CHƯƠNG 4: BÀI TOÁN TỐI ƯU

Môn học: Toán Rời rạc 1

Giảng viên: Nguyễn Kiều Linh
Email: linhnk@ptit.edu.vn
Học viện Công nghệ Bưu chính Viễn thông

Hà Nội, năm 2023
http://www.ptit.edu.vn

Nội dung chính

Giới thiệu bài toán tối ưu

- **Bài toán đếm:** Đếm tất cả các cấu hình tổ hợp thỏa mãn các ràng buộc của bài toán. Phương pháp giải mong muốn là xây dựng được công thức tính nghiệm của bài toán.
- **Bài toán liệt kê:** Xem xét tất cả các cấu hình tổ hợp thỏa mãn các ràng buộc của bài toán. Phương pháp giải thường đưa về một thuật toán vét cạn (ví dụ: thuật toán sinh, thuật toán quay lui,...).
- **Bài toán tối ưu:** Trong số cấu hình tổ hợp thỏa mãn yêu cầu của bài toán, xác định nghiệm có giá trị sử dụng tốt nhất, hay còn gọi là **tối ưu hàm mục tiêu**.

Phát biểu bài toán tối ưu (tối ưu tổ hợp)

- **Tìm cực đại (hoặc cực tiểu):**

$$f(x) \longrightarrow \max \text{ (hoặc } \min)$$

với điều kiện $x \in D$, trong đó D là một tập hữu hạn.

- Hàm $f(x)$ được gọi là hàm mục tiêu của bài toán.
- Mỗi phần tử $x \in D$ được gọi là một phương án, D là tập phương án của bài toán.
- Phương án $x^* \in D$ làm cho hàm mục tiêu có giá trị lớn nhất (nhỏ nhất) được gọi là phương án tối ưu của bài toán.
- $f^* = f(x^*)$ được gọi là giá trị tối ưu của bài toán.

Ví dụ 1

- **Bài toán cái túi - knapsack problem:**
  - Một nhà thám hiểm cần đem theo một cái túi trọng lượng không quá B.
  - Có N đồ vật cần đem theo. Đồ vật thứ i có trọng lượng a_i, có giá trị sử dụng c_i ($i = 1, 2, \ldots, N; a_i, c_i \in Z+$).
  - Hãy tìm cách đưa đồ vật vào túi cho nhà thám hiểm sao cho tổng giá trị sử dụng các đồ vật trong túi là lớn nhất.

Ví dụ 1

- **Tập phương án của bài toán:** Mỗi phương án của bài toán có thể xem như là một xâu nhị phân có độ dài N. Trong đó:
  - $x_i = 1$ ứng với đồ vật i được đưa vào túi.
  - $x_i = 0$ ứng với đồ vật i không được đưa vào túi.
  - Tập các xâu nhị phân $X = (x_1, \ldots, x_N)$ còn phải thỏa mãn điều kiện: tổng trọng lượng không vượt quá B. Nói cách khác, *tập phương án D của bài toán là tập các xâu nhị phân* thỏa mãn:

$$D = \left\{ X = (x_1, x_2, \ldots, x_N) : g(X) = \sum_{i=1}^{N} a_t x_i \leq B; x_i = 0, 1 \right\}$$

Ví dụ 1

- **Hàm mục tiêu của bài toán:**
  - Trong tập phương án $X = (x_1, \ldots, x_N) \in D$, ta cần tìm phương án $X^* = ({x_1}^*, \ldots, {x_N}^*)$ sao cho tổng giá trị sử dụng các đồ vật trong túi là lớn nhất.
  - Tổng giá trị sử dụng các đồ vật được xác định theo công thức:

$$f(X) = \sum_{i=1}^{N} c_i x_i \to \max$$

- **Bài toán có thể phát biểu lại như sau:** Trong số các xâu nhị phân $X = (x_1, \ldots, x_N)$ mà $g(X) \leq B$, tìm vector X^* để hàm $f(X)$ lớn nhất.
- **Đầu vào - Input**:
  - Số lượng đồ vật: N
  - Vector trọng lượng: $A = (a_1, a_2, \ldots, a_N)$
  - Vector giá trị sử dụng: $C = (c_1, c_2, \ldots, c_N)$
- **Đầu ra - Output**:
  - Phương án tối ưu: $X^* = (x_1^*, \ldots, x_N^*) \in D$ để $\mathrm{f}(X^*)$ lớn nhất
  - Giá trị tối ưu của bài toán: $f(X^*)$

$$f(X) = \sum_{i=1}^{N} c_i x_i \rightarrow \max$$

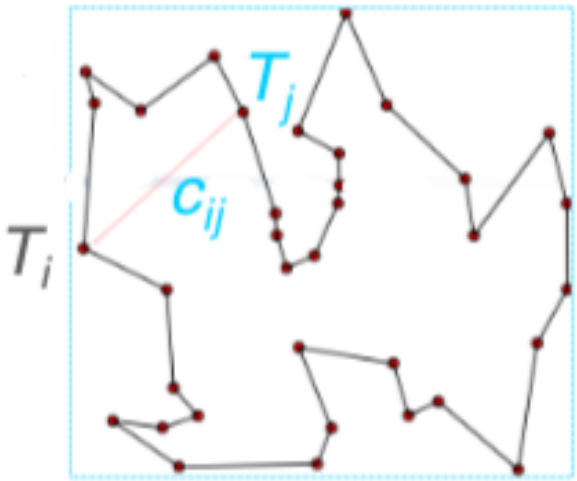
với

$$x \in \mathrm{x} \in D = \left\{ X = (x_1, x_2, .., x_N) : g(X) = \sum_{i=1}^{N} a_i x_i \leq B; x_i = 0, 1 \right\}$$

Ví dụ 2

Bài toán người du lịch

Một người du lịch muốn đi tham quan N thành phố $T_1, T_2, \ldots, T_N$. Xuất phát tại một thành phố nào đó, người du lịch muốn qua tất cả các thành phố còn lại mỗi thành phố đúng một lần rồi trở lại thành phố ban đầu. Gọi c_{ij} là chi phí đi lại từ thành phố T_i đến thành phố T_j. Hãy tìm một hành trình cho người đi du lịch sao cho tổng chi phí là nhỏ nhất.



Ví dụ 2

Tập phương án của bài toán

Không mất tính tổng quát của bài toán, ta cố định xuất phát là thành phố $T_1 = 1$. Khi đó, mỗi hành trình của người du lịch

$$T_1 \to T_2 \to \ldots \to T_N \to T_1$$

được xem như một hoán vị của $1, 2, \ldots, N$. Như vậy, tập phương án D của bài toán là tập các hoán vị của N thành phố.

Hàm mục tiêu của bài toán

Biến của bài toán là

$$x_{ij} = \begin{cases} 1 & \text{có đường đi từ thành phố } i \text{ tới thành phố } j \\ 0 & \text{ngược lại} \end{cases}$$

Để thoả mãn mỗi thành phố được tới đúng 1 lần ta cần có

$$\sum_{i=1, i\neq j}^{n} x_{ij} = 1 \text{ với } j = 1, \ldots, n \text{ và } \sum_{j=1, j\neq i}^{N} x_{ij} = 1 \text{ với } i = 1, \ldots, N.$$

Khi đó tối thiểu hoá tổng chi phí của hành trình, tức là

$$\sum_{i=1}^{N} \sum_{j\neq i, j=1}^{N} c_{ij} x_{ij} \to \min .$$

Ví dụ 3

Bài toán cho thuê máy

Một ông chủ có một chiếc máy cho thuê. Đầu tháng ông nhận được yêu cầu của M khách hàng thuê máy trong N ngày kế tiếp. Mỗi khách hàng i cho biết tập N_i ngày họ cần thuê máy. Ông chủ có quyền hoặc từ chối yêu cầu của khách hàng, hoặc nếu chấp nhận yêu cầu của khách ông phải bố trí máy theo đúng những ngày mà khách yêu cầu.
Hãy tìm phương án thuê máy giúp ông chủ sao cho tổng số ngày thuê máy là nhiều nhất.

Ví dụ 3

```
So khach hang: 10, so ngay may lam viec cua thang: 20
Ma tran don hang:
       1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
KH 1:  _  _  _  _  1  _  _  _  _  _  _  _  _  _  _  _  _  1  _  _
KH 2:  _  1  _  1  _  _  _  _  _  _  _  1  _  1  _  _  _  _  _  _
KH 3:  1  _  _  _  _  1  _  _  _  _  _  _  _  _  1  _  _  1  _  _
KH 4:  _  _  1  _  _  _  _  _  _  1  _  _  _  _  _  _  _  _  1  _
KH 5:  _  1  1  1  _  _  _  _  _  _  _  _  _  _  1  _  _  _  _  _
KH 6:  _  _  _  _  _  _  1  _  _  _  _  _  _  _  _  1  _  _  _  _
KH 7:  _  _  _  _  _  1  _  _  _  _  _  _  _  _  _  _  _  _  1  _
KH 8:  _  _  _  _  _  _  _  _  _  _  _  _  1  _  _  _  _  _  _  _
KH 9:  1  _  _  _  1  _  _  _  _  1  _  _  _  _  _  _  _  _  _  _
KH10:  _  _  _  _  _  _  _  _  _  _  _  1  _  _  _  _  _  _  _  1
So ngay nhieu nhat cho thue may, khong trung lap: 14
                                        1  2  3  4  5  6  7  8  9 10
Lua chon 1 (1: khach duoc chon):        _  _  _  _  1  1  1  1  1  1
Lua chon 2 (1: khach duoc chon):        _  1  1  1  _  1  _  1  _  _
```

Ví dụ 3

- **Tập phương án của bài toán**: Gọi $I = \{1, 2, \ldots, M\}$ là tập chỉ số khách hàng. Khi đó, tập phương án D của bài toán là tập tất cả các tập con của I thỏa mãn:

$$D = \{J \in S \mid N_k \cap N_p = \phi, \forall k, p \in J\}$$

- **Xây dựng hàm mục tiêu**: Ứng với mỗi phương án $J \in D$, tổng số ngày cho thuê máy là:

$$f(J) = \sum_{j \in J} |N_j| \to \max$$

Ví dụ 3

Bài toán có thể được mô tả lại như sau:

- **Đầu vào - Input**:
  - Số lượng khách hàng M
  - Số ngày thuê máy N
  - Ma trận $[0, 1]$ mô tả số ngày thuê máy mỗi khách: $A[M][N]$
- **Đầu ra - Output**:
  - Phương án tối ưu của bài toán: $J^* \in S$
  - Giá trị tối ưu: $f(J^*)$
  - Trong đó:

$$f(J) = \sum_{j \in J} |N_j| \rightarrow \max$$

với $x \in D = \{J \in S \mid N_k \bigcap N_p = \phi, \forall k, p \in J\}$.

Ví dụ 4

Bài toán phân công công việc

Một hệ gồm có N người thực hiện N việc song hành. Biết mỗi người đều có thể thực hiện được N việc kể trên nhưng với chi phí thời gian khác nhau. Biết c_{ij} là thời gian người i thực hiện việc j. Hãy tìm phương án giao việc cho mỗi người sao cho tổng thời gian thực hiện N việc kể trên là nhỏ nhất.

Ví dụ 4

Bài toán phân công công việc

Một hệ gồm có N người thực hiện N việc song hành. Biết mỗi người đều có thể thực hiện được N việc kể trên nhưng với chi phí thời gian khác nhau. Biết c_{ij} là thời gian người i thực hiện việc j. Hãy tìm phương án giao việc cho mỗi người sao cho tổng thời gian thực hiện N việc kể trên là nhỏ nhất.

Ví dụ 4

```
So nguoi (= so viec): 10
Ma tran chi phi thoi gian:
      T1  T2  T3  T4  T5  T6  T7  T8  T9  T10
Ng 1:  3   1   5   6   1   6   5   4   2   1
Ng 2:  7   8   6   1   9   4   9   2   1   2
Ng 3:  9   3   3   2   1   6   4   1   6   8
Ng 4:  1   2   8   3   6   3   4   6   1   7
Ng 5:  6   2   1   5   6   8   5   5   3   1
Ng 6:  4   7   1   8   9   1   4   5   5   3
Ng 7:  7   6   7   3   1   5   5   5   6   1
Ng 8:  1   7   3   8   2   7   1   4   2   6
Ng 9:  8   8   8   8   1   5   6   8   1   7
Ng10:  4   1   3   8   5   5   5   1   6   9
Thoi gian it nhat thuc hien 10 viec: 10
Lua chon 1
N1->v2 N2->v4 N3->v5 N4->v1 N5->v3 N6->v6 N7->v10 N8->v7 N9->v9 N10->v8
Lua chon 2
N1->v5 N2->v4 N3->v8 N4->v1 N5->v3 N6->v6 N7->v10 N8->v7 N9->v9 N10->v2
Lua chon 3
N1->v10 N2->v4 N3->v8 N4->v1 N5->v3 N6->v6 N7->v5 N8->v7 N9->v9 N10->v2
```

Ví dụ 4

- **Tập phương án của bài toán**: Gọi $X = (x_1, x_2, \ldots, x_N)$ là một hoán vị của $1, 2, \ldots, N$. Nếu $x_i = j$ thì ta xem quá trình thứ i được thực hiện việc j. Như' vậy, tập phương án của bài toán chính là tập các hoán vị của $1, 2, \ldots, N$.

  $$D = \{X = (x_1, x_2, \ldots, x_N : (\forall i \neq j) \mid x_i \neq x_j, i, j = 1, 2, \ldots, N\}$$

- **Hàm mục tiêu của bài toán**: Ứng với mỗi phương án $X \in D$, thời gian thực hiện của mỗi phương án là:

$$f(X) = \sum_{i=1}^{N} c[i, x[i]] \rightarrow \min$$

Ví dụ 4

Bài toán có thể được mô tả lại như sau:

- **Đầu vào - Input**:
  - Số lượng quá trình: N
  - Ma trận chi phí thời gian: $C[N][N]$
- **Đầu ra - Output**:
  - Phương án tối ưu của bài toán: $X^* \in D$
  - Giá trị tối ưu: $f(X^*)$
  - Trong đó:

$$D = \{X = (x_1, x_2, \ldots, x_N) : (\forall i \neq j) \mid x_i \neq x_j, i, j = 1, 2, \ldots, N\}$$

$$f(X) = \sum_{i=1}^{N} c[i, x[i]] \rightarrow \min \qquad \text{với } X \in D$$

Nội dung chính

Thuật toán duyệt toàn bộ (Thuật toán vét cạn)

Giả sử D là tập phương án. Ta cần tìm $X^* \in D$ sao cho $f(X^*) \to \max(\min)$. Phương pháp duyệt toàn bộ được tiến hành như sau:

Bước 1 (Khởi tạo):

$XOPT = \emptyset$; //Phương án tối ưu

$FOPT = -\infty(+\infty)$; // Giá trị tối ưu

Bước 2 (Lặp):

for $(X \in D)\{$ //lấy mỗi phần tử trên tập phương án

$S = f(X)$; tính giá trị hàm mục tiêu cho phương án X

if $(FOPT < S)\{$ // Cập nhật phương án tối ưu

$FOPT = S$; // Giá trị tối ưu mới được xác lập

$XOPT = X$; // Phương án tối ưu mới

$\}$

$\}$

Bước 3 (Trả lại kết quả): Return$(XOPT, FOPT)$;

Thuật toán duyệt toàn bộ (Thuật toán vét cạn)

- Ưu điểm: Đơn giản, dễ cài đặt, có thể giải quyết được với mọi bài toán tối ưu.
- Nhược điểm: Chi phí tính toán lớn.

Bài tập

Viết chương trình giải các bài toán dưới đây trên máy tính sử dụng thuật toán duyệt toàn bộ:

1. Bài toán cái túi
2. Bài toán người du lịch
3. Bài toán cho thuê máy
4. Bài toán phân công công việc.

Nội dung chính

Thuật toán nhánh cận (Branch & Bound algorithm)

Bài toán tối ưu tổ hợp được phát biểu lại dưới dạng sau:

$$\min\{f(X) : X \in D\}$$

Trong đó, D là tập hữu hạn các phần tử. Tập D có thể được mô tả như sau:

$$D = \Big\{X = (x_1, x_2, \ldots, x_n) \in A_1 \times A_2 \times \cdots \times A_n : X \text{ thỏa mãn tính chất } P\Big\}$$

Thuật toán nhánh cận (Branch & Bound algorithm)

Xây dựng hàm xác định cận dưới của hàm mục tiêu:

- Thuật toán nhánh cận có thể giải được bài toán đặt ra nếu ta tìm được một hàm g xác định trên tất cả phương án bộ phận của bài toán thỏa mãn bất đẳng thức

$$g(a_1, a_2, \ldots, a_k) \leq \min\left\{f(\dot{X}) : X \in D, x_i = a_i, i = 1, 2, \ldots, k\right\}$$

- Nói cách khác, giá trị của hàm g tại phương án bộ phận cấp k $(a_1, a_2, \ldots, a_k)$ không vượt quá giá trị nhỏ nhất của hàm mục tiêu trên tập con các phương án

$$D(a_1, a_2, \ldots, a_k) = \{X \in D : x_i = a_i, i = 1, 2, \ldots, k\}$$

- Giá trị của hàm $g(a_1, a_2, \ldots, a_k)$ là cận dưới của hàm mục tiêu trên tập $D(a_1, a_2, \ldots, a_k)$. Hàm g được gọi là hàm cận dưới, $g(a_1, a_2, \ldots, a_k)$ gọi là cận dưới của tập $D(a_1, a_2, \ldots, a_k)$.

Hạn chế của các phương án duyệt

Giả sử ta đã có hàm g. Để giảm bớt khối lượng duyệt trên tập phương án trong quá trình liệt kê bằng thuật toán quay lui ta xác định được X^* là phương án làm cho hàm mục tiêu có giá trị nhỏ nhất trong số các phương án tìm được $f^* = f(X^*)$. Ta gọi X^* là phương án tốt nhất hiện có, f^* là giá trị tối ưu hiện tại.

- Nếu

$$f^* < g(a_1, a_2, \ldots, a_k)$$

thì

$$f^* < g(a_1, a_2, \ldots, a_k) \leq \min\{f(X) : X \in D, x_i = a_i, i = 1, 2, \ldots, k\}$$

- Điều này có nghĩa tập $D(a_1, a_2, \ldots, a_k)$ chắc chắn không chứa phương án tối ưu. Trong trường hợp này ta không cần phải triển khai phương án bộ phận $(a_1, a_2, \ldots, a_k)$.
- Tập $D(a_1, a_2, \ldots, a_k)$ cũng bị loại bỏ khỏi quá trình duyệt. Nhờ đó, số các tập cần duyệt nhỏ đi trong quá trình tìm kiếm.

Thuật toán nhánh cận

Thuật toán Branch-And-Bound (k)\{
    for $(a_k \in A_k)$ \{
        if (<chấp nhận a_k>) \{
            $x_k = a_k$;
            if $(k == n)$
                <cập nhật giá trị tối ưu>;
            elif $(g(a_1, a_2, \ldots, a_k) < f^*)$
                Branch-And-Bound $(k+1)$;
        \}
    \}
\}

Xây dựng hàm g

Việc xây dựng hàm g phụ thuộc vào từng bài toán tối ưu tổ hợp cụ thể. Ta cố gắng xây dựng g sao cho:

- Việc tính giá trị của g phải đơn giản hơn việc giải bài toán tối ưu tổ hợp

$$\min\{f(X) : X \in D, x_i = a_i, i = 1, 2, \ldots, k\}$$

- Giá trị của $g(a_1, a_2, \ldots, a_k)$ phải sát với giá trị

$$\min\{f(X) : X \in D, x_i = a_i, i = 1, 2, \ldots, k\}$$

$\Longrightarrow$ Hai yêu cầu này thường đối lập nhau và khó thỏa mãn đông thời trong thực tế

Giải bài toán bằng pp Nhánh cận

Một dạng khác của bài toán cái túi

Giả sử có n loại đồ vật và số lượng đồ vật mỗi loại không hạn chế. Đồ vật loại j có trọng lượng a_j và giá trị sử dụng là c_j ($j = 1, 2, \ldots, n$). Cần xếp các đồ vật này vào một cái túi có trọng lượng b sao cho tổng giá trị sử dụng của các đồ vật trong túi là lớn nhất (mỗi loại đồ vật có thể lấy nhiều lần).

Giải bài toán bằng pp Nhánh cận

Bài toán cái túi có thể được phát biểu tổng quát dưới dạng sau:

- Tìm giá trị lớn nhất của hàm mục tiêu $f(X)$ với $X \in D$.
  Trong đó, $f(X)$ được xác định như dưới đây:

$$D = \left\{ X = (x_1, x_2, \ldots, x_n) : \sum_{i=1}^{n} a_i x_i \leq b, x_i \in Z_+, i = 1, 2, \ldots, n \right\}$$

$$f^* = \max \left\{ f(X) = \sum_{i=1}^{n} c_i x_i : \sum_{i=1}^{n} a_i x_i \leq b, x_i \in Z_+, i = 1, 2, \ldots, n \right\}$$

- Ví dụ về một bài toán cái túi:

$$f(X) = 10x_1 + 5x_2 + 3x_3 + 6x_4 \rightarrow \max,$$
$$5x_1 + 3x_2 + 2x_3 + 4x_4 \leq 8,$$
$$x_j \in Z_+, j = 1, 2, 3, 4.$$

Giải bài toán bằng pp Nhánh cận

Bước 1: Sắp xếp các đồ vật thỏa mãn

$$\frac{c_1}{a_1} \geq \frac{c_2}{a_2} \geq \ldots \geq \frac{c_n}{a_n}$$

Bước 2 (Lặp): Lập trên các bài toán bộ phận cấp $k = 1, 2, \ldots, n$.

- Giá trị sử dụng của k đồ vật trong túi: $\delta_k = \sum_{i=1}^{k} c_i x_i$.

- Trọng lượng còn lại của túi: $b_k = b - \sum_{i=1}^{k} a_i x_i$.

- Cận trên của phương án bộ phận cấp k:

$$g(x_1, x_2, \ldots, x_k) = \delta_k + b_k \frac{c_{k+1}}{a_{k+1}}$$

Bước 3 (Trả lại kết quả): Phương án tối ưu và giá trị tối ưu tìm được

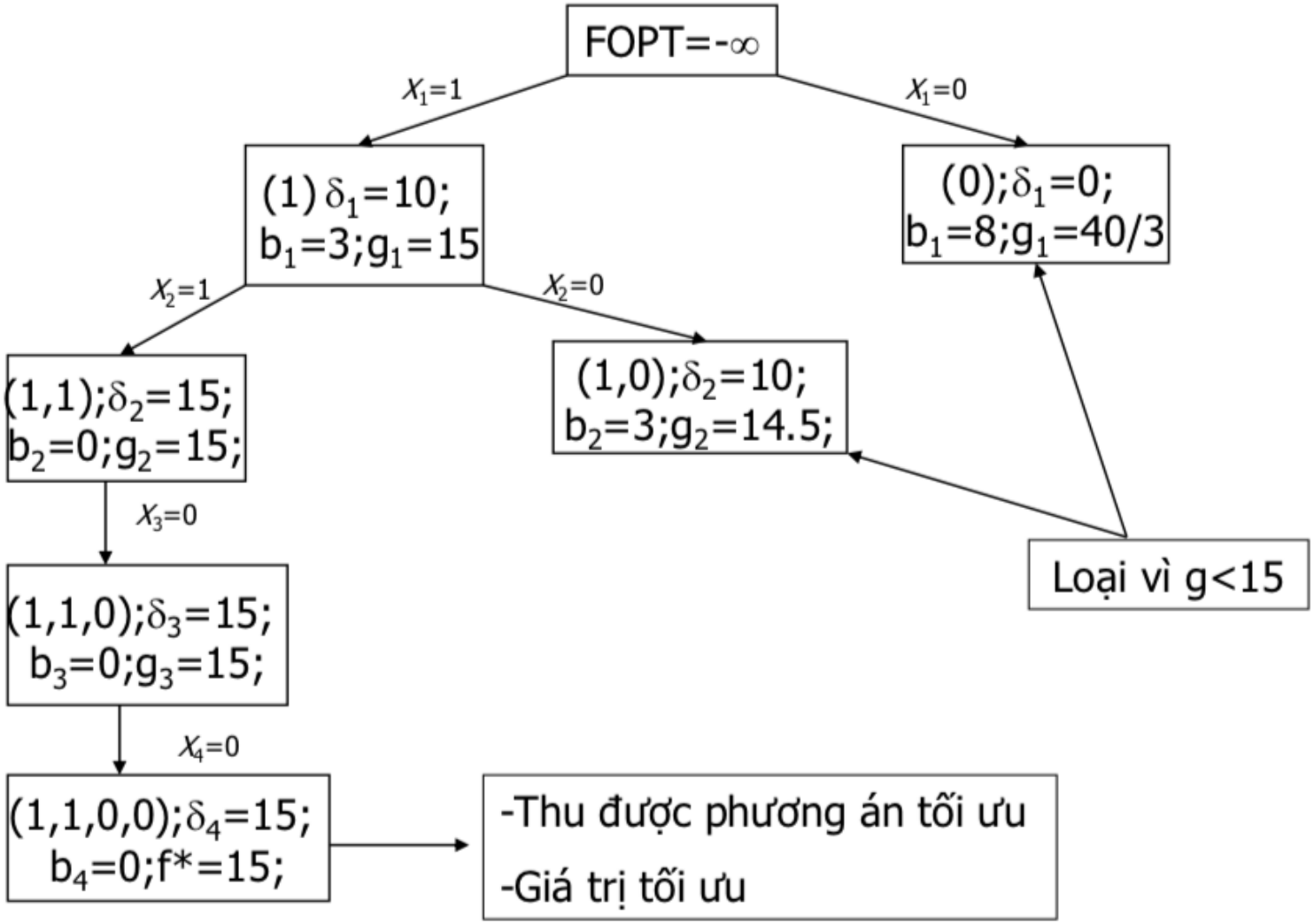
Thuật toán nhánh cận

Thuật toán Branch-And-Bound(k){
    for ($j = \lfloor b_k/a_k \rfloor ; j >= 0; j--$){
        $x[k] = j$;
        $\delta_k = \delta_k + c_k x_k$; $b_k = b_k - a_k x_k$;
        if ($k == n$)
            <cập nhật giá trị tối ưu>;
        elif ($\delta_k + (c_{k+1} b_k)/a_{k+1} >$ FOPT)
            Branch-And-Bound($k+1$);
    }
}

Ví dụ 5

$$f(X) = 10x_1 + 5x_2 + 3x_3 + 6x_4 \rightarrow \max$$
$$5x_1 + 3x_2 + 2x_3 + 4x_4 \leq 8$$
$$x_j \in Z_+, j = 1, 2, 3, 4$$

Ví dụ 5


FOPT=-∞
X1=1
X1=0
(1) δ1=10; b1=3;g1=15
(0);δ1=0; b1=8;g1=40/3
X2=1
X2=0
(1,1);δ2=15; b2=0;g2=15;
(1,0);δ2=10; b2=3;g2=14.5;
X3=0
Loại vì g<15
(1,1,0);δ3=15; b3=0;g3=15;
X4=0
(1,1,0,0);δ4=15; b4=0;f*=15;
-Thu được phương án tối ưu
-Giá trị tối ưu

Bài tập 5

Áp dụng thuật toán nhánh cận giải bài toán cái túi dưới đây, chỉ rõ kết quả theo mỗi bước.

$$f(X) = 7x_1 + 4x_2 + 2x_3 \to \max$$
$$4x_1 + 3x_2 + 2x_3 \leq 6$$
$$x_1, x_2, x_3 \in \{0, 1\}.$$

Bài tập 6

Áp dụng thuật toán nhánh cận giải bài toán cái túi dưới đây, chỉ rõ kết quả theo mỗi bước.

$$f(X) = 5x_1 + x_2 + 8x_3 + x_4 \to \max$$
$$4x_1 + 2x_2 + 7x_3 + x_4 \leq 9,$$
$$x_1, x_2, x_3, x_4 \in \{0, 1\}.$$

Giải bài toán người đi du lịch bằng thuật toán nhánh cận

Bài toán người du lịch có thể được phát biểu tổng quát dưới dạng sau: Tìm giá trị nhỏ nhất của hàm mục tiêu $f(X)$ với $X \in D$. Trong đó, $f(X)$ được xác định như dưới đây:

$$D = \{X = (x_1, x_2, \ldots, x_n) : x_1 = 1 \wedge x_i \neq x_j, i, j = 1, 2, .., n\}$$

$$f^* = \min \left\{ f(X) = \sum_{i=1}^{n-1} c\left[x_i, x_{i+1}\right] + c\left[x_n, x_1\right] : X \in D \right\}$$

Giải bài toán người đi du lịch bằng thuật toán nhánh cận

Ví dụ về ma trận chi phí

0	3	14	18	15
3	0	4	22	20
17	9	0	16	4
6	3	7	0	12
9	15	11	5	0

Gọi $c_{\min} = \min\{c[i,j], i,j = 1,2,\ldots,n, i \neq j\}$ là giá trị nhỏ nhất của ma trận chi phí. Phương pháp đánh giá cận dưới của mỗi bài toán bộ phận cấp k được tiến hành như sau. Giả sử ta đang có hành trình bộ phận qua k thành phố:

$$T_1 \to T_{u_2} \to \ldots \to T_{u_k} \, (T_1 = 1) \, .$$

Khi đó, chi phí của phương án bộ phận cấp k là:

$$\delta = c\,[1, u_2] + c\,[u_2, u_3] + \cdots + c\,[u_{k-1}, u_k] \, .$$

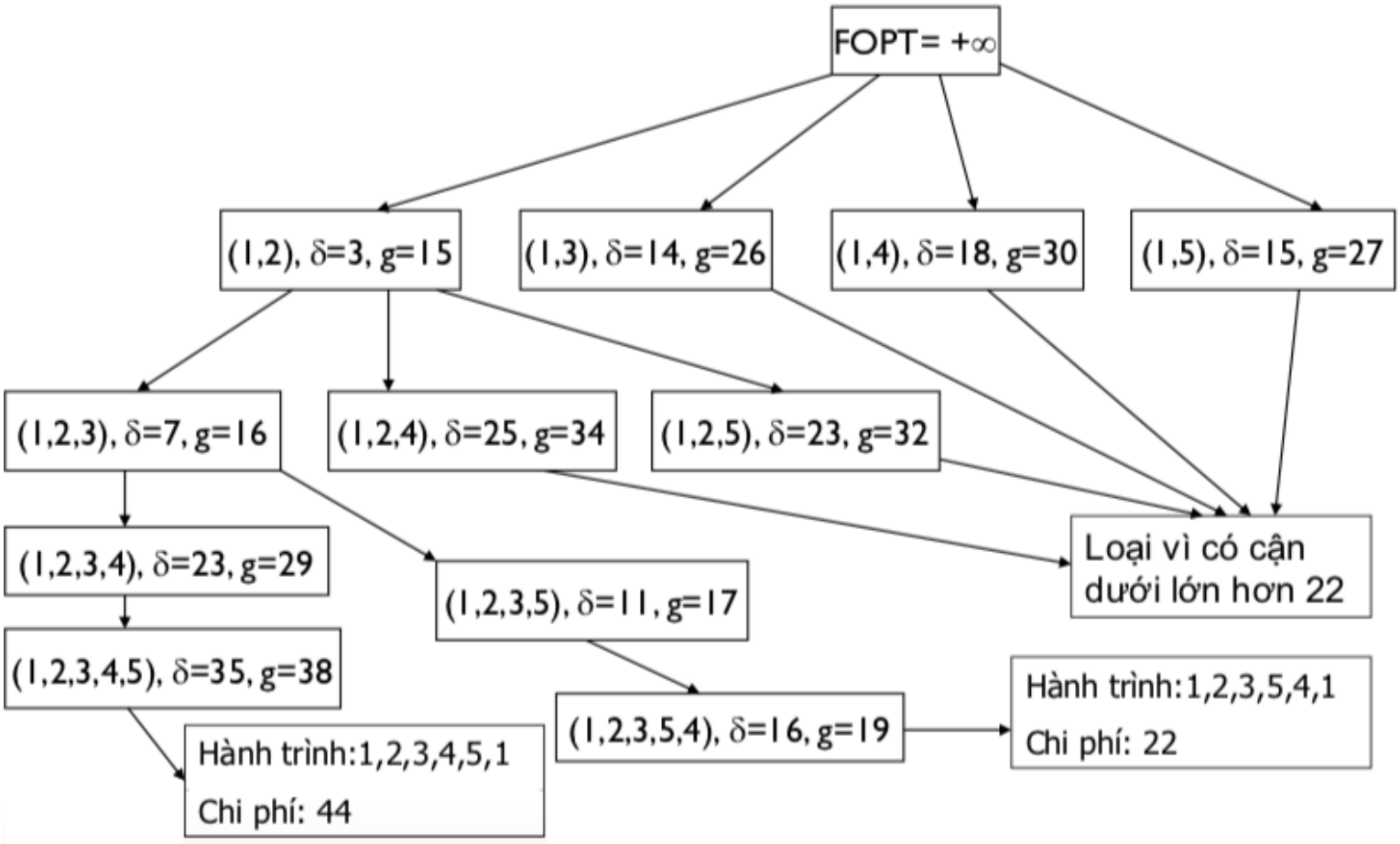
Để phát triển hành trình bộ phận này thành hành trình đầy đủ, ta cần phải qua $n - k$ thành phố nữa rồi quay trở về thành phố số 1. Như vậy, ta cần phải qua $n - k + 1$ đoạn đường nữa. Vì mỗi đoạn đường đều có chi phí không nhỏ hơn $c_{\min}$, nên cận dưới của phương án bộ phận có thể được xác định:

$$g\,(u_1, u_2, \ldots, u_k) = \delta + (n - k + 1) c_{\min} .$$

Giải bài toán người đi du lịch bằng thuật toán nhánh cận

Thuật toán Branch-And-Bound(k){
    for ($j = 2; j <= n; j++$){
        if (chuaxet[j])
            $x[k] = j$; chuaxet [j] = False
            $\delta = \delta + c[x[k-1], x[k]]$
        if ($k == n$)
            <cập nhật giá trị tối ưu>;
        elif ($\delta + (n - k + 1) * c_{min} <$ FOPT)
            Branch-And-Bound($k + 1$);
        chuaxet [j] = True; $\delta = \delta - c[x[k-1], x[k]]$;
    }
}

Giải bài toán người đi du lịch bằng thuật toán nhánh cận



Giải bài toán người đi du lịch bằng thuật toán nhánh cận

Bài tập 8: Lập trình thuật toán nhánh cận giải bài toán đi du lịch.