

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



MÔ HÌNH HÓA TOÁN HỌC (CO2011)

Bài tập lớn

QUY HOẠCH NGẪU NHIÊN VÀ CÁC ỨNG DỤNG

GVHD: Mai Xuân Toàn
Students: Hồ Minh Hưng - 2211361 - Nhóm trưởng
Võ Quốc Huy - 2211303
Mai Hoàng Huỳnh - 2211314
Phạm Duy Hưng - 2211379
Phạm Hoan Huy - 2211264

TP.HỒ CHÍ MINH, THÁNG 12/2023

Mục lục

1	Giới thiệu chung về Quy hoạch và tối ưu hóa ngẫu nhiên	2
1.1	Quy hoạch là gì? Quy hoạch ngẫu nhiên là gì? Tính không chắc chắn?	2
1.2	Các nội dung cơ bản, Giả thiết - Mục tiêu	2
2	Quy hoạch tuyến tính ngẫu nhiên một giai đoạn - Không phản hồi (One-Stage Stochastic linear programming - No recourse, 1-SLP)	4
2.1	CÁCH TIẾP CẬN 1: sử dụng Ràng buộc xác suất và Rủi ro chấp nhận được . .	5
2.2	CÁCH TIẾP CẬN 2: cho các Ràng buộc ngẫu nhiên $T(\alpha) x \leq h(\alpha)$	5
3	Quy hoạch ngẫu nhiên tổng quát có PHẢN HỒI (Generic Stochastic Programming (GSP) with RECOURSE)	6
4	Quy hoạch tuyến tính ngẫu nhiên hai giai đoạn (Two-Stage Stochastic linear programming (2-SLP))	7
4.1	Mô hình QHTT ngẫu nhiên hai giai đoạn có phản hồi (Two-stage SLP Recourse model) - dạng đơn giản	7
4.2	Mô hình QHTT ngẫu nhiên hai giai đoạn có phản hồi (Two-stage SLP Recourse model) - dạng chính tắc	8
5	Câu hỏi 1 BTL 2023	9
5.1	Giới thiệu	9
5.2	Cách giải	10
5.3	Thành lập bài toán cụ thể theo như yêu cầu câu hỏi 1	11
6	Câu hỏi 2 BTL 2023	14
6.1	Đề bài	14
6.2	Giới thiệu	14
6.3	Mục tiêu	15
6.4	Các bước thực hiện	15
6.5	Bảng kí hiệu	15
6.6	Các ràng buộc	16
6.7	Mô hình lập kế hoạch sơ tán ngẫu nhiên hai giai đoạn	17
6.8	Bài toán 1	18
6.8.1	Thuật toán đường đi ngắn nhất liên tiếp	19
6.8.2	Thuật toán nhân sửa (label correcting algorithm)	19
6.8.3	Giải thuật cho đồ thị vô hướng, đa đồ thị	21
6.8.4	Kết quả	21
6.9	Bài toán 2: Bài toán luồng chi phí tối thiểu với thời gian và dung lượng di chuyển của liên kết phụ thuộc vào thời gian	22
6.9.1	Thuật toán tìm đường ngắn nhất phụ thuộc thời gian	23
6.9.2	Đánh giá:	23
7	Ứng dụng II: Quy hoạch ngẫu nhiên trong Viễn thông	24
7.1	Đề bài:	24
7.2	Các bước thực hiện	24

1 Giới thiệu chung về Quy hoạch và tối ưu hóa ngẫu nhiên

1.1 Quy hoạch là gì? Quy hoạch ngẫu nhiên là gì? Tính không chắc chắn?

- **Tối ưu hóa toán học** là lĩnh vực về đưa ra quyết định, thường sử dụng các phương pháp toán học.
- **Quy hoạch ngẫu nhiên** (Stochastic Programming - SP) là lĩnh vực về việc đưa ra quyết định trong các trường hợp *không chắc chắn*.
Có thể xem nó như là 'Quy hoạch (Tối ưu hóa) toán học với các tham số *ngẫu nhiên*'.
- *Quy hoạch tuyến tính ngẫu nhiên* là quy hoạch tuyến tính (tức là hàm mục tiêu của nó tuyến tính) mà trong đó một số thông tin về bài toán xem như chưa chắc chắn.

1.2 Các nội dung cơ bản, Giả thiết - Mục tiêu

Định nghĩa 1 (Quy hoạch tuyến tính (LP) với tham số ngẫu nhiên - SLP).

Một quy hoạch tuyến tính ngẫu nhiên là:

$$\text{Minimize } Z = g(\mathbf{x}) = f(\mathbf{x}) = \mathbf{c}^T \cdot \mathbf{x}, \quad \text{với } A\mathbf{x} = \mathbf{b}, \text{ và } T\mathbf{x} \geq \mathbf{h}$$

với $\mathbf{x} = (x_1, x_2, \dots, x_n)$ (các biến quyết định),
ma trận thực A và vector \mathbf{b} nhất định (cho các điều kiện ràng buộc),
và các tham số ngẫu nhiên T, \mathbf{h} và $T\mathbf{x} \geq \mathbf{h}$ cho biết các ràng buộc về khả năng hoặc xác suất.

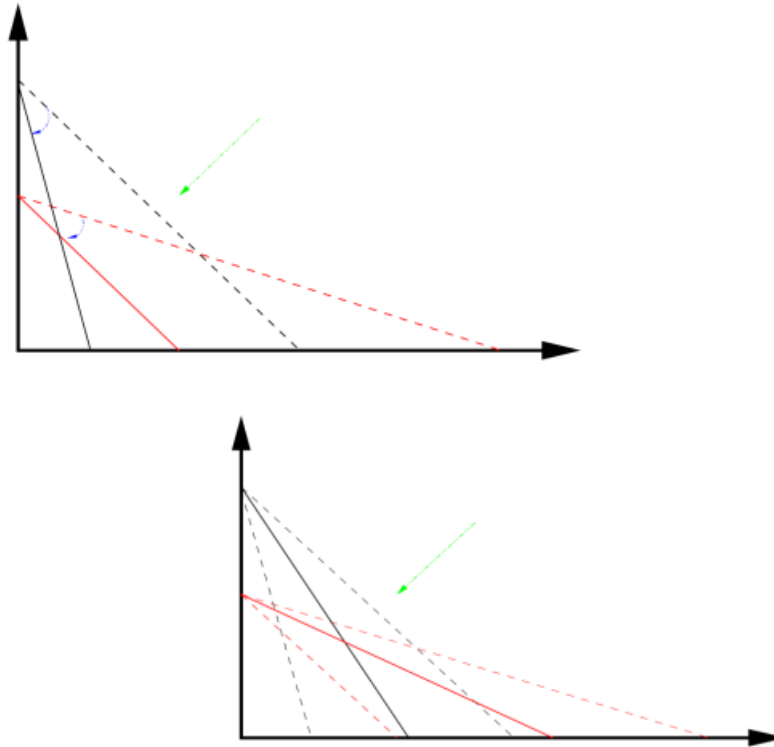
Ví dụ 1. Xét tối ưu hóa sau

$$\text{Minimize } z = x_1 + x_2, \quad \text{với } x_1 \geq 0, x_2 \geq 0. \quad \begin{cases} \omega_1 x_1 + x_2 \geq 7; \\ \omega_2 x_1 + x_2 \geq 4. \end{cases}$$

trong đó tham số ω_1 và ω_2 là biến đồng bộ (ngẫu nhiên), tuân theo phân phối sau:

$Uniform(a, b)$, trong đó $\omega_1 \sim Uniform(1, 4)$ và $\omega_2 \sim Uniform(1/3, 1)$.

- Khi cả $\omega_1 = \omega_2 = 1$ thì hai điều kiện ràng buộc trở thành:
 $x_1 + x_2 = 4$ tạo nên đường màu đỏ, và $x_1 + x_2 = 7$ tạo nên đường nét đứt màu xanh,
có thể thấy được vùng khả thi một cách đầy đủ thông qua mũi tên màu xanh lá. (Hình 1)
- **Vậy làm cách nào để giải bài toán này?**
 1. Phương pháp **wait - and - see (chờ và xem)**: Giả sử có thể chọn các biến quyết định $\mathbf{x} = [x_1, x_2]$ sau khi quan sát vector ngẫu nhiên $\omega = [\omega_1, \omega_2]$?. [biểu trưng một phần cho việc *không chắc chắn về dữ liệu* của bài toán.] Liệu có thể giải mà không cần phải chờ?
 2. Hoàn toàn có thể, giả sử cho $\mathbf{x} = [x_1, x_2]$ **trước khi biết** giá trị của $\omega = [\omega_1, \omega_2]$? Ta cần chọn xem cần phải làm gì với việc không biết ω này. Có hai cách như sau:
(a) Đoán sự không chắc chắn, và (b) Các ràng buộc xác suất (xem lại Khái niệm 1).



Hình 1: Bài toán quy hoạch ngẫu nhiên đơn giản với hai biến quyết định.

- (a) **Đoán sự không chắc chắn:** Ta đoán thử một vài giá trị hợp lý cho ω . Có ba giá trị như vậy, mỗi cái cho biết mức độ ‘rủi ro’
- Unbiased: Chọn giá trị trung bình cho mỗi tham số ω .
 - Pessimistic: Chọn giá trị trong trường hợp xấu nhất cho ω .
 - Optimistic: Chọn giá trị trong trường hợp tốt nhất cho ω .

Chẳng hạn, sử dụng **phương pháp Unbiased**, xem lại ví dụ 1, phân phối đồng nhất $\text{Uniform}(a, b)$ có giá trị trung bình là $\frac{a+b}{2}$, vì vậy $\hat{\omega} = (\frac{5}{2}, \frac{2}{3})$. Quy hoạch lúc này trở thành

$$\text{Minimize } z = x_1 + x_2$$

có giá trị tối ưu $z_1 = \frac{50}{11}$ tại điểm $\hat{x} = (\hat{x}_1, \hat{x}_2) = (18/11, 32/11)$

$$\text{nếu giả sử } \begin{cases} \frac{5}{2}x_1 + x_2 \geq 7; \frac{2}{3}x_1 + x_2 \geq 4; \\ x_1 \geq 0; x_2 \geq 0 \end{cases}$$

Đối với **phương pháp Pessimistic**, $\hat{\omega} = [1, \frac{1}{3}]$

$$\begin{cases} 1x_1 + x_2 \geq 7; 1/3x_1 + x_2 \geq 4; \\ x_1 \geq 0; x_2 \geq 0 \end{cases}$$

$\Rightarrow z_2 = 7$ tại điểm $\hat{x} = (\hat{x}_1, \hat{x}_2) = (0, 7)$

Đối với **phương pháp Optimistic**, $\hat{\omega} = (4, 1)$

$$\begin{cases} 4x_1 + x_2 \geq 7; 1x_1 + x_2 \geq 4; \\ x_1 \geq 0; x_2 \geq 0 \end{cases}$$

$\Rightarrow z_3 = \frac{7}{4}$ tại điểm $\hat{x} = (\hat{x}_1, \hat{x}_2) = (\frac{7}{4}, 0)$

2 Quy hoạch tuyến tính ngẫu nhiên một giai đoạn - Không phản hồi (One-Stage Stochastic linear programming - No recourse, 1-SLP)

Định nghĩa 2 (1 - SLP). Xét quy hoạch $LP(\alpha)$ sau được tham số hóa bởi vector ngẫu nhiên α :

$$\text{Minimize } Z = g(\mathbf{x}) = f(\mathbf{x}) = \mathbf{c}^T \cdot \mathbf{x} = \sum_{j=1}^n c_j x_j$$

phụ thuộc $Ax = b$, (các ràng buộc chắc chắn)

và $Tx \geq h$, (các ràng buộc ngẫu nhiên)

với các giả thiết sau

- Ma trận $T = T(\alpha)$ và vector $h = h(\alpha)$ biểu diễn sự không chắc chắn thông qua các ràng buộc ngẫu nhiên

$$T(\alpha)x \geq h(\alpha) \iff \alpha_1 x_1 + \dots + \alpha_n x_n \geq h(\alpha)$$

- Giá trị (T, h) chưa biết: chúng chưa biết được trước khi một trường hợp của mô hình xảy ra, $h(\alpha)$ chỉ phụ thuộc vào α_j ngẫu nhiên;

- Sự không chắc chắn** được mô tả bởi một phân phối xác suất của các tham số ngẫu nhiên $(\alpha_j) = \alpha$ vì thế quy hoạch tuyến tính xác định là trường hợp suy biến của quy hoạch tuyến tính ngẫu nhiên khi α_j là các hằng số,

- Ta xử lý các bài toán quyết định trong đó vector $\mathbf{x} = x_1, x_2 \dots x_n \in \mathcal{X}$ của các biến quyết định phải có trước khi việc hiện thực hóa vector tham số $\alpha \in \Omega$ được biết.
- Ta thường đặt các giới hạn trên và dưới cho \mathbf{x} thông qua miền $\mathcal{X} = \{x \in \mathbb{R}^n : l \leq x \leq u\}$.

CÁCH TIẾP CẬN: Trong quy hoạch ngẫu nhiên, ta sử dụng một số giả thiết và chân lý.

Giả thiết cơ bản- Chúng ta biết một phân phối xác suất (chung) của dữ liệu. Do đó, cách tiếp cận đầu tiên này đưa đến QHTT với ràng buộc xác suất.

Phân tích kịch bản (The Scenario Analysis-) là cách tiếp cận thứ hai, dù không hoàn hảo nhưng lại hữu ích. Cách này giả định rằng có hữu hạn các quyết định mà tự nhiên có thể tạo ra. Mỗi quyết định khả thi được gọi là **một kịch bản (scenario)**.

2.1 CÁCH TIẾP CẬN 1: sử dụng Ràng buộc xác suất và Rủi ro chấp nhận được

- Ta có thể thay thế $T x \geq h$ bởi các ràng buộc xác suất $\mathbb{P}[T x \geq h] \geq p$ cho một số mức độ tin cậy được quy định $p \in (.5, 1)$, (được chọn bởi người giải bài toán.) QHTT ở Định nghĩa 2 phía trên với các tham số ngẫu nhiên $\alpha = [\alpha_1, \alpha_2, \dots]$ được gọi là **QHTT với ràng buộc xác suất**, viết tắt là **1-SLP**.
- Rủi ro sau đó sẽ được xử lý gọn gàng, nếu định nghĩa một

$$\text{rủi ro chấp nhận được } r_x := \mathbb{P}[\text{Not}(T x \geq h)] = \mathbb{P}[T x \leq h] \leq 1 - p$$

với $(1 - p)$ là **rủi ro chấp nhận được tối đa**.

Ràng buộc xác suất $T x \leq h$ chỉ ra rằng

rủi ro chấp nhận được r_x nhỏ hơn giá trị tối đa xác định của $1 - p \in (0, 1)$.

Định nghĩa 3. QHTT ngẫu nhiên hay **1-SLP với ràng buộc xác suất** được định nghĩa như sau bởi một hệ số ngẫu nhiên $\alpha = (\alpha_1, \alpha_2 \dots \alpha_n)$ trong các ràng buộc xác suất, và một hàm mục tiêu tuyến tính $f(x)$:

$$SP: \min_x Z, \quad Z = f(x) = c^T \cdot x = \sum_{j=1}^n c_j x_j, \quad c_j \in \mathbb{R} \quad (1)$$

$$\text{trong đó } \begin{cases} Ax = b & (x = (x_1, x_2 \dots x_n) \text{ tạo các biến quyết định}) \\ \mathbb{P}[T x = \alpha \cdot x \leq h] \leq (1 - p) & (0 < p < 1). \end{cases}$$

CHÚ Ý: chúng ta sử dụng các vector tham số $\alpha = [\alpha_1, \alpha_2 \dots]$ nói chung, và đặc biệt áp dụng $\omega = [\omega_1, \omega_2, \dots, \omega_S]$ cho các trạng thái s gọi là *các kịch bản (scenarios)*. Ta có thể xử lý mỗi kịch bản $\omega \in \omega$ nhờ vào một tổ hợp các tham số ngẫu nhiên α_i cùng một lúc trong QHTT.

2.2 CÁCH TIẾP CẬN 2: cho các Ràng buộc ngẫu nhiên $T(\alpha) x \leq h(\alpha)$

Sử dụng phân tích kịch bản $T(\alpha) x \leq h(\alpha)$

Với mọi kịch bản $(T^s; h^s), s = 1, \dots, S$, giải được

$$\text{Minimize } \{f(x) = c^T \cdot x; \quad \text{với } Ax = b, T^s x \leq h^s\}$$

Loại quy hoạch này hướng tới một hàm mục tiêu tuyến tính cụ thể trong khi tính toán một hàm xác suất có liên quan đến **các kịch bản khác nhau**. Do đó, ta cần tìm một giải pháp toàn diện bằng cách quan sát các nghiệm kịch bản $x^s (s = 1, \dots, S)$.

Lợi ích: Mỗi bài toán kịch bản là một QHTT.

Bất lợi: phân bố rời rạc \rightarrow mô hình QHTT hỗn hợp số nguyên.

3 Quy hoạch ngẫu nhiên tổng quát có PHẢN HỒI (Generic Stochastic Programming (GSP) with RECOURSE)

Định nghĩa 4: (Quy hoạch ngẫu nhiên hai giai đoạn (generic **2-SP** problem)). Quy hoạch ngẫu nhiên hai giai đoạn (2-SP) được mở rộng từ Định nghĩa 2 có dạng:

$$2-SP: \min_x g(x) \quad \text{với } g(x) = f(x) + \mathbf{E}_\omega[v(x, \omega)] \quad (2)$$

trong đó $x = (x_1, x_2, \dots, x_n)$ là các biến quyết định giai đoạn đầu, $f(x)$ có thể tuyến tính hoặc không, là một phần của hàm **mục tiêu lớn** $g(x)$.

* Giá trị trung bình $Q(x) := \mathbf{E}_\omega[v(x, \omega)]$ của hàm số

$$v : \mathbb{R}^n \times \mathbb{R}^S \rightarrow \mathbb{R}$$

theo ảnh hưởng của kịch bản $\omega \cdot Q(x)$ là giá trị tối ưu của một bài toán hai giai đoạn nào đó

$$\min_{y \in \mathbb{R}^p} q(y) \mid \text{phụ thuộc vào } T \cdot x + W \cdot y = h \quad (3)$$

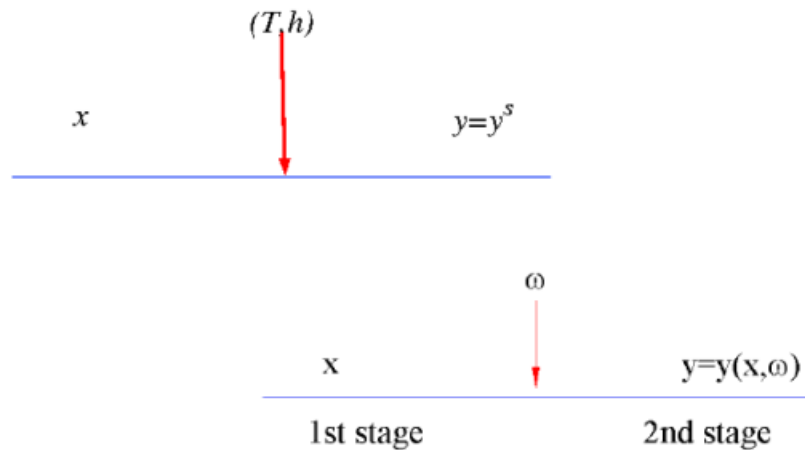
Vector $\alpha = \alpha(\omega)$ và $y = y(\omega)$ có tên là biến quyết định điều chỉnh hoặc khôi phục (correction or recourse decision variables), chỉ được biết sau khi thực hiện thí nghiệm ω .

Tóm lại, chúng ta **Tối thiểu hóa tổng chi phí dự kiến** $g(x) = f(x) + Q(x)$ trong khi thỏa mãn

$$W \cdot y(\omega) = h(\omega) - T(\omega) \cdot x$$

Ở đây W được gọi là ma trận hồi phục (recourse matrix) $m \times p$, và ta bắt đầu với trường hợp đơn giản $m = 1$,

q là vector chi phí hồi phục đơn vị, có chiều giống như y , và $y = y(\omega) \in \mathbb{R}^p$.



Hình 2: Cái nhìn cơ bản về quy hoạch ngẫu nhiên hai giai đoạn

4 Quy hoạch tuyến tính ngẫu nhiên hai giai đoạn (Two-Stage Stochastic linear programming (2-SLP))

4.1 Mô hình QHTT ngẫu nhiên hai giai đoạn có phản hồi (Two-stage SLP Recourse model) - dạng đơn giản

Định nghĩa 5 (QHTT ngẫu nhiên hai giai đoạn có phản hồi: 2-SLPWR). *Quy hoạch tuyến tính ngẫu nhiên hai giai đoạn có phản hồi (2-SLPWR) hoặc chính xác hơn là có phạt hành động khôi phục nói chung, được biểu diễn như sau*

$$2 - SLP : \min_{\mathbf{x} \in \mathbf{X}} \mathbf{c}^T \cdot \mathbf{x} + \min_{\mathbf{y}(\omega) \in \mathbf{Y}} \mathbf{E}_\omega[\mathbf{q} \cdot \mathbf{y}]$$

hoặc nói chung là

$$2 - SLP : \min_{\mathbf{x} \in \mathbf{X}, \mathbf{y}(\omega) \in \mathbf{Y}} \mathbf{E}_\omega[\mathbf{c}^T \cdot \mathbf{x} + v(\mathbf{x}, \omega)] \quad (4)$$

với $v(\mathbf{x}, \omega) := \mathbf{q} \cdot \mathbf{y}$
phụ thuộc vào

$A\mathbf{x} = \mathbf{b}$ Các ràng buộc giai đoạn đầu,

$T(\omega) \cdot \mathbf{x} + W \cdot \mathbf{y}(\omega) = h(\omega)$ Các ràng buộc giai đoạn hai

hoặc gọn hơn là $W \cdot \mathbf{y} = h(\omega) - T(\omega) \cdot \mathbf{x}$

- QHTT ngẫu nhiên này chỉ định 2-SP (ở công thức (2)) thành mục tiêu - một hàm *mục tiêu lớn ngẫu nhiên* cụ thể $g(\mathbf{x})$ có
 1. hàm xác định $f(\mathbf{x})$ - tuyến tính, vẫn đang được tính toán
 2. một hàm xác suất $v(\mathbf{x}, \omega)$ có liên quan đến các kịch bản ω khác nhau.
- $\mathbf{y} = \mathbf{y}(\mathbf{x}, \omega) \in \mathbb{R}_+^p$ có tên là biến hành động phản hồi cho biến \mathbf{x} và là hiện thực hóa của ω

Hành động phản hồi này được xem như là phạt các hành động khôi phục trong QHTT ngẫu nhiên.

Việc phạt khôi phục được biểu diễn thông qua giá trị trung bình $Q(\mathbf{x}) = \mathbf{E}_\omega[v(\mathbf{x}, \omega)]$

Cách tiếp cận 2: Phân tích kịch bản

Giá trị kì vọng $Q(\mathbf{x})$ rõ ràng là dành cho phân phối rời rạc ω

Vì vậy, ta lấy $\Omega = \{\omega_k\}$ là tập hữu hạn có kích thước S (có hữu hạn số lượng các kịch bản $\omega_1, \dots, \omega_S \in \Omega$, với khối lượng xác suất p_k tương ứng).

Bởi $\mathbf{y} = \mathbf{y}(\mathbf{x}, \omega)$ nên giá trị kì vọng của $v(\mathbf{y}) = v(\mathbf{x}, \omega) := \mathbf{q} \cdot \mathbf{y}$ (một q cho tất cả y_k) là

$$Q(\mathbf{x}) = \mathbf{E}_\omega[v(\mathbf{x}, \omega)] = \sum_{k=1}^S p_k q y_k = \sum_{k=1}^S p_k v(\mathbf{x}, \omega_k) \quad (5)$$

trong đó

- p_k là mật độ của kịch bản ω_k , q là phí phạt đơn vị đơn,
- và $q y_k = v(\mathbf{x}, \omega_k)$ - phí phạt của việc sử dụng y_k đơn vị trong pha phục hồi, phụ thuộc vào cả quyết định giai đoạn đầu \mathbf{x} và các kịch bản ngẫu nhiên ω_k .

4.2 Mô hình QHTT ngẫu nhiên hai giai đoạn có phản hồi (Two-stage SLP Recourse model) - dạng chính tắc

Định nghĩa 6: QHTT ngẫu nhiên hai giai đoạn có phản hồi (2-SLPWR) dạng chính tắc có thể biểu diễn như sau

$$\begin{aligned} 2-SLP: \quad & \min_x g(\mathbf{x}) \text{ với} \\ & g(\mathbf{x}) := \mathbf{c}^T \cdot \mathbf{x} + v(\mathbf{y}) \\ & \text{phụ thuộc vào } \mathbf{A}\mathbf{x} = \mathbf{b} \text{ trong đó } \mathbf{x} \in \mathbf{X} \subset \mathbb{R}^n, \mathbf{x} \geq \mathbf{0} \\ & v(\mathbf{z}) (:=) \min_{\mathbf{y} \in \mathbb{R}_+^p} \mathbf{q} \cdot \mathbf{y} \text{ phụ thuộc vào } \mathbf{W} \cdot \mathbf{y} = h(\omega) - T(\omega) \cdot \mathbf{x} =: \mathbf{z} \end{aligned}$$

trong đó $v(\mathbf{y}) := v(\mathbf{x}, \omega)$ là hàm giá trị giai đoạn hai, và $\mathbf{y} = \mathbf{y}(\mathbf{x}, \omega) \in \mathbb{R}_+^p$ là hành động phản hồi cho quyết định \mathbf{x} và hiện thực hóa ω

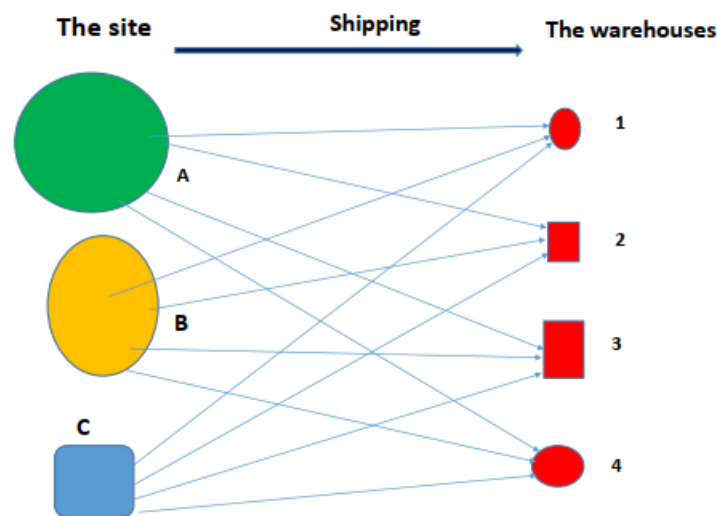
1. Chi phí phản hồi kì vọng của quyết định \mathbf{x} là $Q(\mathbf{x}) := \mathbf{E}_\omega[v(\mathbf{x}, \omega)]$ bằng phương trình (5). Vậy nói chung là, chúng ta tối thiểu hóa tổng chi phí kì vọng $\min_{\mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \mathbb{R}_+^p} \mathbf{c}^T \cdot \mathbf{x} + Q(\mathbf{x})$.
2. Ta thiết kế các biến quyết định thứ hai $\mathbf{y}(\omega)$ vì thế chúng ta có thể điều chỉnh phản ứng với các ràng buộc ban đầu theo một cách thông minh!

$$\mathbf{x} \text{ --- } -T, h, \omega \text{ --- } > \mathbf{y}$$

3. Giá trị tối ưu của bài toán QHTT hai giai đoạn là $v_* = v(\mathbf{y}^*)$, với $\mathbf{y} = \mathbf{y}^*(\mathbf{x}, \omega)$ là nghiệm tối ưu, ở đây, $\mathbf{y}^* \in \mathbb{R}_+^p$. Giá trị tối ưu sau cùng là $\mathbf{c}^T \cdot \mathbf{x}^* + v(\mathbf{y}^*)$

BÀI TOÁN 1: [Công nghiệp - Sản xuất]

Xét một hãng công nghiệp \mathbf{F} có một nhà máy sản xuất được n sản phẩm. Có nhiều bộ phận khác nhau (các bộ phận lắp ráp phụ) được đặt hàng từ m bên thứ ba (the sites). Hình dưới đây cho thấy kế hoạch vận chuyển của hãng công nghiệp \mathbf{F} với $m = 3$ từ các nhà cung cấp và $n = 4$ địa điểm sản xuất (the warehouses).



Một đơn vị sản phẩm i cần $a_{ij} \geq 0$ bộ phận j , trong đó $i = 1, \dots, n$ và $j = 1, \dots, m$. Nhu cầu về sản phẩm được mô hình hóa dưới dạng một vector ngẫu nhiên $\omega = \mathbf{D} = (D_1, D_2, \dots, D_n)$.

Bài toán giai đoạn hai:

Với một giá trị khả kiến $\mathbf{d} = (d_1, d_2, \dots, d_n)$ của vector nhu cầu ngẫu nhiên \mathbf{D} trên kia, ta có thể tìm ra phương án sản xuất tốt nhất bằng cách giải bài toán QHTT ngẫu nhiên sau với các biến quyết định $\mathbf{z} = (z_1, z_2, \dots, z_n)$ - số lượng các đơn vị được sản xuất,

và các biến quyết định **khác** $\mathbf{y} = (y_1, y_2, \dots, y_m)$ - số lượng các bộ phận còn trong kho.

$$LSP : \min_{\mathbf{z}, \mathbf{y}} Z = \sum_{i=1}^n (l_i - q_i) z_i - \sum_{j=1}^m s_j y_j \quad (6)$$

trong đó $s_j < b_j$ (được định nghĩa là giá đặt trước cho mỗi đơn vị bộ phận j), và $x_j, j = 1, \dots, m$ là số lượng bộ phận được đặt trước khi sản xuất.

$$\text{phụ thuộc vào} \begin{cases} y_j = x_j - \sum_{i=1}^n a_{ij} z_i, & j = 1, \dots, m \\ 0 \leq z_i \leq d_i, & i = 1, \dots, n; \quad y_j \geq 0, \quad j = 1, \dots, m \end{cases}$$

Toàn bộ mô hình (của giai đoạn hai) có thể biểu diễn tương đương như sau

$$MODEL = \begin{cases} \min_{\mathbf{z}, \mathbf{y}} Z = \mathbf{c}^T \cdot \mathbf{z} - \mathbf{s}^T \cdot \mathbf{y} \\ \text{với } \mathbf{c} = (c_i := l_i - q_i) \text{ là hệ số chi phí} \\ \mathbf{y} = \mathbf{x} - A^T \mathbf{z}, \text{ trong đó } A = [a_{ij}] \text{ là ma trận có kích thước } n \times m \\ 0 \leq \mathbf{z} \leq \mathbf{d}, \quad \mathbf{y} \geq 0 \end{cases} \quad (7)$$

Quan sát lời giải của bài toán, các vector \mathbf{z}, \mathbf{y} phụ thuộc vào hiện thực hóa \mathbf{d} của nhu cầu ngẫu nhiên $\omega = \mathbf{D}$ cũng như quyết định giai đoạn đầu $\mathbf{x} = (x_1, x_2, \dots, x_m)$.

Bài toán giai đoạn một:

Toàn bộ mô hình 2-SLPWR dựa trên một quy luật phổ biến là **sản xuất \geq nhu cầu**.

Bây giờ áp dụng cách tiếp cận dựa trên phân phối, cho $Q(\mathbf{x}) := \mathbf{E}[Z(\mathbf{z}, \mathbf{y})] = \mathbf{E}_\omega[\mathbf{x}, \omega]$ biểu thị giá trị tối ưu của bài toán (6).

$\mathbf{b} = (b_1, b_2, \dots, b_m)$ được xây dựng từ chi phí đặt trước b_j trên mỗi bộ phận j (trước khi biết nhu cầu). Các đại lượng x_j được xác định từ bài toán tối ưu hóa sau

$$\min g(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \mathbf{b}^T \cdot \mathbf{x} + Q(\mathbf{x}) = \mathbf{b}^T \cdot \mathbf{x} + \mathbf{E}[Z(\mathbf{z})] \quad (8)$$

trong đó $Q(\mathbf{x}) = \mathbf{E}_\omega[Z] = \sum_{i=1}^n p_i c_i z_i$ được lấy từ phân phối xác suất $\omega = \mathbf{D}$.

Phần đầu của hàm mục tiêu biểu diễn **chi phí đặt trước** và \mathbf{x} . Ngược lại, phần thứ hai biểu diễn **chi phí kì vọng** kế hoạch sản xuất tối ưu (7), được đưa ra bởi đại lượng đặt hàng thêm \mathbf{z} , đã sử dụng nhu cầu ngẫu nhiên $\mathbf{D} = \mathbf{d}$ cùng mật độ.

5 Câu hỏi 1 BTL 2023

5.1 Giới thiệu

- Một bài toán quy hoạch tuyến tính ngẫu nhiên (Stochastic linear program) là một bài toán quy hoạch tuyến tính, trong đó:

- Hàm mục tiêu và các phương trình ràng buộc phải ở dạng tuyến tính.
- Phải có các tham số là yếu tố không chắc chắn và biết được xác suất của nó (biến ngẫu nhiên).
- Cách giải bài toán quy hoạch tuyến tính ngẫu nhiên:
Cho các tham số là các biến ngẫu nhiên rời rạc
 - Bước 1: Xác định hàm mục tiêu
 - Bước 2: Xác định các ràng buộc
 - Bước 3: Xác định biến cố có thể xảy ra, ứng với mỗi biến cố, xác định giá trị các tham số ngẫu nhiên và xác suất xảy ra các biến cố đó.
 - Bước 4:
 - * Cách 1: Ứng với từng biến cố, giải bài toán tối ưu bằng phương pháp đơn hình (simplex method). Tìm ra nghiệm tối ưu cho từng bài toán đó.
 - * Cách 2: Tổng hợp các biến cố trong hàm mục tiêu để giải ra nghiệm tối ưu duy nhất cho tất cả các biến cố
 - Bước 5: Kết luận giá trị cuối cùng của nghiệm tối ưu (giá trị kỳ vọng) của bài toán.

5.2 Cách giải

Đề bài: Xác định số lượng sản phẩm cần sản xuất đáp ứng nhu cầu thị trường sao cho lợi nhuận được tối ưu nhất

Giải: Định nghĩa các thành phần trong bài toán như sau:

- $z \in \mathbb{R}^n$: số lượng sản phẩm từng loại (n loại sản phẩm)
- $x \in \mathbb{R}^m$: số lượng nguyên liệu có sẵn từng loại (m loại nguyên liệu)
- $A \in \mathbb{R}^{n \times m}$: ma trận phụ thuộc giữa số sản phẩm cần sản xuất và số nguyên liệu cần sản xuất
Trong đó, phần tử a_{ij} là số nguyên liệu loại j cần để sản xuất một sản phẩm loại i $\rightarrow z = Ax$
- $c \in \mathbb{R}^m$: chi phí mua từng loại nguyên liệu
 - Chi phí mua một loại nguyên liệu M: $c_M = c_i x_i$
 - Chi phí mua tất cả nguyên liệu: $C_M = \sum c_i x_i = c^T x$
- $l \in \mathbb{R}^n$: chi phí để sản xuất một sản phẩm từng loại
 - Chi phí sản xuất loại sản phẩm: $c_P = l_i z_i$
 - Chi phí sản xuất tất cả sản phẩm: $C_P = \sum l_i z_i = l^T z$
- $q \in \mathbb{R}^n$: giá bán cho một sản phẩm từng loại
 - Doanh thu: $R = \sum q_i z_i = q^T z$
 - Lợi nhuận: $P = R - C_M - C_P = q^T z - c^T x - l^T z = (q^T - l^T)z - c^T x$
- $y \in \mathbb{R}^m$: số lượng nguyên liệu còn lại $y = x - A^T z$
- $s \in \mathbb{R}^m$: giá bán cho từng loại nguyên liệu thừa

– Doanh thu từ việc bán nguyên liệu thừa: $R_M = \sum s_i y_i = s^T y$

- Lợi nhuận thực mà công ty thu được: $f(z) = (R - C_M - C_P) + R_M = (q^T - l^T)z - c^T x + s^T y$

$$\text{phụ thuộc vào } \begin{cases} y = x - A^T z \\ 0 \leq z \leq d(\text{nhu cầu - demand}) \\ x \geq 0; y \geq 0 \end{cases}$$

Xét các kịch bản xảy ra:

$d^k \in \mathbb{R}^n$: nhu cầu thị trường trong kịch bản k

p^k : xác suất xảy ra của biến cố k

Trong một biến cố k cụ thể nào đó, ta có bài toán:

$$\text{Max } f(z^k) = (q^T - l^T)z^k - c^T x + s^T y^k$$

$$\text{phụ thuộc vào } \begin{cases} y^k = x - A^T z^k \\ 0 \leq z^k \leq d^k \\ x \geq 0; y^k \geq 0 \end{cases}$$

→ Nghiệm tối ưu: $z = \sum p^k z^k$

→ Giá trị tối ưu: $f(z) = (q^T - l^T)z - c^T x + s^T y$

Biến đổi bài toán thành dạng tổng hợp các kịch bản, ta được:

$$\begin{aligned} \text{Max } f(z^k) &= \sum p^k ((q^T - l^T)z^k - c^T x + s^T y^k) \\ \iff \text{Max } f(z^k) &= \sum p^k ((q^T - l^T)z^k + s^T y^k) - c^T x \\ \text{phụ thuộc vào } &\begin{cases} y^k = x - A^T z^k \\ 0 \leq z^k \leq d^k \\ x \geq 0; y^k \geq 0 \end{cases} \end{aligned}$$

Biến đổi bài toán về dạng chuẩn (standard form):

$$\begin{aligned} \text{Min } -f(z^k) &= c^T x - \sum p^k ((q^T - l^T)z^k + s^T y^k) \\ \text{phụ thuộc vào } &\begin{cases} y^k = x - A^T z^k \\ 0 \leq z^k \leq d^k \\ x \geq 0; y^k \geq 0 \end{cases} \end{aligned}$$

5.3 Thành lập bài toán cụ thể theo như yêu cầu câu hỏi 1

Một công ty F sản xuất thực phẩm gồm 5 loại sản phẩm, 8 loại nguyên liệu được cho như bảng dưới đây, trong đó nhu cầu thị trường là ngẫu nhiên, nhưng theo đề bài nó là dạng phân phối nhị thức (kết cục chỉ xảy ra một trong hai biến cố) và xác suất cho mỗi biến cố là $p = 0.5$.

	Đường	Sữa	Bột	Trái cây	Phô mai	Chi phí sản xuất (l)	Doanh thu/sản phẩm (q)	Nhu cầu 1 (d ¹)	Nhu cầu 2 (d ²)
Bánh đậu	0	0	2	0	1	5000	30000	30	20
Bánh dẻo	1	0	3	0	0	4000	25000	20	10
Bánh thập cẩm	0	0	2	1	0	7000	25000	30	40
Kẹo	3	1	0	1	0	6000	45000	50	50
Nước trái cây	0	0	0	3	1	2000	30000	30	15
Nước có ga	0	0	0	0	3	5000	45000	25	30
Mứt	4	0	1	2	0	1000	40000	15	20
Socola	5	3	1	0	0	3000	70000	10	5
Chi phí mua nguyên liệu (c)	5000	7000	2000	3000	9000	X	X	X	X
Tiền bán nguyên liệu thừa (s)	4000	6000	1000	1000	7000	X	X	X	X
Số nguyên liệu có sẵn (x)	300	100	200	250	150	X	X	X	X

Như các định nghĩa trong bài toán đã nêu ở phần II, ta có:

$$x = (300 \quad 100 \quad 200 \quad 250 \quad 150)$$

$$A = \begin{pmatrix} 0 & 0 & 2 & 0 & 1 \\ 1 & 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 \\ 3 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 0 & 3 \\ 4 & 0 & 1 & 2 & 0 \\ 5 & 3 & 1 & 0 & 0 \end{pmatrix}$$

$$c = (5000 \quad 7000 \quad 2000 \quad 3000 \quad 9000)$$

$$l = \begin{pmatrix} 5000 \\ 4000 \\ 7000 \\ 6000 \\ 2000 \\ 5000 \\ 1000 \\ 3000 \end{pmatrix} \quad q = \begin{pmatrix} 30000 \\ 25000 \\ 25000 \\ 45000 \\ 30000 \\ 45000 \\ 40000 \\ 70000 \end{pmatrix}$$

$$y = x - A^T z$$

Model cụ thể của bài toán sẽ là:

$$\text{Min} - f(z) = c^T x - 0.5((q^T - l^T)z^1 + s^T y^1) - 0.5((q^T - l^T)z^2 + s^T y^2)$$

$$\text{phụ thuộc vào} \begin{cases} y^1 = x - A^T z^1 \\ y^2 = x - A^T z^2 \\ 0 \leq z^1 \leq d^1 \\ 0 \leq z^2 \leq d^2 \\ x \geq 0; y^1 \geq 0; y^2 \geq 0 \end{cases}$$

Giải bài toán bằng Excel, các số liệu được tổ chức theo bảng như sau:

	A	B	C	D	E	F	G	H	I	J	K	L
1		Trường hợp nhu cầu 1	Trường hợp nhu cầu 2									
2												
3	Sản phẩm	Số lượng(z)	Số lượng(z)	Đường	Sữa	Bột	Trái cây	Phô mai	Chi phí sản xuất(l)	Doanh thu/sản phẩm(q)	Nhu cầu 1	Nhu cầu 2
4	Bánh đậu	30	40	0	0	2	0	1	5000	30000	30	40
5	Bánh dẻo	18.33333333	5	1	0	3	0	0	4000	25000	20	10
6	Bánh thập cẩm	30	40	0	0	2	1	0	7000	25000	30	40
7	Kẹo	50	50	3	1	0	1	0	6000	45000	50	50
8	Nước trái cây	30	15	0	0	0	3	1	2000	30000	30	15
9	Nước có gas	25	30	0	0	0	0	3	5000	45000	25	30
10	Mứt	15	20	4	0	1	2	0	1000	40000	15	20
11	Socola	10	5	5	3	1	0	0	3000	70000	10	5
12												
13	Chi phí mua nguyên liệu(c)			5000	7000	2000	3000	9000				
14	Tiền bán nguyên liệu thừa(s)			4000	6000	1000	1000	7000				
15	Số lượng nguyên liệu có sẵn(x)			300	100	200	250	150				
16		Trường hợp nhu cầu 1	Trường hợp nhu cầu 2									
17	Hàm mục tiêu f(z) theo cách 1	2381666.667	2290000						Hàm mục tiêu f(z) theo cách 2	-2335833.333		
18	Điều kiện về nguyên liệu sản xuất			<=					Các điều kiện (tương tự cách 1)			
19	Đường	278.3333333	260			300	300					
20	Sữa	80	65			100	100					
21	Bột	200	200			200	200					
22	Trái cây	200	175			250	250					
23	Phô mai	135	145			150	150					
24	Điều kiện về nhu cầu			<=		Nhu cầu 1	Nhu cầu 2					
25	Bánh đậu	30	40			30	40					
26	Bánh dẻo	18.33333333	5			20	10					
27	Bánh thập cẩm	30	40			30	40					
28	Kẹo	50	50			50	50					
29	Nước trái cây	30	15			30	15					
30	Nước có gas	25	30			25	30					
31	Mứt	15	20			15	20					
32	Socola	10	5			10	5					

Trong đó:

- Hàm mục tiêu ở ô B17 = SUMPRODUCT((J4:J11-I4:I11)*B4:B11)-SUMPRODUCT(D15:H15*D13:H13)+(D15-SUMPRODUCT(B4:B11*D4:D11))*D14+(E15-SUMPRODUCT(B4:B11*E4:E11))*E14+(F15-SUMPRODUCT(B4:B11*F4:F11))*F14+(G15-SUMPRODUCT(B4:B11*G4:G11))*G14+(H15-SUMPRODUCT(B4:B11*H4:H11))*H14
- Hàm mục tiêu ở ô C17 = SUMPRODUCT((J4:J11-I4:I11)*C4:C11)-SUMPRODUCT(D15:H15*D13:H13)+(D15-SUMPRODUCT(C4:C11*D4:D11))*D14+(E15-SUMPRODUCT(C4:C11*E4:E11))*E14+(F15-SUMPRODUCT(C4:C11*F4:F11))*F14+(G15-SUMPRODUCT(C4:C11*G4:G11))*G14+(H15-SUMPRODUCT(C4:C11*H4:H11))*H14
- Hàm mục tiêu ở ô J17 = SUMPRODUCT(D15:H15*D13:H13) - 0.5*(SUMPRODUCT((J4:J11-I4:I11)*B4:B11) + (D15-SUMPRODUCT(B4:B11*D4:D11))*D14 + (E15-SUMPRODUCT(B4:B11*E4:E11))*E14+(F15-SUMPRODUCT(B4:B11*F4:F11))*F14 + (G15-SUMPRODUCT(B4:B11*G4:G11))*G14+(H15-SUMPRODUCT(B4:B11*H4:H11))*H14) - 0.5*(SUMPRODUCT((J4:J11-I4:I11)*C4:C11) + (D15-SUMPRODUCT(C4:C11*D4:D11))*D14 + (E15-SUMPRODUCT(C4:C11*E4:E11))*E14+(F15-SUMPRODUCT(C4:C11*F4:F11))*F14 + (G15-SUMPRODUCT(C4:C11*G4:G11))*G14+(H15-SUMPRODUCT(C4:C11*H4:H11))*H14)

Sử dụng chức năng Solver ở mục Data ta giải được bài toán với kết luận như sau:

33	KẾT LUẬN: để công ty có lợi nhuận cao nhất thì công ty nên sản xuất như sau	
34	Sản phẩm	Số lượng
35	Bánh đậu	35
36	Bánh dẻo	11.66666667
37	Bánh thập cẩm	35
38	Kẹo	50
39	Nước trái cây	22.5
40	Nước có gas	27.5
41	Mứt	17.5
42	Socola	7.5
43	Lợi nhuận tốt nhất có thể đạt được	2335833.333

Nhận xét: khi giải bài toán bằng 2 cách như đã nêu ở phần II, ta thu được kết quả giống nhau.

6 Câu hỏi 2 BTL 2023

6.1 Đề bài

Tìm hiểu, triển khai và xác minh tính hiệu quả của thuật toán đã nghiên cứu hoặc giải quyết mô hình lập kế hoạch sơ tán ngẫu nhiên hai giai đoạn trên mạng lưới nhỏ, với cách tiếp cận thiết kế thí nghiệm. Mô phỏng sẽ có tối đa 50 nút và 100 liên kết.

6.2 Giới thiệu

Các sự kiện cực đoan (do thiên tai hoặc do con người gây ra) như động đất, lũ lụt, lốc xoáy,... có thể ảnh hưởng nghiêm trọng đến mạng lưới và dịch vụ viễn thông ở các khu vực đô thị. Lập kế hoạch sơ tán hợp lý sao cho những thảm họa là một hành động quan trọng và cần được suy nghĩ cẩn thận (sớm) và thực hiện (vào ngày nào, thời gian nào), đặc biệt khi có sự không chắc chắn.

Bài tập lớn này sẽ nghiên cứu rõ hơn bài báo của L. Wang, “A two-stage stochastic programming framework for evacuation planning in disaster responses,”(2020) về giải pháp kế hoạch sơ tán cụ thể cho những người bị ảnh hưởng từ khu vực nguy hiểm sang khu vực an toàn. Vì có rất nhiều tình huống cũng như những thiệt hại phát sinh trong thảm họa do đó vấn đề sơ tán này nên được coi là một vấn đề quy hoạch ngẫu nhiên. Quy hoạch ngẫu nhiên có phản hồi được sử dụng để tìm ra các quyết định không dự đoán được phải được thực hiện trước khi biết được các thực hiện của một số biến ngẫu nhiên sao cho tổng chi phí kỳ vọng của các hành động hậu quả có thể là nhỏ nhất. Dựa trên phân tích trên, vấn đề sơ tán trong bài tập lớn này được xây dựng thành một mô hình quy hoạch ngẫu nhiên hai giai đoạn dựa trên kịch bản để đại diện cho sự ngẫu nhiên xuất phát từ cường độ và ảnh hưởng.

6.3 Mục tiêu

Lập kế hoạch sơ tán người dân khỏi những khu vực bị ảnh hưởng bởi các thiên tai như lũ lụt, động đất, núi lửa,... Mô hình quy hoạch ngẫu nhiên hai giai đoạn xem xét cả lựa chọn trước đường sơ tán (trước thảm họa) và thích ứng (sau thảm họa) để cung cấp kế hoạch sơ tán trước mắt cho người bị ảnh hưởng từ các khu vực nguy hiểm sang các khu vực an toàn. Không giống như các mô hình xác định, nó liên quan đến sự khả dụng của các thời gian di chuyển và khả năng liên kết ngẫu nhiên dựa trên kịch bản. Trong giai đoạn thứ nhất, giả sử rằng người bị ảnh hưởng không thể thu thập được thông tin về mức độ thiên tai và mức độ hư hại của đường và sau một khoảng thời gian nhất định, họ có thể thu thập được thông tin mạng lưới đường bộ chính xác thông qua một số thiết bị giám sát thời gian thực sau đó thực hiện kế hoạch sơ tán nhanh cho tất cả các mức độ thảm họa. Các quyết định giai đoạn thứ hai (hậu quả) được phụ thuộc vào các quyết định giai đoạn đầu, được thực hiện sau khi thực hiện các thời gian di chuyển và khả năng ngẫu nhiên. Do đó, mục tiêu là thực hiện kế hoạch sơ tán trước khi tối ưu hóa trong giai đoạn đầu, dưới các điều kiện không chắc chắn sẽ phải đối mặt trong giai đoạn thứ hai. Hơn nữa, thay vì quyết định vị trí cơ sở và đặt trước hàng hóa cứu trợ khẩn cấp, kế hoạch này trình bày đường sơ tán cụ thể cho người bị ảnh hưởng.

6.4 Các bước thực hiện

- Bước 1: Đề xuất mô hình lập trình ngẫu nhiên hai giai đoạn và lựa chọn lộ trình trước và sau thảm họa để cung cấp kế hoạch sơ tán cho những người bị ảnh hưởng đến nơi an toàn.
- Bước 2: Xây dựng kế hoạch di chuyển rõ ràng của những người dân bị ảnh hưởng bằng mô hình dòng chi phí tối thiểu từ đó chia thành 2 bài toán nhỏ là luồng có chi phí nhỏ nhất và bài toán phụ thuộc vào thời gian của nó.
- Bước 3: Để thu được nghiệm gần tối ưu, khung thuật toán tối ưu hóa đạo hàm riêng được sử dụng để giảm dần khoảng cách tương đối của giới hạn trên và dưới, tức là các giá trị mục tiêu của mô hình gốc và mô hình độc lập; và hai bài toán con được giải quyết bằng thuật toán đường liên tục ngắn nhất.

6.5 Bảng kí hiệu

Kí hiệu	Ý nghĩa
V	tập hợp các nút (nodes)
A	tập hợp các liên kết (links)
i, j	vị trí của các nút, $i, j \in V$
(i, j)	vị trí của các liên kết, $(i, j) \in A$
s	số thứ tự của kịch bản
S	tổng số kịch bản
v	giá trị cung cấp của nút nguồn
T	tổng số khoảng thời gian
u_{ij}	dung lượng trên liên kết (i, j)
$u_{ij}^s(t)$	dung lượng trên liên kết (i, j) trong kịch bản s ở thời gian t
$c_{ij}^s(t)$	thời gian di chuyển trên liên kết (i, j) trong kịch bản s ở thời gian t
μ_s	xác suất trong kịch bản s

Biến quyết định	Ý nghĩa
x_{ij}^s	luồng trên liên kết (i,j)
y_{ij}^s	luồng trên liên kết (i,j) trong kịch bản s ở thời gian t

6.6 Các ràng buộc

Giai đoạn 1:

Trong giai đoạn đầu tiên, một sơ tán khả thi cần được xác định từ vùng nguy hiểm đến vùng an toàn. Mỗi luồng trên mỗi liên kết cần thỏa mãn ràng buộc về cân bằng luồng (tổng luồng chạy vào một nút bằng tổng luồng chạy ra khỏi nút đó)

$$\sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{j,i} = d_i$$

trong đó d_i là tham số, với $d_i = \begin{cases} v, i = s \\ -v, i = t \\ 0, \text{ các trường hợp còn lại} \end{cases}$

Ngoài ra luồng còn phải thỏa mãn ràng buộc về dung lượng tối đa (luồng trên mỗi liên kết không vượt quá dung lượng tối đa của liên kết đó)

$$0 \leq x_{ij} \leq u_{ij}, \forall (i,j) \in A$$

Lưu ý: Phạt liên kết (link penalty) $p_{ij} \in A$ là một giá trị được gán cho mỗi liên kết để đánh giá mức độ không mong muốn của việc sử dụng liên kết đó trong một cách sơ tán. Hàm liên kết có thể định nghĩa là

$$f(\mathbf{X}) = \sum_{(i,j) \in A} p_{ij} x_{ij}$$

$$\text{trong đó } \mathbf{X} := (x_{ij})_{(i,j) \in A}$$

Giai đoạn 2:

Ở giai đoạn 2, những người bị ảnh hưởng sẽ nhận được các đường đi ứng với thời gian sơ tán tối thiểu theo thông tin thảm họa thời gian thực. Trước thời điểm đó, những người bị ảnh hưởng sẽ được sơ tán theo kế hoạch sơ tán giai đoạn một quy định lúc trước; nói cách khác, kế hoạch sơ tán ở các kịch bản khác nhau trước thời điểm đều giống như tiên nghiệm. Theo thảo luận ở trên, các ràng buộc cho kế hoạch sơ tán của từng kịch bản trước ngưỡng thời gian có thể được xây dựng như sau:

$$y_{ij}^s(t) = x_{ij}, \quad \forall t \leq \tilde{T}, (i,j) \in A, s = 1, 2, \dots, S$$

Các cặp ràng buộc thể hiện mối quan hệ giữa liên kết vật lý và cung không gian - thời gian trong kế hoạch sơ tán, nghĩa là nếu có các luồng truy cập trên liên kết (i,j) ví dụ $x_{ij} = 2$ thì luồng truy cập trên mỗi liên kết trước thời điểm 2, $y_{ij}^s(t) = 2$. Nói cách khác, trước ngưỡng đó, kế hoạch sơ tán trong mỗi kịch bản của giai đoạn thứ hai giống như kế hoạch sơ tán ban đầu ở giai đoạn đầu tiên. Sau đây, một mô hình giai đoạn hai được thiết lập với mục tiêu nhằm giảm thiểu tổng thời gian di tản của những người bị ảnh hưởng khỏi khu vực nguy hiểm ở mỗi kịch bản:

$$Q(Y, s) = \min \sum_{(i,j) \in A_s} c_{ij}^s \cdot y_{ij}^s(t) \quad (9)$$

Tùy thuộc vào

$$\sum_{(i_t, j_{t'}) \in A_s} y_{ij}^s(t) - \sum_{(j_{t'}, i_t) \in A_s} y_{ij}^s(t') = d_i^s(t), \forall i \in V, t \in \{0, 1, \dots, T\}, s = 1, 2, \dots, S \quad (10)$$

$$0 \leq y_{ij}^s(t) \leq u_{ij}^s(t), \forall (i, j) \in A_s, t \in \{0, 1, \dots, T\}, s = 1, 2, \dots, S \quad (11)$$

$$\sum_{t \leq \tilde{T}} y_{ij}^s(t) = x_{ij}, (i, j) \in A, s = 1, 2, \dots, S \quad (12)$$

- Hàm mục tiêu (9) là thời gian sơ tán tổng thể của các phương tiện giao thông trong kịch bản giảm thiểu s
- Ràng buộc (10) là ràng buộc cân bằng luồng
- Ràng buộc (11) là ràng buộc về dung lượng tối đa
- Ràng buộc (12) là một ràng buộc cặp để đảm bảo rằng kế hoạch sơ tán trong từng kịch bản ở giai đoạn thứ hai trước ngưỡng thời gian giống như kế hoạch tiên nghiệm ở giai đoạn đầu.

6.7 Mô hình lập kế hoạch sơ tán ngẫu nhiên hai giai đoạn

Cần có được một kế hoạch sơ tán vững chắc trong giai đoạn đầu tiên bằng cách đánh giá các kế hoạch thích ứng ở giai đoạn thứ hai. Để đạt được mục tiêu này, kế hoạch sơ tán giai đoạn một với thời gian sơ tán tổng thể dự kiến của đường sơ tán ứng với từng kịch bản và xác suất xảy ra của từng kịch bản s giả định là $\mu_s, s = 1, 2, \dots, S$. Để giảm thiểu mức phạt đối với kế hoạch sơ tán trước đó và tổng thời gian sơ tán dự kiến của mỗi kế hoạch sơ tán ứng theo kịch bản, mô hình kế hoạch sơ tán hai giai đoạn trong môi trường phụ thuộc thời gian và ngẫu nhiên được xây dựng như sau:

$$\left\{ \begin{array}{l} \min \sum_{(i,j) \in A} p_{ij} x_{ij} + \sum_{s=1}^S \mu_s \cdot Q(Y, s) \\ s.t. \\ \sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = d_i, \forall i \in V \\ 0 \leq x_{ij} \leq u_{ij}, \forall (i, j) \in A \\ \text{trong đó} \\ Q(Y, s) = \min \sum_{(i,j) \in A_s} c_{ij}^s(t) \cdot y_{ij}^s(t) \\ s.t. \\ \sum_{(i_t, j_{t'}) \in A_s} y_{ij}^s(t) - \sum_{(j_{t'}, i_t) \in A_s} y_{ij}^s(t') = d_i^s(t), \forall i \in V, \\ t \in \{0, 1, \dots, T\}, s = 1, 2, \dots, S \\ 0 \leq y_{ij}^s(t) \leq u_{ij}^s(t), \forall (i, j) \in A, t \in \{0, 1, \dots, T\}, s = 1, 2, \dots, S \\ \sum_{t \leq \tilde{T}} y_{ij}^s(t) = x_{ij}, (i, j) \in A, s = 1, 2, \dots, S \end{array} \right.$$

$$\left\{ \begin{array}{l} \min \sum_{(i,j) \in A} p_{ij} x_{ij} + \sum_{s=1}^S (\mu_s \cdot \sum_{(i,j) \in A_s} c_{ij}^s(t) \cdot y_{ij}^s(t)) \\ s.t. \\ \sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = d_i, \forall i \in V \\ 0 \leq x_{ij} \leq u_{ij}, \forall (i,j) \in A \\ \sum_{(i_t, j_{t'}) \in A_s} y_{ij}^s(t) - \sum_{(j_{t'}, i_t) \in A_s} y_{ij}^s(t') = d_i^s(t), \forall i \in V, \\ t \in \{0, 1, \dots, T\}, s = 1, 2, \dots, S \\ 0 \leq y_{ij}^s(t) \leq u_{ij}^s(t), \forall (i,j) \in A, t \in \{0, 1, \dots, T\}, s = 1, 2, \dots, S \\ \sum_{t \leq \bar{T}} y_{ij}^s(t) = x_{ij}, (i,j) \in A, s = 1, 2, \dots, S \end{array} \right. \quad (13)$$

Để giải quyết mô hình (13) cần một mô hình lập trình nguyên vốn chứa hai loại biến quyết định là $X := \{x_{ij}\} (i, j \in A)$ và $Y := \{y_{ij}^s(t)\} (i, j \in A, s \in \{1, 2, \dots, S\}, t \in \{1, 2, \dots, T\})$ Ta cần chia mô hình thành 2 bài toán nhỏ bằng phương pháp song phương nhằm nối lỏng ràng buộc ghép vào hàm mục tiêu bằng cách sử dụng phương pháp nối lỏng Lagrange. Cụ thể, mô hình ban đầu được phân tách thành một bài toán luồng có chi phí nhỏ nhất và một bài toán phụ thuộc vào thời gian, mà có thể được giải quyết hiệu quả bằng các thuật toán chính xác. Vấn đề then chốt của phương pháp nối lỏng Lagrange là tìm ra cận trên của bài toán cần quan tâm. Các giải pháp nối lỏng, ngẫu nhiên, là khả thi cho bài toán ban đầu.

6.8 Bài toán 1

$$\left\{ \begin{array}{l} \min SP1(\alpha) = \sum_{(i,j) \in A} (p_{ij} - \sum_{s=1}^S \sum_{t \leq \bar{T}} \alpha_{ij}^s(t)) x_{ij} \\ s.t. \\ \sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = d_i, \forall i \in V \\ 0 \leq x_{ij} \leq u_{ij}, \forall (i,j) \in A \end{array} \right.$$

Cho một mạng G bao gồm n đỉnh và m cạnh mỗi cạnh có dung lượng $A(e)$, chi phí trên mỗi luồng dọc theo cạnh là $C(e)$ (số nguyên) và cân bằng đỉnh. Tổng chi phí của bất kì luồng f có thể định nghĩa là:

$$A(f) = \sum_{e \in E} A(e) f(e)$$

Đối với một giá trị nhất định K , chúng ta phải tìm một luồng có đại lượng này và trong số tất cả các luồng có đại lượng này, chúng ta phải chọn luồng có chi phí thấp nhất. Nhiệm vụ này được gọi là bài toán luồng chi phí tối thiểu.

Nhiệm vụ này có thể giải quyết bằng thuật toán đường đi ngắn nhất liên tiếp.

Đường đi ngắn nhất liên tiếp được xây dựng trên mạng luồng chung $G = (V, E)$ với dung lượng $f(\cdot)$, giới hạn dưới, chi phí $C(\cdot)$ trên các cạnh và cân bằng (dòng vào bằng dòng ra tại bất kỳ nút nào). Giả sử các điều dưới đây được thỏa mãn

1. Các cạnh có dung lượng không âm và giới hạn dưới bằng 0:

$$0 \leq f(u, v) \leq c(u, v) \quad \forall (u, v)$$

2. Mọi luồng f phải thỏa mãn ràng buộc bảo toàn số luồng vào bằng số luồng ra tại mỗi nút:

$$\sum_{v \in V} f(v, u) = \sum_{v \in V} f(u, v) \Leftrightarrow \sum_{v \in V} f(v, u) - \sum_{v \in V} f(u, v) = 0, \quad u \neq s, t$$

6.8.1 Thuật toán đường đi ngắn nhất liên tiếp

Mục tiêu của bài toán này là sẽ tìm kiếm giá trị lớn nhất của luồng và đồng thời tối ưu hóa hàm mục tiêu cho trước.

- Bước 1: Chọn biến x làm một luồng khả thi, tức là một luồng thỏa mãn các ràng buộc về dung lượng, đối xứng, và bảo toàn luồng. Luồng này cũng phải có chi phí nhỏ nhất trong các luồng khả thi có cùng giá trị luồng.
- Bước 2: Kiểm tra xem luồng đã đạt giá trị mong muốn hay chưa, hoặc có còn đường đi nào có chi phí nhỏ hơn trong mạng lưới dư hay không. Mạng lưới dư là một mạng lưới được tạo ra bằng cách thêm các cung ngược lại vào mạng lưới ban đầu, với chi phí và dung lượng phù hợp. Nếu luồng đã đạt giá trị mong muốn v , hoặc không còn đường đi nào có chi phí nhỏ hơn, thuật toán kết thúc. Nếu không, tìm đường đi ngắn nhất với luồng lớn nhất trong mạng lưới dư, bằng cách sử dụng thuật toán nhãn sửa, và sau đó chuyển sang Bước 3. Hàm $A(x), C(x), U(x)$ trong phần còn lại mạng có thể được định nghĩa là:

$$A(x) = \{(i, j) | (i, j) \in A, x_{ij} < u_{ij}\} \cup \{(j, i) | (j, i) \in A, x_{ji} > 0\}$$

$$C(x) = \begin{cases} c_{ij}, & (i, j) \in A, x_{ij} < u_{ij} \\ -c_{ji}, & (j, i) \in A, x_{ji} > 0 \end{cases}$$

$$U(x) = \begin{cases} u_{ij}, & (i, j) \in A, x_{ij} < u_{ij} \\ x_{ji}, & (j, i) \in A, x_{ji} > 0 \end{cases}$$

- Bước 3: Tăng luồng dọc theo đường đi có chi phí nhỏ nhất, bằng cách thêm hoặc bớt luồng trên các cung thuộc đường đi đó. Nếu giá trị luồng tăng không vượt quá giá trị mong muốn, quay lại Bước 2.

Lưu ý rằng trong vòng lặp cuối cùng của thuật toán, chỉ cần tăng luồng thêm một lượng sao cho giá trị luồng cuối không vượt quá K .

6.8.2 Thuật toán nhãn sửa (label correcting algorithm)

Đầu vào: Cho $G = (V, E)$ là một đồ thị (có thể vô hướng hoặc có hướng) và $\{a_{ij} : (i, j) \in E\}$ là tập hợp các trọng số không âm trên các cạnh. Chúng ta phân biệt hai nút trong đồ thị, cụ thể là điểm gốc $s \in V$ và điểm đích $t \in V$. Nút $j \in V$ được gọi là nút con của nút $i \in V$ nếu có cạnh từ i đến j . Lưu ý rằng nếu đồ thị vô hướng và (i, j) là một cạnh, thì i là con của j và j là con của i . Mục tiêu của chúng ta là tìm đường đi ngắn nhất từ s đến t .

Ý tưởng đằng sau các thuật toán sửa nhãn là tìm dần dần các đường đi ngắn hơn từ nguồn gốc của mọi nút khác i . Điều này đạt được bằng cách duy trì nhãn di cho mỗi nút i , nhãn này biểu thị giới hạn trên của khoảng cách ngắn nhất từ điểm gốc s đến nút i . Nhãn d_t của đích t được duy trì trong một biến gọi là Upper. Như chúng ta sẽ thấy, biến Upper đóng vai trò một vai trò đặc biệt trong thuật toán. Khi thuật toán tiến triển, nếu một đường đi từ điểm gốc đến nút i được phát hiện và khoảng cách của nó nhỏ hơn d_i , thì nhãn của nút i sẽ được sửa. Để tạo thuận lợi cho việc ghi sổ, thuật toán duy trì một danh sách các nút được gọi là Open. Danh sách Open chứa các nút sẽ được kiểm tra bởi thuật toán và là ứng cử viên có thể được đưa vào con đường ngắn nhất. Khi danh sách mở trống, thuật toán kết thúc.

- Bước 1: Cho $d_s = 0$, $d_i = \infty$ với $\forall i \in V \setminus \{s, t\}$ và $Upper = \infty$. Tạo danh sách $Open = \{s\}$
- Bước 2: Xóa a node i khỏi Open
- Bước 3: Với mỗi con j của i ta thực hiện:
 - Nếu $d_i + a_{ij} < \min(d_j, Upper)$, $d_j = d_i + a_{ij}$ và cho i thành nút cha của j
 - Nếu j khác đích t, thêm j vào danh sách Open nếu j chưa xuất hiện. Ngược lại nếu j là đích t, $Upper = d_i + a_{it}$
- Bước 4: Nếu danh sách Open rỗng kết thúc thuật toán nếu không quay lại bước 2.

Về mặt hình thức, thuật toán sửa nhãn nói chung được đưa như trên. Ở bước 3 của thuật toán, chúng ta gặp một đường đi từ s đến j qua i và chúng ta kiểm tra xem đường dẫn này (i) có ngắn hơn đường đi được xem xét trước đây từ s đến j (tức là nếu $d_i + a_{ij} < d_j$) và (ii) có khả năng dẫn đến a đường đi ngắn nhất từ s đến t (tức là nếu $d_i + a_{ij} < Upper$) hay không. Nếu vậy thì nhãn d_j bị giảm và nút j được đặt vào danh sách Open sao cho các đường đi qua j và tới các phần tử con của j có thể được kiểm tra sau.

1. Trong suốt thuật toán, nhãn d_i của nút i tùy ý là ∞ (tương ứng trong trường hợp tôi chưa vào danh sách Open) hoặc đó là độ dài của một số đường dẫn từ s đến i bao gồm các nút đã vào danh sách Open ít nhất một lần.
2. Nếu $d_i < \infty$, thì có thể thu được đường đi từ s đến i có độ dài di bằng cách truy tìm các nút cha bắt đầu với cha mẹ của i.
3. Thuật toán sẽ kết thúc sau một số bước hữu hạn. Hơn nữa, khi nó kết thúc, biến Upper sẽ bằng khoảng cách ngắn nhất từ điểm gốc đến đích t (trường hợp này có ít nhất một đường đi từ s tới t), hoặc bằng ∞ (trường hợp không có đường đi từ s đến t).
4. Nút j mà $d_i + a_{ij} \geq Upper$ ở bước 3 chắc chắn sẽ không được đưa vào danh sách Open trong lần lặp hiện tại và có thể không được đưa vào danh sách trong bất kỳ lần lặp tiếp theo nào. Do đó, số lượng nút vào danh sách Open có thể nhỏ hơn đáng kể so với tổng số nút. Đặc biệt, thuật toán sửa nhãn có thể được hiệu quả hơn thuật toán DP cơ bản.

Lưu ý rằng ở bước 2 của Thuật toán 1, có một số quyền tự do trong việc chọn nút nào để loại bỏ từ danh sách Open trong mỗi lần lặp. Mỗi lựa chọn tương ứng với một cách triển khai khác nhau của thuật toán sửa nhãn. Một số lựa chọn nổi tiếng hơn được đưa ra như sau:

1. Thuật toán BFS: Danh sách Open được duy trì dưới dạng hàng đợi FIFO, tức là nút luôn được xóa khỏi đầu Open và mỗi nút vào Open được đặt tại phần dưới cùng của Open.
2. Thuật toán DFS: Danh sách Open được duy trì dưới dạng hàng đợi LIFO, tức là nút luôn được xóa khỏi đầu Open và mỗi nút vào Open được đặt tại đỉnh của Open.
3. Thuật toán Tìm kiếm tốt nhất - Đầu tiên: Trong mỗi lần lặp, một nút có nhãn nhỏ nhất sẽ bị xóa khỏi danh sách. Đây được gọi là phương pháp Dijkstra. Có thể chỉ ra rằng trong phương pháp này, mỗi nút sẽ chỉ vào Open tối đa một lần

6.8.3 Giải thuật cho đồ thị vô hướng, đa đồ thị

Trong trường hợp đồ thị vô hướng hay đa đồ thị không khác nhau nhiều về mặt khái niệm của thuật toán trên. Tuy nhiên, việc thực hiện thuật toán sẽ khó khăn hơn một chút.

- Đầu tiên, luồng cho từng cạnh phải giữ riêng biệt.
- Thứ hai, khi tìm kiếm đường đi ngắn nhất, cần phải tính đến việc điều quan trọng là cạnh nào trong số nhiều cạnh được sử dụng trong đường dẫn. Do đó, thay vì mảng tổ tiên thông thường, chúng ta phải lưu trữ số cạnh mà chúng ta xuất phát cùng với mảng tổ tiên.
- Thứ ba, khi luồng tăng dọc theo một cạnh nhất định thì cần phải giảm luồng dọc theo cạnh sau. Vì chúng ta có nhiều cạnh nên chúng ta phải lưu trữ số cạnh của cạnh đảo ngược cho mỗi cạnh.

6.8.4 Kết quả

Đánh giá:

Thuật toán Dijkstra yêu cầu ít lần lặp hơn Bellman-Ford. Tuy nhiên, Bellman-Ford yêu cầu ít công việc hơn cho mỗi lần lặp. Ta có thể dừng thuật toán của Dijkstra sau khi đã xác định xong đến nút đích. Ta không thể sử dụng thuật toán của Dijkstra nếu có chi phí liên kết âm. Còn Bellman-Ford vẫn hoạt động.

Độ phức tạp: phụ thuộc vào cấp số nhân của đầu vào, trong trường hợp xấu nhất nó có thể đẩy tới đa 1 đơn vị lưu lượng trên 1 lần lặp, lấy $O(F)$ lặp đi lặp lại để tìm ra luồng kích thước có chi phí tối thiểu F , làm cho tổng thời gian chạy là $O(F.T)$. T là thời gian cần để tìm đường đi ngắn nhất từ nguồn tới đích.

Nếu thuật toán Bellman-Ford được sử dụng cho việc này, nó sẽ làm cho thời gian chạy là $O(mn)$. Cũng có thể sửa đổi thuật toán Dijkstra để nó chỉ cần $O(nm)$, tiền xử lý là bước đầu tiên và sau đó thực hiện trong mỗi lần lặp $O(m \log n)$, làm cho thời gian tổng thể là $O(mn + Fm \log n)$. Đây là một trình tạo biểu đồ, trong đó thuật toán như vậy sẽ yêu cầu $O(2^{n/2} n^2 \log n)$

Thuật toán Dijkstra đã sửa đổi sử dụng cái gọi là tiềm năng từ thuật toán Johnson. Có thể kết hợp ý tưởng của thuật toán này và thuật toán Dinic để giảm số lần lặp từ F đến $\min(F, nC)$ với C là chi phí tối đa được tìm thấy giữa các cạnh.

Ưu điểm:

- Thuật toán này có thể áp dụng cho các đồ thị có trọng số âm, miễn là không có chu trình âm. Điều này làm cho thuật toán này linh hoạt hơn so với các thuật toán chỉ áp dụng cho các đồ thị có trọng số dương, như thuật toán Dijkstra.
- Thuật toán này có thể tận dụng được các cấu trúc đặc biệt của đồ thị, như đồ thị thưa hay đồ thị đầy đủ, để cải thiện hiệu suất. Ví dụ, nếu đồ thị là thưa, ta có thể dùng cấu trúc dữ liệu hàng đợi ưu tiên để tìm đường đi ngắn nhất nhanh hơn. Nếu đồ thị là đầy đủ, ta có thể dùng ma trận để lưu trữ và cập nhật các nhãn.
- Thuật toán này có thể dễ dàng được mở rộng để giải quyết các biến thể của bài toán tối thiểu luồng chi phí, như bài toán luồng cực đại chi phí nhỏ nhất, bài toán luồng cân bằng, hay bài toán luồng có giới hạn.

Nhược điểm:

- Thuật toán này chỉ áp dụng được cho các đồ thị không có chu trình âm. Nếu đồ thị có chu trình âm, thuật toán sẽ không hội tụ và không tìm được lời giải. Do đó, ta cần kiểm tra trước xem đồ thị có chu trình âm hay không, hoặc sử dụng các thuật toán khác như Cycle canceling hay Out-of-kilter.

- Thuật toán này có thể bị ảnh hưởng bởi sự thay đổi của trọng số của các cạnh. Nếu trọng số của một cạnh giảm, thuật toán có thể tiếp tục chạy và cập nhật lời giải. Nhưng nếu trọng số của một cạnh tăng, thuật toán sẽ phải khởi tạo lại và chạy lại từ đầu.

6.9 Bài toán 2: Bài toán luồng chi phí tối thiểu với thời gian và dung lượng di chuyển của liên kết phụ thuộc vào thời gian

$$\begin{cases} \min SP2(\alpha, s) = \sum_{(i,j) \in A} \left(\sum_{t \in \{0,1,\dots,T\}} \mu_s \cdot c_{ij}^s(t) + \sum_{t \leq \tilde{T}} \alpha_{ij}^s(t) \right) y_{ij}^s(t) \\ s.t \\ \sum_{(i_t, j_{t'}) \in A_s} y_{ij}^s(t) - \sum_{(j_{t'}, i_t) \in A_s} y_{ij}^s(t') = d_t^s(t), \forall i \in V, t \in \{0, 1, \dots, T\} \\ 0 \leq y_{ij}^s(t) \leq u_{ij}^s(t), \forall (i, j) \in A, t \in \{0, 1, \dots, T\} \end{cases} \quad (14)$$

Với mỗi kịch bản $s \in \{1, 2, 3, \dots, S\}$ bài toán (14) có cấu trúc với chi phí phụ thuộc thời gian $c_{ij}^s(t)$, dung lượng liên kết $u_{ij}^s(t)$ và chi phí tổng quát $g_{ij}^s(t)$. Khoảng thời gian T được chia thành hai giai đoạn thời gian, chi phí tổng quát được định nghĩa là hàm từng phần:

$$g_{ij}^s(t) = \begin{cases} \mu_s \cdot c_{ij}^s(t) + \alpha_{ij}^s(t), & t \leq \tilde{T} \\ \mu_s \cdot c_{ij}^s(t), & \tilde{T} < t \leq T \end{cases}$$

Vì bài toán 2 là bài toán phụ thuộc vào thời gian vì vậy chúng ta cần phải thay đổi ở bước 2.

Đầu tiên chúng ta cần chỉnh sửa các tham số $A(y(t))$, $C(y(t))$, $U(y(t))$ trong mạng dư $N(y(t))$ như sau:

$$\begin{aligned} A_s(y(t)) &= \{(i_t, j_{t'}) | (i_t, j_{t'}) \in A_s, y_{ij}^s < u_{ij}^s\} \cup \{(j_{t'}, i_t) | (j_{t'}, i_t) \in A_s, y_{ji}^s > 0\}, s = 1, 2, \dots, S \\ c_{ij}^s(y(t)) &= \begin{cases} c_{ij}^s(t), (i_t, j_{t'}) \in A_s, y_{ij}^s(t) < u_{ij}^s(t), t \in \{0, 1, \dots, T\} \\ -c_{ji}^s(t'), (j_{t'}, i_t) \in A_s, \forall \{t' \in \{0, 1, \dots, T\} | y_{ji}^s(t') > 0\}, \\ s = 1, 2, \dots, S \\ T, (j_{t'}, i_t) \in A_s, \forall \{t' \in \{0, 1, \dots, T\} | y_{ji}^s(t') = 0\} \end{cases} \\ u_{ij}^s(y(t)) &= \begin{cases} u_{ij}^s(t) - y_{ij}^s(t), (i_t, j_{t'}) \in A_s, y_{ij}^s(t) < u_{ij}^s(t), t \in \{1, 2, \dots, T\} \\ y_{ji}^s(t), (j_{t'}, i_t) \in A_s, \forall \{t' \in \{0, 1, \dots, T\} | y_{ji}^s(t') > 0\}, \\ s = 1, 2, \dots, S \\ 0, (j_{t'}, i_t) \in A_s, \forall \{t' \in \{0, 1, \dots, T\} | y_{ji}^s(t') = 0\} \end{cases} \end{aligned}$$

Tiếp theo chúng ta cần sửa đổi thuật toán sửa nhãn theo (Ziliaskopoulos and Mahmassani, 1992) để tìm luồng chi phí tối thiểu phụ thuộc vào thời gian trong mạng dư.

Thuật toán tìm đường ngắn nhất phụ thuộc thời gian được giới thiệu trong bài báo của (Ziliaskopoulos and Mahmassani, 1992) tính toán đường đi ngắn nhất phụ thuộc thời gian từ tất cả các nút trong mạng đến một nút đích cho mỗi bước thời gian trong một khoảng thời gian nhất định trong một mạng với chi phí cạnh phụ thuộc thời gian. Khác với các thuật toán phụ thuộc thời gian khác, phương pháp này có thể xử lý các mạng trong đó chi phí đi lại không nhất thiết phải là thời gian đi lại chính nó. Thuật toán này dựa trên nguyên tắc tối ưu của Bellman. Nó rời rạc hóa khoảng thời gian quan tâm thành các khoảng thời gian nhỏ và tính toán các đường đi hoạt động ngược lại từ nút đích.

6.9.1 Thuật toán tìm đường ngắn nhất phụ thuộc thời gian

Giải thuật này có thể được giải thích như sau:

- Bước 1: Tạo một danh sách SE rỗng.
- Bước 2: Chèn nút nguồn (source node) N vào danh sách SE.
- Bước 3: Lặp lại cho đến khi danh sách SE trống:
 - Xóa nút đầu tiên (current node) khỏi danh sách SE.
 - Duyệt qua tất cả các nút J có thể đến được từ nút hiện tại (current node).
 - * Gán next node bằng J.
 - * Gán insert in SE list bằng FALSE.
 - * Duyệt qua tất cả các bước thời gian (time step) t từ 1 đến M.
 - * Tính thời gian di chuyển (travel time) từ next node đến current node tại bước thời gian t.
 - * Tính nhãn mới (new label) bằng cách cộng nhãn của current node tại bước thời gian t cộng với thời gian di chuyển.
 - * Nếu nhãn của next node tại bước thời gian t lớn hơn nhãn mới, thì cập nhật nhãn của next node bằng nhãn mới và gán insert in SE list bằng TRUE.
 - * Nếu insert in SE list bằng TRUE, thì chèn next node vào danh sách SE.
- Bước 4: Sau khi kết thúc vòng lặp, ta có thể truy vết đường đi ngắn nhất từ nút đích (destination node) đến nút nguồn (source node) bằng cách sử dụng các con trỏ đường dẫn đã lưu trữ.

6.9.2 Đánh giá:

Độ phức tạp:

Độ phức tạp thời gian là $O(N^2M)$, trong đó N là số đỉnh, M là số bước thời gian. Đây là một độ phức tạp cao, đặc biệt khi đồ thị có nhiều đỉnh và số bước thời gian lớn.

Độ phức tạp không gian của thuật toán là $O(NM)$, trong đó N là số đỉnh, M là số bước thời gian.

Ưu điểm:

- Thuật toán có thể xử lý được đồ thị có trọng số âm
- Thuật toán tìm đường đi ngắn nhất phụ thuộc vào thời gian cũng có thể tìm được đường đi ngắn nhất trong đồ thị có hướng và có chu kỳ

Nhược điểm:

- Thuật toán có thể không tìm được đường đi ngắn nhất nếu đồ thị có chu trình âm
- Thuật toán cần phải biết trước các thông tin về đồ thị, như số đỉnh, số cạnh, trọng số của các cạnh, số bước thời gian. Điều này có thể không phù hợp với một số bài toán thực tế, khi các thông tin này có thể thay đổi theo thời gian hoặc không được biết trước.

Bằng cách giải theo phương pháp nối lỏng Lagrange, giá trị tối ưu cho mô hình với một tập vector nhân tử Lagrange có thể biểu diễn dưới dạng sau:

$$Z_{LR}^*(\alpha) = Z_{SP1}^*(\alpha) + Z_{SP2}^*(\alpha)$$

Rõ ràng, giá trị tối ưu trên là giới hạn dưới của giá trị mục tiêu tối ưu của mô hình (13) ban đầu. Để có được giải pháp tốt hơn, cần có giới hạn dưới gần với giá trị mục tiêu tối ưu của mô hình ban đầu. Nghĩa là, cần đạt được giới hạn dưới lớn nhất có thể và biểu thức được đưa ra như sau:

$$Z_{LD}(\alpha^*) = \max_{\alpha \geq 0} Z_{LR}(\alpha)$$

7 Ứng dụng II: Quy hoạch ngẫu nhiên trong Viễn thông

7.1 Đề bài:

Lập kế hoạch cho một dịch vụ thông tin dựa trên Internet và viết báo cáo về các bước chính của quá trình phát triển mô hình đó, sử dụng lý thuyết quy hoạch ngẫu nhiên và quy trình ra quyết định Markov

7.2 Các bước thực hiện

- Bước 1: Mô hình giảm thiểu chi phí trong một giai đoạn
Chúng ta bắt đầu bằng việc chỉ xem xét một giai đoạn quyết định và có đầy đủ thông tin về nhu cầu cũng như những điều không chắc chắn khác. Mặc dù những giả định này rất phi thực tế, nhưng mô hình kết quả sẽ tạo tiền đề cho những mô hình thực tế hơn. Trong cách thiết lập này, cho rằng chương trình triển khai phải đáp ứng đầy đủ nhu cầu đã biết. Vì giá dịch vụ cố định nên doanh thu cố định. Vì lý do này, cách duy nhất có thể tác động đến lợi nhuận là giảm thiểu chi phí.

$i = 1 \dots n$		Chỉ số của các vùng tạo nên một lãnh thổ. Dân số người dùng tồn tại ở mỗi khu vực tạo ra nhu cầu.
$j = 1 \dots m$		Chỉ số các vùng có thể có máy chủ.
y_j		Biến ngẫu nhiên mang giá trị 1 nếu quyết định đặt một máy chủ tại khu vực j , 0 cho các trường hợp còn lại
x_{ij}		Lượng nhu cầu từ khu vực i được phục vụ bởi máy chủ đặt ở vị trí j
f_j		Phí cài đặt một máy chủ ở khu vực j
c_{ij}		Chi phí phục vụ một đơn vị nhu cầu từ vùng i bởi máy chủ ở khu vực j
d_i		Nhu cầu phát sinh tại khu vực i
g_j		Dung lượng của máy chủ được đặt tại khu vực j

Như vậy, bài toán tìm kiếm triển khai máy chủ $y = (y_1, y_2, \dots, y_m)$ và phân nhóm sử dụng các máy chủ $x = \{x_{ij}\}, i = 1, \dots, n; j = 1, \dots, m$ sẽ trở thành:

$$\min_{i,j} \sum_{j=1}^m f_j y_j + \sum_{j=1}^m \sum_{i=1}^n c_{ij} x_{ij} \quad (15)$$

$$\sum_{j=1}^m x_{ij} \geq d_i, \forall i = 1, \dots, n \quad (16)$$

$$\sum_{i=1}^n x_{ij} \leq g_j y_j, \forall j = 1, \dots, m \quad (17)$$

Khi đó, $y_j \in \{0, 1\}$ và $x_{ij} \geq 0$. Ở đây số hạng đầu tiên trong (15) biểu hiện chi phí cố định để cài đặt máy chủ tại khu vực y_i , số hạng thứ hai biểu hiện chi phí biến đổi để phục vụ cho nhu cầu. Ràng buộc (16) bảo đảm về sự thỏa mãn nhu cầu và ràng buộc cuối cùng (17) thể hiện các ràng buộc về năng lực. Đây là mô hình cơ sở nhất đóng vai trò là điểm khởi đầu để phát triển mô hình quy hoạch ngẫu nhiên với các kịch bản khác nhau về nhu cầu trong tương lai và số lượng giai đoạn triển khai lớn hơn.

- Bước 2: Mô hình tối thiểu hóa chi phí ngẫu nhiên hai giai đoạn
Hai giai đoạn triển khai được xét đến: Giai đoạn 1 hiện tại với nhu cầu đã biết và Giai đoạn 2 trong tương lai với nhu cầu không chắc chắn được mô tả bằng một số kịch bản hữu hạn. Mỗi kịch bản được mô tả bằng giá trị nhu cầu ở các khu vực khác nhau trong Giai đoạn 2 và xác suất xảy ra kịch bản này. Các quyết định của Giai đoạn 2 bao gồm việc triển khai bổ sung các máy chủ và phân bổ lại nhu cầu cho các máy chủ để đáp ứng nhu cầu đã được biết đến. Mô hình tuân theo khuôn khổ quy hoạch ngẫu nhiên có phản hồi.

Kí hiệu bổ sung

$r=1$	R	Các trường hợp cần nhu cầu
	d_i^r	Nhu cầu phát sinh của vùng i theo trường hợp r
	p^r	Xác suất xảy ra trường hợp r
	z_j^r	Biến nhị phân với 1 là đã đặt máy chủ tại j và 0 trong trường hợp còn lại
	x_{ij}^r	Lượng nhu cầu từ khu vực i được phục vụ bởi máy chủ j trong trường hợp r
	α	Hệ số chiết khấu chi phí giai đoạn 2 đến thời điểm hiện tại

Mỗi trường hợp được đặc trưng bởi một cặp (d^r, p^r) , trong đó $d^r = (d_1^r, \dots, d_n^r)$

Mô hình: Bài toán tìm kiếm triển khai máy chủ giai đoạn 1 $y = (y_1, \dots, y_m)$ và phân công nhóm sử dụng máy chủ $x = \{x_{ij}\}, i = 1 : n; j = 1 : m$ sẽ trở thành:

$$\min_{x, y} \sum_{j=1}^m f_j y_j + \sum_{j=1}^m \sum_{i=1}^n c_{ij} x_{ij} + \alpha \sum_{r=1}^R p^r Q(r, y) \quad (18)$$

Tùy thuộc vào 2 ràng buộc (16) và (17). Số hạng thứ 3 trong hàm mục tiêu (18) cho biết chiết khấu khi triển khai giai đoạn 2 tính trung bình theo các kịch bản (scenarios), các chi phí liên quan đến trường hợp nằm ở hàm $Q(r, y)$ và phụ thuộc vào quyết định y triển khai giai đoạn một. Những chi phí này có được từ việc giải bài toán phản hồi cho từng kịch bản r .

$$Q(r, y) = \min_{x^r, z^r} \sum_{j=1}^m f_j z_j^r + \sum_{j=1}^m \sum_{i=1}^n c_{ij} x_{ij}^r \quad (19)$$

$$\sum_{j=1}^5 x_{ij}^r \geq d_i^r \quad \forall i = 1, \dots, n \quad (20)$$

$$\sum_{i=1}^n x_{ij}^r \leq g_j(y_j + z_j^r) \quad \forall j = 1, \dots, m \quad (21)$$

Tương tự như ràng buộc (16) và (17) và chọn triển khai giai đoạn 2: $z^r = (z_1^r, \dots, z_m^r)$ và chỉ định mới các nhóm khu vực dùng máy chủ $x^r = \{x_{ij}^r\}, i = 1 : n; j = 1 : m$ từ việc giảm thiểu chi phí triển khai cố định và chi phí dịch vụ thay đổi cho một kịch bản xác định r . Đây có thể là một bài toán khó về mặt số học vì nó chứa các biến nhị phân. Tuy nhiên, công nghệ tối ưu hóa hiện đại cho phép giải quyết các trường hợp không tầm thường và quan trọng trong thực tế.

- Bước 3: Mô hình tối đa hóa lợi nhuận ngẫu nhiên hai giai đoạn với việc định giá.
Trong môi trường cạnh tranh được bãi bỏ quy định, tối đa hóa lợi nhuận là mục tiêu phù hợp hơn việc giảm thiểu chi phí mạng lưới. Về cơ bản, nó trở nên khác biệt so với việc giảm thiểu chi phí đơn giản khi các quyết định về giá được xem xét đồng thời với các quyết định triển khai. Các mô hình trở nên phức tạp hơn vì giá cả ảnh hưởng đến nhu cầu và sự phụ thuộc này tạo ra sự phi tuyến tính. Tuy nhiên, phân tích có ý nghĩa cũng khả thi trong trường hợp này. Đầu tiên, hãy bắt đầu bằng cách xác định mô hình nhu cầu tuyến tính mở rộng khung kịch bản đã giải thích trước đó.

Kí hiệu bổ sung

h_0	giá tham chiếu dịch vụ trong giai đoạn 1
d_{i0}	cầu tham chiếu tại khu vực i trong giai đoạn 1 tương ứng với giá tham chiếu h_0
w_i	độ co giãn của cầu tại khu vực i trong giai đoạn 1
h	mức giá tăng so với giá tham chiếu trong giai đoạn 1
h^r_0	giá tham chiếu dịch vụ trong giai đoạn 2 trong trường hợp r
d^r_{i0}	cầu tham chiếu tại khu vực i trong giai đoạn 2 tương ứng với giá tham chiếu h^r_0 trong trường hợp r
w^r_i	độ co giãn của cầu tại khu vực i trong giai đoạn 2 trong trường hợp r
h^r	mức giá tăng so với giá tham chiếu trong giai đoạn 2 trong trường hợp r

Mô hình nhu cầu:

Đây là phần quan trọng của mô hình lợi nhuận. Hãy xem xét việc triển khai Giai đoạn 1. Giá dịch vụ bằng $h_0 + h$ và quyết định giá bao gồm việc chọn mức tăng giá h , có thể dương hoặc âm. Giả sử rằng nhu cầu d_i ở vùng i trong giai đoạn này phụ thuộc vào giá dịch vụ theo một số hàm $d_i = f_i(h_0 + h)$, và ở vùng lân cận điểm h_0 , sự phụ thuộc này có thể được tuyến tính hóa thông qua

$$d_i = d_{i0} - w_i h \quad (22)$$

Các mối quan hệ mô tả tương tự hành vi nhu cầu của giai đoạn 2 cho từng trường hợp $r = 1 : R$. Mỗi trường hợp được định nghĩa trong trường hợp này bởi một bộ $(d^r_{i0}, w^r_i, h^r, p^r)$, trong đó xác định sự phụ thuộc của cầu vào giá theo quan hệ (22) đối với một trường hợp r nhất định.

Mô hình quyết định: Tìm mức tăng giai đoạn 1 cho giá dịch vụ h , Bài toán tìm kiếm triển khai máy chủ giai đoạn 1 $y = (y_1, \dots, y_m)$ và phân công nhóm sử dụng máy chủ $x = \{x_{ij}\}, i = 1 : n; j = 1 : m$ sẽ trở thành:

$$\max_{h,x,y} W(h) - C(y,x) + \alpha \sum_{r=1}^R p^r P(r,y) \quad (23)$$

$$w_i h + \sum_{j=1}^m x_{ij} \geq d_{i0}, \quad i = 1 : n, \quad (24)$$

Và với ràng buộc (17). Ở đây $W(h)$ là hàm doanh thu trong giai đoạn 1,

$$W(h) = \sum_{i=1}^n (h + h_i)(d_0 - w_i h) \quad (25)$$

Và $C(x,y)$ là hàm chi phí trong giai đoạn một được xác định theo (15). Phần tử thứ 3 trong hàm (23) đại diện cho lợi nhuận trung bình trong giai đoạn 2 theo các trường hợp và được chiết khấu theo giá trị hiện tại trong đó $P(r,y)$ là lợi nhuận giai đoạn hai trong trường hợp r . Nó được coi là giá trị tối ưu của bài toán có phản hồi sau:

$$P(r,y) = \max_{h^r, x^r, z^r} W(r, h^r) - C(r, z^r, x^r) \quad (26)$$

$$w_i^r h^r + \sum_{j=1}^m x_{ij}^r \geq d_{i0}^r, \quad i = 1 : n, \quad (27)$$

Tùy thuộc vào các ràng buộc bổ sung (21). Ở đây $W(r, h^r)$ là doanh thu trong giai đoạn 2 trong trường hợp r thu được tương tự như (25), và $C(r, z^r, x^r)$ là chi phí trong giai đoạn 2 trong trường hợp r được xác định theo (19).

Có một đặc điểm quan trọng của mô hình này không có trong các mô hình (15)-(17) và (18)-(21). Trong khi (18)-(21) có thể được chuyển đổi thành quy hoạch tuyến tính nguyên hỗn hợp bằng cách xem xét tương đương, thì không thể chuyển đổi như vậy đối với mô hình (23)-(27). Điều này là do doanh thu $W(r, h)$ và $W(r, h^r)$ phụ thuộc phi tuyến tính vào các biến quyết định h và h^r . Ngay cả trong trường hợp đơn giản nhất của mô hình cầu tuyến tính (22), sự phụ thuộc này là bậc hai. Do đó, các kỹ thuật số chuyên dụng nên được sử dụng trong trường hợp này, với các phương pháp phân rã là khả thi nhất.

Đánh giá cơ hội đầu tư, lựa chọn thực tế. Một trong những ứng dụng quan trọng nhất của mô hình (23)-(27) là đánh giá khả năng sinh lời của dự án đầu tư bao gồm việc triển khai dịch vụ mới. Những phát triển gần đây về tài chính doanh nghiệp cho thấy tầm quan trọng của việc đánh giá các phương án thực tế để đánh giá chính xác các dự án công nghiệp. Trong khi đối với các ngành truyền thống hơn, kỹ thuật đánh giá trực tiếp có thể tương tự như đánh giá các lựa chọn tài chính, thì đối với các ngành đổi mới có dự án độc đáo, những phương pháp như vậy rất khó áp dụng. Các mô hình quy hoạch ngẫu nhiên có thể biểu diễn cho một giải pháp thay thế hợp lý để đánh giá tùy chọn thực tế. Vì thế, chúng ta có thể sử dụng mô hình (23)-(27) cho mục đích này. Cụ thể, chúng ta hãy đánh giá các phương án mở rộng, nâng cấp công nghệ, từ bỏ hoặc chuyển đổi một phần cơ sở hạ tầng.

Tùy chọn để mở rộng (tùy chọn chờ và xem). Tùy chọn này đã được nhúng trong mô hình (23)-(27), có khả năng thêm các máy chủ bổ sung trong Giai đoạn 2 tùy theo phản ứng của thị trường. Giá trị của tùy chọn này có thể được tính như sau. Kí hiệu P^* là giá trị tối ưu của mô hình (23)-(27). Đây là giá trị của dự án có tùy chọn mở rộng. Giá trị \hat{P} của cùng một dự án không có tùy chọn mở rộng có được bằng cách giải cùng một mô hình với các biến nhị phân z^r cố định bằng 0 cho tất cả các kịch bản. Rõ ràng $P \leq P^*$. Giá trị chênh lệch của lựa chọn là $P^* - P$.

Tùy chọn nâng cấp công nghệ. Đây là một lựa chọn có giá trị vì nó có thể thay đổi đáng kể việc đánh giá dự án. Ví dụ nổi tiếng nhất là mạng di động GSM, mạng này bắt đầu phát triển trước khi có công nghệ sản xuất điện thoại di động. Để đánh giá phương án này, cần phải xem xét kỹ hơn cách thức phát triển công nghệ có thể ảnh hưởng đến các thành phần khác nhau của mô hình (23)-(27). Ví dụ, phát triển công nghệ có thể dẫn đến giảm chi phí cố định cho việc cài đặt máy chủ và/hoặc tăng dung lượng máy chủ có thể có trong Giai đoạn 2. Trong trường hợp này, cần đưa các tính năng này vào định nghĩa của các trường hợp; f_j^r là chi phí cố định để thiết lập máy chủ tại vị trí j theo trường hợp r và g_i^r là dung lượng của máy chủ đặt tại vị trí i theo trường hợp r .

Mô hình thay đổi như sau. Phần (23)-(24) vẫn giữ nguyên vì nó mô tả Giai đoạn 1 sẽ được triển khai bằng công nghệ đã biết. Phần (26)-(27) có giới hạn công suất được sửa đổi để thay thế (21)

$$\sum_{i=1}^n x_{ij}^r \leq g_j y_j + g_j^r z_j^r, \quad j = 1 : m \quad (28)$$

Điều khoản chi phí $C(r, z^r, x^r)$ từ (26) là

$$C(r, z^r, x^r) = \min_{z^r, x^r} \sum_{j=1}^m f_j^r z_j^r + \sum_{j=1}^m \sum_{i=1}^n c_{ij}^r x_{ij}^r \quad (29)$$

Mô hình (23)-(27) đã được giải quyết bằng (28)-(29), sẽ cho giá trị P^{**} của dự án với tùy chọn nâng cấp công nghệ. Giá trị này được so sánh với giá trị P^* của dự án không tùy chọn, và sự chênh lệch $P^{**} - P^*$ là giá trị của tùy chọn.

Lựa chọn từ bỏ. Đây là một lựa chọn có giá trị khi phản ứng từ thị trường không chắc chắn. Nếu nhu cầu không bắt kịp, việc cắt giảm chi phí bảo trì ở những khu vực có nhu cầu yếu và có thể thu hồi một phần chi phí cố định bằng cách bán hoặc cho thuê cơ sở hạ tầng máy chủ là điều hợp lý. b_j^r là chi phí bảo trì cho máy chủ tại vị trí j trong Giai đoạn 2 theo trường hợp r , β_j^r là phần chi phí cố định có thể được phục hồi bằng cách từ bỏ máy chủ tại vị trí j trong trường hợp r và u_j^r là biến nhị phân bằng 1 nếu máy chủ ở vị trí j bị bỏ trong trường hợp r .

Mô hình thay đổi như sau. Một lần nữa, phần (23)-(24) đề cập đến giai đoạn 1 vẫn giữ nguyên. Phần (26)-(27) có giới hạn về khả năng (capacity) được sửa đổi thay thế cho (21)

$$\sum_{i=1}^n x_{ij}^r \leq g_j (y_j + z_j^r - u_j^r), \quad j = 1 : m, r = 1 : R \quad (30)$$

Và thêm các ràng buộc từ bỏ

$$u_j^r \leq y_j, \quad j = 1 : m, r = 1 : R \quad (31)$$

Số hạng thu nhập $W(r, h^r)$ và số hạng chi phí $C(r, z^r, x^r)$ từ (23) là

$$W(r, h^r) = \sum_{i=1}^n (h_0^r + h^r) (d_{i0}^r - w_i^r h^r + \sum_{j=1}^n \beta_j^r f_j u_j^r) \quad (32)$$

$$C(r, z^r, x^r) = \sum_{j=1}^m (f_j z_j^r + b_j^r (y_j + z_j^r - u_j^r) + \sum_{i=1}^n c_{ij} x_{ij}^r) \quad (33)$$

Mô hình (23)-(27) đã được giải quyết bằng việc thay thế (30)-(33), sẽ mang lại giá trị P^{++} của dự án với tùy chọn từ bỏ cơ sở hạ tầng. Giá trị này được so sánh với giá trị P^+ của dự án thu được bằng cách giải cùng một mô hình với các biến được đặt bằng 0, tương ứng với việc đánh giá mà không có tùy chọn từ bỏ. Sự chênh lệch $P^{++} - P^+$ là giá trị của tùy chọn từ bỏ.

Tài liệu tham khảo

- [1] Man Van Minh Nguyen, Khuong An Nguyen. (2023). *MATHEMATICAL MODELING: Methods and Application*, Mahidol University
- [2] YouTube. (2018). *Excel sử dụng công cụ Solver giải bài tập quy hoạch tuyến tính*, tại: <https://youtu.be/vPDW8dHPhNs?si=V9T21BxJp8zWQiIr>
- [3] Stein W. Wallace, William T. Ziemba. (2005). *Applications of stochastic programming - Society for Industrial and Applied Mathematics - Mathematical Programming Society*, Saunder School of Business, University of British Columbia, Vancouver, Canada
- [4] Wang, L., Yang, L., Gao, Z. (2016). *The constrained shortest path problem with stochastic correlated link travel times. European Journal of Operational Research*, mục 255(1), 43 - 57
- [5] Zealint. (2018). *MINIMUM COST FLOW PART TWO: ALGORITHMS*, tại: <https://www.topcoder.com/thrive/articles/Minimum%20Cost%20Flow%20Part%20Two:%20Algorithms?fbclid=IwAR3rVSGwIUrPrXDWnz-BvTXRly2Rz-P0w-sBGdFkqXj3Ri6Kggr9RxZRxE>
- [6] *Minimum Cost Flow Problem*, tại: <https://sboyles.github.io/teaching/ce377k/class14.pdf?fbclid=IwAR1AERDDnDbqG6dLLpcoQxDGbp9aOCwrh1wzGk1VqykyuxsrN6Nkr31v1w>
- [7] Li Wang. (2020). *A two-stage stochastic programming framework for evacuation planning in disaster responses*, School of Modern Post, Beijing University of Posts and Telecommunications, China
- [8] *Minimum-cost flow-Successive shortest path algorithm*, tại: https://cp-algorithms.com/graph/min_cost_flow.html?fbclid=IwAR0BURkoMjfk-411ZWfdKeelx-Op3j44adPpvyIwo0LPK4vXwloD9LyKe2c