Toán rời rạc 1

Discrete mathematics 1

Bài 5: Bài toán tối ưu

Optimization

Nội dung Bài 5

1. Giới thiệu một số bài toán tối ưu
2. Thuật toán duyệt toàn bộ -
   brute force, exhaustive search
3. Thuật toán nhánh cận -
   Branch and Bound algorithm – B&B
4. Bài tập

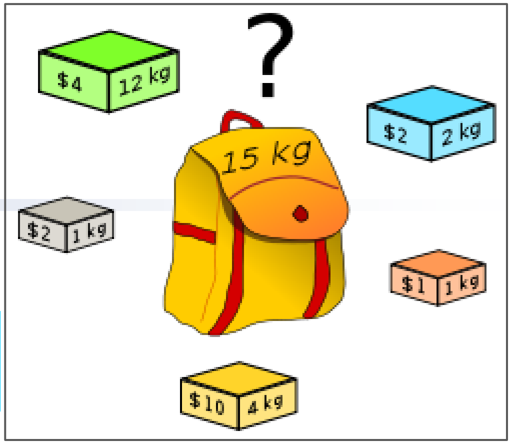
Giới thiệu bài toán tối ưu

- 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 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, hãy lựa chọn nghiệm có giá trị sử dụng tốt nhất - tối ưu hàm mục tiêu.

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

- Tìm cực đại (hoặc cực tiểu) của:
  $f(x) \rightarrow Max$ (hoặc 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 dạng 0 / 1 (không lấy / lấy 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, . . . , 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 (2/4)

- 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:

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

Ví dụ 1 (3/4)

- Hàm mục tiêu của bài toán:
  Trong tập phương án $X = (x_1, . . . , x_N) \in D$,
  ta cần tìm phương án $X^* = (x_1{}^*, . . . , x_N{}^*)$ 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:

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

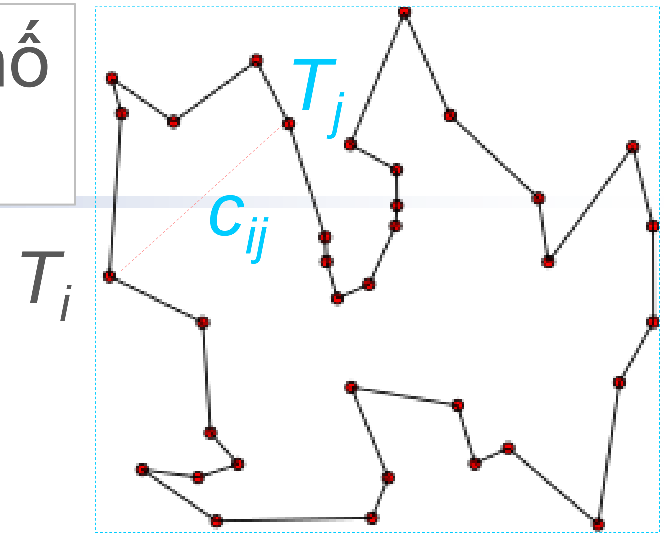
Ví dụ 1 (4/4)

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

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

Ví dụ 2 (1/4)

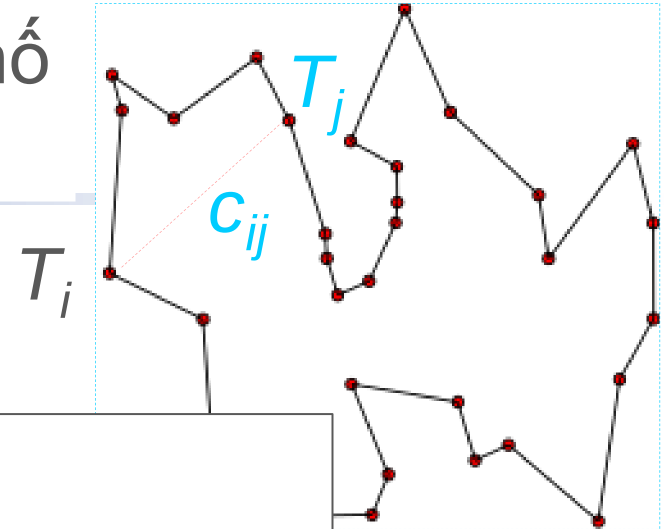
T_i, T_j: thành phố
c_{ij}: chi phí



- 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 (1/4)

T_i, T_j: thành phố
c_{ij}: chi phí



- Bài toán người du lịch:

```
So thanh pho: 5
Ma tran chi phi:
       TP1  TP2  TP3  TP4  TP5
TP1      0    3   14   18   15
TP2      3    0    4   22   20
TP3     17    9    0   16    4
TP4      6    2    7    0   12
TP5      9   15   11    5    0
Chi phi toi uu 22:
Hanh trinh:
 1 ->  2 ->  3 ->  5 ->  4 ->  1
```

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 có dạng:
  $T_1 \to T_2 \to \ldots \to T_N \to T_1$ được xem như một hoán vị của 1, 2, … , 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 thỏa mãn:
  $X = (x_1, x_2, \ldots, x_N)$ với $x_1 = 1$.

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

Ví dụ 2 (3/4)

- Hàm mục tiêu của bài toán:
  Ứng với mỗi phương án $X = (x_1, . . . , x_N) \in D$,
  chi phí đi lại từ thành phố thứ i đến thành phố i+1 là: $C[X[i]][X[i{+}1]]$ với (i = 1, 2, . . . , 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 xác định 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^*)

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

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

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 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 (1/3)

- Bài toán cho thuê máy:

```
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 (2/3)

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

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

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

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

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:

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

với x ∈

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

Ví dụ 4 (1/3)

- 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 (1/3)

- Bài toán phân công công việc:

```
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 (2/3)

- Tập phương án của bài toán:
  Gọi $X = (x_1, x_2, ..., x_N)$: một hoán vị của 1, 2, ..., N.
  Nếu $x_i = j$ thì ta có người i được thực hiện việc j.
  Như vậy, tập phương án D của bài toán chính là tập các hoán vị của 1, 2, . . . , N thỏa mãn:

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

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

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

Ví dụ 4 (3/3)

- Bài toán có thể mô tả lại như sau:
  - Đầu vào - Input:
    - Số lượng người: 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:

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

với x ∈

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

Nội dung Bài 5

1. Giới thiệu một số bài toán tối ưu

2. Thuật toán duyệt toàn bộ -
   brute force, exhaustive search

3. Thuật toán nhánh cận -
   Branch and Bound algorithm – B&B

4. Bài tập

Phương pháp: Duyệt toàn bộ

- Giả sử D là tập phương án, ta cần tìm $X* \in D$ sao cho $f(X^*) \rightarrow$ max (hoặc min).
  Duyệt toàn bộ được tiến hành như sau

```
Bước 1 (khởi tạo):
    XOPT = ∅;                       // XOPT: Phương án tối ưu
    FOPT = − ∞ (+ ∞);               // FOPT: Giá trị tối ưu, − ∞: tìm max,
Bước 2 (lặp):                       // + ∞: tìm min
    for (X ∈ 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);
```

Duyệt toàn bộ: Đặc điểm

- 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 1 Duyệt toàn bộ

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

  Bài toán cái túi dạng 0 / 1 - mỗi thứ hoặc không lấy, hoặc lấy 1 (vào từ file, ra màn hình).

```
So loai do vat: 4
Trong luong tui: 9
Vector trong luong:  4  2  7  1
Vector gia tri su dung:  5  1  8  1
Ket qua toi uu:   9.0
Phuong an toi uu:  0  0  1  1
```

Bài tập 2 Duyệt toàn bộ

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

  Bài toán người du lịch (vào từ file, ra màn hình).

```
So thanh pho: 5
Ma tran chi phi:
   0    3   14   18   15
   3    0    4   22   20
  17    9    0   16    4
   6    2    7    0   12
   9   15   11    5    0
Chi phi toi uu 22:
Hanh trinh:  1 ->  2 ->  3 ->  5 ->  4 ->  1
```

Bài tập 3 Duyệt toàn bộ

- Viết chương trình giải bài toán cho thuê máy

```
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  _  _
```

Bài tập 4 Duyệt toàn bộ

- Viết chương trình giải bài toán phân việc

```
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
```

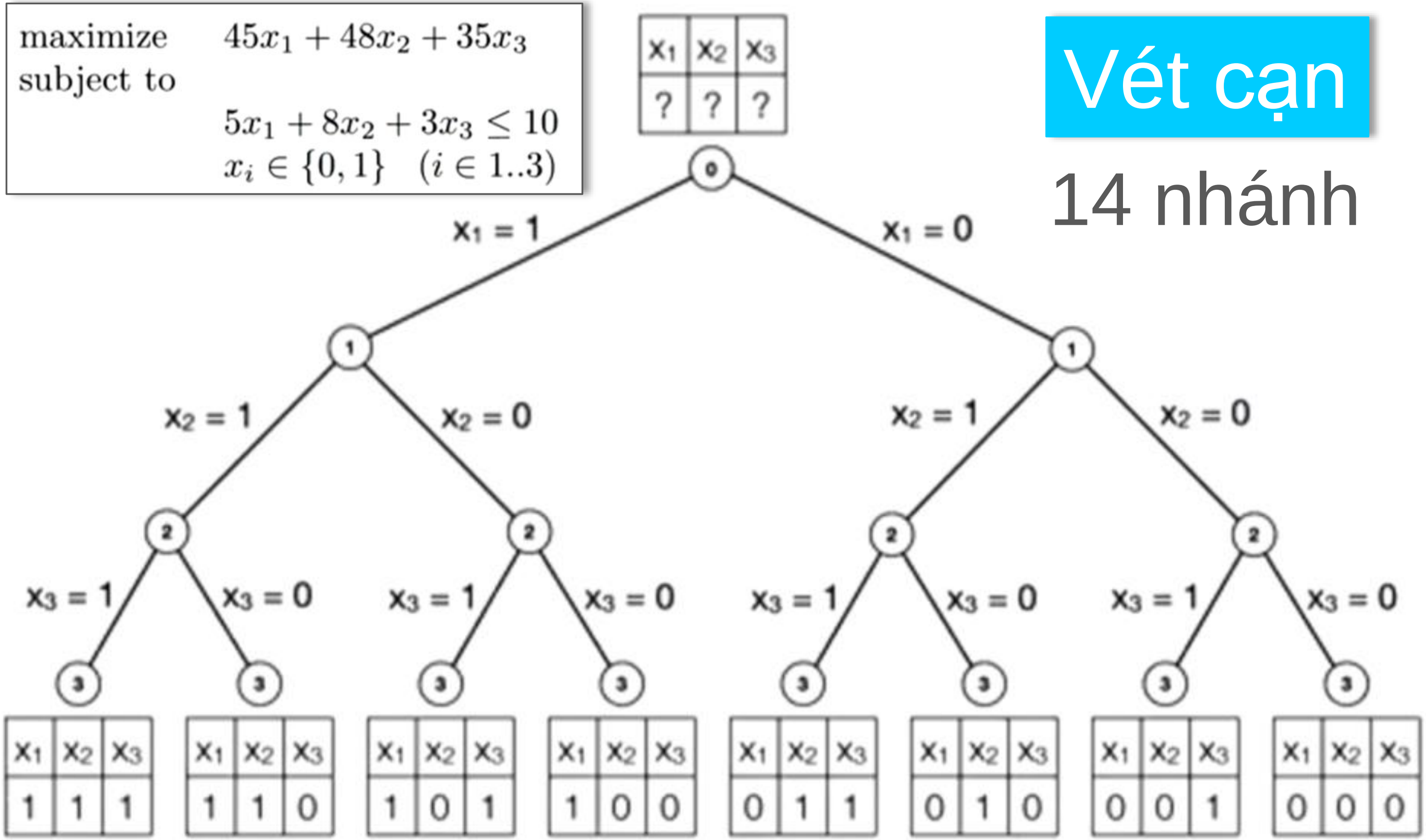
Nội dung Bài 5

1. Giới thiệu một số bài toán tối ưu
2. Thuật toán duyệt toàn bộ -
   brute force, exhaustive search
3. Thuật toán nhánh cận -
   Branch and Bound algorithm – B & B
4. Bài tập

Thuật toán nhánh cận - B & B

- B & B – Branch and Bound?

$$\begin{aligned} \text{maximize} \quad & 45x_1 + 48x_2 + 35x_3 \\ \text{subject to} \quad & \\ & 5x_1 + 8x_2 + 3x_3 \leq 10 \\ & x_i \in \{0,1\} \quad (i \in 1..3) \end{aligned}$$



Vét cạn

14 nhánh

Thuật toán nhánh cận - B & B

f(X)

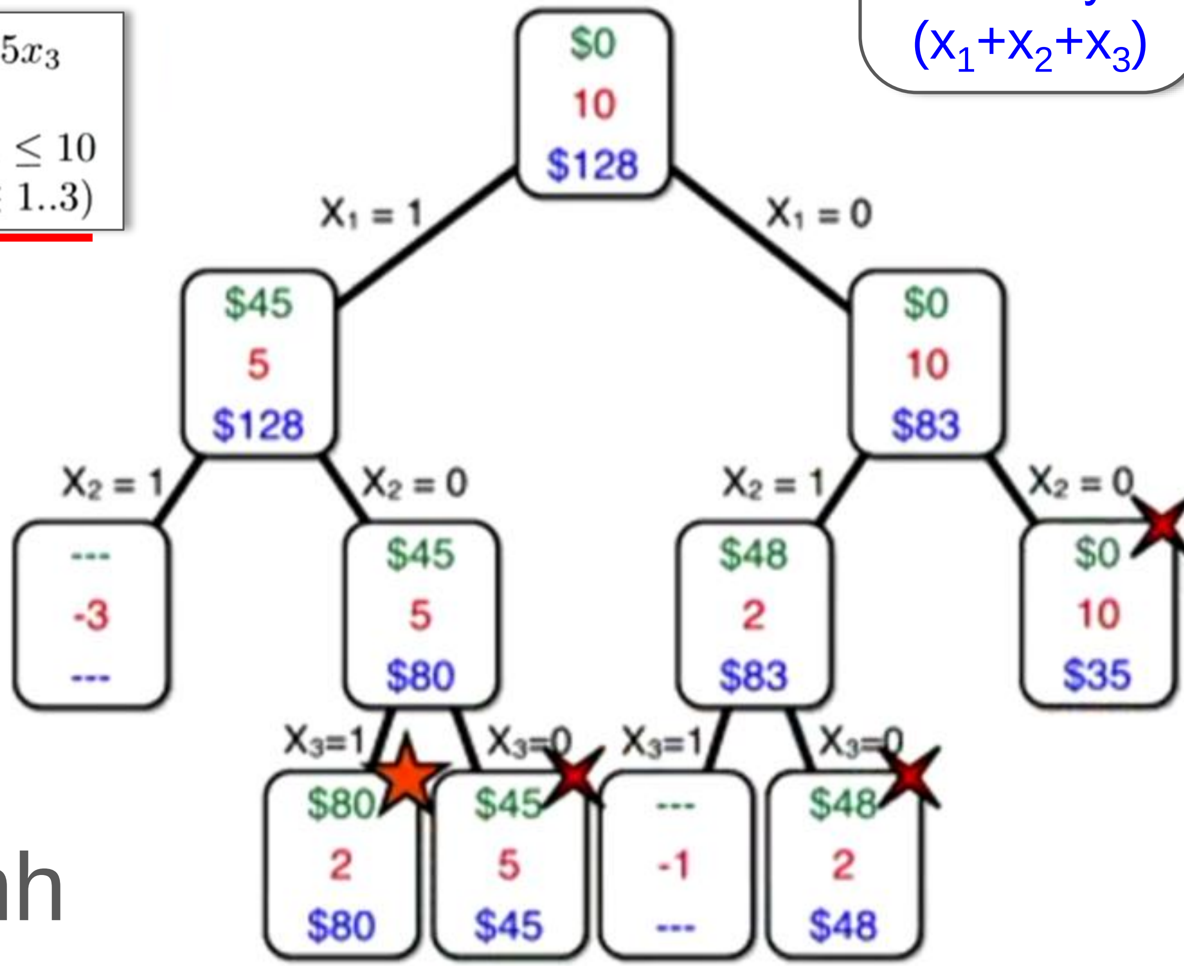
Cân nặng còn lại

Tối đa có thể lấy $(x_1+x_2+x_3)$

- B & B – Branch and Bound?

$$
\begin{array}{ll}
\text{maximize} & 45x_1 + 48x_2 + 35x_3 \\
\text{subject to} & \\
 & 5x_1 + 8x_2 + 3x_3 \leq 10 \\
 & x_i \in \{0,1\} \quad (i \in 1..3)
\end{array}
$$

i	c	a
1	45	5
2	48	8
3	35	3



n = 3

B = 10

12 nhánh

Thuật toán nhánh cận - B & B

f(X)

Cân nặng còn lại

Tối đa có thể lấy (chia nhỏ đồ vật)

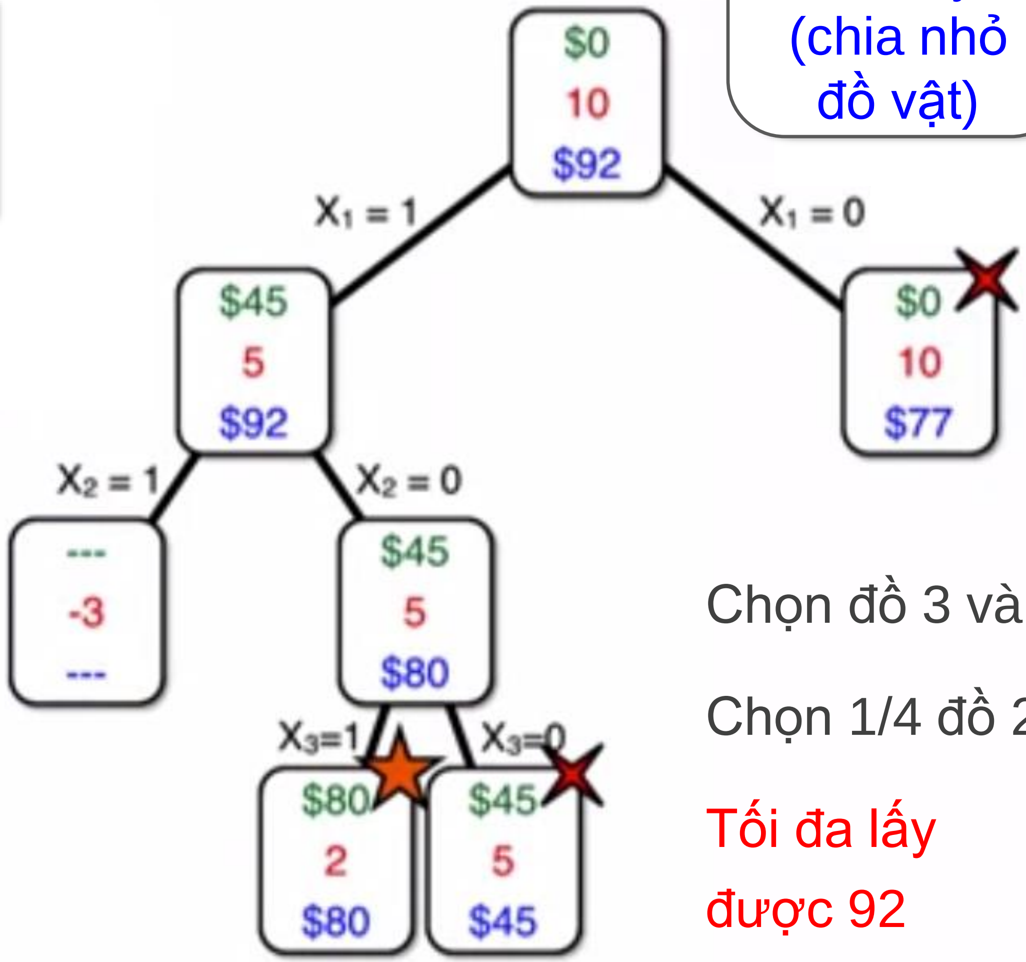
- B & B – Branch and Bound?

$$\begin{array}{ll} \text{maximize} & 45x_1 + 48x_2 + 35x_3 \\ \text{subject to} & \\ & 5x_1 + 8x_2 + 3x_3 \leq 10 \\ & 0 \leq x_i \leq 1 \quad (i \in 1..3) \end{array}$$

i	c	a
1	45	5
2	48	8
3	35	3

n = 3
B = 10

8 nhánh



Chọn đồ 3 và 1

Chọn 1/4 đồ 2

Tối đa lấy được 92

Thuật toán nhánh cận - B & B

- B & B – Branch and Bound?

$$\begin{array}{ll} \text{maximize} & 45x_1 + 48x_2 + 35x_3 \\ \text{subject to} & \\ & 5x_1 + 8x_2 + 3x_3 \le 10 \\ & 0 \le x_i \le 1 \quad (i \in 1..3) \end{array}$$

i	c	a
1	45	5
2	48	8
3	35	3

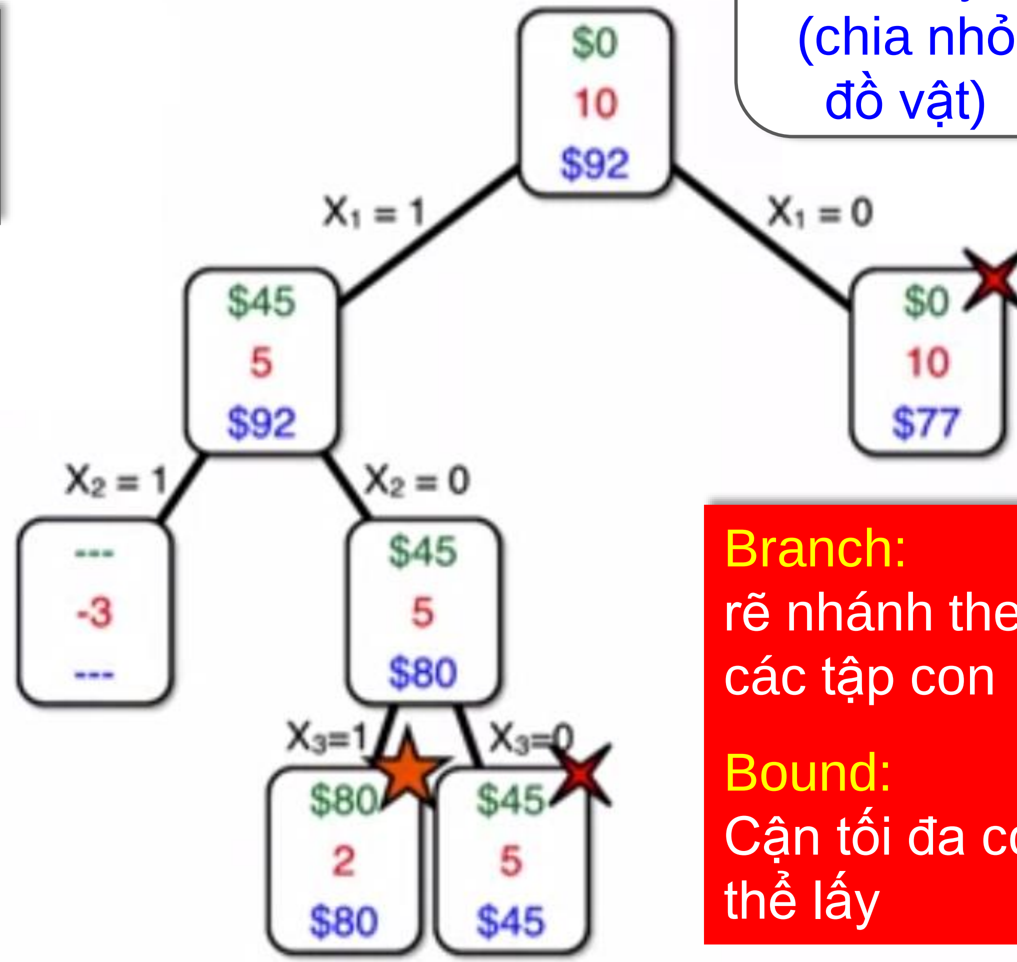
n = 3
B = 10

8 nhánh

f(X)

Cân nặng còn lại

Tối đa có thể lấy (chia nhỏ đồ vật)



Branch:
rẽ nhánh theo các tập con

Bound:
Cận tối đa có thể lấy

Thuật toán nhánh cận - B & B

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

- Tìm: $\max\{f(X) : X \in D\}$
- Trong đó, D là tập hữu hạn các phần tử.
  Tập D có thể được mô tả như sau:
  $D = \{X = (x_1, x_2, \ldots, x_n) \in A_1 \times A_2 \times \cdots \times A_n \mid X \text{ thỏa mãn tính chất } P\}$

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

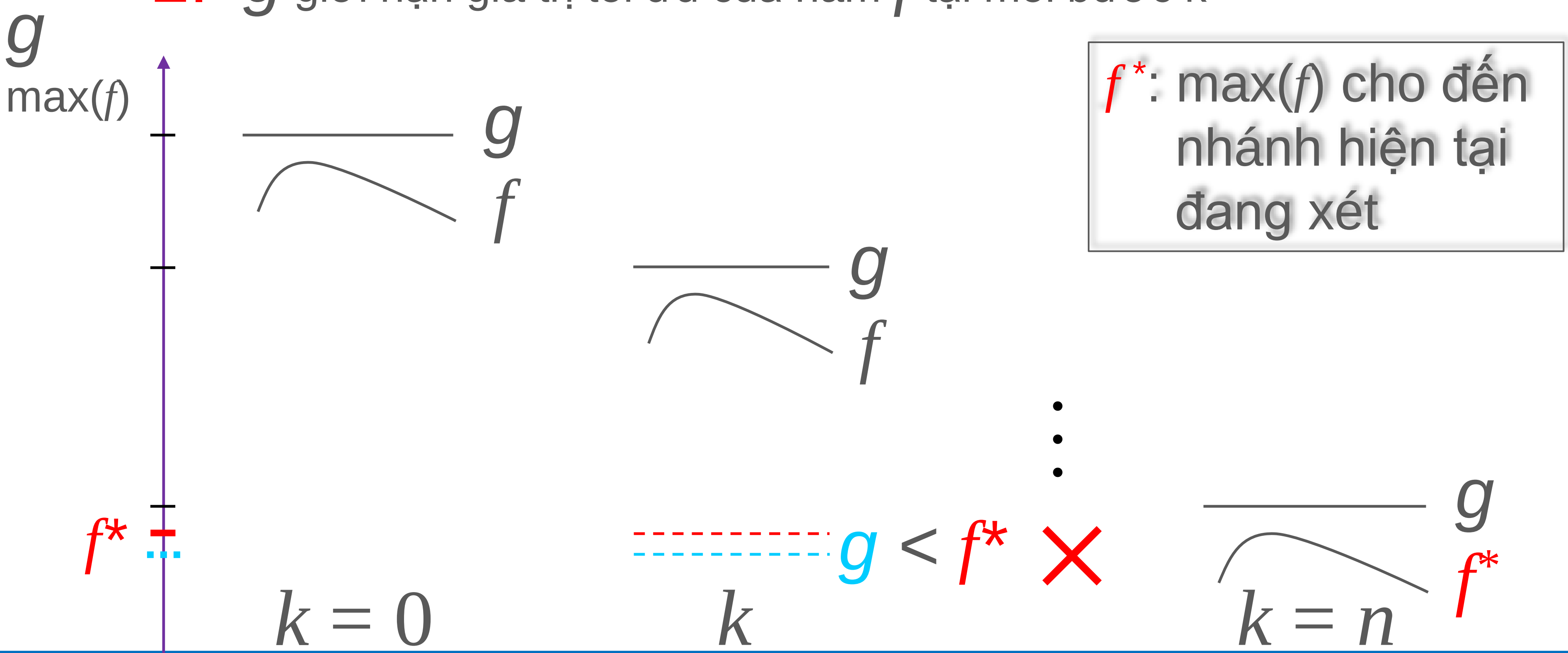
- Thuật toán nhánh cận có thể giải được bài toán đặt ra nếu ta tìm được một hàm g xác định trên tất cả phương án bộ phận của bài toán và:
  $g(a_1, a_2, \ldots, a_k) \geq \max\{f(X): X \in D, x_i = a_i, i = 1, 2, \ldots, k\}$

Thuật toán nhánh cận - B & B

- Hay, giá trị của hàm g tại phương án bộ phận cấp k $(a_1, a_2, \ldots, a_k)$ không nhỏ hơn giá trị lớn 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 trên của hàm mục tiêu trên tập $D(a_1, a_2, \ldots, a_k)$.
  Hàm g được gọi là hàm cận trên,
  $g(a_1, a_2, \ldots, a_k)$ gọi là cận trên của tập $D(a_1, a_2, \ldots, a_k)$.
- Bài toán tìm cực tiểu là bài toán tìm cực đại của hàm $-f(x)$

Thuật toán nhánh cận - B & B

- Một vài tính chất quan trọng của hàm *g*:
  1. *g* tạo thành một dãy không tăng theo k.
  2. *g* giới hạn giá trị tối ưu của hàm *f* tại mỗi bước k



B & B - 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 quay lui ta xác định được:
    X^* - phương án làm cho hàm mục tiêu có giá trị lớn nhất trong số các phương án tìm được: $f^* = f(X^*)$.
    X^* được gọi là phương án tốt nhất hiện có; f^*: kỷ lục hiện tại.
  - Nếu: $f^* > g(a_1, a_2, \ldots, a_k)$ thì $f^* > g(a_1, a_2, \ldots, a_k) \geq \max\{f(X): X \in D, x_i = a_i, i = 1, 2, \ldots, k\}$
    Nghĩa là tập $D(a_1, a_2, \ldots, a_k)$ chắc chắn không chứa phương án tối ưu
    $\rightarrow$ không cần triển khai phương án bộ phận $(a_1, a_2, \ldots, a_k)$
    $\rightarrow$ tập $D(a_1, a_2, \ldots, a_k)$ cũng loại bỏ khỏi quá trình duyệt
    $\rightarrow$ số các tập cần duyệt nhỏ đi trong quá trình tìm kiếm.

B & B – giả mã tìm max

```
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 - FOPT>;
                        else
                                if ( g(a_1, a_2, . . . , a_k) > f*)
                                        Branch_and_Bound (k+1);
                }
        }
}
```

B & B – 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:

    max $\{f(X) : X \in D, x_i = a_i, i = 1, 2, \ldots, k\}$
  - Giá trị của $g(a_1, a_2, \ldots, a_k)$ phải sát với giá trị

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

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

B & B – Bài toán cái túi dạng tổng quát (1/4)

- Dạng tổng quát 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 c_j (j = 1, 2, … , n).
  Cần đưa các đồ vật này vào một cái túi sao cho:
  - trọng lượng các đồ là b và
  - tổng giá trị sử dụng của các đồ vật trong túi là lớn nhất hay hàm giá trị sử dụng $\rightarrow$ max.
  Mỗi loại đồ vật có thể lấy nhiều lần.

B & B – Bài toán cái túi dạng tổng quát (2/4)

- Bài toán cái túi có thể được phát biểu tổng quát dưới dạng sau:
  Tìm giá trị max của hàm mục tiêu $f(X)$ với $X \in D$ và $f(X)$ 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:

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

B & B – Bài toán cái túi dạng tổng quát (3/4)

- 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):
  Lặp 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

B & B – Bài toán cái túi dạng tổng quát (4/4)

Branch_and_Bound (k) { *// c_k/a_k đã được sắp xếp giảm dần*

for (j = b_k/a_k ; 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)

<Ghi nhận kỷ lục nếu δ_k > FOTP >;

else

if ($\delta_k + (c_{k+1} * b_k)/a_{k+1}$ > FOPT)

Branch_and_Bound (k+1);

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

}

}

$$\delta_k = \sum_{i=1}^{k} c_i x_i \qquad g(x_1, x_2, \ldots, x_k) = \delta_k + b_k \frac{c_{k+1}}{a_{k+1}} \qquad b_k = b - \sum_{i=1}^{k} a_i x_i$$

FOPT: Kỷ lục

B & B – Ví dụ 1 (1/2)

- Giải bài toán cái túi tổng quát đã nêu sử dụng thuật toán nhánh cận

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

- Lời giải:

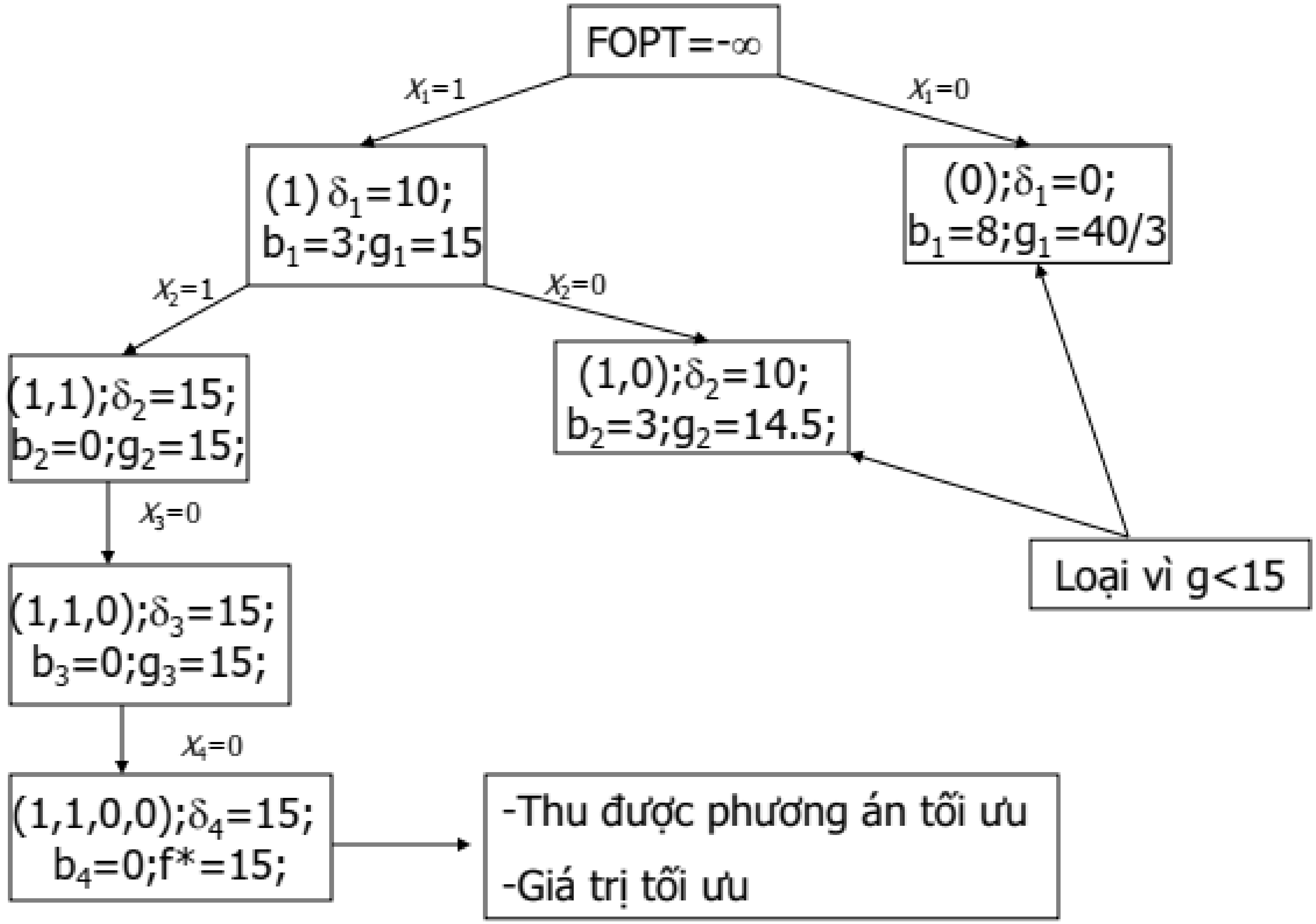
  Thông tin về một phương án bộ phận trên cây được ghi theo ô vuông theo thứ tự sau:

(1) δ_1=10; b_1=3; g_1=15

  - Đầu tiên là các thành phần của phương án bộ phận thứ k: $X = (x_1, x_2, .., x_k)$
  - Tiếp đến δ là giá trị của các đồ vật theo phương án bộ phận thứ k
  - Kế tiếp là b_k tương ứng với trọng lượng còn lại của túi theo phương án bộ phận thứ k
  - Cuối cùng là cận trên g_k theo phương án bộ phận thứ k

B & B – Ví dụ 1 (2/2)

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



$$\delta_k = \sum_{i=1}^{k} c_i x_i \qquad g(x_1, x_2, \ldots, x_k) = \delta_k + b_k \frac{c_{k+1}}{a_{k+1}} \qquad b_k = b - \sum_{i=1}^{k} a_i x_i$$

FOPT: Kỷ lục

Bài tập 1 B & B

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

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

Bài tập 2 B & B

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

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

B & B – Bài toán người đi du lịch (1/4)

- Dạng tổng quát:
  Tìm giá trị nhỏ nhất của hàm mục tiêu $f(X)$ với $X \in D$ và $f(X)$ được xác định như sau:

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

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

- Một ví dụ
  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

B & B – Bài toán người đi du lịch (2/4)

- Gọi:

  c_{min} = min{$c[i, j]$, i , j = 1, 2, . . . , 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:

    $T1 \rightarrow Tu_2 \rightarrow . . . \rightarrow Tu_k$ (với $T1 = 1$).

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

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

B & B – Bài toán người đi du lịch (2/4)

- 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 T1.

  → Cần phải qua $n - k + 1$ đoạn đường nữa.

  Vì mỗi đoạn đường đều có chi phí không nhỏ hơn c_{min},

  → Cận dưới của phương án bộ phận có thể được xác định: $g(u_1, u_2, \ldots, u_k) = \delta + (n - k + 1) \times c_{min}$.

B & B – Bài toán người đi du lịch (3/4)

```
Branch_and_Bound (k) {
        for (j = 2; j <= n; j++){
                if (unused[j]){                              // Chưa xét
                        x[k] = j; unused[j] = 0;             // Đã xét
                        δ = δ + c[ x[k − 1], x[k]];
                        if ( k == n) Ghi_Nhan_Ky_Luc();
                        else if (δ + (n − k + 1)×c_min < FOPT)
                                    Branch_and_Bound (k+1);
                        unused[j] = 1;                       // Chưa xét
                        δ = δ − c[ x[k − 1], x[k]];
                }
        }
}
```

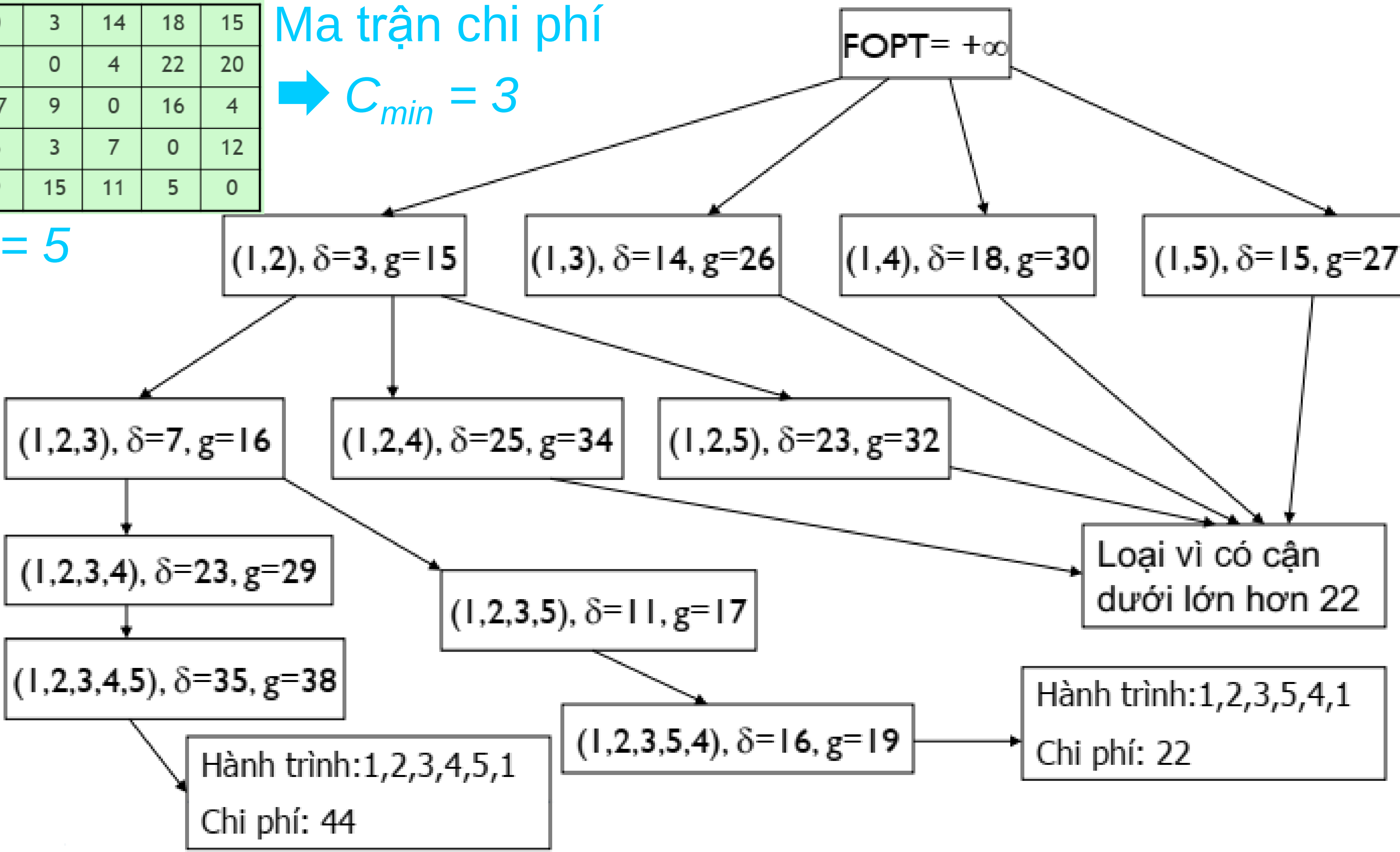
B & B – Bài toán người đi du lịch (4/4)

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

Ma trận chi phí

➡ $C_{min} = 3$

$n = 5$



FOPT: Kỷ lục; $\delta = \delta + c[x[k-1], x[k]]$; $g = \delta + (n - k + 1) \times c_{min}$

Kết thúc Bài 5

- Câu hỏi và thảo luận