3	
1	2 40	1 10	5 ☐	50
2	3 ☐	4 60	3 ☐	60
3	4 ☐	6 15	6 55	70
Tổng cầu	40	85	55	180

☐ Theo phương pháp cước phí nhỏ nhất

- Chọn ô $(i, j) = (1, 2)$ có cước phí nhỏ nhất là $c_{12} = 1$. Vì $s_1 = 50 < 85 = d_2$ nên $x_{12} = 50$. Khi đó $d_2 = 85 - 50 = 30$. Loại hàng 1 vì hết khả năng cung cấp.
- Chọn ô $(i, j) = (2, 1)$ có cước phí nhỏ nhất là $c_{21} = 3$. Vì $d_1 = 40 < 60 = s_2$ nên $x_{21} = 40$. Khi đó $s_2 = 60 - 40 = 20$. Loại cột 1 vì hết khả năng cung cấp.
- Chọn ô $(i, j) = (2, 3)$ có cước phí nhỏ nhất là $c_{23} = 3$. Vì $s_2 = 20 < 55 = d_3$ nên $x_{23} = 20$. Khi đó $d_3 = 55 - 20 = 35$. Loại hàng 2 vì hết khả năng cung cấp.
- Từ các ô đã có suy ra được: $x_{32} = 35$ và $x_{33} = 35$.
- Ta lập được bảng sau:

☐ Theo thuật toán thế vị:

Kho (i)	Điểm bán lẻ (j)			Tổng cung
	1	2	3	
1	^{2} $\square$	^{1} 50	^{5} $\square$	50
2	^{3} 40	^{4} $\square$	^{3} 20	60
3	^{4} $\square$	^{6} 35	^{6} 35	70
Tổng cầu	40	85	55	180

- Ta sẽ sử dụng phương án ban đầu là ma trận thu được ở phương pháp cước phí nhỏ nhất
- Chọn $u_1 = 0 \longrightarrow v_2 = 1 \longrightarrow u_3 = 5 \longrightarrow v_3 = 1 \longrightarrow u_2 = 2 \longrightarrow v_1 = 1$.
- Tính các hệ số ước lượng: $\Delta_{11} = -1$; $\Delta_{13} = -4$; $\Delta_{22} = -1$; $\Delta_{31} = 2$. Còn lại bằng 0.
- Ta lập được bảng sau:

Kho (i)	Điểm bán lẻ (j)			Tổng cung	u_i
	1	2	3		
1	^{2} $\boxed{-1}$	^{1} 50	^{5} $\boxed{-4}$	50	0
2	^{3} 40	^{4} $\boxed{-1}$	^{3} 20	60	2
3	^{4} $\boxed{+2}$	^{6} 35	^{6} 35	70	5
Tổng cầu	40	85	55	180	
v_j	1	1	1		

- Chọn ô có hệ số ước lượng dương lớn nhất là $\Delta_{31} = 2$. Chọn các ô x_{21}; x_{23}; x_{33}. Giá trị nhỏ nhất của ô x_{21} và x_{33} là 35, loại ô x_{33} khỏi bảng. Tăng ô x_{31} và x_{23} thêm 35 và giảm x_{33} đi 35. Ta lập bảng mới với các thế vị và hệ số ước lượng mới.
- Chọn $u_1 = 0 \longrightarrow v_2 = 1 \longrightarrow u_3 = 5 \longrightarrow v_1 = -1 \longrightarrow u_2 = 4 \longrightarrow v_3 = -1$.
- Tính các hệ số ước lượng: $\Delta_{11} = -3$; $\Delta_{13} = -6$; $\Delta_{22} = 1$; $\Delta_{33} = -2$. Còn lại bằng 0.
- Ta lập được bảng sau:

Kho (i)	Điểm bán lẻ (j)			Tổng cung	u_i
	1	2	3		
1	^{2} $\boxed{-3}$	^{1} 50	^{5} $\boxed{-6}$	50	0
2	^{3} 5	^{4} $\boxed{+1}$	^{3} 55	60	4
3	^{4} 35	^{6} 35	//////// ////////	70	5
Tổng cầu	40	85	55	180	
v_j	1	-1	1		

- Chọn ô có hệ số ước lượng dương lớn nhất là $\Delta_{22} = 1$. Chọn các ô x_{21}; x_{31}; x_{32}. Giá trị nhỏ nhất của ô x_{21} và x_{32} là 5, loại ô x_{21} khỏi bảng. Tăng ô x_{22} và x_{31} thêm 5 và giảm x_{33} đi 5. Ta lập bảng mới với các thế vị và hệ số ước lượng mới.

- Chọn $u_1 = 0 \longrightarrow v_2 = 1 \longrightarrow u_3 = 5 \longrightarrow v_1 = -1 \longrightarrow u_2 = 3 \longrightarrow v_3 = 0$.
- Tính các hệ số ước lượng: $\Delta_{11} = -3$; $\Delta_{13} = -5$; $\Delta_{21} = -1$; $\Delta_{33} = -1$. Còn lại bằng 0.
- Ta lập được bảng sau:

Kho (i)	Điểm bán lẻ (j)			Tổng cung	u_i
	1	2	3		
1	2 $\boxed{-3}$	1 50	5 $\boxed{-6}$	50	0
2	//////// ////////	4 5	3 55	60	3
3	4 40	6 30	//////// ////////	70	5
Tổng cầu	40	85	55	180	
v_j	-1	1	0		

- Vì các hệ số ước lượng đều âm nên đây chính là phương án tối ưu của bài toán theo thuật toán thế vị.

□ **Kết quả kiểm tra đúng theo đáp án của phần mềm.**

2.1.3 Giá trị đầu vào thứ hai

Nhap ma tran gia: [1 2 4 3 ; 2 3 2 7 ; 3 5 6 4]

Nhap ma tran cung: [60 70 20]

Nhap ma tran cau: [30 40 30 50]

Đây chính là giá trị đầu vào của bài toán được trình bày ở **Ví dụ** 1.4 trong thuật toán **Thế vị**

2.1.4 Kết quả thu được:

Kết quả thu được khi chạy chương trình

```
Nhap ma tran gia : [1 2 4 3 ; 2 3 2 7 ; 3 5 6 4]
Nhap ma tran cung : [60 70 20]
Nhap ma tran cau : [30 40 30 50]
Can bang cung va cau
Ma tran phan phoi:
    30    30     0     0
     0    10    30    30
     0     0     0    20

Chi phi toi uu theo phuong phap Tay Bac:
   470

Ma tran phan phoi:
    30    30     0     0
     0    10    30    30
     0     0     0    20

Cuoc phi nho nhat tinh theo phuong phap Cuoc Phi Nho Nhat la:
   470

---------------------------------------------------
Phuong Phap Cuoc Phi Nho Nhat toi uu hon Phuong Phap Tay Bac
Ma tran phan phoi
    30     0     0    30
     0    40    30     0
     0     0     0    20

Chi phi toi uu hoa theo thuat toan the vi la
   380

Thuat toan the vi la thuat toan dua ra dap an toi uu nhat
```

Hình I.2: Kết quả thu được khi chạy chương trình

IV. Nguồn trích dẫn

Trích dẫn *et. al.* [**?**, **?**, **?**, **?**, **?**]