Học viện Công nghệ Bưu chính Viễn thông
Khoa Công nghệ thông tin 1

Toán rời rạc 1

Bài toán tối ưu

Nội dung

- Giới thiệu bài toán
- Thuật toán duyệt toàn bộ
- Thuật toán nhánh cận
- Bài tập

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 một 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 (thuật toán sinh, thuật toán quay lui,…).

- **Bài toán tối ưu:** Trong số cấu hình tổ hợp thỏa mãn yêu cầu của bài toán, hãy lựa chọn nghiệm có giá trị sử dụng tốt nhất (tối ưu hàm mục tiêu).

Phát biểu bài toán tối ưu

- Tìm cực đại (hoặc cực tiểu) của $f(x) \rightarrow Max(Min)$ với điều kiện $x \in D$, trong đó D là tập hợp 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 (1/4)

- **Bài toán cái túi**: 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,..., N; a_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 (2/4)

- **Tập phương án của bài toán:** Mỗi phương án của bài toán là một xâu nhị phân có độ dài N. Trong đó, $x_i = 1$ tương ứng với đồ vật i được đưa vào túi, $x_i = 0$ tương 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 được xác định như công thức dưới đây.

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

Ví dụ 1 (3/4)

- **Hàm mục tiêu của bài toán:** Ứng với mỗi 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 sau:

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

Ví dụ 1 (4/4)

- **Bài toán có thể phát biểu lại như sau:** Trong số các xâu nhị phân $X = (x_1, \ldots, x_N)$ thỏa mãn điều kiện $g(X) \le B$, hãy tìm vector X^* để hàm $f(X)$ có giá trị 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 của bài toán: $X^* = (x_1^*, \ldots, x_N^*) \in D$ để $f(X^*)$ lớn nhất
  - Giá trị tối ưu của bài toán: $f(X^*)$

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

$$f(X) = \sum_{i=1}^{N} c_i x_i \to \max \quad \text{với } X \in D$$

Ví dụ 2 (1/4)

- **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. Biết 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 có tổng chi phí là nhỏ nhất.

Ví dụ 2 (2/4)

- **Tập phương án của bài toán:** Không hạn chế 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 $T_1 \rightarrow T_2 \rightarrow \ldots \rightarrow T_N \rightarrow T_1$ được xem như một hoán vị của $1,2,\ldots,N$ là $X = (x_1, x_2, \ldots, x_N)$, trong đó $x_1 = 1$. Như vậy, tập phương án D của bài toán là tập các hoán vị $X = (x_1, x_2, \ldots, x_N)$ với $x_1 = 1$.

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

Ví dụ 2 (3/4)

- **Hàm mục tiêu của bài toán:** Ứng với mỗi phương án $X = (x_1, \ldots, x_N) \in D$, chi phí đi lại từ thành phố thứ i đến thành phố j là $C[X[i]][X[i+1]]$ $(i = 1, 2, \ldots, N-1)$. Sau đó ta quay lại thành phố ban đầu với chi phí là $C[X[N]][X[1]]$. Như vậy, tổng chi phí của toàn bộ hành trình được xác1định theo công thức:

$$f(X) = \sum_{i=1}^{N-1} c_{x_i x_{i+1}} + c_{x_N x_1} \rightarrow \min$$

Ví dụ 2 (4/4)

- Bài toán có thể mô tả lại như sau
- **Đầu vào (Input):**
  - Số lượng thành phố: N
  - Ma trận chi phí cấp N:$\{c_{ij}\}$
- **Đầu ra (Output):**
  - Phương án tối ưu của bài toán: $X^* = (x_1^*, \ldots, x_N^*) \in D$ để f(X^*) nhỏ nhất
  - Giá trị tối ưu của bài toán: f(X^*)

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

$$f(X) = \sum_{i=1}^{N-1} c_{x_i x_{i+1}} + c_{x_N x_1} \rightarrow \min \quad \text{với } X \in D$$

Ví dụ 3 (1/3)

- **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 cho N ngày kế tiếp. Mỗi khách hàng i cho biết tập Ni 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 (2/3)

- Tập phương án của bài toán: Gọi $I = \{ 1, 2, . . . , M\}$ là tập chỉ số khách hàng, S là tập của tất các tập con của I. Khi đó, tập các phương án cho thuê máy là:

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

- 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 (3/3)

- **Bài toán có thể 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:

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

$$f(J) = \sum_{j \in J} |N_j| \to \max \qquad \text{với } X \in D$$

Ví dụ 4 (1/3)

- **Bài toán phân công công việc:** Một hệ gồm có N quá trình thực hiện N việc song hành. Biết mỗi quá trình đều có thể thực hiện được N việc kể trên nhưng với chi phí thời gian khác nhau. Biết c_{ij} là thời gian quá trình i thực hiện việc j. Hãy tìm phương án giao việc cho mỗi quá trình sao cho tổng thời gian thực hiện N việc kể trên là ít nhất.

Ví dụ 4 (2/3)

- **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 $1, 2, \ldots, N$.

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

- **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 (3/3)

- Bài toán có thể 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:

$$D = \left\{X = (x_1, x_2, ..., x_N) : \left(\forall i \neq j\right) | \, x_i \neq x_j, i, j = 1,2,...,N\right\}$$

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

Nội dung

- Giới thiệu bài toán
- Thuật toán duyệt toàn bộ
- Thuật toán nhánh cận
- Bài tập

Thuật toán duyệt toàn bộ

- Giả sử D là tập phương án. Ta cần tìm $X* \in D$ sao cho $\boldsymbol{f(X*)}$ $\longrightarrow$ **max(min)**

Phương pháp duyệt toàn bộ được tiến hành như sau:

<u>Bước 1</u> (*Khởi tạo*)**:**

$XOPT = \emptyset$;//Phương án tối ưu

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

<u>Bước 2</u>(*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

}

}

<u>Bước 3</u> (*Trả lại kết quả*):

Return($XOPT$, $FOPT$);

Đặc điểm thuật toán duyệt toàn bộ

- **Ưu điểm:**
  - Đơn giản dễ cài đặt
  - Có thể thực hiện trên 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:
  - Bài toán cái túi
  - Bài toán người du lịch
  - Bài toán cho thuê máy
  - Bài toán phân công công công việc

Nội dung

- Giới thiệu bài toán
- Thuật toán duyệt toàn bộ
- Thuật toán nhánh cận
- Bài tập

Thuật toán nhánh cận

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

  Tìm $\min\{f(X) : X \in D\}$

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

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

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

- 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 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 kỷ lục 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 (giả mã)

```
Thuật toán Branch_And_Bound (k) {
    for (a_k ∈ A_k){
        if(<chấp nhận a_k>){
            x_k = a_k;
            if ( k == n )
                <Cập nhật kỷ lục>;
            else if ( g(a_1, a_2, ..., 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

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

Hai yêu cầu này thường đối lập nhau
và khó thỏa mãn đồng thời trong thực tế

Giải bài toán cái túi bằng thuật toán nhánh cận (1/4)

- **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, … , n). Cần chất 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 cái túi bằng thuật toán nhánh cận (2/4)

- 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:

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

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

- Ví dụ về một bài toán cái túi:

$$\begin{aligned} f(X) = 10x_1 + 5x_2 + 3x_3 + 6x_4 &\to \max, \\ 5x_1 + 3x_2 + 2x_3 + 4x_4 &\le 8, \\ x_j \in Z_+, j &= 1,2,3,4. \end{aligned}$$

Giải bài toán cái túi bằng thuật toán nhánh cận (3/4)

- **Bước 1**: Sắp xếp các đồ vật thỏa mãn

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

- **Bước 2** (Lặp): Lặp trên các bài toán bộ phận cấp $k = 1, 2, \dots, n$:
  - 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:

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

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

Giải bài toán cái túi bằng thuật toán nhánh cận (4/4)

Thuật toán Branch_And_Bound (k) {

    for (j = b_k/a_k ; j>=0; j--){

        x[k] =j;

        $\delta_k = \delta_k + c_k x_k;$ $\quad b_k = b_k - a_k x_k;$

        if (k==n) <Ghi nhận kỷ lục>;

        else if (δ_k + (c_{k+1}*b_k)/a_{k+1} >FOPT)

            Branch_And_Bound(k+1);

        $\delta_k = \delta_k - c_k x_k;$ $\quad b_k = b_k + a_k x_k;$

    }

}

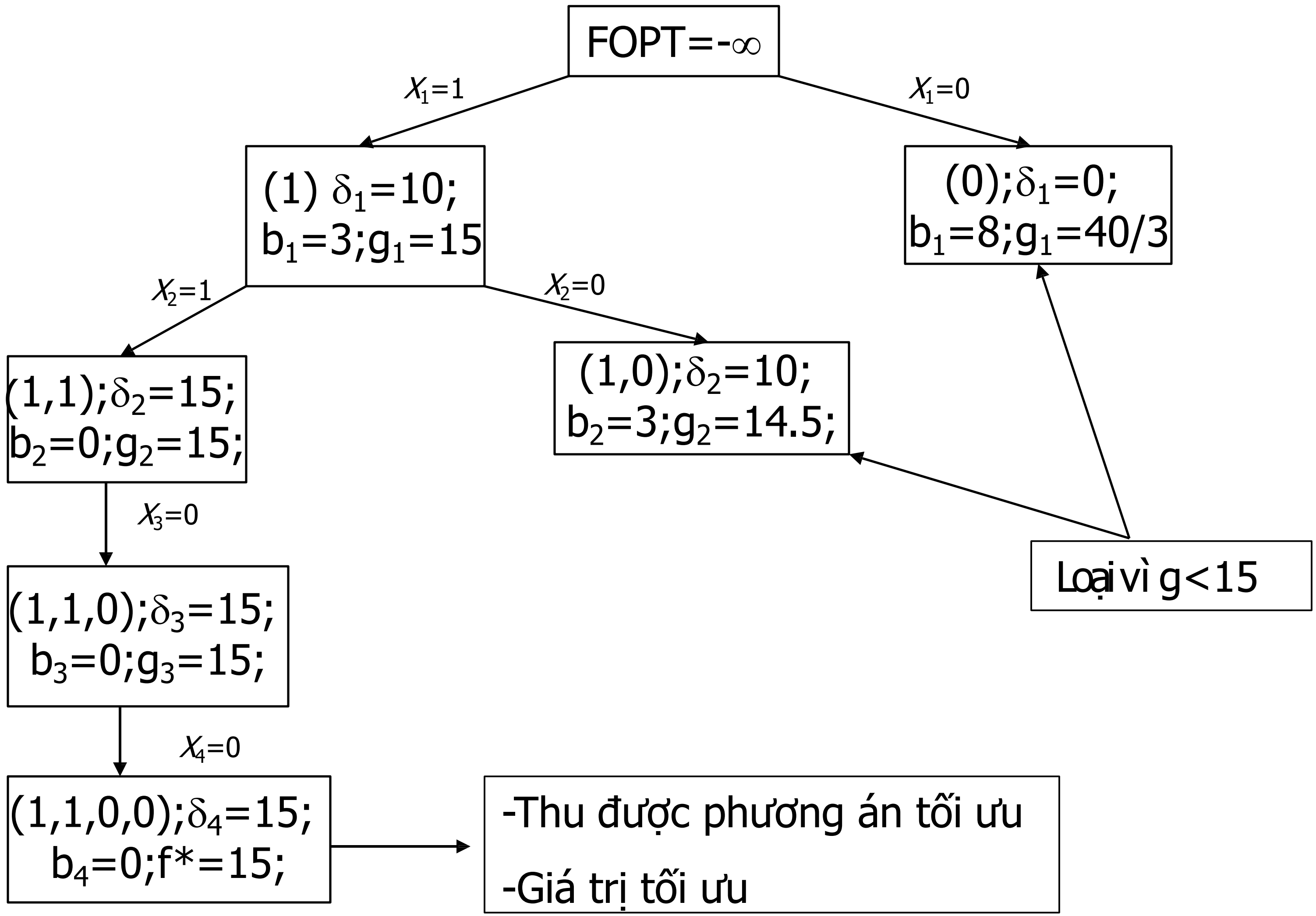
Ví dụ 5 (1/2)

$$f(X) = 10x_1 + 5x_2 + 3x_3 + 6x_4 \to \max,$$

$$5x_1 + 3x_2 + 2x_3 + 4x_4 \le 8,$$

$$x_j \in Z_+,\ j = 1,2,3,4.$$

Ví dụ 5 (2/2)


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 1

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 2

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

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

Giải bài toán người du lịch bằng thuật toán nhánh cận (1/4)

- 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:

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

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

- 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

Giải bài toán người du lịch bằng thuật toán nhánh cận (2/4)

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:

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

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 du lịch bằng thuật toán nhánh cận (3/4)

```
Thuật toán Branch_And_Bound (k) {
        for (j = 2; j<=n; j++){
                if ( chuaxet[j]) {
                        x[k] = j; chuaxet[j] = False;
                        δ = δ + c[x[k − 1], x[k]];
                        if (k==n) Ghi_Nhan();
                        else if (δ + (n − k + 1) *c_min < FOPT)
                                Branch_And_Bound (k+1);
                        chuaxet[j] = True; δ = δ − c[x[k − 1], x[k]];
                }
        }
}
```

Giải bài toán người du lịch bằng thuật toán nhánh cận (4/4)