

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA TOÁN - CƠ - TIN HỌC

Trịnh Văn Hải

PHƯƠNG PHÁP PHÂN RÃ DANTZIG-WOLFE
GIẢI BÀI TOÁN QUY HOẠCH KÍCH THƯỚC LỚN

KHÓA LUẬN TỐT NGHIỆP HỆ ĐẠI HỌC CHÍNH QUY

Ngành: Toán - Tin ứng dụng

Người hướng dẫn: ThS. Trần Đình Quốc

Hà Nội - 2008

LỜI CẢM ƠN

Trước khi trình bày nội dung chính của khóa luận, em xin bày tỏ lòng biết ơn sâu sắc tới Thạc sỹ Trần Đình Quốc người đã tận tình hướng dẫn để em có thể hoàn thành khóa luận này.

Em cũng xin bày tỏ lòng biết ơn chân thành tới toàn thể các thầy cô giáo trong khoa Toán - Cơ - Tin học, Đại học Khoa Học Tự Nhiên, Đại Học Quốc Gia Hà Nội đã dạy bảo em tận tình trong suốt quá trình học tập tại khoa.

Nhân dịp này em cũng xin được gửi lời cảm ơn chân thành tới gia đình, bạn bè đã luôn bên em, cổ vũ, động viên, giúp đỡ em trong suốt quá trình học tập và thực hiện khóa luận tốt nghiệp.

Hà Nội, ngày 19 tháng 05 năm 2008

Sinh viên

Trịnh Văn Hải

Mục lục

Chương 1. Một số kiến thức liên quan	6
1.1. Một số kết quả của giải tích lồi và bài toán quy hoạch tuyến tính	6
1.1.1. Tập lồi và tập lồi đa diện.....	6
1.1.2. Một số kết quả trong giải tích lồi	8
1.2. Bài toán quy hoạch tuyến tính gốc và đối ngẫu.....	9
1.2.1. Bài toán quy hoạch tuyến tính gốc và đối ngẫu.	9
1.2.2. Phương pháp đơn hình giải bài toán QHTT.....	11
1.3. Bài toán QHTT kích thước lớn	17
1.3.1. Ma trận thưa và vấn đề lưu trữ ma trận thưa kích thước lớn.....	17
1.3.2. Bài toán quy hoạch kích thước lớn và có cấu trúc.	20
Chương 2. Phương pháp phân rã Dantzig-Wolfe giải bài toán kích thước lớn	23
2.1. Bài toán quy hoạch tuyến tính Wolfe tổng quát	23
2.1.1. Phát biểu bài toán và các tính chất.....	23
2.1.2. Trường hợp bài toán phụ có phương án không bị chặn.	30
2.2. Nguyên lý phân rã Dantzig-Wolfe(D-W)	30
2.2.1. Ý tưởng của phương pháp phân rã Dantzig-Wolfe.....	31
2.2.2. Thuật toán Dantzig-Wolfe và phương án xuất phát	37
2.3. Áp dụng phương pháp Dantzig-Wolfe cho một số bài toán.....	39
2.3.1. Hệ khối góc (block-angular system).....	40
2.3.2. Bài toán cấu trúc bậc thang (staircase structured problems).....	41
2.3.3. Ví dụ bằng số minh họa thuật toán Dantzig-Wolfe.....	42

Chương 3. Thực thi thuật toán phân rã Dantzig - Wolfe.....	51
3.1.Chuyển bài toán về dạng thực thi được trên máy tính.....	51
3.1.1. Chuyển bài toán QHTT về dạng chính tắc.....	52
3.1.2. Loại bỏ các ràng buộc thừa và ràng buộc mâu thuẫn.	53
3.1.3. Kiểm tra cấu trúc của bài toán.	53
3.1.4. Xử lý kết quả.	54
3.2.Triển khai thuật toán.....	55
3.2.1. Chọn mô hình triển khai thuật toán.	55
3.2.2. Cấu trúc chương trình.	55
3.2.3. Một số thủ tục cơ bản của chương trình.	57
3.3.Các bước triển khai thực thi thuật toán.	57

LỜI MỞ ĐẦU

Ngày nay, tối ưu hóa đã trở thành một lĩnh vực rất phát triển, góp phần quan trọng trong việc ứng dụng khoa học công nghệ vào cuộc sống và sản xuất. Quy hoạch tuyến tính (QHTT) là một lĩnh vực của tối ưu hóa đã được phát triển từ những năm đầu của thế kỷ 20, đến nay toàn bộ lý thuyết toán học cho lĩnh vực này có thể nói là đã rất hoàn thiện. Mặc dù vậy, ứng dụng của QHTT vẫn đóng một vai trò rất quan trọng trong việc giải quyết các bài toán ứng dụng trong cuộc sống và kỹ nghệ. Bài toán QHTT có thể ứng dụng trực tiếp vào các lĩnh vực như sản xuất với mô hình "lập kế hoạch sản xuất", vào giao thông vận tải với mô "bài toán vận tải", vào quản lý con người với mô hình "phân việc"... hoặc nó có thể ứng dụng gián tiếp như những bài toán con trong các phương pháp, các thuật toán giải các bài toán tối ưu phi tuyến, bài toán điều khiển... Đối với các ứng dụng kể trên, bài toán QHTT thường có kích thước lớn, do đó việc xử lý chúng như thông thường là điều không thể. Do đó, việc thiết kế những thuật toán theo hướng giải quyết các bài toán lớn là một trong những vấn đề vẫn đang được quan tâm xử lý hiện nay.

Các bài toán QHTT có kích cỡ trung bình, việc sử dụng phương pháp đơn hình của Dantzig hay các phương pháp điểm trong một cách trực tiếp là rất hiệu quả và tin cậy. Tuy nhiên, qua thực tiễn tính toán và áp dụng, nhiều lớp bài toán kích thước lớn xuất hiện trong nhiều ứng dụng lại có những cấu trúc riêng trên các ràng buộc, đặc biệt là ma trận ràng buộc thường có cấu trúc đường chéo, chéo khối... và thường là những ma trận thưa.

Mục đích của khóa luận này là nhằm tìm hiểu một phương pháp giải quyết các bài toán QHTT kích thước lớn, có cấu trúc. Phương pháp được đề cập ở đây là phương pháp phân rã Dantzig-Wolfe. Trên thực tế, phương pháp phân rã Dantzig-Wolfe là một phương pháp khá tổng quát và được phát triển khá mạnh từ khi xuất hiện không chỉ trong QHTT mà cả trong quy hoạch lồi và các bài toán tối ưu nói chung. Nhìn chung, phương pháp phân rã Dantzig-Wolfe nhằm mục đích phân rã một bài toán QHTT có kích thước lớn thành một số các bài toán có kích thước nhỏ hơn bằng một số phép biến đổi. Sau đó áp dụng phương pháp đơn hình để giải quyết các bài toán phân rã và sử dụng nghiệm của các bài toán phân rã để tạo nên nghiệm cho bài toán

ban đầu.

Khóa luận tập trung làm rõ một số vấn đề sau: Trình bày ý tưởng của phương pháp phân rã Dantzig-Wolfe, các khái niệm và tính chất liên quan đến phương pháp, nội dung phương pháp và cuối cùng là các ví dụ cũng như kết quả tính toán bằng số để minh họa cho phương pháp phân rã Dantzig-Wolfe.

Bố cục của khóa luận bao gồm 3 chương và một phụ lục:

- Chương 1 của khóa luận trình bày tóm tắt một số kết quả đã biết trong giải tích lồi, thuật toán đơn hình, các định lý và kết quả cơ bản liên quan đến khóa luận. Cuối chương trình bày một số phương pháp lưu trữ ma trận.
- Chương 2 của khóa luận tập trung trình bày ý tưởng, các khái niệm và tính chất và nội dung cơ bản của phương pháp phân rã Dantzig-Wolfe giải bài toán quy hoạch tuyến tính lớn có cấu trúc. Cuối chương là một ví dụ bằng số minh họa kết quả tính toán.
- Chương 3 trình bày quan điểm thực thi của phương pháp phân rã Dantzig-Wolfe.
- Phụ lục trình bày một số module cơ bản trong lập trình thuật toán.

Do thời gian thực hiện khóa luận không nhiều, kiến thức còn hạn chế nên khi làm khóa luận không tránh khỏi những hạn chế và sai sót. Tác giả mong nhận được sự góp ý và những ý kiến phản biện của quý thầy cô và bạn đọc. Xin chân thành cảm ơn!

Hà Nội, ngày 19 tháng 05 năm 2008

Sinh viên

Trịnh Văn Hải

Chương 1

Một số kiến thức liên quan

1.1. Một số kết quả của giải tích lồi và bài toán quy hoạch tuyến tính

1.1.1. Tập lồi và tập lồi đa diện

Kí hiệu \mathbb{R}^n là không gian Euclide thực n chiều, một phần tử $x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$ là một vector cột của \mathbb{R}^n . Cho hai điểm $a = (a_1, \dots, a_n)^T$ và $b = (b_1, \dots, b_n)^T$. Khi đó, đường thẳng đi qua hai điểm a và b là tập có dạng $\{x \in \mathbb{R}^n : x = \lambda a + (1 - \lambda)b, \lambda \in \mathbb{R}\}$, còn tập $[a, b] := \{x \in \mathbb{R}^n : x = \lambda a + (1 - \lambda)b, \lambda \in [0, 1]\}$ gọi là đoạn thẳng nối hai điểm a và b . Tập $H := \{x \in \mathbb{R}^n : a^T x = b\}$ với $a \neq 0, a \in \mathbb{R}^n, b \in \mathbb{R}$ gọi là một siêu phẳng trong \mathbb{R}^n . Một siêu phẳng H chia không gian \mathbb{R}^n thành hai nửa không gian. Tập $H_{\leq} := \{x \in \mathbb{R}^n : a^T x \leq b\}$ gọi là nửa không gian đóng.

Định nghĩa 1.1.1. Giả sử $C \subseteq \mathbb{R}^n$ là một tập hợp khác rỗng.

- Tập C gọi là tập lồi nếu với mọi $x, y \in C$ và mọi $\lambda \in [0, 1]$ ta có

$$\lambda x + (1 - \lambda)y \in C$$

- Giả sử x^1, \dots, x^k là k điểm của C khi đó

$$x = \sum_{i=1}^k \lambda_i x^i, \quad \text{với } \lambda_i \geq 0, \sum_{i=1}^k \lambda_i = 1, \forall i = 1, \dots, k \quad (1.1.1)$$

gọi là một tổ hợp lồi của hệ vector $\{x^1, \dots, x^k\}$.

Cho $M \subseteq \mathbb{R}^n$, ta nói bao lồi của một tập hợp M là giao của tất cả các tập lồi chứa M , được kí hiệu là $\text{conv}(M)$. Rõ ràng bao lồi của M là tập lồi nhỏ nhất chứa M . Một tập C là lồi khi và chỉ khi nó chứa mọi tổ hợp lồi hữu hạn các điểm của C .

Một trong những tập lồi quan trọng (trong giải tích lồi, tối ưu (đặc biệt là quy hoạch tuyến tính)) là tập lồi đa diện. Tập lồi P được gọi là tập lồi đa diện nếu C có dạng

$$P := \{x \in \mathbb{R}^n \mid Ax \leq b\} \quad (1.1.2)$$

trong đó $A = (a_{ij})_{m \times n}, b \in \mathbb{R}^m$. Hay nói cách khác, P là giao của hữu hạn các nửa không gian đóng. Trường hợp đặc biệt của tập lồi đa diện là đa diện chính tắc có dạng như sau:

$$P_0 := \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\} \quad (1.1.3)$$

trong đó ma trận $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$ và giả thiết A có hạng đủ (tức là $\text{rank}A = m \leq n$).

Định nghĩa 1.1.2.

- Tập C được gọi là một nón nếu với mọi $x \in C$ và mọi $\lambda \geq 0$ ta có $\lambda x \in C$.
- Tập C được gọi là nón lồi nếu C vừa là nón vừa là tập lồi.

Bao nón lồi của tập M là giao của tất cả các nón lồi chứa M , ký hiệu là $\text{conv}(M)$ và nó cũng là tập nón lồi nhỏ nhất chứa C . Nói C gọi là nón lồi đa diện nếu C vừa là nón vừa là tập lồi đa diện. Mỗi nón C có một điểm $x = 0$ gọi là gốc (mũi) của nón C . Tập $r := \{x \in C : x = x^0 + \lambda d\}$ gọi là một tia xuất phát từ x^0 theo hướng $d \neq 0$. Một nón C sẽ chứa các tia xuất phát từ $x^0 = 0$.

Cho một tập lồi đa diện $P = \{x \in \mathbb{R}^n : Ax = b\}$ với $\text{rank}A = m \leq n$. Khi đó tập $K := \{d \mid Ad \geq 0\}$ là một nón lồi đa diện, gọi là nón lồi sinh ra bởi các hướng cực biên của P (như định nghĩa dưới đây).

Định nghĩa 1.1.3.

- Điểm x^0 là điểm cực biên (hai đỉnh) của tập lồi đa diện P nếu nó không là điểm trong của bất kỳ đoạn nào nối hai điểm khác nhau của P , tức là $\nexists x', x'' \in P$ và $x' \neq x''$ sao cho

$$x^0 = \alpha x' + (1 - \alpha)x''$$

với α nào đó thuộc $(0, 1)$.

- Mỗi tập con L của đa diện lồi P gọi là một cạnh nếu với mọi $x, y \in P, \lambda \in (0, 1)$ mà $\lambda x + (1 - \lambda)y \in L$ thì $[x, y] \subseteq L$ và L chứa trọn trong một đường thẳng. Một cạnh vô hạn gọi là một tia cực biên. Hướng của tia cực biên này gọi là hướng cực biên của P .

Chú ý rằng, tập lồi đa diện đóng, bị chặn thì không có hướng cực biên. Số đỉnh và số hướng cực biên của tập lồi đa diện P là hữu hạn.

1.1.2. Một số kết quả trong giải tích lồi

Phần này trình bày một số kết quả đã được đề cập trong giải tích lồi có liên quan đến khóa luận. Trước hết, cần nhắc lại một trong những định lý quan trọng là định lý biểu diễn tập lồi đa diện.

Định lý 1.1.1. Cho P là tập lồi đa diện trong \mathbb{R}^n với tập đỉnh là $V := \{v^i : i = 1, \dots, p\}$ và tập các hướng cực biên là $E := \{d^j : j = 1, \dots, q\}$. Khi đó mọi điểm $x \in P$ đều có thể biểu diễn dưới dạng:

$$x = \sum_{i=1}^p \lambda_i v^i + \sum_{j=1}^q \gamma_j d^j, \quad (1.1.4)$$

trong đó $\sum_{i=1}^p \lambda_i = 1, \lambda_i \geq 0, i = 1, \dots, p$ và $\gamma_j \geq 0, j = 1, \dots, q$.

Hiển nhiên nếu P là đa diện lồi bị chặn thì biểu diễn chỉ còn lại

$$x = \sum_{i=1}^p \lambda_i v^i.$$

Một tập lồi đa diện khác rỗng, không chứa trọn một đường thẳng thì luôn có điểm cực biên và số điểm cực biên là hữu hạn. Để kiểm tra một điểm x^0 có là điểm cực biên của tập lồi đa diện, ta có định lý sau.

Định lý 1.1.2. Giả sử P_0 là một tập lồi đa diện chính tắc dạng

$$P_0 := \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\} \quad (1.1.5)$$

trong đó ma trận $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ và giả thiết A có hạng đủ (tức là $\text{rank} A = m \leq n$). Điểm x^0 là một điểm cực biên (đỉnh) của P_0 nếu và chỉ nếu tập các véc tơ cột $B := \{A_j : j \in J_+(x^0)\}$ của ma trận A là độc lập tuyến tính, trong đó

$$J_+(x^0) := \{j \in \{1, \dots, n\} : x_j^0 > 0\}. \quad (1.1.6)$$

1.2. Bài toán quy hoạch tuyến tính gốc và đối ngẫu

1.2.1. Bài toán quy hoạch tuyến tính gốc và đối ngẫu.

Bài toán quy hoạch tuyến tính (QH TT) tổng quát có thể được phát biểu dưới dạng:

$$\begin{aligned} \min(\max)\{f(x) := \sum_{j=1}^n c_j x_j\} \\ \text{thỏa mãn: } D := \begin{cases} \sum_{j=1}^n a_{ij} x_j = b_i, & i = 1, \dots, m_1, \\ \sum_{j=1}^n a_{ij} x_j \leq b_i, & i = m_1 + 1, \dots, m_2, \\ \sum_{j=1}^n a_{ij} x_j \geq b_i, & i = m_2 + 1, \dots, m, \\ l_j \leq x_j \leq u_j, & j = 1, \dots, n. \end{cases} \end{aligned} \quad (1.2.7)$$

trong đó x_j gọi là các biến, c_j gọi là thành phần của véc tơ hệ số hàm mục tiêu (hàm giá), a_{ij} gọi là hệ số ràng buộc, b_i gọi là hệ số vế phải, $l_j < u_j$ lần lượt gọi là các cận dưới và cận trên (giới hạn dưới và trên) của biến x_j ($i = 1, \dots, m, j = 1, \dots, n$).

Để nghiên cứu tính chất và các phương pháp giải bài toán quy hoạch tuyến tính (1.2.7) người ta thường chuyển bài toán này về một trong hai dạng chính tắc và chuẩn tắc. Trong khóa luận này chỉ đề cập đến bài toán quy hoạch tuyến tính dạng chính tắc như sau:

$$\begin{aligned} \min\{f(x) := c^T x\} \\ \text{thỏa mãn: } D_p := \begin{cases} Ax = b \\ x \geq 0. \end{cases} \end{aligned} \quad (1.2.8)$$

trong đó $x = (x_1, \dots, x_n)^T$ gọi là các biến cần tối ưu, $c = (c_1, \dots, c_n)^T$ là véc tơ hàm mục tiêu, ma trận $A = (a_{ij})_{m \times n}$ là ma trận hệ số ràng buộc và $b = (b_1, \dots, b_m)^T$

gọi là véc tơ vế phải. Như thông thường, hàm f gọi là hàm mục tiêu, tập D_p gọi là tập ràng buộc. Ma trận A được giả thiết là có hạng đủ, $\text{rank}A = m \leq n$ và bài toán (1.2.8) gọi là bài toán QHTT gốc, ký hiệu là (P).

Một điểm $x \in D_p$ gọi là một điểm (hay phương án) chấp nhận được. Điểm $x^* \in D_p$ gọi là nghiệm (hay phương án tối ưu) của bài toán (1.2.8) nếu $f(x^*) \leq f(x)$ với mọi $x \in D_p$. Vì D_p là tập lồi đa diện, nên $x \in D_p$ là đỉnh của D_p thì x gọi là phương án cực biên (phương án cơ sở). Nếu x^* là điểm cực biên (đỉnh) của D_p và tối ưu thì x^* gọi là phương án cực biên tối ưu.

Cho một phương án cơ sở (hay một đỉnh) x , ký hiệu $J_+(x) := \{j \in \{1, \dots, n\} : x_j > 0\}$ gọi là tập chỉ số cơ sở của x . Nếu $|J_+(x)| = m$ thì x gọi là phương án không suy biến, còn nếu $|J_+(x)| < m$ thì x gọi là phương án suy biến. Theo định lý 1.1.2 thì tập hợp các véc tơ

$$B_+(x) := \{A_j, |; j \in J_+(x)\} \quad (1.2.9)$$

gồm các véc tơ cột của A sẽ là một hệ độc lập tuyến tính. Nếu $r := |J_+(x)| < m$ thì ta bổ sung thêm $m - r$ véc tơ còn lại của A vào $B_+(x)$ sao cho ta thu được hệ gồm m véc tơ cột của A độc lập tuyến tính, ký hiệu là $B(x)$. Hệ véc tơ $B(x)$ gọi là hệ véc tơ cơ sở (hay gọi tắt là cơ sở) của phương án cực biên x . Thông thường để ngắn gọn, người ta thường gọi $J(x)$ là tập chỉ số cơ sở thay cho gọi cơ sở $B(x)$.

Lập hàm Lagrange cho bài toán QHTT gốc (P) như sau:

$$L(x, y, s) := c^T x + y^T (b - Ax) + s^T x \quad (1.2.10)$$

trong đó y và $s \geq 0$ là các nhân tử Lagrange. Khi đó bài toán đối ngẫu (dạng Lagrange) của bài toán gốc (P) sẽ có dạng (sẽ ký hiệu là (D)):

$$\begin{aligned} & \max\{g(y, s) := b^T y\} \\ \text{thỏa mãn: } D_d := & \begin{cases} A^T y + s = c \\ s \geq 0. \end{cases} \end{aligned} \quad (1.2.11)$$

Biến s gọi là biến bù, g gọi là hàm mục tiêu đối ngẫu và D_d gọi là miền ràng buộc đối ngẫu. Hiển nhiên D_d cũng là tập lồi đa diện và bài toán (1.2.11) cũng là bài toán QHTT.

Với mọi bộ ba (x, y, s) sao cho $x \in D_p$ và $(y, s) \in D_d$ ta đặt

$$\tau(x, y, s) := f(x) - g(y, s) = s^T x, \quad (1.2.12)$$

gọi là khoảng trống đối ngẫu. Theo định lý đối ngẫu yếu thì $\tau(x, y, s) \geq 0$, và nếu $\tau(x, y, s) = 0$ thì (x, y, s) sẽ là nghiệm của cặp bài toán gốc đối ngẫu (P)-(D). Còn theo định lý đối ngẫu mạnh thì nếu x^* là nghiệm tối ưu của bài toán gốc (P) thì sẽ tồn tại nghiệm tối ưu (y^*, s^*) của bài toán đối ngẫu (D) sao cho $\tau^* = \tau(x^*, y^*, s^*) = 0$ và ngược lại.

1.2.2. Phương pháp đơn hình giải bài toán QHTT.

Một trong những phương pháp nổi tiếng và hiệu quả là phương pháp đơn hình, được G. B. Dantzig phát minh ra năm 1947. Phần này sẽ trình bày tóm tắt lại tư tưởng cơ bản và nội dung của phương pháp đơn hình mà sẽ sử dụng chính trong khóa luận.

Trước hết ta chỉ ra một tính chất quan trọng của bài toán quy hoạch tuyến tính là nghiệm sẽ nằm ở điểm cực biên.

Bổ đề 1.2.1. *Giả sử bài toán QHTT gốc (1.2.8) có nghiệm tối ưu thì nó sẽ có nghiệm tối ưu x^* nằm ở đỉnh.*

Do tính chất đặc biệt của bài toán QHTT nên thuật toán đơn hình đã tận dụng rất hiệu quả các tính chất này để tạo ra một thuật toán rất hiệu quả. Đặc biệt là các tính chất:

- Miền ràng buộc của bài toán QHTT là một tập lồi đa diện với số điểm cực biên là hữu hạn.
- Nếu bài toán QHTT có nghiệm tối ưu thì sẽ có nghiệm tối ưu nằm ở đỉnh.

Ý tưởng của thuật toán đơn hình mô tả như sau:

Bước 1: Xuất phát từ một đỉnh x^0 của miền ràng buộc.

Bước 2: Nếu x^0 là nghiệm tối ưu, dừng thuật toán. Nếu không chuyển sang Bước 3.

Bước 3: Từ x^0 tìm cách di chuyển đến đỉnh kế tiếp theo của miền ràng buộc tốt hơn đỉnh x^0 (theo nghĩa giá trị hàm mục tiêu nhỏ hơn).

Bước 4: Lặp lại Bước 2, 3 với x^0 thay bằng x^1 .

Do số đỉnh của miền ràng buộc là hữu hạn, nên nếu bài toán có nghiệm, sau hữu hạn bước ta sẽ tìm được đỉnh tối ưu. Có nhiều vấn đề cần giải quyết trong phương pháp đơn hình, bao gồm:

- Tìm đỉnh xuất phát, vấn đề này thường được giải quyết dựa vào phương pháp hai pha hoặc đánh thuế (hai phương pháp này cũng cho biết miền ràng buộc có rỗng hay không).
- Kiểm tra bài toán có nghiệm hay vô nghiệm (có bị chặn dưới hay không).
- Kiểm tra đỉnh x^0 có là tối ưu hay không?
- Từ đỉnh x^0 làm thế nào di chuyển đến đỉnh x^1 tốt hơn x^0 ?

Với ý tưởng như trên, thuật toán đơn hình được mô tả như sau:

Thuật toán đơn hình.

- **Đầu vào:** Ma trận $A = (a_{ij})_{m \times n}$, véc tơ b , véc tơ c . Phương án cơ sở x^0 và cơ sở tương ứng $J(x^0)$.
- **Đầu ra:** Phương án cơ sở tối ưu x^* và giá trị mục tiêu tối ưu $f(x^*)$ hoặc chỉ ra bài toán không có nghiệm tối ưu (tức là hàm mục tiêu không bị chặn dưới).
- **Thuật toán:**

Bước khởi tạo:

1. Tìm một phương án cơ sở xuất phát x^0 ứng với cơ sở xuất phát $J_0 := B(x^0)$.

2. Tính các hệ số khai triển $Z = (z_{jk})$ và các ước lượng Δ_k theo các công thức tương ứng sau

$$\begin{cases} A_k = \sum_{j \in J_0} z_{jk} A_j & \forall j = \overline{1, n} \\ \Delta_k = \sum_{j \in J_0} z_{jk} c_j - c_k & \forall k \notin J_0 \\ \Delta_k = 0 & \forall k \in J_0 \end{cases}$$

Bước 1: Kiểm tra tiêu chuẩn tối ưu.

1. Nếu $\Delta_k \leq 0$ với mọi $k \notin J_0$ thì x^0 là phương án tối ưu. Kết thúc thuật toán.
2. Nếu $\exists \Delta_k > 0$, chuyển sang bước 2.

Bước 2: Kiểm tra tính bị chặn của hàm mục tiêu.

Với mỗi $k \notin J_0$ mà $\Delta_k > 0$ ta kiểm tra các hệ số khai triển $Z_k = (z_{jk})$.

1. Nếu có một $\Delta_k > 0$ mà tất cả các hệ số khai triển $z_{jk} \leq 0$, ($\forall j \in J_0$) thì kết luận hàm mục tiêu không bị chặn dưới. Bài toán không có phương án hữu hạn. Kết thúc thuật toán.
2. Nếu với mọi $k \notin J_0$ mà tồn tại ít nhất một hệ số $z_{jk} > 0$ thì tiến hành tìm phương án mới x^1 tốt hơn x^0 bằng cách chuyển sang bước 3.

Bước 3: Tìm phương án mới

1. Chọn véc tơ A_s đưa vào cơ sở: Có thể bất kỳ $s \notin J$ sao cho $\Delta_s > 0$. Thông thường chọn s sao cho Δ_s lớn nhất

$$\Delta_s = \max\{\Delta_k > 0 \mid k \notin J_0\}$$

2. Chọn véc tơ A_r đưa ra khỏi cơ sở theo quy tắc:

$$\theta := \frac{x_r^0}{z_{rs}} = \min\left\{\frac{x_j^0}{z_{js}} \mid z_{js} > 0\right\}$$

3. Tính phương án mới x^1 và giá trị hàm mục tiêu mới theo công thức:

$$x^1 = \begin{cases} 0 & \text{với } k \notin J_0, k \neq s \\ \frac{x_r^0}{z_{rs}} & \text{với } k = s \\ x_j^0 - \frac{x_r^0}{z_{rs}} z_{js} & \text{với } j \in J_0 \end{cases}$$

$$f(x^1) = f(x^0) - \frac{x_r^0}{z_{rs}} \Delta_s$$

Bước 4: Tính các hệ số khai triển và ước lượng mới theo công thức

$$z_{jk}^1 = \begin{cases} \frac{z_{rk}}{z_{rs}} & \text{nếu } j = s \\ z_{jk} - \frac{z_{rk}}{z_{rs}} z_{js} & \text{nếu } j \in J_0, j \neq s \end{cases}$$

$$\Delta_k^1 = \Delta_k - \frac{z_{rk}}{z_{rs}} \Delta_s$$

Bước 5: Quay về bước 1 với phương án cơ sở mới x^1 và cơ sở mới $J_1 := J(x^1)$.

Để thuận tiện cho việc "thực hành" thuật toán đơn hình giải bài toán QHTT dạng chuẩn tắc. Ta sử dụng một bảng gọi là **Bảng đơn hình** gồm $n + 3$ cột và $m + 3$ hàng như sau:

Cơ sở	c_J	Phương án	1	2	\dots	k	\dots	n
J		x_J	c_1	c_2	\dots	c_k	\dots	c_n
J_1	c_{j_1}	x_{j_1}	$z_{j_1 1}$	$z_{j_1 2}$	\dots	$z_{j_1 k}$	\dots	$z_{j_1 n}$
J_2	c_{j_2}	x_{j_2}	$z_{j_2 1}$	$z_{j_2 2}$	\dots	$z_{j_2 k}$	\dots	$z_{j_2 n}$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
J_m	c_{j_m}	x_{j_m}	$z_{j_m 1}$	$z_{j_m 2}$	\dots	$z_{j_m k}$	\dots	$z_{j_m n}$
		$f(x)$	Δ_1	Δ_2	\dots	Δ_k	\dots	Δ_n

Lưu ý phần bảng dành cho các hệ số khai triển z_{jk} :

- Các cột ứng với các $j \in J_0$ sẽ là các véc tơ đơn vị với hệ số 1 nằm trên dòng với chỉ số j .
- Với $k \notin J_0$ thì cột k của bảng đơn hình là hệ số khai triển của A_k của ma trận A theo cơ sở J_0 . Ta ký hiệu cột này là Z_k nghĩa là $A_k = B_{J_0} Z_k$ hay $Z_k = (B_{J_0})^{-1} A_k$, ở đây B_{J_0} là ma trận cơ sở ($B_{J_0} = \{A_j \mid j \in J_0\}$).

Đặc biệt khi ta chọn được một ma trận B_{J_0} có dạng ma trận đơn vị thì hệ số khai triển trên các cột j với $j \in J_0$ sẽ chính là cột véc tơ đơn vị $z^{jk} = e_j$, còn các hệ số khai triển trên các cột $k \notin J_0$ chính là $z_{jk} = A_k$.

- Dòng ước lượng là dòng cuối cùng của bảng và được tính bởi $\Delta_k = c_{J_0}^T Z_k - c_k$ và $\Delta_j = 0, (\forall j \in J_0)$.
- Giá trị hàm mục tiêu chính là $f(x) = c_{J_0}^T x_{J_0}$.

Thuật toán nêu trên gọi là thuật toán đơn hình gốc, ngoài ra người ta còn xây dựng các thuật toán đơn hình khác như: thuật toán đơn hình đối ngẫu, thuật toán đơn hình gốc đối ngẫu. Một số phiên bản khác của thuật toán đơn hình cũng được đề cập trong các tài liệu về QHTT.

Một trong những chi tiết quan trọng trong phương pháp đơn hình là: *Từ nghiệm x^* của bài toán gốc (P), ta có xây dựng lại được nghiệm đối ngẫu hay không?*. Phương pháp đơn hình gốc - đối ngẫu sẽ cho phép thu được bộ ba nghiệm (x^*, y^*, s^*) cho cặp bài toán gốc, đối ngẫu. Trên thực tế, ta có thể xuất phát từ nghiệm của bài toán gốc (P) là x^* với cơ sở A_{J^*} , ta có thể thu được nghiệm của bài toán đối ngẫu (D) như sau:

- Giả sử x^* là nghiệm của bài toán gốc (P) ứng với cơ sở tối ưu A_{J^*} . Khi đó ta có:

$$x_{J^*}^* = A_{J^*}^{-1} b,$$

với $x_{J^*}^* = (x_j^*)_{j \in J^*}$ và $(x_j^*)_{j \in J^*} = (0)$.

- Khi đó A_{J^*} cũng sẽ là cơ sở đối ngẫu của bài toán đối ngẫu (D) và nghiệm đối ngẫu (y^*, s^*) được tính theo công thức:

$$y^* = (A_{J^*}^{-1})^T c_{J^*}, \quad s^* = c - A^T y^*,$$

trong đó $c_{J^*} = (c_j)_{j \in J^*}$.

- Khi đó khoảng trống đối ngẫu sẽ là: $\tau^* = (x^*)^T s^* = 0$. Theo định lý về độ lệch bù thì nếu $x_j^* > 0$ thì $s_j^* = 0$, do đó dễ thấy $s_{J^*}^* = 0$.

Trong thực tế, bài toán QHTT thường không phải là bài toán dạng chuẩn tắc với véc tơ vế phải b không âm, nên ta không thể có ngay được phương án xuất phát x^0 để thực hiện phương pháp đơn hình. Do vậy người ta thường sử dụng một trong hai phương pháp: Phương pháp đơn hình hai pha và phương pháp đánh thuế. Trong khóa luận này sẽ sử dụng phương pháp đơn hình hai pha với nội dung được tóm tắt như sau:

Trước hết, đối với bài toán gốc (P), không mất tính tổng quát ta có thể xem các $b_i \geq 0$ với mọi $i = \overline{1, m}$. Nếu trái lại ta nhân hai vế của ràng buộc thứ i với -1 . Ta lập bài toán phụ sau

$$\begin{aligned} \min\{f_a(u) := \sum_{j=1}^m u_{n+j}\} & \quad (1.2.13) \\ \text{thoả mãn } D_a := & \begin{cases} \sum_{j=1}^n a_{ij}x_j + u_{n+i} = b_i, & (i = \overline{1, m}) \\ x_j \geq 0, & (j = \overline{1, n}) \\ u_{n+i} \geq 0, & (i = \overline{1, m}) \end{cases} \end{aligned}$$

Các biến u_{n+i} với $(i = \overline{1, m})$ gọi là các biến giả.

Nếu ta ký hiệu $e = (1, 1, \dots, 1)^T$ là véc tơ gồm m thành phần là 1, $u = (u_{n+1}, u_{n+2}, \dots, u_{n+m})^T$ và E là ma trận đơn vị cấp m thì ta có thể viết bài toán trên dưới dạng

$$\begin{aligned} \min\{f_a(u) := e^T u\} & \quad (1.2.14) \\ \text{thoả mãn } D_a := & \begin{cases} Ax + Eu = b \\ x \geq 0, u \geq 0 \end{cases} \end{aligned}$$

Khi đó, quan hệ giữa hai bài toán (1.2.13) và (1.2.14) được chỉ ra như sau:

- Bài toán (1.2.14) có một phương án cơ sở xuất phát là $(x, u)^T := (0, b)^T$.
- Bài toán (1.2.8) có phương án chấp nhận được khi và chỉ khi bài toán phụ (1.2.14) có phương án tối ưu $(\bar{x}, \bar{u})^T$ với tất cả các biến giả $\bar{u}_{n+i} = 0$, $(i = \overline{1, m})$.

Do đó phương pháp đơn hình hai pha được thực hiện như sau:

Pha 1: Lập bài toán phụ cho bài toán (P), giải bài toán phụ bằng phương pháp đơn hình. Nếu bài toán phụ vô nghiệm hoặc có nghiệm không là nghiệm chấp nhận của bài toán (P), dừng thuật toán. Ngược lại, chuyển sang pha 2.

Pha 2: Sử dụng thuật toán đơn hình giải bài toán (P) với phương án xuất phát thu được từ pha 1.

1.3. Bài toán QHTT kích thước lớn

1.3.1. Ma trận thưa và vấn đề lưu trữ ma trận thưa kích thước lớn

. Các bài toán QHTT trong thực tế ứng dụng thường có kích thước lớn. Bài toán QHTT có thể xuất hiện từ các lĩnh vực ứng dụng trực tiếp như: giao thông vận tải, xây dựng kế hoạch sản xuất, chế biến, quản lý nhân lực... Ngoài ra bài toán QHTT có thể xuất hiện trong các phương pháp toán học và tin học như một bài toán phụ, chẳng hạn xuất hiện trong các phương pháp tuyến tính hóa của quy hoạch phi tuyến, trong các phương pháp nhánh và cận... Các bài toán xuất hiện như thế thường có kích thước lớn, thậm chí là rất lớn. Tuy nhiên có một thuận lợi ở đây là ma trận ràng buộc A của các bài toán này thường có cấu trúc đặc biệt và thưa. Các ma trận thưa dạng đặc biệt như ma trận đường chéo, ma trận vết, ma trận tam giác, ma trận khối, chéo khối... Việc xử lý các ma trận này đòi hỏi phải có các kỹ thuật riêng, vì nếu xử lý như thông thường sẽ gây ra lãng phí bộ nhớ máy tính và tăng thời gian tính toán, thậm chí là không khả thi trên thực tế. Chẳng hạn, với ma trận thực (6 byte) A có kích thước $m \times n$ với $m = 10^6$ và $n = 10^8$ thì nếu lưu trữ như thông thường ta cần tới $Q = 6 \times 6 \times m \times n = 6 \times 10^{14} \text{ byte} \approx 558793 \text{Gb}$. Trong khi đó ma trận A có thể chỉ có cỡ $m + n$ phần tử khác không chẳng hạn, thì ta chỉ cần $6 \times (m + n) \approx 578 \text{Mb}$. Do đó để xử lý các bài toán kích thước lớn, người ta phải sử dụng kỹ thuật nén ma trận. Thông thường các ma trận có số phần tử lớn hơn 10^6 có thể xem là ma trận kích thước lớn.

Định nghĩa 1.3.1. *Ma trận thưa là dạng ma trận có chứa nhiều phần tử bằng 0.*

Bao nhiêu phần tử 0 gọi là nhiều và được coi là ma trận thưa. Để định lượng, đối với ma trận $A = (a_{ij})_{m \times n}$, ta gọi

$$d = \frac{nz}{m \times n} \times 100,$$

trong đó nz là số phần tử khác không của ma trận A . Số d gọi là mật độ của ma trận A . Thông thường, người ta có thể xem các ma trận có mật độ dưới 50 là các ma trận thưa. Trên thực tế, các bài toán ứng dụng thường có ma trận mà mật độ $d < 50$ (tức là có tới 50% phần tử bằng 0).

Như vậy để lưu trữ ma trận thưa, người ta chỉ lưu các phần tử khác không, số ô nhớ cần để lưu trữ chỉ cần khoảng nz phần tử. Tùy vào cấu trúc của ma trận, người ta lựa chọn các cách lưu trữ, nén ma trận khác nhau. Đối với ma trận thưa bất kỳ có ba cách lưu trữ cơ bản như sau:

1. Nén theo hàng,
2. Nén theo cột,
3. Nén theo khối (block).

1. Nén theo hàng. Để nén theo hàng, người ta sử dụng cấu trúc dữ liệu gồm 3 thành phần (**val**, **col_ind**, **row_ptr**). Trong đó **val** dùng để lưu trữ giá trị của các phần tử khác không, **col_ind** lưu trữ chỉ số cột còn **row_ptr** là con trỏ chỉ hàng.

2. Nén theo cột. Để nén theo cột, người ta sử dụng cấu trúc dữ liệu gồm 3 thành phần (**val**, **row_ind**, **col_ptr**). Trong đó **val** dùng để lưu trữ giá trị của các phần tử khác không, **row_ind** lưu trữ chỉ số hàng còn **col_ptr** là con trỏ chỉ cột.

3. Nén theo khối. Đối với các ma trận có các khối dạng đặc biệt, chẳng hạn như dải theo hàng hoặc theo cột, thì người ta sửa đổi hai Để nén theo hàng, người ta sử dụng cấu trúc dữ liệu gồm 3 thành phần (**val**, **col_ind**, **row_ptr**). Trong đó **val** dùng để lưu trữ giá trị của phương pháp 1 và 2 để nén các dạng ma trận này. Thay vì **val** chứa một phần tử thì nó có thể chứa một khối.

Phương pháp lưu trữ nén theo hàng và cột mô tả chi tiết như sau:

Lưu trữ ma trận bởi mảng

Ta lưu trữ các ma trận bằng 3 mảng một chiều :

1. mảng **val()** : lưu trữ các giá trị khác 0 của ma trận.
2. mảng **col()** : lưu trữ chỉ số cột của các giá trị tương ứng.

3. mảng `row()` : lưu trữ chỉ số hàng của các giá trị tương ứng.

Minh họa ta xét ma trận:

$$A = \begin{pmatrix} 1 & 2 & 0 & 0 & 3 \\ 4 & 5 & 6 & 0 & 0 \\ 0 & 7 & 8 & 0 & 9 \\ 0 & 0 & 0 & 10 & 0 \\ 11 & 0 & 0 & 0 & 12 \end{pmatrix}$$

val()	1	2	3	4	5	6	7	8	9	10	11	12
row()	0	0	0	1	1	1	2	2	2	3	4	4
col()	0	1	4	0	1	2	1	2	4	3	0	4

a. Nén ma trận theo hàng

Từ cách lưu trữ ma trận trên ta nhận thấy ở mảng `row()` có nhiều phần tử giống nhau và chỉ hơn kém nhau một đơn vị từ đó ta có cách nén làm giảm kích cỡ của mảng `row()` bằng cách chỉ lưu các giá trị thay đổi ở mảng `row()` và lưu vào một mảng `row-ptr()` với chú ý các giá trị ở mảng `row()` đã được sắp tăng dần thành các khối giá trị các khối này hơn kém nhau 1 đơn vị .Bằng cách giữ nguyên hai mảng `val()` và `col()` tạo ra một mảng mới `row-ptr()` có kích cỡ nhỏ hơn mảng `row()` ban đầu ta được một kiểu lưu trữ mới.

Minh họa bởi ma trận A :

row-ptr()	0	3	6	9	10	12						
val()	1	2	3	4	5	6	7	8	9	10	11	12
col()	0	1	4	0	1	2	1	2	4	3	0	4

b. Nén ma trận theo cột

Tương tự với phương pháp nén ma trận theo hàng ta thao tác với mảng `col()` giữ lại mảng `val()` và mảng `row()` tạo ra một mảng mới từ mảng `col()` là `col-ptr()` ta có một cách lưu trữ ma trận khác

col-ptr()	0	3	6	8	9	12						
val()	1	4	11	2	5	7	6	8	10	3	9	12
col()	0	1	4	0	1	2	1	2	3	0	2	4

1.3.2. Bài toán quy hoạch kích thước lớn và có cấu trúc.

Bài toán QHTT kích thước lớn là một trong những thách thức đối với những người làm ứng dụng và xử lý tính toán. Tuy nhiên hiện nay có rất nhiều phương pháp để xử lý các bài toán dạng này, chẳng hạn như: phương pháp đối ngẫu, phương pháp giảm số chiều, phương pháp phân rã. Một trong những phương pháp hiệu quả và tiện lợi nhất là phương pháp phân rã (decomposition method). Đối với một số bài toán có cấu trúc dạng đặc biệt, việc sử dụng phương pháp phân rã giúp ta chuyển các bài toán kích thước lớn về các bài toán kích thước nhỏ hơn. Trên thực tế, cấu trúc của các bài toán rất khác nhau, tùy thuộc vào bản chất của bài toán hoặc tùy thuộc vào cách chuyển về mô hình QHTT. Khóa luận này chỉ đề cập đến một số mô hình cụ thể dưới đây:

a. Mô hình chéo khối. Trường hợp đơn giản nhất đề cập ở đây là trường hợp cấu trúc khối có dạng như sau:

$$\begin{aligned} \min \left\{ \begin{array}{l} f(x) := \sum_{j=1}^{n_1} c_j x_j + \sum_{j=n_1+1}^n c_j x_j = z \\ \sum_{j=1}^{n_1} A_{ij} x_j = b_i, \quad i = 1, \dots, m_1 \\ \sum_{j=n_1+1}^n A_{ij} x_j = b_i, \quad i = m_1 + 1, \dots, m \\ x_j \geq 0, \quad j = 1, \dots, n. \end{array} \right. \end{aligned} \quad (1.3.15)$$

Không có liên quan nào giữa hai khối ma trận trong bài toán quy hoạch tuyến tính 1.3.15 vì vậy ta có thể giải bài toán 1.3.15 bằng cách giải hai bài toán quy hoạch tuyến tính riêng biệt.

Giả sử có thể chia bài toán ban đầu thành k khối các khối này độc lập với nhau thì độ phức tạp của bài toán sẽ giảm đi $\frac{1}{k^2}$ so với bài toán ban đầu.

b. Mô hình chéo khối không hoàn toàn. Hệ khối góc (block - angular system) 1.3.16 là một dạng của 1.3.15, nó có k khối độc lập và một tập các ràng buộc

liên quan. Tuy nhiên có một dải ràng buộc liên quan trên cùng.

$$\begin{aligned}
 \min\{ & (c^0)^T x^0 + (c^1)^T x^1 + \dots + (c^k)^T x^k = z\} \\
 & A^0 x^0 + A^1 x^1 + \dots + A^k x^k = b \\
 & F^1 x^1 = f^1 \\
 & \dots \\
 & F^k x^k = f^k \\
 & x^i \geq 0, \quad i = 0, 1, 2, \dots, k.
 \end{aligned} \tag{1.3.16}$$

Một ứng dụng của hệ khối góc ví dụ một công ty với K nhà máy độc lập, $k = 1, 2, \dots, K$. Mỗi nhà máy có một số ràng buộc mà ràng buộc này độc lập với cá ràng buộc của các nhà máy khác. Nhưng các nhà máy này có chung ngân quỹ và chung một hàm mục tiêu. Trong 1.3.16, x^k là véc tơ thể hiện mức chi phí của nhà máy thứ k . x^0 thể hiện mức chi phí của cơ quan điều hành, nó không thể hiện hoạt động của nhà máy cụ thể nào. Phương trình đầu tiên là hàm mục tiêu, dòng thứ hai là m ràng buộc biểu diễn sự chia sẻ tài nguyên chung của các nhà máy, dòng thứ ba gồm m_1 ràng buộc chỉ liên quan đến nhà máy thứ nhất, dòng cuối cùng là m_k ràng buộc chỉ liên quan đến nhà máy thứ k . Cách làm phổ biến của các nhà kinh tế là ban đầu gán bất kì giá cho các tài nguyên và tối ưu hóa hoạt động của các nhà máy tùy theo giá của các tài nguyên nó sử dụng khi hoạt động. Tổng nhu cầu về tài nguyên mà cơ quan điều hành và các nhà máy sử dụng bằng b . Với các tài nguyên đang có vấn đề trở thành tìm một thuật toán để điều chỉnh chi phí một cách hợp lý. Trong phần này chúng ta sẽ trình bày bằng cách nào để làm được điều này thông qua một số hữu hạn bước lặp của phép phân rã Dantzig Wolfe.

c. Mô hình bậc thang. Trên thực tế phép phân tích còn được sử dụng xử lý hệ bậc thang (staircase) khác với hệ 1.3.16, ở hệ bậc thang các bước trước phải sử dụng cùng một số tài nguyên vào hoặc ra của bước sau. Ví dụ hệ 1.3.17 là hệ bậc thang

với 4 bước:

$$\begin{aligned}
 \min\{ & (c^0)^T x^0 + (c^1)^T x^1 + (c^2)^T x^2 + (c^3)^T x^3 + (c^4)^T x^4 = z\} \\
 & A^{11} x^1 = b^1 \\
 & A^{21} x^1 + A^{22} x^2 = b^2 \\
 & A^{32} x^2 + A^{33} x^3 = b^3 \\
 & A^{43} x^3 + A^{44} x^4 = b^4 \\
 & x^k \geq 0, \quad k = 1, \dots, 4.
 \end{aligned} \tag{1.3.17}$$

Hệ bậc thang thường sử dụng trong quá trình xuyên thời gian mà các hoạt động của mỗi giai đoạn trực tiếp ảnh hưởng hoặc bị ảnh hưởng bởi giai đoạn trước và sau nó mà không bị ảnh hưởng bởi các giai đoạn khác. Hệ thống đó xuất hiện trong sản xuất mà sản phẩm ở mỗi giai đoạn bị ảnh hưởng bởi giai đoạn trước và nó thì ảnh hưởng tới giai đoạn sau.

Trong một số bài toán các khối ma trận con A^{ii} dọc đường chéo chính hoặc đường chéo phụ thường có thể giống nhau. Nếu điều đó xảy ra thì sẽ có rất nhiều tiện lợi.

d. Mô hình tam giác. Một dạng phổ biến khác của hệ này mà có thể xử lý bằng phương pháp phân rã Dantzig Wolfe là hệ khối tam giác dưới (lower block triangular).

$$\begin{aligned}
 \min\{ & (c^0)^T x^0 + (c^1)^T x^1 + (c^2)^T x^2 + (c^3)^T x^3 + (c^4)^T x^4 = z\} \\
 & A^{11} x^1 = b^1 \\
 & A^{21} x^1 + A^{22} x^2 = b^2 \\
 & A^{31} x^1 + A^{32} x^2 + A^{33} x^3 = b^3 \\
 & A^{41} x^1 + A^{42} x^2 + A^{43} x^3 + A^{44} x^4 = b^4 \\
 & x^k \geq 0, \quad k = 1, \dots, 4.
 \end{aligned} \tag{1.3.18}$$

Ngoài những mô hình cụ thể trên, ta có thể xét các mô hình khác như: dạng đường chéo, ma trận các đường chéo, ma trận hình sao, ma trận dải...

Chương 2

Phương pháp phân rã

Dantzig-Wolfe giải bài toán kích thước lớn

Phương pháp phân rã Dantzig - Wolfe là sự kết hợp giữa ý tưởng về việc giải quyết một bài toán QHTT tổng quát do Philip Wolfe đề xuất và phương pháp phân rã của Dantzig. Do vậy, để xem xét tận gốc của vấn đề, trước hết cần trình bày lại bài toán QHTT tổng quát của Wolfe.

2.1. Bài toán quy hoạch tuyến tính Wolfe tổng quát

2.1.1. Phát biểu bài toán và các tính chất

Xuất phát từ những bài toán ứng dụng thực tế, các hệ số A, b và c của bài toán QHTT (2.1.1) thường thay đổi trong một tập nào đó. Điều này cho phép việc ứng dụng vào thực tế trở nên mềm dẻo hơn, phù hợp với ứng dụng thực tế hơn. Khi các hệ số A, b, c thay đổi trong các tập lồi, bài toán này trở thành bài toán QHTT tổng quát được Philip Wolfe đề xuất.

Bài toán QHTT Wolfe tổng quát được phát biểu như sau:

$$\begin{aligned} \min\{f(x) = c^T x\} & \quad (2.1.1) \\ \text{thỏa mãn } D := & \begin{cases} Ax = b \\ x \geq 0 \end{cases} \end{aligned}$$

trong đó các hệ số $\begin{pmatrix} c_j \\ A_j \end{pmatrix} \in C_j$ với mọi $j = 1, \dots, n$. Ở đây, C_j là các tập lồi nào đó trong \mathbb{R}^{m+1} và A_j là véc tơ cột thứ j của ma trận A . Tổng quát hơn, người ta cũng có thể xem xét bài toán (2.1.1) với hệ số b_i thay đổi, tức $b \in C_b$ nào đó với C_b là tập lồi trong \mathbb{R}^m .

Vì các véc tơ $\begin{pmatrix} c_j \\ A_j \end{pmatrix}$ chọn tùy ý trong các tập C_j , nên theo định lý biểu diễn tập lồi, với mỗi j , véc tơ $\begin{pmatrix} c_j \\ A_j \end{pmatrix}$ có thể biểu diễn qua tập đỉnh và tập hướng cực biên của C_j . Trước hết ta sẽ chứng minh một định lý tổng quát cho bài toán Wolfe tổng quát.

Định lý 2.1.1. *Bài toán QHTT Wolfe tổng quát (2.1.1) là tương đương với bài toán QHTT Wolfe tổng quát sau:*

$$\begin{aligned} \min\{\hat{f}(x, x^t) := \sum_{j=1}^n (c_j x_j + \sum_{t=1}^{T_j} c_j^t x_j^t)\} & \quad (2.1.2) \\ \text{thỏa mãn } D := & \begin{cases} \sum_{j=1}^n (a_{ij} x_j + \sum_{t=1}^{T_j} a_{ij}^t x_j^t) = b_i, \\ x_j \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n. \end{cases} \end{aligned}$$

trong đó $\begin{pmatrix} c_j \\ A_j \end{pmatrix}$ chọn tùy ý trong C_j và $\begin{pmatrix} c_j^t \\ A_j^t \end{pmatrix}$ là T_j các điểm cố định trong C_j .

Chú ý rằng bài toán (2.1.2) có số biến nhiều hơn bài toán (2.1.1). Do vậy tính tương đương ở đây thể hiện theo nghĩa, tập nghiệm của hai bài toán có thể suy ra nhau và giá trị hàm mục tiêu là bằng nhau.

Chứng minh. Giả sử $x_j = x_j^*$ và $\begin{pmatrix} c_j \\ A_j \end{pmatrix} = \begin{pmatrix} c_j^* \\ A_j^* \end{pmatrix}$ với $j = 1, \dots, n$ là nghiệm tối ưu của bài toán (2.1.1) với giá trị $f(x^*) = f^*$ và giả sử $x_j = \hat{x}_j, x_j^t = \hat{x}_j^t$ và $\begin{pmatrix} c_j \\ A_j \end{pmatrix} = \begin{pmatrix} \hat{c}_j \\ \hat{A}_j \end{pmatrix}, \begin{pmatrix} c_j^t \\ A_j^t \end{pmatrix} = \begin{pmatrix} \hat{c}_j^t \\ \hat{A}_j^t \end{pmatrix}$, với $j = 1, \dots, n$ là nghiệm tối ưu của bài toán (2.1.2) với giá trị hàm mục tiêu tối ưu $\hat{f}(\hat{x}) = \hat{f}$.

Dễ thấy $x_j = x_j^*, x_j^t = 0$ là nghiệm chấp nhận của bài toán (2.1.2) nên ta có $f^* \leq \hat{f}$. Mặt khác, nghiệm tối ưu của bài toán (2.1.2) có thể viết lại như là nghiệm chấp nhận

của bài toán (2.1.1) dưới dạng:

$$\min\left\{\sum_{j=1}^n \bar{c}_j \bar{x}_j\right\}$$

$$\text{thỏa mãn } \begin{cases} \sum_{j=1}^n \bar{a}_{ij} \bar{x}_j = b_i, & i = 1, \dots, m, \\ \bar{x}_j \geq 0, & j = 1, \dots, n, \end{cases}$$

trong đó

$$\bar{x}_j = \hat{x}_j + \sum_{t=1}^{T_j} \bar{x}_j^t,$$

$$\begin{pmatrix} \bar{c}_j \\ \bar{A}_j \end{pmatrix} = \begin{cases} \begin{pmatrix} \hat{c}_j \\ \hat{A}_j \end{pmatrix} \frac{\hat{x}_j}{\bar{x}_j} + \begin{pmatrix} \hat{c}_j^t \\ \hat{A}_j^t \end{pmatrix} \frac{\bar{x}_j^t}{\bar{x}_j} & \text{với } \bar{x}_j \neq 0, \\ \begin{pmatrix} \hat{c}_j \\ \hat{A}_j \end{pmatrix} & \text{với } \bar{x}_j = 0. \end{cases}$$

Điều này có nghĩa là $\bar{f} \leq f^*$. Do vậy $\bar{f} = f^*$. \square

Bây giờ để đơn giản việc nghiên cứu, ta hạn chế một số các hệ số $\begin{pmatrix} c_j \\ A_j \end{pmatrix}$ thay đổi trong các tập C_j một số các hệ số khác cố định, trong trường hợp này $|C_j| = 1$ (tập hợp chỉ gồm 1 điểm). Giả sử ta xét bài toán QHTT tổng quát có dạng:

$$\begin{aligned} \min\{ & c^T x + y_{0,n+1} x_{n+1} = z \\ & Ax + y_{\bullet,n+1} x_{n+1} = b, \\ & (x, x_{n+1}) = (x_1, x_2, \dots, x_{n+1}) \geq 0. \end{aligned} \quad (2.1.3)$$

trong đó $A = (a_{ij})_{m \times n}$ và các hệ số $\begin{pmatrix} y_{0,n+1} \\ y_{\bullet,n+1} \end{pmatrix} \in C_{n+1}$ là tập đơn hình có dạng:

$$C_{n+1} = \left\{ y_{i,n+1} \mid \sum_{i=0}^m \alpha_i y_{i,n+1} = 1, y_{i,n+1} \geq 0, \text{ với } i = 0, \dots, m \right\} \quad (2.1.4)$$

Khi đó sử dụng định lý biểu diễn tập lồi, ta sẽ biểu diễn mỗi điểm $u \in C_{n+1}$ qua tập đỉnh của nó

$$(u_0, u) = \sum_{i=0}^m \alpha_i y_{i,n+1}. \quad (2.1.5)$$

Sau đó thay vào bài toán (2.1.3) ta sẽ thu được một bài toán QHTT với $n + m + 1$ biến $x, \alpha_0, \dots, \alpha_m$. Khi đó ta có định lý sau:

Định lý 2.1.2. Bài toán QHTT tổng quát dạng (2.1.3) là tương đương với bài toán QHTT thông thường

$$\begin{aligned} \min\{ & c^T x + u_0 & & = w \} \\ & Ax + u & & = b \\ & \sum_{i=0}^m \alpha_i u_{i,n+1} & & = x_{n+1} \\ & x_1, x_2, \dots, x_{n+1} \geq 0, & & u_0, u_1, \dots, u_m \geq 0 \end{aligned} \quad (2.1.6)$$

trong đó $A = (a_{ij})_{m \times n}$ và $u_i = y_{i,n+1} x_{n+1}$ với giả thiết rằng phương án tối ưu cho (2.1.6) có $x_{n+1} = x_{n+1}^* > 0$

Chứng minh. Ta sẽ chứng minh mỗi phương án tối ưu của (2.1.3) tìm được một phương án tối ưu của (2.1.6) và ngược lại. Trước hết ta sẽ chỉ ra phương án tối ưu của (2.1.3) là chấp nhận được cho (2.1.6). Giả sử x^* , $y_{i,n+1}^*$ và z^* là tối ưu cho bài toán (1.2.12). Đặt $z = z^*$, $x_j = x_j^*$, $j = 1, \dots, n+1$ với $x_{n+1}^* \geq 0$, và $y_{i,n+1} = y_{i,n+1}^*$, $i = 0, \dots, m$. Khi đó nó là một phương án tối ưu chấp nhận được cho (2.1.6). Thật vậy, giá trị $x_j = x_j^*$, $j = 1, \dots, n+1$ và $u_i = y_{i,n+1}^* x_{n+1}^*$, $i = 0, \dots, m+1$, rõ ràng là chấp nhận được cho (2.1.3) nghĩa là:

$$\begin{aligned} \min\{ & c^T x^* + y_{0,n+1}^* x_{n+1}^* & & = w^* \} \\ & Ax^* + y_{\bullet,n+1}^* x_{n+1}^* & & = b, \\ & \sum_{i=0}^m \alpha_i y_{i,n+1}^* x_{n+1}^* & & = x_{n+1}^*. \end{aligned} \quad (2.1.7)$$

Do đó ta có $z^* \geq w^*$. Tiếp theo giả sử rằng w^* không tối ưu nhưng tồn tại tập các giá trị $w = \bar{w}$, $x_j = \bar{x}_j$ $j = 1, \dots, m+1$ và $u_i = \bar{u}_i$, $i = 0, \dots, m+1$ tập này là tối ưu chấp nhận được cho (2.1.6). Vì $\bar{x}_{n+1} > 0$ theo giả thiết, ta tính $\bar{y}_{i,n+1} = \bar{u}_{n+1} / \bar{x}_{n+1}$ với $i = 0, \dots, m$. Phương án $x_j = \bar{x}_j$, $j = 1, \dots, n+1$ với $y_{i,n+1} = \bar{y}_{i,n+1}$, $i = 0, \dots, m$ rõ ràng là chấp nhận được cho (2.1.3) và vì vậy $\bar{w} \leq z^*$. Do đó phương án tối ưu hai bài toán là tương đương. \square

Tiếp theo, để minh họa cho phương pháp Wolfe giải bài toán QHTT dạng (2.1.3), ta xem xét ví dụ sau:

Ví dụ 2.1.1. Xét bài toán xác định bởi hệ phương trình (2.1.8)

$$\begin{aligned} \min \quad & \{6x_1 \quad +4x_2 \quad +x_3 \quad +y_{04}x_4 \quad := z\} \\ & x_1 \quad +x_2 \quad -4x_3 \quad +y_{14}x_4 = 5 \\ & -x_1 \quad +x_2 \quad -x_3 \quad +y_{24}x_4 = 1 \\ & x_1 \geq 0 \quad x_2 \geq 0 \quad x_3 \geq 0 \quad x_4 \geq 0 \end{aligned} \quad (2.1.8)$$

với các hệ số $y_{\bullet 4} = (y_{04}, y_{14}, y_{24})$ với $y_{i4} \geq 0$ không cố định được chọn tùy ý trong đơn hình

$$C_4 = \{y_{\bullet 4} \mid 3y_{04} + y_{14} + 2y_{24} = 2 \text{ với } y_{i4} \geq 0 \quad i = 0, 1, 2\} \quad (2.1.9)$$

và các hệ số còn lại cố định.

Giả sử ta có một cơ sở xuất phát của các biến $(-z, x_1, x_2)$, phương án cơ sở chấp nhận được cho dưới dạng:

$$(-z) = -24, \quad x_1 = 2, \quad x_2 = 3, \quad x_3 = x_4 = 0.$$

Bây giờ sử dụng thuật toán đơn hình để kiểm tra xem phương án đã cho có tối ưu không, ta tìm các nhân tử Lagrange, tức các biến đối ngẫu π ứng với phương án đã cho bằng việc giải phương trình $B^T \pi = c_B$, tức $\pi = (B^{-1})^T c_j$ với B là cơ sở ứng với phương án đã cho ở trên. Giải ra ta có $\pi = (5, -1)^T$ và tính các ước lượng \bar{c}_3 và \bar{c}_4 với $\bar{c}_j = c_j - \pi^T A_{\bullet j}$ ta thu được:

$$\bar{c}_3 = 18, \quad \bar{c}_4 = y_{04} - 5y_{14} + y_{24}$$

trong đó $(y_{04}, y_{14}, y_{24}) \in C_4$. Theo thuật toán đơn hình, nếu các ước lượng $\bar{c}_3, \bar{c}_4 \geq 0$ thì phương án đang xét là tối ưu. Trong trường hợp này vì $\bar{c}_3 = 20 > 0$ nên chỉ còn c_4 là phụ thuộc vào tập C_4 . Để kiểm tra \bar{c}_4 , ta tìm giá trị nhỏ nhất của \bar{c}_4 trên tập C_4 bằng việc giải bài toán QHTT:

$$\begin{aligned} \min \quad & y_{04} \quad -5y_{14} \quad +y_{24} \quad = \bar{c}_4 \\ & 3y_{04} \quad +y_{14} \quad +2y_{24} \quad = 2 \\ & y_{i4} \geq 0 \quad i = 0, 1, 2 \end{aligned} \quad (2.1.10)$$

Ta thu được $y_{04} = 0, y_{14} = 2, y_{24} = 0$ và $\bar{c}_4 = -10$. Do đó vẫn tồn tại điểm trong C_4 sao cho $\bar{c}_4 < 0$, tức phương án đang xét

$$(-z) = -24, \quad x_1 = 2, \quad x_2 = 3, \quad x_3 = x_4 = 0.$$

chưa phải là phương án tối ưu cho bài toán (2.1.8).

Để cải thiện phương án của bài toán (2.1.8), ta đưa biến x_4 vào cơ sở và tìm một biến đưa ra khỏi cơ sở để có cơ sở mới. Trong trường hợp này, theo quy tắc của phương pháp đơn hình, biến x_1 sẽ đưa ra khỏi cơ sở, ta có cơ sở mới là $\{2, 4\}$ với giá trị

$$(-z) = -4, \quad x_2 = 1, \quad x_4 = 2, \quad x_1 = x_3 = 0$$

Tuy nhiên để bảo toàn biến y_{i4} thu được ở bước trước của thuật toán, thay vì xem xét bài toán (2.1.8), theo Định lý 2.1.1, ta xem xét bài toán tương đương với nó có dạng:

$$\begin{array}{rccccrc} 6x_1 & +4x_2 & +x_3 & +0x_4^{(1)} & +y_{04}x_4 & = z(\min) \\ x_1 & +x_2 & -4x_3 & +2x_4^{(1)} & +y_{14}x_4 & = 5 \\ -x_1 & +x_2 & -x_3 & +0x_4 & +y_{24}x_4 & = 1 \\ x_1 \geq 0, & x_2 \geq 0, & x_3 \geq 0, & x_4 \geq 0, & x_4^{(1)} \geq 0 & \end{array} \quad . \quad (2.1.11)$$

ở đây $y_{\bullet 4} \in C_4$.

Như ở trên, ta có phương án cơ sở cho bài toán (2.1.11) là

$$(-z) = -4, \quad x_2 = 1, \quad x_4^{(1)} = 2, \quad x_1 = x_3 = x_4 = 0$$

và phương án đối ngẫu (hay nhân tử Lagrange) sẽ là $\pi = (0, 4)^T$. Khi đó các ước lượng cho hệ số sẽ là:

$$\bar{c}_1 = 10, \quad \bar{c}_3 = 5, \quad \bar{c}_4 = y_{04} - 4y_{24}.$$

Các giá trị \bar{c}_1, \bar{c}_3 là không âm. Do vậy ta chỉ cần kiểm tra c_4 . Để kiểm tra c_4 có nhỏ hơn 0 hay không ta giải bài toán QHTT phụ:

$$\begin{array}{rccccrc} \min & y_{04} & & -4y_{24} & = & \bar{c}_4 \\ & 3y_{04} & +y_{14} & +2y_{24} & = & 2 \\ & & & & & y_{i4} \geq 0. \end{array} \quad . \quad (2.1.12)$$

Bài toán này có cùng tập ràng buộc là đơn hình C_4 như bài toán phụ ở bước trước, chỉ khác nhau hàm mục tiêu. Giải bài toán (2.1.12) ta thu được $\bar{c}_4 = -4, y_{04} = 0, y_{14} = 0, y_{24} = 1$. Do $\bar{c}_4 = -4 < 0$ nên phương án

$$(-z) = -4, \quad x_2 = 1, \quad x_4^{(1)} = 2, \quad x_1 = x_3 = x_4 = 0$$

vẫn chưa tối ưu cho bài toán (2.1.8). Khi đó biến x_4 đưa vào cơ sở và loại bỏ x_2 ra khỏi cơ sở được phương án chấp nhận được mới là:

$$(-z) = -4, \quad x_4^{(1)} = \frac{5}{2}, \quad x_4 = 1, \quad x_1 = x_2 = x_3 = 0$$

Tiếp tục lập bài toán QHTT mới tương đương với (2.1.8) mà y_{i4} được xét lại:

$$\begin{array}{rcccccc} 6x_1 & +4x_2 & +x_3 & +0x_4^{(1)} & +0x_4^{(2)} & +y_{04}x_4 & = z(\min) \\ x_1 & +x_2 & -4x_3 & +2x_4^{(1)} & +0x_4^{(2)} & +y_{14}x_4 & = 5 \\ -x_1 & +x_2 & -x_3 & +0x_4^{(1)} & +1x_4^{(2)} & +y_{24}x_4 & = 1 \\ x_1 \geq 0, & x_2 \geq 0, & x_3 \geq 0, & x_4 \geq 0, & x_4^{(1)} \geq 0, & x_4^{(2)} \geq 0. & \end{array} \quad (2.1.13)$$

Phương án cơ sở chấp nhận được mới là:

$$(-z) = 0, \quad x_4^{(1)} = -\frac{5}{2}, \quad x_4^{(2)} = 1, \quad x_1 = x_2 = x_3 = x_4 = 0$$

Tương ứng với nó là phương án đối ngẫu (hay nhân tử Lagrange) thu được sẽ là $\pi = (0, 0)^T$ và các ước lượng tính được là $\bar{c}_1 = 6$, $\bar{c}_2 = 4$, $\bar{c}_3 = 1$, $\bar{c}_4 = y_{04}$. Tiếp theo để kiểm tra $\bar{c}_4 = y_{04}$, ta giải bài toán phụ:

$$\begin{array}{rcl} \min & y_{04} & = \bar{c}_4 \\ & 3y_{04} + y_{14} + y_{24} & = 2 \\ & y_{i4} & \geq 0 \end{array} \quad (2.1.14)$$

Giải bài toán này ta thu được $\bar{c}_4 = 0$, $y_{04} = 0$, $y_{14} = 2$, $y_{24} = 0$. Vì tất cả các ước lượng của bài toán (2.1.13) đều không âm cho mọi giá trị của $y_{\bullet 4}$ nên phương án đang xét là tối ưu cho (2.1.13). Để khôi phục lại nghiệm bài toán gốc (2.1.8) ta thấy rằng:

$$z = 0, \quad x_1 = x_2 = x_3 = 0, \quad x_4 = x_4^{(1)} = x_4^{(2)} = \frac{7}{2}$$

và tính được:

$$\begin{pmatrix} y_{04} \\ y_{14} \\ y_{24} \end{pmatrix} = y_{\bullet 4}^{(1)} \frac{x_4^{(1)}}{x_4^{(1)} + x_4^{(2)}} + y_{\bullet 4}^{(2)} \frac{x_4^{(2)}}{x_4^{(1)} + x_4^{(2)}} = \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix} \cdot \frac{5}{7} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \cdot \frac{2}{7} = \begin{pmatrix} 0 \\ 10/7 \\ 2/7 \end{pmatrix}.$$

2.1.2. Trường hợp bài toán phụ có phương án không bị chặn.

Chú ý rằng, trong phương pháp Wolfe, ta cần phải giải liên tiếp các bài toán QHTT phụ trên tập ràng buộc C_j . Có thể xảy ra trường hợp tập ràng buộc này không bị chặn, và khi đó bài toán phụ có thể không bị chặn dưới. Khi giải bài toán phụ, sẽ có hai trường hợp:

- 1) Nếu một phương án cực biên $y_{\bullet s} = y_{\bullet s}^e \in C_s$ thu được với $y_{\bullet s}^e$ là điểm cực biên, thì nếu cước phí \bar{c}_s là âm thì đưa nó vào cơ sở của bài toán gốc.
- 2) Nếu một lớp các phương án $y_{\bullet s} = y_{\bullet s}^e + \theta y_{\bullet s}^h \in C_s$ với $y_{\bullet s}^e$ là điểm cực biên, $y_{\bullet s}^h$ là các hướng cực biên, với θ là một tham số vô hướng. Thì chỉ cần đưa các hướng cực biên $y_{\bullet s}^h$ vào cơ sở của bài toán gốc. Cơ sở cho việc xử lý này là ta viết lại $y_{\bullet s}x_s$ như sau:

$$y_{\bullet s}x_s = (y_{\bullet s}^e + \theta y_{\bullet s}^h)x_s = \left(\frac{1}{\theta}y_{\bullet s}^e + y_{\bullet s}^h\right)\theta x_s$$

. Thì do $\theta > 0$ tùy ý, nên hướng cực biên này sẽ làm cho giá trị của mục tiêu \bar{c}_s giảm tới mức âm.

2.2. Nguyên lý phân rã Dantzig-Wolfe(D-W)

Nguyên lý phân rã Dantzig-Wolfe dựa trên định lý biểu diễn tập lồi đa diện. Ta đã biết rằng, trong giải tích lồi, một tập lồi đa diện P có số điểm cực biên $V(P)$ và số hướng cực biên $E(P)$ là hữu hạn, và mọi điểm x trong tập lồi đa diện P đều có thể biểu diễn qua tổ hợp lồi các điểm cực biên và tổ hợp không âm các hướng cực biên của nó, nghĩa là:

$$x = \sum_{v^i \in V(C)} \alpha_i v^i + \sum_{e^j \in E(C)} \beta_j e^j, \quad (2.2.15)$$

trong đó $\alpha_i, \beta_j \geq 0$ và $\sum_i \alpha_i = 1$.

Đối với tập lồi đa diện P có dạng:

$$P := \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\} \quad (2.2.16)$$

thì tập hợp các hướng vô hạn của P sẽ là $Ay = 0, y \geq 0$. Do đó ta định nghĩa hướng cực biên chuẩn hóa như sau:

Định nghĩa 2.2.1. Các hướng cực biên chuẩn hóa của tập lồi đa diện P cho bởi (2.2.16) sẽ là các nghiệm cơ sở của hệ phương trình:

$$\begin{cases} Ay = 0 \\ e^T y = 1 \\ y \geq 0 \end{cases} \quad (2.2.17)$$

với $e^T = (1, 1, \dots, 1)^T$ (véc tơ các thành phần đều là 1).

Chú ý rằng, nếu $A = m < n$ thì số nghiệm cơ sở (hay số nghiệm độc lập tuyến tính) của hệ này là $n - m$ và thỏa mãn điều kiện không âm và có tổng các thành phần bằng 1. Hiển nhiên số hướng cực biên chuẩn hóa luôn hữu hạn.

2.2.1. Ý tưởng của phương pháp phân rã Dantzig-Wolfe

a. Ý tưởng chung. Giả sử ta có một bài toán QHTT kích thước lớn với ràng buộc dạng $Ax = b, x \geq 0$. Phương pháp phân rã D-W thực hiện như sau:

- Ta sẽ phân rã ràng buộc này thành các ràng buộc dạng $A^k x = b^k$ với $k = 1, \dots, p$ nào đó. Hiển nhiên nếu A có dạng chéo khối với p khối, ta có thể phân rã bài toán thành p bài toán QHTT độc lập có kích thước nhỏ hơn.
- Khi A có dạng tùy ý, ta có thể xem bài toán QHTT thỏa mãn ràng buộc $A^1 x = b$ và thỏa mãn thêm ràng buộc $A^k x = b^k$ với $k = 2, \dots, p$ và $x \geq 0$. Ta xem các ràng buộc thêm này như một tập lồi đa diện, và các điểm trong đó thỏa mãn định lý biểu diễn tập lồi.
- Biểu diễn biến x qua các điểm cực biên và hướng cực biên chuẩn hóa của tập lồi đa diện cần thỏa mãn.
- Thay x vào bài toán chỉ có ràng buộc $A^1 x = b^1$, thu được bài toán với các biến mới α_i, β_j .

- Hạn chế số đỉnh, số hướng cực biên chuẩn hóa, giải quyết bài toán hạn chế. Nếu thỏa mãn điều kiện tối ưu cho bài toán ban đầu dừng lại. Ngược lại, thêm một đỉnh mới hoặc hướng cực biên chuẩn hóa mới và lặp lại việc giải bài toán đó.

Chú ý rằng, để tăng hiệu quả của phương pháp phân rã D-W, điều quan trọng là khai thác cấu trúc của ma trận ràng buộc. Tùy thuộc vào cấu trúc của ma trận A mà người ta lựa chọn cách phân rã bài toán một cách hợp lý nhằm giảm tối đa kích thước bài toán con và bài toán con được giải quyết hiệu quả. Thậm chí trong trường hợp ma trận là chéo khối, thì việc giải bài toán con có thể làm một cách độc lập, song song trên các máy tính khác nhau.

b. Nội dung phương pháp phân rã Dantzig - Wolfe. Để đơn giản cách trình bày, ta xét bài toán QHTT chuẩn tắc dạng

$$\begin{aligned} \min\{ & c^T x & = z \\ & Ax & = b \\ & x & \geq 0, \end{aligned} \quad (2.2.18)$$

trong đó A là ma trận cỡ $m \times n$. Trước hết, ta phân rã tập $D := \{Ax = b, x \geq 0\}$ bởi hai phần $A^1x = b^1$ và $A^2x = b^2, x \geq 0$ với A^1, A^2 là các ma trận cỡ $m_1 \times n$ và $m_2 \times n$, tương ứng. Bài toán (2.2.18) trở thành

$$\begin{aligned} \min\{ & c^T x & = z \\ & A^1x & = b^1 \\ & A^2x & = b^2 \\ & x & \geq 0. \end{aligned} \quad (2.2.19)$$

Hay, ta có thể viết lại như sau

$$\begin{aligned} \min\{ & c^T x & = z \\ & A^1x & = b^1 \\ & x & \geq 0 \end{aligned} \quad (2.2.20)$$

thỏa mãn thêm ràng buộc

$$P := \begin{cases} A^2x = b^2 \\ x \geq 0 \end{cases} \quad (2.2.21)$$

Hiển nhiên P là một tập lồi đa diện chính tắc, ta gọi số đỉnh của P là $V(P) := \{u^i \mid i = 1, \dots, L\}$ và số hướng cực biên chuẩn hóa của P là $E(P) := \{v^j \mid j = 1, \dots, M\}$. Khi đó, theo định lý biểu diễn tập lồi (Định lý 1.1.1) mọi điểm trong P đều có thể biểu diễn dưới dạng:

$$x = \sum_{i=1}^L \alpha_i v^i + \sum_{j=1}^M \beta_j e^j, \quad (2.2.22)$$

trong đó $\alpha_i, \beta_j \geq 0$ và $\sum_{i=1}^L \alpha_i = 1$ với $i = 1, \dots, L, j = 1, \dots, M$. Thay vào (2.2.20) ta thu được bài toán có dạng:

$$\begin{aligned} \min \{ & \sum_{i=1}^L (c^T u^i) \alpha_i + \sum_{j=1}^M (c^T v^j) \beta_j = z \}, \\ & \sum_{i=1}^L (A^1 u^i) \alpha_i + \sum_{j=1}^M (A^1 v^j) \beta_j = b^1, \\ & \sum_{i=1}^L \alpha_i = 1, \quad . \\ & \alpha_i \geq 0, \quad i = 1, \dots, L, \\ & \beta_j \geq 0, \quad j = 1, \dots, M. \end{aligned} \quad (2.2.23)$$

Bằng cách đặt

$$\begin{aligned} G^i &= A^1 u^i, & H^j &= A^1 v^j, \\ g_i &= c^T u^i, & h_j &= c^T v^j. \end{aligned} \quad (2.2.24)$$

Bài toán (2.2.23) được viết gọn dưới dạng:

$$\begin{aligned} \min \{ & \sum_{i=1}^L g_i \alpha_i + \sum_{j=1}^M h_j \beta_j = z \}, \\ & \sum_{i=1}^L G^i \alpha_i + \sum_{j=1}^M H^j \beta_j = b^1 \quad : \pi, \\ & \sum_{i=1}^L \alpha_i = 1 \quad : \gamma, \quad . \\ & \alpha_i \geq 0, \quad i = 1, \dots, L, \\ & \beta_j \geq 0, \quad j = 1, \dots, M. \end{aligned} \quad (2.2.25)$$

Bài toán (2.2.25) là một bài toán QHTT theo các biến α_i và β_j với $i = 1, \dots, L, j = 1, \dots, M$. Bài toán (2.2.25) được gọi là bài toán **Full Master Program**. Rõ ràng, nếu số đỉnh $V(P)$ và số hướng cực biên $E(P)$ của tập P là lớn thì bài toán (2.2.25) sẽ có kích thước lớn. Ở đây $\pi \in \mathbb{R}^{m_1}$ và $\gamma \in R$ là các biến đối ngẫu (hay nhân tử Lagrange) tương ứng với các ràng buộc của bài toán (2.2.25).

Định nghĩa 2.2.2. Hai bài toán QHTT (2.2.18) và (2.2.25) được gọi là tương đương, nếu x là phương án chấp nhận của bài toán (2.2.18) thì tồn tại α_i và β_j với $i = 1, \dots, L$, $j = 1, \dots, M$ là phương án chấp nhận của bài toán (2.2.25) sao cho

$$x = \sum_{i=1}^L \alpha_i u^i + \sum_{j=1}^M \beta_j v^j$$

và ngược lại.

Với cách định nghĩa hai bài toán tương đương như định nghĩa 2.2.2, ta có định lý sau:

Định lý 2.2.1. Bài toán QHTT chính tắc (2.2.18) là tương đương với bài toán QHTT **Full Master Program** (2.2.25) theo nghĩa tương đương được định nghĩa như trong định nghĩa 2.2.2.

Hiển nhiên định lý này cũng cho ta biết, nếu x^* là phương án tối ưu của (2.2.18) thì tồn tại α_i^* và β_j^* với $i = 1, \dots, L$, $j = 1, \dots, M$ là phương án tối ưu của bài toán (2.2.25) sao cho

$$x^* = \sum_{i=1}^L \alpha_i^* u^i + \sum_{j=1}^M \beta_j^* v^j$$

và ngược lại.

Thuật toán phân rã D-W không giải trực tiếp bài toán **Full Master Program** (2.2.25) mà giải trên bài toán hạn chế sẽ được định nghĩa dưới đây. Ta gọi

$$L_{\max} := \{1, 2, \dots, L\}, \quad M_{\max} := \{1, 2, \dots, M\},$$

và \bar{M}, \bar{L} là những tập con của M_{\max} và L_{\max} tương ứng. Khi đó ta định nghĩa.

Định nghĩa 2.2.3. Restricted Master Program

Bài toán thu được từ (2.2.25) bằng cách hạn chế các tập định của P trên tập \bar{M} và hạn chế các hướng cực biên của P trên \bar{L} gọi là bài toán **Restricted Master Program**, nghĩa là:

$$\begin{aligned} \min \{ & \sum_{i \in \bar{L}} g_i \alpha_i + \sum_{j \in \bar{M}} h_j \beta_j = z \quad \}, \\ & \sum_{i \in \bar{L}} G^i \alpha_i + \sum_{j \in \bar{M}} H^j \beta_j = b^1 \quad : \pi, \\ & \sum_{i=1}^L \alpha_i = 1 \quad : \gamma, \cdot \\ & \alpha_i \geq 0, \quad i \in L, \\ & \beta_j \geq 0, \quad j \in M. \end{aligned} \quad (2.2.26)$$

Câu hỏi tiếp theo là từ nghiệm bài toán hạn chế (2.2.26) làm thế nào để kiểm tra xem đã là tối ưu cho bài toán cần giải chưa? Để trả lời câu hỏi này, ta xuất phát từ tiêu chuẩn tối ưu trong phương pháp đơn hình. Chú ý rằng, bài toán **Full Master Program** và bài toán hạn chế (2.2.26) đều là các bài toán QHTT chính tắc.

Nếu ta ký hiệu $(\bar{\pi}, \bar{\gamma})$ là phương án cơ sở đối ngẫu thì, thì nó sẽ thỏa mãn hệ ràng buộc đối ngẫu dạng:

$$\begin{cases} \gamma + (G^i)^T \pi = g_i, & i = 1, 2, \dots, k \\ (H^j)^T \pi = h_j, & j = 1, 2, \dots, l. \end{cases} \quad (2.2.27)$$

Phương pháp đơn hình sẽ chọn véc tơ mới đưa vào cơ sở dựa trên quy tắc là cho hàm mục tiêu giảm nhiều nhất, nghĩa là, ta chọn chỉ số $i = i^*$ và $j = j^*$ từ điều kiện:

$$g_{i^*} - (G^{i^*})^T \bar{\pi} - \bar{\gamma} = \min_{i=1, \dots, L} \{g_i - (G^i)^T \bar{\pi}\} - \bar{\gamma} \quad (2.2.28)$$

và

$$h_{j^*} - (H^{j^*})^T \bar{\pi} = \min_{j=1, \dots, M} \{h_j - (H^j)^T \bar{\pi}\} \quad (2.2.29)$$

Khi đó véc tơ cột của ma trận ràng buộc ứng với chỉ số (i^*, j^*) sẽ được đưa vào cơ sở. Ta định nghĩa độ lệch đối ngẫu bởi:

$$\bar{\rho} = c - (A^1)^T \bar{\pi} \quad (2.2.30)$$

và thay thế $G^i = A^1 u^i$ và $H^j = A^1 v^j$ để thu được:

$$\begin{aligned} \min_{i=1, \dots, L} \{g_i - (G^i)^T \bar{\pi}\} - \bar{\gamma} &= \min_{i=1, \dots, L} \{c^T u^i - (A^1 u^i)^T \bar{\pi}\} - \bar{\gamma} \\ &= \min_{i=1, \dots, L} \{(c^T - (A^1)^T \bar{\pi})^T u^i\} - \bar{\gamma} \\ &= \min_{i=1, \dots, L} \{\bar{\rho}^T u^i\} - \bar{\gamma}. \end{aligned}$$

Tương tự

$$\min_{j=1, \dots, M} \{h_j - (H^j)^T \bar{\pi}\} = \min_{j=1, \dots, M} \{\bar{\rho}^T v^j\} \quad (2.2.31)$$

Đến đây ta vẫn chưa biết có điểm cực biên hai hướng cực biên chuẩn hóa nào mà có giá âm. Để xác định xem (2.2.30) có giá âm hay không, ta không phải đi ước lượng trên toàn bộ số đỉnh của P mà thay vào đó, ta giải một bài toán QHTT dạng:

$$\begin{aligned} \min \{ \bar{\rho}^T x &= z_2 + \bar{\gamma} \} \\ A^2 x &= b^2 \\ x &\geq 0 \end{aligned} \quad (2.2.32)$$

với $\bar{\rho} = c - (A^1)^T \bar{\pi}$ thỏa mãn (2.2.30) và $\bar{\gamma}$ là biến đối ngẫu ứng với ràng buộc lồi của bài toán **Restricted Master Program** để xác định nghiệm $x = u^*$ là điểm cực biên của P .

Sau khi sử dụng phương pháp đơn hình mà phương án tối ưu là một phương án cơ sở chấp nhận $x = u^*$, thì ta tìm được cột với cước phí thu gọn nhỏ nhất cho bài toán **Full Master Program**. Nếu u^* là tối ưu, và nếu $\min z_2 \leq 0$ ta bổ sung vào cho bài toán **Restricted Master Program** thêm cột mới bởi

$$\begin{pmatrix} G^* \\ g^* \\ 1 \end{pmatrix} = \begin{pmatrix} A^1 u^* \\ c^T u^* \\ 1 \end{pmatrix} \quad (2.2.33)$$

với chỉ số $*$ nhằm thể hiện việc đánh số lại các chỉ số cho phù hợp với bài toán **Restricted Master Program** mới thu được. Tiếp theo lại tối ưu lại bài toán hạn chế thu được này. Nếu $\min z_2 = 0$ thì tất cả cước phí thu gọn với bài toán Full Master Program (2.2.25) không âm. Trong trường hợp này từ phương án tối ưu của bài toán Full Master Program và (2.2.22) có thể tìm được phương án tối ưu cho bài toán ban đầu (2.2.18).

Trong trường hợp còn lại, giả sử giải bài toán (2.2.32) bằng phương pháp đơn hình cho ta một phương án cực biên thuần nhất $x^h = v^*$. Trong trường hợp này, theo lý thuyết, ta không cần ước lượng tất cả các hướng cực biên chuẩn hóa (2.2.21) mà có thể tìm được hướng cực biên tốt nhất bằng cách giải bài toán QHTT:

$$\begin{aligned} \min \{ \bar{\rho}^T x &= z_2^h \} \\ A^2 x &= 0 \\ e^T x &= 1 \\ x &\geq 0 \end{aligned} \quad (2.2.34)$$

với $e = (1, \dots, 1)^T$, $\bar{\rho}$ thỏa mãn (2.2.30) và z_2^h là giá trị tối ưu tương ứng với nghiệm của nó. Tuy nhiên, việc giải quyết bài toán này sẽ là không cần thiết vì sẽ tăng chi phí tính toán cho thuật toán. Trên thực tế, ta chỉ cần chọn một hướng cực biên chuẩn hóa $x^h = v^*$ nào đó của (2.2.21) mà làm cho ước lượng véc tơ giá là âm. Khi đó ta

đưa cột

$$\begin{pmatrix} H^* \\ h^* \\ 0 \end{pmatrix} = \begin{pmatrix} A^1 v^* \\ c^T v^* \\ 0 \end{pmatrix} \quad (2.2.35)$$

vào cơ sở của bài toán thu gọn để thu được bài toán mới và giải lại bài toán này.

Chú ý 2.2.1. Trong thủ tục đơn hình, để tính các ước lượng của các véc tơ A_k , gọi là $\Delta_k = \sum_{j \in J} z_{jk} c_j - c_k$. Nếu $\Delta_k \leq 0$ với mọi k thì phương án đang xét là tối ưu. Đây chính là tiêu chuẩn tối ưu của thuật toán đơn hình mà ta đã sử dụng trong thuật toán này. Trên thực tế, ta không tính các ước lượng này thông qua các hệ số z_{jk} mà thông qua các biến đối ngẫu π, γ như công thức đề cập trong chương 1.

2.2.2. Thuật toán Dantzig-Wolfe và phương án xuất phát

a. Thuật toán Dantzig-Wolfe. Thuật toán phân rã Dantzig - Wolfe được mô tả như sau:

Đầu vào: Dữ liệu của bài toán QHTT: Véc tơ giá c , Ma trận ràng buộc A^1 véc tơ vế phải b^1 , tập lồi đa diện P với tập đỉnh \bar{L} và tập hướng cực biên \bar{L} xuất phát cho bài toán *Restricted Master Program*.

Đầu ra: Khẳng định bài toán (2.2.18) vô nghiệm hoặc tìm được một nghiệm tối ưu x^* .

Thuật toán:

1. Từ bài toán *Restricted Master Program* với một phương án cơ sở chấp nhận được xuất phát và một cột không cơ sở.
2. Giải bài toán *Restricted Master Program*. Nếu tìm được một nghiệm tối ưu thì đi đến bước 3 nếu khác thông báo bài toán gốc không bị chặn và dừng.
3. Nghiệm cơ sở chấp nhận được tối ưu của bài toán *Restricted Master Program* cùng với nhân tử đơn hình $(\bar{\pi})$ giúp ta loại ra cột không cơ sở.
4. Giải bài toán con (2.2.32).

5. Nếu từ (2.2.32) tìm được phương án cơ sở chấp nhận được tối ưu và $\min z_2 < 0$ thì cột (2.2.33) được thêm vào bài toán *Restricted Master Program* và quá trình tiếp tục ở bước 2.
6. Nếu từ (2.2.32) ta thu được một hướng cực biên thuần nhất không âm thì cột (2.2.35) được thêm vào bài toán *Restricted Master Program* quá trình tiếp tục ở bước 2.
7. Nếu $z_2 = 0$, từ phương án này tìm được phương án tối ưu cho bài toán gốc. Phương án tối ưu được cho bởi (2.2.21) với $\alpha_1, \alpha_2, \dots$ và $\beta_1, \beta_2, \beta_3, \dots$ là phương án tối ưu cho bài toán *Restricted Master Program* cuối cùng. Nó chính là phương án tối ưu chấp nhận được cho bài toán *Full Master Program*.

Chú ý rằng, một phương án cơ sở tối ưu của bài toán *Restricted Master Program* cũng là phương án tối ưu cơ sở cho bài toán *Full Master Program* nếu

$$\min z_2 = \bar{\gamma}. \quad (2.2.36)$$

Công thức này suy ra từ điều kiện (2.2.32). Khi đó độ lệch đối ngẫu $\bar{\rho}^T x^* = 0$.

Bên cạnh đó, nếu bài toán QHTT (2.2.18) là không suy biến thì thuật toán D-W sẽ dừng sau hữu hạn bước lặp. Hoặc tìm được phương án tối ưu hoặc chỉ ra một hướng cực biên mà hàm mục tiêu giảm ra $-\infty$.

b. Phương án xuất phát của thuật toán Dantzig-Wolfe.

Trong phần này chúng ta sẽ trình bày phương pháp tìm phương án xuất phát của thuật toán Dantzig-Wolfe. Trước hết phải ta cần phải xác định một đỉnh của tập P là u^1 . Việc này có thể được làm bằng cách sử dụng Pha 1 của thuật toán đơn hình để giải bài toán phụ dạng.

$$\begin{aligned} \min \quad & \sum_{i=1}^{m_2} v_i = f(u, v) \\ \sum_{j=1}^n a_{ij}^2 u_j + v_i &= b^2 \\ u_j \geq 0, v_i \geq 0, i &= 1, \dots, m_1, j = 1, \dots, n. \end{aligned} \quad (2.2.37)$$

Sau khi có một đỉnh u^1 của tập P , ta đặt $G^1 = A^1 u^1$ ta thiết lập bài toán *Restricted*

Master Program với biến giả ξ_i .

$$\begin{aligned} \min \quad & \sum_{i=1}^{m_1} \xi_i = w \\ & G^1 \alpha_1 + \sum_{i=1}^{m_1} \pm e^i \xi_i = b^1 \\ & \alpha_1 = 1 \\ & \alpha_1 \geq 0, \quad \xi_i \geq 0, \quad i = 1, \dots, m_1 \end{aligned} \tag{2.2.38}$$

trong đó e^i là cột thứ i của ma trận đơn vị $m_1 \times m_1$ chú ý rằng α_1 phải bằng 1. Do đó trong đẳng thức thứ 2 là $+e^i$ nếu $b_i^1 - G_i^1 \geq 0$ và $-e^i$ nếu $b_i^1 - G_i^1 < 0$. Biến $\alpha_1, \xi_1, \xi_2, \dots, \xi_{m_1}$ tạo ra tập biến cơ sở cho bài toán (2.2.37). Pha I của bài toán được giải bằng phương pháp Dantzig-Wolfe. Sau pha I ta tìm được một phương án chấp nhận được cho bài toán gốc hoặc có thể kết luận bài toán gốc không có phương án tối ưu. Nếu pha I kết thúc với một phương án chấp nhận được thì tất cả các biến giả sẽ có giá trị bằng 0. Sau pha I các hệ số của hàm mục tiêu được thay thế cho các cột trong bài toán *Restricted Master Program* và giải lại bài toán *Restricted Master Program*. Quá trình được tiếp tục thực hiện lặp lại.

2.3. Áp dụng phương pháp Dantzig-Wolfe cho một số bài toán

Thuật toán phân rã D-W có thể áp dụng cho các bài toán QHTT tổng quát như đã đề cập ở trên. Tuy nhiên phương pháp này rất hiệu quả khi áp dụng cho những bài toán QHTT kích thước lớn và có cấu trúc. Dưới đây ta sẽ áp dụng phương pháp này cho một số trường hợp cụ thể và đưa ra một ví dụ minh họa bằng số.

2.3.1. Hệ khối góc (block-angular system)

Phép phân rã Dantzig-Wolfe để xử lý bài toán khối góc có dạng (2.3.39):

$$\begin{aligned} \min\{ & (c^0)^T x^0 + (c^1)^T x^1 + \dots + (c^k)^T x^k = z \} \\ \text{ràng buộc } & A^0 x^0 + A^1 x^1 + \dots + A^k x^k = b \\ & F^1 x^1 = f^1 \\ & \dots \\ & F^k x^k = f^k \end{aligned} \quad (2.3.39)$$

$$x^i \geq 0, \quad i = 0, 1, 2, \dots, k.$$

Bài toán (2.3.39) được phân rã dưới dạng.

$$\begin{aligned} \min\{ & (c^0)^T x^0 + (c^1)^T x^1 + \dots + (c^k)^T x^k = z \} \\ \text{ràng buộc } & A^0 x^0 + A^1 x^1 + \dots + A^k x^k = b \\ & x^0 \geq 0 \end{aligned} \quad (2.3.40)$$

và thỏa mãn thêm các ràng buộc :

$$(S_k) : \quad F^k x^k = f^k, \quad x^k \geq 0, \quad \text{với } k = 1, \dots, K,$$

trong đó x^k là một véc tơ thành phần con của x có số chiều tương thích với F^k . Như vậy trong trường hợp này tập P được phân ra thành K tập S_k có số chiều thấp hơn. Ta có thể tìm tập đỉnh và tập hướng cực biên của S_k dễ dàng hơn.

Giả sử các nghiệm chấp nhận được và các hướng cực biên chuẩn hóa cho S_1 đến S_k của (2.3.1) là tồn tại. Trên thực tế số nghiệm này là rất lớn khó có thể tính toán tất cả một cách thô sơ. Từ kết quả của định lý biểu diễn tập lồi (Định lý 1.1.1), các nghiệm $x^k \geq 0$ cho (S_k) với $k = 1, \dots, K$ có thể viết dưới dạng:

$$x^k = \sum_{i=1}^{L_k} \alpha_{ki} u^{ki} + \sum_{j=1}^{M_k} \beta_{kj} v^{kj}$$

với

$$\begin{aligned} \sum_{i=1}^{L_k} \alpha_{ki}, \quad \alpha_{ki} \geq 0, \quad i = 1, \dots, L_k, \\ \beta_{kj} \geq 0, \quad j = 1, \dots, M_k, \end{aligned}$$

trong đó u^{ki} với $i = 1, \dots, L_k$ là tập hữu hạn các phương án cơ sở chấp nhận được cho S_k và v_{kj} với $j = 1, \dots, M_k$ là tập hữu hạn các hướng cực biên chuẩn hóa cho S_k . Ngược lại tất cả các nghiệm biểu diễn bởi (2.3.1) là chấp nhận được cho S_k .

2.3.2. Bài toán cấu trúc bậc thang (staircase structured problems)

Để đơn giản, ta xét bài toán QHTT dạng bậc thang với 4 bậc như sau

$$\begin{aligned}
 & \min\{(c^0)^T x^0 + (c^1)^T x^1 + (c^2)^T x^2 + (c^3)^T x^3 + (c^4)^T x^4 = z\} \\
 \text{Ràng buộc} \quad & A^{11}x^1 = b^1 \\
 & A^{21}x^1 + A^{22}x^2 = b^2 \\
 & A^{32}x^2 + A^{33}x^3 = b^3 \\
 & A^{43}x^3 + A^{44}x^4 = b^4 \\
 & x^k \geq 0, \quad k = 1, \dots, 4.
 \end{aligned} \tag{2.3.41}$$

Sử dụng phép phân rã D-W, ta phân rã bài toán như sau:

$$\begin{aligned}
 & \min(c^0)^t x^0 + (c^1)^t x^1 + (c^2)^t x^2 + (c^3)^t x^3 + (c^4)^t x^4 = z \\
 \text{Ràng buộc} \quad & A^{11}x^1 = b^1.
 \end{aligned} \tag{2.3.42}$$

thỏa mãn thêm ràng buộc

$$P := \begin{cases} A^{21}x^1 + A^{22}x^2 = b^2 \\ A^{32}x^2 + A^{33}x^3 = b^3 \\ A^{43}x^3 + A^{44}x^4 = b^4 \\ x^k \geq 0 \quad k = 1, \dots, 4. \end{cases} \tag{2.3.43}$$

Tiếp theo sử dụng định lý biểu diễn tập lồi (Định lý 1.1.1) ta có biểu diễn:

$$x = \begin{pmatrix} x^1 \\ x^2 \\ x^3 \\ x^4 \end{pmatrix} = \sum_{i=1}^L \alpha_i \begin{pmatrix} u^{i1} \\ u^{i2} \\ u^{i3} \\ u^{i4} \end{pmatrix} + \sum_{j=1}^M \beta_j \begin{pmatrix} v^{j1} \\ v^{j2} \\ v^{j3} \\ v^{j4} \end{pmatrix}$$

với

$$\sum_{i=1}^{L_1} \alpha_i = 1, \quad \alpha_i \geq 0 \quad \beta_j \geq 0, \quad j = 1, \dots, M$$

và $u^i = (u^{i1}, u^{i2}, u^{i3}, u^{i4})$ với $i = 1, \dots, L$ là tập đầy đủ các nghiệm cơ sở chấp nhận được của (2.3.43), còn $v^j = (v^{j1}, v^{j2}, v^{j3}, v^{j4})$ với $j = 1, \dots, M$ là tập đầy đủ các

hướng cực biên chuẩn hóa của (2.3.43). Khi đó, ta thu được bài toán *Full Master Program* như sau:

$$\begin{aligned}
 \min \quad & \sum_{i=1}^L [(c^1)^T u^{i1} + (c^2)^T u^{i2} + (c^3)^T u^{i3} + (c^4)^T u^{i4}] \alpha_i + \\
 & \sum_{j=1}^M [(c^1)^T v^{j1} + (c^2)^T v^{j2} + (c^3)^T v^{j3} + (c^4)^T v^{j4}] \beta_j = z \\
 \text{ràng buộc: } \quad & \sum_{i=1}^L A^{11} u^{i1} \alpha_i + \sum_{j=1}^M A^{11} v^{j1} \beta_j = b^1 \quad (2.3.44) \\
 & \sum_{i=1}^L \alpha_i = 1 \\
 & \alpha_i \geq 0 \quad i = 1, \dots, L, \beta_j \geq 0 \quad j = 1, \dots, M.
 \end{aligned}$$

Ta sử dụng giá trị tối ưu π^1 của bài toán *Restricted Master Program* xác định mục tiêu cho bài toán (2.3.43) bài toán với mục tiêu được điều chỉnh là:

$$\begin{aligned}
 \min \quad & (c^1 - (\pi^1)^T A^{11})^T x^1 + (c^2)^T x^2 + (c^3)^T x^3 + (c^4)^T x^4 = z \\
 \text{Ràng buộc} \quad & A^{21} x^1 + A^{22} x^2 = b^2 \\
 & A^{32} x^2 + A^{33} x^3 = b^3 \quad (2.3.45) \\
 & A^{43} x^3 + A^{44} x^4 = b^4 \\
 x^k \geq 0, \quad & k = 1, \dots, 4.
 \end{aligned}$$

Bài toán này đã có dạng tương tự như bài toán ban đầu, ở các "bậc thang". Do đó, ta tiếp tục xử lý bài toán này lặp lại các bước như trên với số bậc giảm đi 1.

2.3.3. Ví dụ bằng số minh họa thuật toán Dantzig-Wolfe

Xét bài toán quy hoạch tuyến tính dạng

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	
1	2	3	4	5	6	1	2	3	4	5	6	7	-10	= $z(\min)$
3	2	1	6	5	4	8	5	7	3	4	1	1	2	= 64
1	8	3	7	1	4	5	2	5	3	2	6	3	4	= 63
1	1	1												= 3
			1	1	1									= 4
1			1											= 2
	1			1										= 1
		1			1									= 4
						1	1	1						= 4
									1	1	1			= 5
						1			1					= 3
							1			1				= 3
								1			1			= 3
												1	-1	= 1

trong đó hàng thứ nhất chứa $x_j \geq 0$ với $j = 1, \dots, 14$ là biến cần tối ưu. Hàng thứ 2 chứa hệ số hàm mục tiêu, các hàng còn lại là ma trận ràng buộc và véc tơ vế phải. Đây là bài toán dạng chính tắc, có cấu trúc đặc biệt với ba khối sau cùng là tách biến hoàn toàn.

Các nhóm biến được phân rã như sau: $x^1 = (x_1, x_2, \dots, x_6)^T$, $x^2 = (x_7, \dots, x_{12})^T$ và $x^3 = (x_{13}, x_{14})^T$. Ta gọi ba nhóm ràng buộc này là **sub1**, **sub2** và **sub3** với ba nhóm biến x^1 , x^2 và x^3 tương ứng như sau:

$$\begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \end{bmatrix}, \quad \begin{bmatrix} x_7 & x_8 & x_9 \\ x_{10} & x_{11} & x_{12} \end{bmatrix} \quad \text{và} \quad \begin{bmatrix} x_{13} \\ x_{14} \end{bmatrix}$$

Khi đó ta biểu diễn lại bài toán theo ba nhóm biến trên dưới dạng sau (bằng cách

đánh số lại chỉ số các biến):

$$\begin{aligned}
 (c^1)^T x^1 + (c^2)^T x^2 + (c^3)^T x^3 &= z(\min) \\
 A^1 x^1 + A^2 x^2 + A^3 x^3 &= b \\
 F^1 x^1 &= f^1 \\
 &F^2 x^2 &= f^2 \\
 &&F^3 x^3 &= f^3 \\
 x^k &\geq 0 && k = 1, 2, 3.
 \end{aligned} \tag{2.3.46}$$

Gọi các tập

$$P_1 = \{x^1 \in \mathbb{R}^6 : F^1 x^1 = f^1\}, \quad P_2 = \{x^2 \in \mathbb{R}^6 : F^2 x^2 = f^2\}, \quad \text{và} \quad P_3 = \{x^3 \in \mathbb{R}^2 : F^3 x^3 = f^3\}.$$

Khi đó, các đỉnh và hướng cực biên của P_1 (ứng với **sub1**) là:

$$\begin{bmatrix} 2 & 1 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} 3 \\ 2 & 1 & 1 \end{bmatrix}$$

Tương tự, ta tìm được các phương án cơ sở cho P_2 hay **sub2** là:

$$\begin{bmatrix} 1 & 3 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} 1 & 3 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} 1 & 3 \\ 2 & 3 \end{bmatrix}$$

Cuối cùng, P_3 hay **sub3** có một nghiệm cơ sở là $x_{13} = 1, x_{14} = 0$ và một hướng cực biên là $x_{13} = 1, x_{14} = 1$.

Thiết lập bài toán *Full Master Problem* cho bài toán trên, ta thu được:

α_1^1	α_2^1	α_3^1	α_4^1	α_1^2	α_2^2	α_3^2	α_4^2	α_5^2	α_6^2	α_1^3	β_1^3	
28	28	28	28	33	33	33	33	38	33	7	-3	= $z(\min)$
24	24	24	24	32	40	45	43	33	47	1	3	= 64
26	18	36	28	35	39	38	30	32	32	3	7	= 63
1	1	1	1									= 1
				1	1	1	1	1	1			= 1
										1		= 1

Tiếp theo, ta sẽ trình bày cụ thể phương pháp phân rã Dantzig-Wolfe cho bài toán này. Bắt đầu ta tìm một nghiệm cơ sở chấp nhận được cho ba bài toán con. Nếu

không tồn tại nghiệm cơ sở chấp nhận được cho một trong ba bài toán con thì kết luận bài toán gốc là không có phương án chấp nhận được. Xét các bài toán:

sub1:

$$\begin{array}{rcccccccc}
 \min & 1x_1 & +2x_2 & +3x_3 & +4x_4 & +5x_5 & +6x_6 & = z_1 \\
 \text{ràng buộc} & x_1 & +x_2 & +x_3 & & & & = 3 \\
 & & & & x_4 & +x_5 & +x_6 & = 4 \\
 & x_1 & & & +x_4 & & & = 2 \quad . \quad (2.3.47) \\
 & & x_2 & & & +x_5 & & = 1 \\
 & & & x_3 & & & +x_6 & = 4 \\
 & x_1 \geq 0 & x_2 \geq 0 & x_3 \geq 0 & x_4 \geq 0 & x_5 \geq 0 & x_6 \geq 0 &
 \end{array}$$

sub2:

$$\begin{array}{rcccccccc}
 \min & 1x_7 & +2x_8 & +3x_9 & +4x_{10} & +5x_{11} & +6x_{12} & = z_2 \\
 \text{ràng buộc} & x_7 & +x_8 & +x_9 & & & & = 4 \\
 & & & & x_{10} & +x_{11} & +x_{12} & = 5 \\
 & x_7 & & & +x_{10} & & & = 3 \quad . \quad (2.3.48) \\
 & & x_8 & & & +x_{11} & & = 3 \\
 & & & x_9 & & & +x_{12} & = 3 \\
 & x_7 \geq 0 & x_8 \geq 0 & x_9 \geq 0 & x_{10} \geq 0 & x_{11} \geq 0 & x_{12} \geq 0 &
 \end{array}$$

sub3:

$$\begin{array}{rcccc}
 \min & 7x_{13} & -10x_{14} & = z_3 \\
 \text{ràng buộc} & x_{13} & -x_{14} & = 1 \quad . \quad (2.3.49) \\
 & x_{13} \geq 0 & x_{14} \geq 0 &
 \end{array}$$

Nghiệm cơ sở chấp nhận được cho ba bài toán con là:

$$\begin{array}{l}
 \mathbf{sub1} : u_1^{11} = 2, \quad u_2^{11} = 1, \quad u_4^{11} = 0, \quad u_6^{11} = 4 \\
 \mathbf{sub2} : u_2^{21} = 1, \quad u_3^{21} = 3, \quad u_4^{21} = 3, \quad u_5^{21} = 2 \\
 \mathbf{sub3} : u_1^{31} = 1
 \end{array}$$

Pha 2 mục tiêu được ước tính bởi:

$$c^1 = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{pmatrix}, c^2 = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{pmatrix}, c^3 = \begin{pmatrix} 7 \\ -10 \end{pmatrix}$$

Pha 1 mục tiêu được ước tính bởi:

$$w^1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, w^2 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, w^3 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Và ma trận chuyển đổi hệ số của bài toán Restricted Master Problem xác định bởi:

$$A^1 = \begin{pmatrix} 3 & 2 & 1 & 6 & 5 & 4 \\ 1 & 8 & 3 & 7 & 1 & 4 \end{pmatrix}$$

$$A^2 = \begin{pmatrix} 8 & 5 & 7 & 3 & 4 & 1 \\ 5 & 2 & 5 & 3 & 2 & 6 \end{pmatrix}$$

$$A^3 = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

kết quả thu được là:

$$G^{11} = A^1 u^{11} = \begin{pmatrix} 24 \\ 26 \end{pmatrix}$$

$$G^{21} = A^2 u^{21} = \begin{pmatrix} 43 \\ 30 \end{pmatrix}$$

$$G^{31} = A^3 u^{31} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$$

và

$$g^{11} = (w^1)^T u^{11} = 0$$

$$g^{21} = (w^2)^T u^{21} = 0$$

$$g^{31} = (w^3)^T u^{31} = 0$$

Pha 1 bài toán Restricted Master Problem là:

$$\begin{array}{rllllll}
 \min & 0\alpha_{11} & +0\alpha_{21} & +0\alpha_{31} & +1x_{13} & +1x_{14} & = w \\
 \text{ràng buộc} & 28\alpha_{11} & +43\alpha_{21} & +\alpha_{31} & -x_{13} & & = 64 \\
 & 26\alpha_{11} & +30\alpha_{21} & +3\alpha_{31} & & +x_{14} & = 63 \\
 & \alpha_{11} & & & & & = 1 \\
 & & \alpha_{21} & & & & = 1 \\
 & & & \alpha_{31} & & & = 1 \\
 & \alpha_{ki} \geq 0 & & & & & x_j \geq 0.
 \end{array} \quad (2.3.50)$$

Sau khi giải pha 1 của bài toán Restricted Master Problem (2.3.50) tất cả 5 biến là cơ sở và chúng ta thu được nhân tử:

$$\bar{\pi} = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \text{ và } \bar{\gamma} = \begin{pmatrix} -2 \\ 13 \\ -2 \end{pmatrix}$$

sử dụng các nhân tử ta tính được cước phí điều chỉnh cho mục tiêu của bài toán con:

$$\begin{aligned}
 \bar{\rho}^1 &= w^1 - (A^1)^T \bar{\pi} = (2 \quad -6 \quad -2 \quad -1 \quad 4 \quad 0)^T \\
 \bar{\rho}^2 &= w^2 - (A^2)^T \bar{\pi} = (3 \quad 3 \quad 2 \quad 0 \quad 2 \quad -5)^T \\
 \bar{\rho}^3 &= w^3 - (A^3)^T \bar{\pi} = (-2 \quad -2)^T
 \end{aligned}$$

Sau khi giải ba bài toán con với mục tiêu đã được điều chỉnh ta thu được các nghiệm:

$$\begin{aligned}
 \text{sub1} &: u_2^{12} = 1, \quad u_3^{12} = 2, \quad u_4^{12} = 2, \quad u_6^{12} = 2 \quad z_1 = -12 \\
 \text{sub2} &: u_1^{22} = 1, \quad u_2^{22} = 3, \quad u_4^{22} = 2, \quad u_6^{22} = 3 \quad z_2 = -3 \\
 \text{sub3} &: v_1^{32} = 1, \quad v_2^{32} = 1. \text{ nghiệm thuần nhất.}
 \end{aligned}$$

Do

$$z_1 < \bar{\gamma}_1, \quad z_2 < \bar{\gamma}_2 \quad z_3 < \bar{\gamma}_3$$

vì vậy chúng ta sẽ bổ xung bài toán Restricted Master Problem với các hệ số sau:

$$G^{12} = A^1 u^{12} = \begin{pmatrix} 24 \\ 36 \end{pmatrix}$$

$$G^{22} = A^2 u^{22} = \begin{pmatrix} 32 \\ 35 \end{pmatrix}$$

$$H^{32} = A^3 v^{32} = \begin{pmatrix} 3 \\ 7 \end{pmatrix}$$

và

$$g^{12} = (w^1)^T u^{12} = 0$$

$$g^{22} = (w^2)^T u^{22} = 0$$

$$h^{32} = (w^3)^T v^{32} = 0$$

Pha 1 mới của bài toán Restricted Master Problem là:

$$\begin{array}{rllllllll} \min & 0\alpha_{11} & +0\alpha_{12} & +0\alpha_{21} & +0\alpha_{22} & +0\alpha_{31} & +0\alpha_{32} & +1x_{13} & +1x_{14} & = w \\ \text{ràng buộc} & 28\alpha_{11} & +24\alpha_{12} & +43\alpha_{21} & +32\alpha_{22} & +\alpha_{31} & +3\beta_{31} & -x_{13} & & = 64 \\ & 26\alpha_{11} & +36\alpha_{12} & +30\alpha_{21} & +35\alpha_{22} & +3\alpha_{31} & +7\beta_{31} & & +x_{14} & = 63 \\ & \alpha_{11} & +\alpha_{12} & & & & & & & = 1 \\ & & & \alpha_{21} & +\alpha_{22} & & & & & = 1 \\ & & & & & \alpha_{31} & & & & = 1 \end{array}$$

$$\alpha_{ki} \geq 0 \quad \beta_{31} \geq 0 \quad x_j \geq 0.$$

(2.3.51)

Sau khi giải pha 1 này ta thu được một nghiệm cơ sở với các biến giả đã ra khỏi cơ sở chúng ta tiếp tục pha 2 với các hệ số được thay đổi như sau:

$$g^{11} = (c^1)^T u^{11} = 28$$

$$g^{21} = (c^2)^T u^{21} = 28$$

$$g^{31} = (c^3)^T u^{31} = 33$$

$$g^{12} = (c^1)^T u^{12} = 33$$

$$g^{22} = (c^2)^T u^{22} = 7$$

$$h^{32} = (c^3)^T v^{32} = -3$$

ở pha 2 này nhân tử là:

$$\bar{\pi} = \begin{pmatrix} -0.1630 \\ -0.3587 \end{pmatrix} \quad \text{và} \quad \bar{\gamma} = \begin{pmatrix} 41.2391 \\ 50.7717 \\ 8.2391 \end{pmatrix}$$

sử dụng các nhân tử ta tính được cước phí điều chỉnh cho mục tiêu của bài toán con:

$$\begin{aligned}\bar{\rho}^1 &= c^1 - (A^1)^T \bar{\pi} = (1.8477 \quad 5.1956 \quad 4.2391 \quad 7.4889 \quad 6.1737 \quad 8.0868)^T \\ \bar{\rho}^2 &= c^2 - (A^2)^T \bar{\pi} = (4.0975 \quad 3.5324 \quad 5.9345 \quad 5.5651 \quad 6.3694 \quad 8.3152)^T \\ \bar{\rho}^3 &= c^3 - (A^3)^T \bar{\pi} = (8.2391 \quad -8.2391)^T\end{aligned}$$

Sau khi giải ba bài toán con với mục tiêu đã được điều chỉnh ta thu được các nghiệm:

$$\begin{aligned}\mathbf{sub1} : u_1^{13} &= 2, \quad u_3^{13} = 1, \quad u_5^{13} = 1, \quad u_6^{13} = 3 \quad z_1 = 38.3686 \\ \mathbf{sub2} : u_2^{23} &= 3, \quad u_3^{23} = 1, \quad u_4^{23} = 4, \quad u_6^{23} = 2 \quad z_2 = -3 \\ \mathbf{sub3} : u_1^{33} &= 1, \quad z_3 = 8.3291\end{aligned}$$

Do

$$z_1 < \bar{\gamma}_1, \quad z_2 < \bar{\gamma}_2 \quad z_3 = \bar{\gamma}_3$$

vì vậy chúng ta chuyển đổi và thêm nghiệm của **sub1** và **sub2** vào trong bài toán

Restricted Master Problem với hệ số chuyển đổi là:

$$\begin{aligned}G^{13} &= A^1 u^{13} = \begin{pmatrix} 24 \\ 18 \end{pmatrix} \\ G^{23} &= A^2 u^{23} = \begin{pmatrix} 33 \\ 32 \end{pmatrix}\end{aligned}$$

và

$$\begin{aligned}g^{13} &= (c^1) T u^{13} = 28 \\ g^{23} &= (c^2) T u^{23} = 33\end{aligned}$$

Bài toán Restricted Master Problem mới là:

$$\begin{array}{rcccccccc} \min & 28\alpha_{11} & +28\alpha_{12} & +28\alpha_{13} & +33\alpha_{21} & +33\alpha_{22} & +33\alpha_{23} & +7\alpha_{31} & +(-3)\alpha_{32} & = z \\ \text{ràng buộc} & 28\alpha_{11} & +24\alpha_{12} & +24\alpha_{13} & +43\alpha_{21} & +32\alpha_{22} & +33\alpha_{23} & +\alpha_{31} & +3\beta_{31} & = 64 \\ & 26\alpha_{11} & +36\alpha_{12} & +18\alpha_{13} & +30\alpha_{21} & +35\alpha_{22} & +32\alpha_{23} & +3\alpha_{31} & +7\beta_{31} & = 63 \\ & \alpha_{11} & +\alpha_{12} & +\alpha_{13} & & & & & & = 1 \\ & & & & \alpha_{21} & +\alpha_{22} & +\alpha_{23} & & & = 1 \\ & & & & & & & \alpha_{31} & & = 1 \\ & \alpha_{ki} \geq 0 & \beta_{31} \geq 0 & x_j \geq 0. & & & & & & \end{array} \quad (2.3.52)$$

Nhân tử mới là:

$$\bar{\pi} = \begin{pmatrix} -0.0789 \\ -0.3948 \end{pmatrix} \quad \text{và} \quad \bar{\gamma} = \begin{pmatrix} 37.000 \\ 48.2367 \\ 8.2632 \end{pmatrix}$$

một lần nữa sử dụng các nhân tử ta tính được cước phí điều chỉnh cho mục tiêu của bài toán con:

$$\bar{\rho}^1 = c^1 - (A^1)^T \bar{\pi} = (1.6315 \quad 5.3164 \quad 4.2633 \quad 7.2370 \quad 5.7893 \quad 7.8940)^T$$

$$\bar{\rho}^2 = c^2 - (A^2)^T \bar{\pi} = (3.6052 \quad 3.1841 \quad 5.5263 \quad 5.4211 \quad 7.8940 \quad 8.4477)^T$$

$$\bar{\rho}^3 = c^3 - (A^3)^T \bar{\pi} = (8.2632 \quad 8.2632)^T$$

Sau khi giải ba bài toán con với mục tiêu đã được điều chỉnh ta thu được các nghiệm:

$$\mathbf{sub1} : u_1^{14} = 2, \quad u_3^{14} = 1, \quad u_5^{14} = 1, \quad u_6^{14} = 3 \quad z_1 = 37.000$$

$$\mathbf{sub2} : u_2^{24} = 1, \quad u_3^{24} = 3, \quad u_4^{24} = 3, \quad u_5^{24} = 2 \quad z_2 = 48.6327$$

$$\mathbf{sub3} : u_1^{33} = 1, \quad z_3 = 8.2632$$

Vì

$$z_1 = \bar{\gamma}_1, \quad z_2 = \bar{\gamma}_2 \quad z_3 = \bar{\gamma}_3$$

nên phương án này là tối ưu. Thay trở lại bài toán ban đầu với phương án tối ưu này ta thu được $f(x) = 68$ là kết quả tối ưu của hàm mục tiêu.

Chương 3

Thực thi thuật toán phân rã

Dantzig - Wolfe

3.1. Chuyển bài toán về dạng thực thi được trên máy tính.

Các mô hình bài toán QHTT phát sinh trong thực tế thường ở dạng tổng quát. Do vậy, để thực thi phương pháp phân rã Dantzig-Wolfe trên máy tính, ta cần xử lý bài toán này qua các bước tiền xử lý. Đối với phương pháp phân rã Dantzig - Wolfe, việc thực thi trên máy tính chỉ được thực hiện trên bài toán QHTT chính tắc và bài toán *Restricted Master Program*. Có nhiều giai đoạn trong việc xử lý này, chẳng hạn ta sẽ thực thi các khâu cơ bản sau:

- Chuyển bài toán về dạng chính tắc.
- Loại bỏ các ràng buộc dư thừa và tính không chấp nhận của bài toán (tức là tập ràng buộc mâu thuẫn).
- Xác định cấu trúc của ma trận ràng buộc.
- Chọn cách phân rã bài toán.

3.1.1. Chuyển bài toán QHTT về dạng chính tắc

Trên thực tế, các bài toán QHTT trong ứng dụng thường có dạng tổng quát như sau:

$$\begin{aligned}
 f(x) &:= \sum_{j=1}^n c_j x_j \rightarrow \min(\max) \\
 \sum_{j=1}^n a_{ij} x_j &\geq b_i, \quad i = 1, \dots, m_1 \\
 \sum_{j=1}^n a_{ij} x_j &\leq b_i, \quad i = m_1 + 1, \dots, m_2 \\
 \sum_{j=1}^n a_{ij} x_j &= b_i, \quad i = m_2 + 1, \dots, m \\
 x_j &\geq 0 && j = 1, \dots, n_1 \\
 x_j &\leq 0 && j = n_1 + 1 \dots, n_2 \\
 x_j &\text{ tự do, } && j = n_2 + 1, \dots, n
 \end{aligned}$$

Ta biết rằng, mọi bài toán QHTT tổng quát đều có thể đưa về dạng chính tắc bằng các phép biến đổi tương đương sau:

$$\begin{aligned}
 f(x^*) = \min\{f(x), x \in D\} &\leftrightarrow -f(x^*) = \max\{-f(x), x \in D\} \\
 \sum_{j=1}^n a_{ij} x_j \geq b_i &\leftrightarrow \sum_{j=1}^n (-a_{ij}) x_j \leq -b_i \\
 \sum_{j=1}^n a_{ij} x_j = b_i &\leftrightarrow \begin{cases} \sum_{j=1}^n a_{ij} x_j \geq b_i \\ \sum_{j=1}^n a_{ij} x_j \leq b_i \end{cases} \\
 \sum_{j=1}^n a_{ij} x_j \leq b_i &\leftrightarrow \sum_{j=1}^n a_{ij} x_j + x_{n+i} = b_i; \quad x_{n+i} \geq 0 \\
 \sum_{j=1}^n a_{ij} x_j \geq b_i &\leftrightarrow \sum_{j=1}^n a_{ij} x_j - x_{n+i} = b_i; \quad x_{n+i} \geq 0
 \end{aligned}$$

Trường hợp biến x_j không bị ràng buộc về dấu có thể thay bằng hiệu hai biến không âm dưới dạng

$$x_j = x'_j - x'_{n+j}; \quad x'_j \geq 0, x'_{n+j} \geq 0$$

Từ các nhận xét trên ta thấy rằng bất kì một bài toán quy hoạch tuyến tính nào cũng có thể đưa về dạng chính tắc sau đây

$$\begin{aligned}
 f(x) &= \sum_{j=1}^n c_j x_j \rightarrow \min \\
 \sum_{j=1}^n a_{ij} x_j &= b_i \quad i = 1, \dots, m \quad . \\
 x_j &\geq 0 \quad j = 1, \dots, n
 \end{aligned} \tag{3.1.1}$$

3.1.2. Loại bỏ các ràng buộc thừa và ràng buộc mâu thuẫn.

Đối với bài toán QHTT có thể có những ràng buộc dư thừa, chẳng hạn $x_1 + x_2 = 5$ và $2x_1 + 2x_2 = 10$ là hai ràng buộc cùng tồn tại, khi đó một trong hai ràng buộc là dư thừa. Ta có thể loại bỏ một trong hai ràng buộc này. Ngoài ràng buộc dư thừa dạng này, còn có các ràng buộc dư thừa dưới dạng $x_1 \geq 0$ và $x_1 \geq -2$, khi đó ràng buộc $x_1 \geq -2$ là dư thừa, ta có thể loại bỏ.

Ràng buộc mâu thuẫn là các ràng buộc làm cho tập nghiệm của bài toán trở nên rỗng, chẳng hạn có hai ràng buộc $x_1 + x_2 = 5$ và $x_1 + x_2 = -6$, khi đó hai ràng buộc này là mâu thuẫn, bài toán của ta sẽ có tập nghiệm là rỗng. Hoặc ràng buộc $x_1 \geq 0$ và $x_1 < 0$ là cặp ràng buộc mâu thuẫn. Điều này sẽ làm cho biến x_1 trở nên vô nghĩa. Người ta có thể vẫn giải quyết bài toán bằng cách xem xét lại sự hợp lý của một trong các ràng buộc mâu thuẫn này nhằm giữ lại ràng buộc hợp lý và loại bỏ đi những ràng buộc không hợp lý hoặc không cần thiết trong thực tế ứng dụng.

Trong ứng dụng việc xử lý các ràng buộc dư thừa và ràng buộc mâu thuẫn có thể thực hiện với các phép biến đổi đại số, chẳng hạn phép khử Gauss trong đại số tuyến tính. Nhưng nếu bài toán có kích thước lớn thì việc xử lý trở nên khó khăn. Nếu bài toán có cấu trúc đặc biệt, ta có thể phân rã và xử lý từng phần riêng rẽ, chẳng hạn bài toán có cấu trúc góc khối như ví dụ ở mục 2.3.3 ở Chương 2, ta có thể xử lý thành 3 nhóm biến $x^1 = (x_1, \dots, x_6)^T$, $x^2 = (x_7, \dots, x_{12})^T$ và $x^3 = (x_{13}, x_{14})^T$ một cách hoàn toàn riêng rẽ.

Phép loại bỏ ràng buộc dư thừa nhằm đưa ma trận ràng buộc A về ma trận có hạng đủ, nghĩa là $\text{rank}A = m \leq n$. Việc này giúp cho các thuật toán đơn hình làm việc đúng đắn.

3.1.3. Kiểm tra cấu trúc của bài toán.

Đối với bài toán lớn, việc xác định kích thước và cấu trúc của ma trận ràng buộc là rất quan trọng, nó đóng vai trò quyết định trong các phương pháp giải quyết bài toán. Cấu trúc của bài toán sẽ quyết định các bước sau đây trong quá trình thiết kế thuật toán và chương trình:

- Lựa chọn phương pháp phân rã Dantzig-Wolfe, tức là lựa chọn cách phân rã

tập P để có số đỉnh và số hướng cực biên hợp lý, không quá lớn, có thể thực thi tính toán dễ dàng.

- Lựa chọn phương pháp lưu trữ ma trận. Điều này rất quan trọng trong xây dựng chương trình, vì nó quyết định đến hiệu suất thực thi: thời gian tính toán, không gian lưu trữ, độ chính xác, tính ổn định và bền vững của thuật toán...
- Lựa chọn cấu trúc dữ liệu đầu vào. Đối với bài toán lớn không thể nhập dữ liệu bằng tay, không thể lưu trữ dữ liệu dưới dạng file text thông thường, mà cần phải thiết kế một số cấu trúc dữ liệu riêng. Cấu trúc dữ liệu phổ biến cho bài toán QHTT hiện nay là định dạng MPS.
- Xây dựng các phép toán. Đối với bài toán QHTT, một trong những phép toán cơ bản nhất là phép toán nhân ma trận và véc tơ. Khi bài toán kích thước lớn, ma trận thưa, ta không thể dùng các phép nhân ma trận với véc tơ thông thường, nó sẽ làm giảm tốc độ tính toán. Ngoài ra do cấu trúc dữ liệu lưu trữ của ma trận được lựa chọn riêng, nên việc thiết kế lại các phép toán cơ bản là khâu không thể thiếu trong xây dựng chương trình.

Mặc dù bài toán QHTT có nhiều dạng cấu trúc khác nhau, có nhiều bước xử lý. Tuy nhiên, khóa luận này chỉ đề cập đến việc giải quyết bài toán QHTT có cấu trúc cho hai bài toán có cấu trúc đặc biệt là hệ khối góc (block angular system) và bài toán cấu trúc bậc thang (staircase structured problems).

3.1.4. Xử lý kết quả.

Các thuật toán sau khi thực thi sẽ cho kết quả của bài toán trung gian, tức là bài toán QHTT chính tắc, nghiệm của bài toán này chưa phải là kết quả chúng ta mong muốn, nghiệm chúng ta cần là nghiệm của bài toán QHTT tổng quát ban đầu. Do vậy, việc xử lý kết quả đầu ra cũng là một khâu cơ bản trong thực thi thuật toán trên máy tính. Việc khôi phục nghiệm của bài toán ban đầu từ nghiệm của bài toán QHTT chính tắc thường được thực hiện bằng một số phép biến đổi đại số đơn giản. Do vậy khâu xử lý này thường ít tốn kém về chi phí thực thi như: thời gian, không gian bộ nhớ lưu trữ...

3.2. Triển khai thuật toán

3.2.1. Chọn mô hình triển khai thuật toán.

Thuật toán phân rã Dantzig-Wolfe đã được xây dựng về mặt lý thuyết, song để triển khai chi tiết, ta cần xử lý rất nhiều các công việc: Chọn ngôn ngữ lập trình, chọn cấu trúc dữ liệu, thiết kế các thủ tục con... Để triển khai chương trình máy tính cho phương pháp phân rã Dantzig-Wolfe, trong khóa luận này đã đưa ra một số lựa chọn chi tiết như sau:

- Môi trường lập trình được dùng ở đây là Visual Basic (trên hệ điều hành Windows XP). Ưu điểm của môi trường này là có hướng đối tượng và dễ thực thi lập trình, dễ sử dụng hệ thống giao diện có sẵn. Có thể tích hợp với các thư viện có sẵn nhằm tận dụng tối đa tài nguyên sẵn có.
- Các thủ tục xử lý được thực hiện một cách tuần tự, chưa được song song hóa ở các bước xử lý bài toán con.
- Dữ liệu đầu vào được lưu trữ ở các file Excel.

3.2.2. Cấu trúc chương trình.

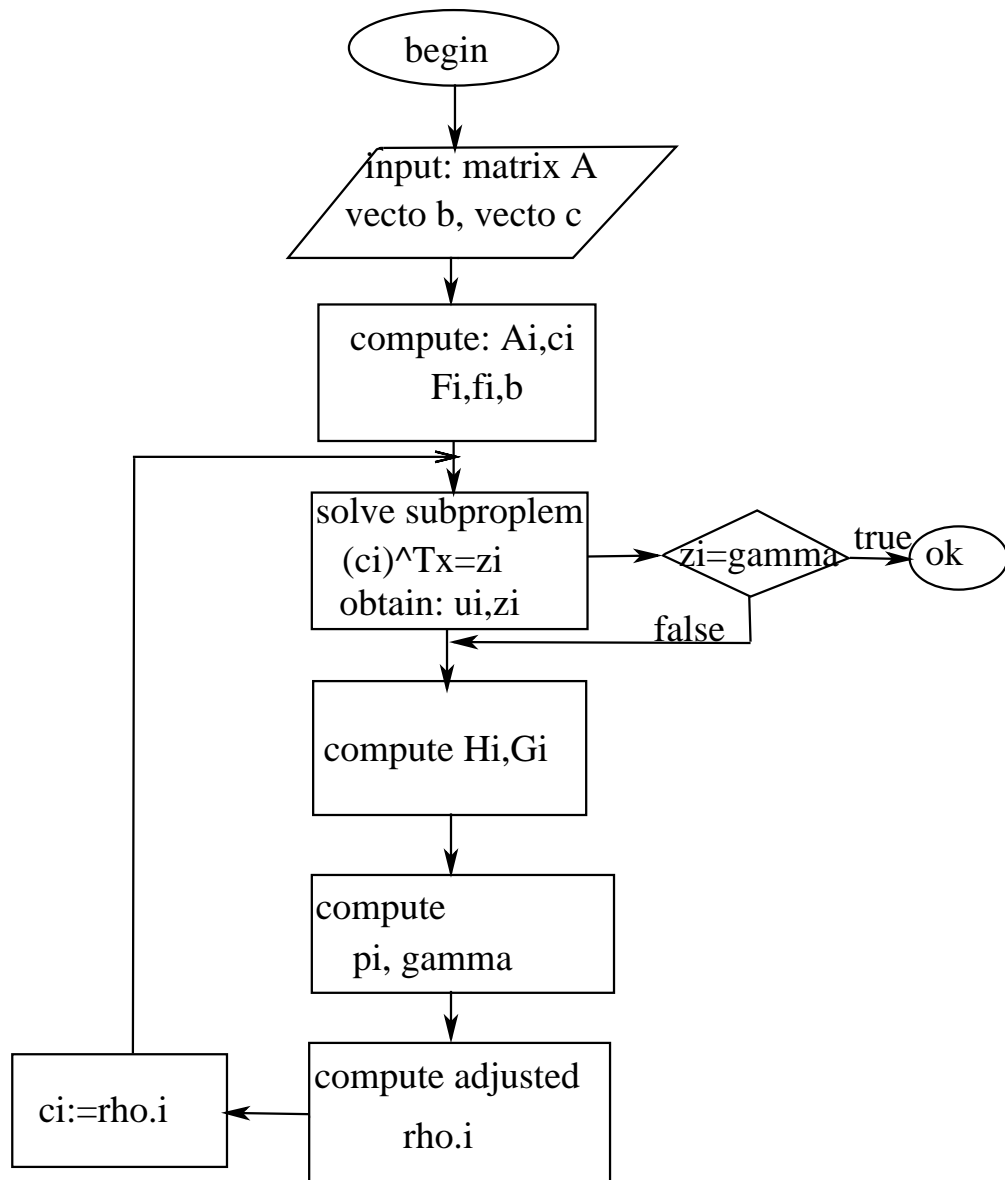
Chương trình sẽ bao gồm một số Module cơ bản như sau:

1. *Module đọc dữ liệu đầu vào:* Dữ liệu đọc từ file sau đó được lưu trữ vào bộ nhớ, dưới dạng ma trận và véc tơ.
2. *Module chuyển đổi bài toán:* Module này làm nhiệm vụ chuyển đổi bài toán về dạng chính tắc, thực thi loại bỏ ràng buộc dư thừa, ràng buộc mâu thuẫn, xây dựng bài toán phụ.
3. *Module thuật toán đơn hình:* Đây là module cơ bản thể hiện thuật toán đơn hình giải bài toán QHTT, module này sẽ được gọi nhiều lần trong thuật toán để giải quyết các bài toán phụ và bài toán con.

4. *Module xử lý kết quả:* Module này nhằm khôi phục lại nghiệm cho bài toán ban đầu từ nghiệm giải được và xuất kết quả ra file hoặc màn hình quan sát.

5. *Module chính:* Đây là module kết nối tất cả các giai đoạn của phương pháp phân rã Dantzig-Wolfe.

Sơ đồ khối của thuật toán



Hình 3.1: sơ đồ thuật toán

3.2.3. Một số thủ tục cơ bản của chương trình.

Chương trình thể hiện phương pháp phân rã Dantzig-Wolfe có nhiều module khác nhau, được thiết kế riêng rẽ thuận tiện cho việc kiểm tra, sửa lỗi, bảo trì... Dưới đây là một số module cơ bản nhất:

1. Private Function Doc_m(textfile As String) As Integer 'Hàm đọc số ràng buộc từ tệp.
2. Private Function DocMT(textfile As String, mask As String) As String 'Hàm đọc ma trận từ tệp.
3. Private Function DocVT(textfile As String, mask As String) As String 'Hàm đọc vec tơ từ tệp.
4. Private Function phan_ra(strA As String) As String 'Hàm phân rã ma trận A.
5. Private Function so_khoi(strPR As String) As Integer 'Hàm tìm số khối xuất hiện trong ma trận A.
6. Private Sub GiaiBT_DZW(textfile As String) 'Thủ tục giải bài toán D-W.

3.3. Các bước triển khai thực thi thuật toán.

Với mục đích đưa những thuật toán có sẵn vào những ứng dụng thực tế, khóa luận này sẽ trình bày tóm tắt lại những khâu cơ bản trong quá trình triển khai thực thi một thuật toán nhằm giải quyết các bài toán ứng dụng cụ thể. Tùy thuộc vào mục đích ứng dụng, thời gian thực hiện, chi phí, quy mô của bài toán ứng dụng, mà người ta lựa chọn những quy trình thực hiện khác nhau. Chẳng hạn dưới đây là một số bước cơ bản khi triển khai áp dụng:

- **Bước 1.** Xây dựng mô hình định tính cho bài toán bằng cách tham biến (tham số) hoá các thành phần của bài toán thực tế. Nói cách khác là lượng hoá hoặc hình thức hoá các đối tượng thực tế bằng các ký hiệu hoặc lược đồ... Trong một số trường hợp cần phân chia, hoặc hạn chế hay rút gọn bài toán thực tế thì mới có thể mô hình hoá được dưới dạng mô hình toán học.

- **Bước 2.** Xây dựng mô hình toán học, bằng cách liên kết các đối tượng theo các mối quan hệ thành các công thức tường minh hoặc quan hệ cụ thể. Đồng thời với nó là xác định mục tiêu (một hoặc nhiều mục tiêu) cho bài toán thực tế và lượng hoá nó thành hàm mục tiêu cho bài toán tối ưu.
- **Bước 3.** Sử dụng các công cụ của toán học, cụ thể là lý thuyết tối ưu để giải quyết bài toán. Bước này cần thực hiện một cách khoa học và tuân thủ theo yêu cầu đặt ra ban đầu. Đó là phân loại bài toán, nghiên cứu các tính chất định tính, sự tồn tại nghiệm, xây dựng thuật toán giải (chương trình hoá bằng chương trình máy tính hoặc xây dựng được một lược đồ tính toán (thủ công)), đánh giá hiệu quả của phương pháp trên các phương diện: thời gian, kinh phí, phương tiện, độ chính xác, độ ổn định và tính tin cậy...
- **Bước 4.** Áp dụng vào bài toán thực tế, phân tích và kiểm định với bài toán thực tế, đánh giá kết quả và mức độ phù hợp. Có thể xảy ra hai khả năng.
 1. Mô hình tính toán là phù hợp với thực tế. Khi đó cần tổng kết lại (thành tài liệu nghiên cứu và hướng dẫn) và có thể đưa ra áp dụng rộng rãi. Sau đó cần tính đến vấn đề kiểm định và bảo trì cũng như hướng phát triển của hệ thống trong tương lai.
 2. Mô hình tính toán không phù hợp với thực tế. Khi đó cần rà soát lại các bước thực hiện để tìm ra nguyên nhân của nó. Có thể có nhiều nguyên nhân, trong đó có các nguyên nhân chủ yếu thường gặp sau: Do lựa chọn mô hình hoá toán học chưa phù hợp, công cụ giải quyết chưa phù hợp (chưa có công cụ đủ mạnh), sai số của dữ liệu (do quá trình đo đạc, quan sát, ảnh hưởng của môi trường...), độ chính xác và độ ổn định của thuật toán chưa được kiểm định... Do vậy cần tìm cách để khắc phục hoặc tìm hướng mới để giải lại bài toán.

Phụ lục

Phụ lục này sẽ trình bày một số modules cơ bản của thuật toán đơn hình gốc của Dantzig, và một số module của phương pháp phân rã Dantzig-Wolfe dưới dạng ngôn ngữ Visual Basic, phiên bản 6.0.

Cụ thể sẽ trình bày các thủ tục sau:

1. Modul thuật toán đơn hình gốc của Dantzig

noindent1. Modul thuật toán đơn hình gốc của Dantzig

```
Private Sub QHTT()  
    'Bo xung vao ma tran A  
    Dim i, j As Integer  
    Dim b_temp(30) As Double  
    Dim a_temp(100, 100) As Double  
    For i = 0 To m - 1  
        a(i + 1, 0) = b(i + 1) 'Cot 0 ma tran a  
    Next  
    For i = 0 To n - 1  
        a(0, i + 1) = c(i + 1)  
    Next  
  
    'Phuong phap don hinh  
    Dim toi_uu As Boolean  
    toi_uu = False  
    Dim var_lap As Integer  
    var_lap = 0  
    While toi_uu = False  
        'Tinh cac uoc luong delta  
        For j = 1 To n  
            a(m + 1, j) = 0  
            For i = 1 To m
```

```

        a(m + 1, j) = a(m + 1, j) + acb(i) * a(i, j)
    Next
    a(m + 1, j) = a(m + 1, j) - a(0, j)
'Dong thu m+1 trong ma tran a luu tru cac gia tri delta
Next

' Hien thi bang don hinh
display (var_lap * (m + 2) + 1)

'Tim delta max va s
Dim s As Integer
s = 1
Dim delta As Double
delta = a(m + 1, s)
For j = 1 To n
    If a(m + 1, j) > delta Then
        delta = a(m + 1, j)
        s = j
    End If
Next
'MsgBox (CStr(delta))

' Kiem tra dieu kien toi uu
If delta > 0 Then
    toi_uu = False
Else
    toi_uu = True
    GoTo Ketthuc
End If

'Kiem tra vo nghiem
Dim voNo As Boolean
voNo = False
For j = 1 To n
    If a(m + 1, j) > 0 Then
        dem = 1
        Do While dem <= m And a(dem, j) <= 0

```

```

        dem = dem + 1
    Loop
    If dem > m Then
        voNo = True
        MsgBox ("BAI TOAN VO NGHIEM")
        Exit Sub
    Else
        voNo = False
    End If
End If
Next

'Tim an loai ra
Dim r As Integer
r = 1
If a(r, s) < 0 Then
    r = r + 1
End If
For i = r To m
    If a(i, s) > 0 Then
        For j = 1 To n
            If a(r, j) / a(r, s) > a(i, j) / a(i, s) Then
                r = i
            End If
        Next
    End If
Next
End If
Next

'Bien doi bang
acb(r) = a(0, s)
cb(r) = s
Dim ars, ais As Double
ars = a(r, s)
'Bien doi cac phan tu con lai
For i = 1 To m
    For j = 1 To n
        If i <> r Then

```

```

        b_temp(i) = b(i) - (b(r) / ars) * a(i, s)
        a_temp(i, j) = a(i, j) - (a(r, j) / ars) * a(i, s)
    Else
        a_temp(r, j) = a(r, j) / ars
        b_temp(r) = b(r) / ars
    End If
Next
Next

For i = 1 To m
    For j = 1 To n
        a(i, j) = a_temp(i, j)
        a(i, 0) = b_temp(i)
        b(i) = b_temp(i)
    Next
Next

```

Ketthuc:

```
var_lap = var_lap + 1
```

```
Wend
```

```
End Sub
```

2. Modul thuật toán phân rã Dantzig-Wolfe.

```
Private Function Doc_m(textfile As String) As Integer
```

```
    Dim m As Integer
```

```
    'Doc du lieu tu tep
```

```
    Doc_m = m
```

```
End Function
```

```
-----
Private Function Doc_n(textfile As String) As Integer
```

```
    Dim n As Integer
```

```
    'Doc du lieu tu tep gan vao bien n
```

```
    Doc_n = n
```

```
End Function
```

```
-----
Private Function DocMT(textfile As String, mask As String) As String
```

```

Dim strA As String
'Doc du lieu tu tep luu cac phan tu cua mang vao StrA
'Cac phan tu cua chuoi strA cach nhau boi dau cach
'Cac dong cua ma tran cach nhau boi dau (,)
DocMT = strA
End Function
-----
Private Function DocVT(textfile As String, mask As String) As String
Dim strVT As String
'Doc du lieu tu tep
'Luu du lieu doc duoc la mot vec to vao strVT
'Cac phan tu cua strVT cach nhau boi dau cach
DocVT = strVT
End Function
-----
Private Function phan_ra(strA As String) As String
Dim strPR As String
'strPR lu tru cac diem phan ra
'Cac diem phan ra cach nhau boi dau (,)
'Diem dau la (0 0) va diem cuoi la (m n)
'Moi diem la mot cap gom 2 so la chi so hang va chi so cot cua ma tran A
'Cac phan tu cua strPR cac nhau boi dau cach
phan_ra = strPR
End Function
-----
Private Function so_khoi(strPR As String) As Integer
'Tim so khoi phan ra
Dim N_khoi As Integer
'So khoi phan ra bang so diem phan ra - 2
so_khoi = N_khoi '= so dau (,)
End Function
-----
Private Sub GiaiBT_DZW(textfile As String)
Dim m, n, k, i, j, N_khoi As Integer
m = Doc_m(textfile)
n = Doc_n(textfile)
Dim A(100, 100) As Double ' khai bao ma tran A

```



```

Dim b(100) As Double ' Khai bao vec to b
Dim c(100) As Double ' Khai bao vecto c
ReDim A(m, n) '
ReDim c(n)
ReDim b(m)

Dim strA, strb, strc, strPR As String
strA = DocMT(textfile, "a=")
strb = DocVT(textfile, "b=")
strc = DocVT(textfile, "c=")
Dim strtemp(n) As String
strtemp = Split(strA, ",", -1, 1)
Dim temp
For k = 1 To UBound(strtemp)
    temp = Split(strtemp(k), " ", -1, 1)
    For i = 1 To m
        For j = 1 To n
            A(i, j) = temp(i)
        Next
    Next
Next
Next
b = Split(strb, " ", -1, 1)
c = Split(strc, " ", -1, 1)
strPR = phan_ra(strA)
N_khoi = so_khoi(strPR)
' Khai bao
Dim Ak, fk, ck, bk
For k = 1 To N_khoi
    ' Tinh Ak, Fk, ck, bk
Next
Dim u(n), v(n)
Dim toi_uu, lap As Boolean
toi_uu = False
lap = False
While toi_uu = False
    For k = 1 To N_khoi
        ' Giai cac bai toan con Subk
        ' Output tinh duoc cac uk, vk
    
```

```

Next
'Chon co so xuat phat la wk la cac vec to 0
'Tim G, H
Dim numberOfRowAk As Integer
For k = 1 To UBound(uk)
    For i = 1 To numberOfRowAk
        tg1 = 0
        tg2 = 0
        For j = 1 To UBound(uk)
            tg1 = tg1 + Ak(i, j) * uk(j)
            tg2 = tg2 + Ak(i, j) * vk(j)
        Next
        gk(i) = tg1
        Hk(i) = tg2
    Next
Next
Next
'Tim gk
For k = 1 To N_khoi
    tg = 0
    For i = 1 To UBound(wk)
        tg = tg + chuyevi(wk(i)) * uk(i)
    Next
    gk = tg
Next
' Xay dung bai toan Restricted Master
If lap = False Then
    'Giai bai toan Restricted Master (RMas)'
    If RMas = "VoNo" Then
        'Ket thuc: Ket luan bai toan goc vo nghiem
    Else
        'Tinh pi va gamma
    End If
Else
    'Chen Gk va Hk moi vao bai toan RMas ta duoc bai toan NewRMas
    'Giai bai toan NewRMas
    If NewRMas = "VoNo" Then
        'Ket thuc: Ket luan bai toan toan goc vo nghiem
    
```

```

Else
    'Tinh pi va gamma
End If
End If
' Tinh rho
For k = 1 To N_khoi
    For i = 1 To UBound(wk)
        tg = 0
        For j = 1 To UBound(pi)
            tg = tg - chuyenvi(Ak(i, j)) * pi
        Next
        rhok(i) = tg + wk(i)
    Next
Next
'Gan c = rho
For k = 1 To N_khoi
    For i = 1 To UBound(ck)
        ck(i) = rhok(i)
    Next
Next
'Giai bai toan Subk
'Output: uk va zk
Next
'Kiem tra dieu kien toi uu
For k = 1 To N_khoi
    If z(k) < gamma(k) Then
        ' Tinh lai Gk, Hk
        dem = dem + 1
        lap = True
    End If
Next
If dem = 0 Then
    toi_uu = True
    ' Ket luan phuong an toi uu cua bai toan goc
End If
Loop

```

End Sub

Kiểm thử kết quả ví dụ trình bày ở chương 2 bằng thuật toán đơn hình

Input

```
m := 13
n := 14
c[i] := 1 2 3 4 5 6 1 2 3 4 5 6 7 -10
b[i] := 64 63 3 4 2 1 4 4 5 3 3 3 1
a[i,j] := 3 2 1 6 5 4 8 5 7 3 4 1 1 2
a[i,j] := 1 8 3 7 1 4 5 2 5 3 2 6 3 4
a[i,j] := 1 1 1 0 0 0 0 0 0 0 0 0 0 0
a[i,j] := 0 0 0 1 1 1 0 0 0 0 0 0 0 0
a[i,j] := 1 0 0 1 0 0 0 0 0 0 0 0 0 0
a[i,j] := 0 1 0 0 1 0 0 0 0 0 0 0 0 0
a[i,j] := 0 0 1 0 0 1 0 0 0 0 0 0 0 0
a[i,j] := 0 0 0 0 0 0 1 1 1 0 0 0 0 0
a[i,j] := 0 0 0 0 0 0 0 0 0 1 1 1 0 0
a[i,j] := 0 0 0 0 0 0 1 0 0 1 0 0 0 0
a[i,j] := 0 0 0 0 0 0 0 1 0 0 1 0 0 0
a[i,j] := 0 0 0 0 0 0 0 0 1 0 0 1 0 0
a[i,j] := 0 0 0 0 0 0 0 0 0 0 0 0 1 -1
```

Ta có một số bước lặp minh họa bằng bảng đơn hình như sau. Tại bước lặp thứ 15 bài toán đã đạt giá trị tối ưu khi tất cả các biến giả bằng 0 và giá trị tối ưu đạt được là $f(x) = 68$ ta cũng được kết quả này khi giải với thuật toán Dantzig-Wolfe trong ví dụ minh họa ở chương 2.

he so	co so	P/A	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	Lamda
M	15	64	3	2	1	6	5	4	8	5	7	3	4	1	1	2	8
M	16	63	1	8	3	7	1	4	5	2	5	3	2	6	3	4	12.6
M	17	3	1	1	1	0	0	0	0	0	0	0	0	0	0	0	-
M	18	4	0	0	0	1	1	1	0	0	0	0	0	0	0	0	-
M	19	2	1	0	0	1	0	0	0	0	0	0	0	0	0	0	-
M	20	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	-
M	21	4	0	0	1	0	0	1	0	0	0	0	0	0	0	0	-
M	22	4	0	0	0	0	0	0	1	1	1	0	0	0	0	0	4
M	23	5	0	0	0	0	0	0	0	0	0	1	1	1	0	0	-
M	24	3	0	0	0	0	0	0	1	0	0	1	0	0	0	0	3
M	25	3	0	0	0	0	0	0	0	1	0	0	1	0	0	0	-
M	26	3	0	0	0	0	0	0	0	0	1	0	0	1	0	0	-
M	27	1	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	-
	F(x)	0	-1	-2	-3	-4	-5	-6	-1	-2	-3	-4	-5	-6	-7	10	
		160	6	12	6	15	8	10	15	9	14	8	8	9	5	5	

Hình 3.2: Bảng đơn hình ở bước lặp thứ 1

he so	co so	P/A	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	Lamda
M	15	40	3	2	1	6	5	4	0	5	7	-5	4	1	1	2	6.67
M	16	48	1	8	3	7	1	4	0	2	5	-2	2	6	3	4	6.86
M	17	3	1	1	1	0	0	0	0	0	0	0	0	0	0	0	-
M	18	4	0	0	0	1	1	1	0	0	0	0	0	0	0	0	4
M	19	2	1	0	0	1	0	0	0	0	0	0	0	0	0	0	2
M	20	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	-
M	21	4	0	0	1	0	0	1	0	0	0	0	0	0	0	0	-
M	22	1	0	0	0	0	0	0	0	1	1	-1	0	0	0	0	-
M	23	5	0	0	0	0	0	0	0	0	0	0	1	1	1	0	-
1	24	3	0	0	0	0	0	0	1	0	0	1	0	0	0	0	-
M	25	3	0	0	0	0	0	0	0	1	0	0	1	0	0	0	-
M	26	3	0	0	0	0	0	0	0	0	1	0	0	1	0	0	-
M	27	1	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	-
	F(x)	3	-1	-2	-3	-4	-5	-6	0	-2	-3	-3	-5	-6	-7	10	
		115	6	12	6	15	8	10	0	9	14	-7	8	9	5	5	

Hình 3.3: Bảng đơn hình tại bước lặp thứ 2

he so	co so	P/A	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	Lamda
M	15	28	-3	2	1	0	5	4	0	5	7	-5	4	1	1	2	4
M	16	34	-6	8	3	0	1	4	0	2	5	-2	2	6	3	4	6.8
M	17	3	1	1	1	0	0	0	0	0	0	0	0	0	0	0	-
M	18	2	-1	0	0	0	1	1	0	0	0	0	0	0	0	0	-
4	19	2	1	0	0	1	0	0	0	0	0	0	0	0	0	0	-
M	20	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	-
M	21	4	0	0	1	0	0	1	0	0	0	0	0	0	0	0	-
M	22	1	0	0	0	0	0	0	0	1	1	-1	0	0	0	0	1
M	23	5	0	0	0	0	0	0	0	0	0	1	1	1	0	0	-
1	7	3	0	0	0	0	0	0	1	0	0	1	0	0	0	0	-
M	25	3	0	0	0	0	0	0	0	1	0	0	1	0	0	0	-
M	26	3	0	0	0	0	0	0	0	0	1	0	0	1	0	0	3
M	27	1	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	-
	F(x)	11	3	-2	-3	0	-5	-6	0	-2	-3	-3	-5	-6	-7	10	
		85	-9	12	6	0	8	10	0	9	14	-7	8	9	5	5	

Hình 3.4: Bảng đơn hình ở bước lặp thứ 3

he so	co so	P/A	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	Lamda
5	5	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	-
5	11	1.75	0	0	0	0	0	0	0	0	0.25	0	1	0	0	0.75	-
3	3	1.7	0	2.6	1	0	0	0	0	0	-0.7	0	0	0	0	1.1	-
6	6	2.3	0	-2.6	0	0	0	1	0	0	0.7	0	0	0	0	-1.1	-
1	1	1.3	1	-1.6	0	0	0	0	0	0	0.7	0	0	0	0	-1.1	-
4	4	0.7	0	1.6	0	1	0	0	0	0	-0.7	0	0	0	0	1.1	-
M	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-
7	13	1	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	-
2	8	1.25	0	0	0	0	0	0	0	1	-0.25	0	0	0	0	-0.75	-
1	7	2.75	0	0	0	0	0	0	1	0	1.25	0	0	0	0	0.75	-
M	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-
6	12	3	0	0	0	0	0	0	0	0	1	0	0	1	0	0	-
4	10	0.25	0	0	0	0	0	0	0	0	-1.25	1	0	0	0	-0.75	-
	F(x)	68	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Hình 3.5: bảng đơn hình ở bước lặp thứ 15

KẾT LUẬN

Trong khóa luận này em đã trình bày tư tưởng nội dung, trường hợp áp dụng cho thuật toán phân rã Dantzig-Wolfe và một số thuật toán là đơn hình đơn hình đối ngẫu để phục vụ mục đích giải bài toán quy hoạch kích thước lớn. Song song là ví dụ bằng số minh họa cho phương pháp cuối cùng là lập trình với thuật toán đơn hình đơn hình bằng ngôn ngữ VB phục vụ ở một số bước của thuật toán. Đóng góp chính của khóa luận bao gồm:

- 1 Tìm hiểu và trình bày lại nội dung phương pháp Dantzig-Wolfe.
- 2 Đề cập phân tích một số dạng bài toán có kích thước lớn trong thực tế áp dụng phương pháp Dantzig-Wole.
- 3 Lập trình với thuật toán đơn hình để phục vụ một số bước của thuật toán ngoài ra còn có tác dụng kiểm thử với các bài toán nhỏ.

Tuy nhiên do thời gian thực hiện khóa luận không nhiều còn có những sai sót em rất mong nhận được sự góp ý của quý thầy cô và bạn đọc.

Tài liệu tham khảo

- [1] Phan Quốc Khánh - Trần Huệ Nương, *Quy hoạch tuyến tính*, NXB Giáo dục (2000) 457 trang.
- [2] Nguyễn Ngọc Thắng - Nguyễn Đình Hóa, *Quy hoạch tuyến tính*, NXB Đại học Quốc Gia Hà Nội (2006) 148 trang.
- [3] Danzig G.B and Thapa M.N, *Linear programming 2 - Theory and Extensions*, Springer Verlag, New York Berlin, Heideiberg, (2003)
- [4] Jorge Nocedal Stephen J.Wright, *Numerical Optimization*, Springer (2006).