ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Thuật Toán Tham Lam

THUẬT TOÁN ỨNG DỤNG

Các mô hình giải bài căn bản

Các phương pháp căn bản xây dựng lời giải bài toán:

- Duyệt toàn bộ
- Chia để trị
- Qui hoạch động
- Tham lam

1. Sơ đồ thuật toán tham lam
   - Giới thiệu chung
   - Sơ đồ chung
   - Chứng minh tính tối ưu
2. Bài toán đổi tiền
3. Bài toán cái túi
4. Tập đoạn thẳng không giao nhau

Thuật toán tham lam

- Các thuật toán tham lam là một phương pháp tiếp cận căn bản để xây dựng thuật toán giải hầu hết các dạng bài toán khác nhau
- Tại mỗi bước, quyết định được đưa ra dựa ngay vào thông tin đang có, và trong tương lai sẽ không xem xét lại tác động của các quyết định đã nhận trong quá khứ
- Do đó các thuật toán tham lam rất dễ đề xuất, và thông thường không đòi hỏi nhiều thời gian tính. Tuy nhiên, các thuật toán dạng này thường không cho kết quả tối ưu !
- Ứng dụng đưa ra lời giải chấp nhận được cho các bài toán tối ưu trong thực tế có độ khó cao

Một số bài toán điển hình của thuật toán tham lam

- Bài toán đổi tiền
  - Đổi giá trị tiền x bởi các đồng mệnh trị 1, 5, 10, 25
  - Đối với các mệnh giá khác thì sao? Ví dụ 1, 5, 7
- Bài toán cây khung nhỏ nhất
  - Thuật toán Prim:
    - ★ Xuất phát từ cây là một đỉnh bất kỳ, mỗi bước bổ sung vào cây hiện tại đỉnh gần cây nhất
  - Thuật toán Kruskal:
    - ★ Sắp xếp các cạnh theo trọng số giảm dần
    - ★ Mỗi bước bổ sung vào tập đang xây dựng một cạnh theo thứ tự sắp xếp mà không tạo ra chu trình trong tập
- Bài toán đường đi ngắn nhất: Thuật toán Dijkstra:
  - Mỗi bước cố định đỉnh có nhãn nhỏ nhất
- Mã Huffman

Sơ đồ chung: Mô tả bài toán

- Lời giải cần tìm có thể mô tả như là bộ gồm hữu hạn các thành phần thoả mãn các điều kiện bài toán
- Thông thường giải quyết bài toán tối ưu: Tìm lời giải với giá trị hàm mục tiêu lớn nhất (hoặc nhỏ nhất)
- Xác định tập các ứng cử viên có thể lựa chọn làm các thành phần của lời giải

Sơ đồ chung

- Xuất phát từ lời giải rỗng, thuật toán xây dựng lời giải của bài toán theo từng bước, ở mỗi bước sẽ chọn một phần tử từ tập ứng cử viên và bổ sung vào lời giải hiện có
- Hàm `Solution`(S) nhận biết tính chấp nhận được của lời giải S
- Hàm `Select`(C) chọn từ tập C ứng cử viên có triển vọng nhất để bổ sung vào lời giải hiện có
- Hàm `Feasible`($S \cup x$) kiểm tra tính chấp nhận được của lời giải bộ phận $S \cup x$

Sơ đồ chung: Thuật toán

GREEDY-ALGORITHM()

```
// C : Tap_Cac_Ung_Vien
S = ∅; // S : Loi_Giai_Xay_Dung_Theo_Thuat_Toan
while (C ≠ ∅) && (Solution(S)==false) {
      x ← Select(C);
      C = C \ x;
      if (Feasible(S ∪ x)==true)
          S = S ∪ x;
}
if (Solution(S)==true)
    return S;
```

Chứng minh tính tối ưu

- Để chỉ ra thuật toán không đúng đắn chỉ cần đưa ra một phản ví dụ: Một bộ dữ liệu mà đối với nó thuật toán không cho lời giải đúng
- Chứng minh tính đúng đắn của thuật toán khó hơn nhiều

Lập luận biến đổi (Exchange Argument)

Giả sử cần chứng minh thuật toán A cho lời giải tối ưu:

- Gọi $A(I)$ là lời giải tìm được bởi thuật toán A đối với bộ dữ liệu I, còn O là lời giải tối ưu của bài toán đối với bộ dữ liệu này
- Cần tìm cách xây dựng phép biến đổi ϕ cho phép biến đổi O thành lời giải O' sao cho
  - O' cũng tốt không kém gì O (nghĩa là O' vẫn là tối ưu);
  - O' **giống** với $A(I)$ nhiều hơn O

Lập luận biến đổi: Chứng minh bằng phản chứng

Giả sử A không đúng đắn, khi đó phải tìm được bộ dữ liệu I sao cho $A(I)$ khác với lời giải tối ưu của bài toán:

- Gọi O là lời giải tối ưu giống với $A(I)$ nhất;
- Do $A(I)$ không là lời giải tối ưu nên $A(I) \neq O$;
- Khi đó sử dụng phép biến đổi ϕ ta có thể biến đổi O thành O' sao cho: O' vẫn là tối ưu đồng thời O' giống với $A(I)$ hơn là O
  $\Rightarrow$ Mâu thuẫn với giả thiết O là lời giải tối ưu giống với $A(I)$ nhất !

Lập luận biến đổi: Chứng minh trực tiếp

Giả sử có O là lời giải tối ưu, ta biến đổi O thành lời giải tối ưu O' giống với $A(I)$ hơn là O:

- Nếu $O' = A(I)$ thì $A(I)$ chính là lời giải tối ưu;
- Ngược lại, ta lại lặp lại phép biến đổi đối với O' để thu được lời giải tối ưu O'' giống với $A(I)$ hơn là O',
- ...
- Ta thu được dãy O', O'', O''', ...;
- Mỗi phần tử của dãy này đều là phương án tối ưu, và mỗi phần tử sau lại giống với $A(I)$ hơn phần tử trước nó
  $\Rightarrow$ Dãy này phải kết thúc ở $A(I)$. Vậy $A(I)$ cũng là tối ưu !

Đổi tiền

Bài toán

Có các đồng tiền mệnh giá: 1, 5, 10, 25, 100 (xu), hãy tìm phương pháp đổi một lượng tiền x sử dụng một số lượng ít nhất các đồng tiền

Hình: Đổi 34¢

Đổi tiền: Thuật toán

Thuật toán Người Thu Ngân

Ở mỗi bước, trả đồng tiền mệnh giá lớn nhất không vượt quá lượng tiền cần đổi

Hình: Đổi 2.89$

Đổi tiền: Thuật toán

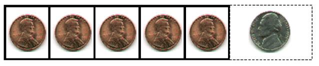
CASHIERS-ALGORITHM($x, C_1, C_2, \ldots, C_n$)

```
SORT(n đồng tiền theo thứ tự: C1 < C2 < ... < Cn)
S ← ∅      // S là tập các đồng tiền được chọn
while (x > 0) {
    k ← đồng tiền mệnh giá lớn nhất ck sao cho ck ≤ x
    if (không tìm được k)
        return "no solution"
    else
        x ← x − ck
        S ← S ∪ {k}
    return S
}
```

Thuật toán Người Thu Ngân có cho cách đổi tiền tối ưu không?

Đổi tiền: Các tính chất của lời giải tối ưu

Bổ đề 1: Số lượng đồng 1¢ $\leq$ 4.



CM: Thay năm đồng 1¢ bởi một đồng 5¢

Bổ đề 2: Số lượng đồng 5¢ $\leq$ 1

CM: Thay hai đồng 5¢ bởi một đồng 10¢

Bổ đề 3: Số lượng đồng 25¢ $\leq$ 3.

CM: Thay ba đồng 25¢ bởi một đồng 1$

Đổi tiền: Các tính chất của lời giải tối ưu

Bổ đề 4: **Số lượng đồng 5¢ + Số lượng đồng 10¢** ≤ 2

CM:

- Thay 3 đồng 10¢ và 0 đồng năm¢ bởi 1 đồng 25¢ và 1 đồng năm¢;
- Thay 2 đồng 10¢ và 1 đồng 5¢ bởi 1 đồng 25¢;
- Nhắc lại Bổ đề 2: có tối đa 1:đồng 5¢

k	c_k	Tất cả lời giải tối ưu phải thoả mãn	Giá trị tối đa với các mệnh giá $1, 2, \ldots, k-1$ trong bất kỳ OPT
1	1	$P \leq 4$	-
2	5	$N \leq 1$	4
3	10	$N + D \leq 2$	4+5=9
4	25	$Q \leq 3$	20+4=24
5	100	không giới hạn	75+24=99

Đổi tiền: Chứng minh tính tối ưu

Định lí

Thuật toán Người Thu Ngân (NTN) là tối ưu đối với dãy mệnh giá: 1,5,10,25,100

CM: (qui nạp theo x)

- Xét cách đổi tối ưu $c_k \leq x < c_k + 1$: thuật toán NTN sẽ chọn đồng tiền k
- Ta khẳng định bất kì lời giải tối ưu nào cũng phải chọn đồng tiền k:
  - Nếu không, cần sử dụng các đồng tiền $c_1, \ldots, c_{k-1}$ để đổi x
  - Bảng trên đã cho thấy không tồn tại lời giải nào đổi được như vậy !
- Bài toán đưa về đổi $x - c_k$ ¢. Theo qui nạp, thuật toán NTN cho lời giải tối ưu !

k	c_k		
1	1	$P \leq 4$	-
2	5	$N \leq 1$	4
3	10	$N + D \leq 2$	4+5=9
4	25	$Q \leq 3$	20+4=24
5	100	không giới hạn	75+24=99

Đổi tiền: Mở rộng

Câu hỏi: Thuật toán NTN có là tối ưu cho dãy mệnh giá khác?

Trả lời:

- KHÔNG. Thuật toán NTN không cho lời giải tối ưu với các mệnh giá tem của Bưu chính Hoa Kỳ: 1,10,21,34,70,100,350,1225,1500. Phản ví dụ: 140¢
  - Thuật toán NTN: 140=100+34+1+1+1+1+1+1.
  - Lời giải tối ưu: 140=70+70.
- KHÔNG. Thậm chí thuật toán NTN có thể không tìm được lời giải chấp nhận được nếu $c_1 > 1$: 7, 8, 9. Phản ví dụ: 15¢
  - Thuật toán NTN: 15=9+???.
  - Lời giải tối ưu: 15=7+8.

Bài toán cái túi

Phát biểu bài toán

- Có n đồ vật. Đồ vật i có trọng lượng W_i và giá trị C_i, $i = 1, \ldots, n$.
- Yêu cầu: Tìm cách chất các đồ vật này vào cái túi có dung lượng là b sao cho tổng trọng lượng của các đồ vật được chất vào túi là không quá b, đồng thời tổng giá trị của chúng là lớn nhất
- Ký hiệu $S = \{1, 2, \ldots, n\}$ tập chỉ số các đồ vật. Bài toán đặt ra là: Tìm $I \subset S$ sao cho

$$\sum_{i \in I} W_i \leq b, \sum_{i \in I} C_i \rightarrow \max$$

Bài toán cái túi: Tham lam 1

Greedy1: Sắp xếp các đồ vật theo thứ tự không tăng của giá trị

- Lần lượt xét các đồ vật theo thứ tự đã sắp, và xếp đồ vật đang xét vào túi nếu như dung lượng còn lại của cái túi đủ chứa nó (tức là tổng trọng lượng của các đồ vật đã xếp vào túi và trọng lượng của đồ vật đang xét là không vượt quá b)

Bài toán cái túi: Tham lam 1

Phản ví dụ với bộ dữ liệu sau

- Số lượng đồ vật $n = 3$
- Trọng lượng cái túi $b = 19$

Đồ vật	1	2	3
Giá trị C_i	20	16	8
Trọng lượng W_i	14	6	10

- Greedy1 cho lời giải: $I_1 = \{1\}$ với giá trị 20
- Trong khi đó, ta có lời giải tốt hơn là $I^* = \{2, 3\}$ với giá trị 24

Bài toán cái túi: Tham lam 2

Greedy2: Sắp xếp các đồ vật theo thứ tự không giảm của trọng lượng

- Lần lượt xét các đồ vật theo thứ tự đã sắp, và chất đồ vật đang xét vào túi nếu như dung lượng còn lại của cái túi đủ chứa nó (tức là tổng trọng lượng của các đồ vật đã xếp vào túi và trọng lượng của đồ vật đang xét là không vượt quá b)

Bài toán cái túi: Tham lam 2

Phản ví dụ với bộ dữ liệu sau

- Số lượng đồ vật $n = 3$
- Trọng lượng cái túi $b = 11$

Đồ vật	1	2	3
Giá trị C_i	10	16	28
Trọng lượng W_i	5	6	10

- Greedy2 cho lời giải: $I_2 = \{1, 2\}$ với giá trị 26
- Trong khi đó, ta có lời giải tốt hơn là $I^* = \{3\}$ với giá trị 28

Bài toán cái túi: Tham lam 3

Greedy3: Sắp xếp các đồ vật theo thứ tự không tăng của giá trị một đơn vị trọng lượng (c_i/w_i)

Nghĩa là

$$\frac{C_{i_1}}{W_{i_1}} \leq \frac{C_{i_2}}{W_{i_2}} \leq \dots \frac{C_{i_n}}{W_{i_n}}$$

- Lần lượt xét các đồ vật theo thứ tự đã sắp, và chất đồ vật đang xét vào túi nếu như dung lượng còn lại của cái túi đủ chứa nó (tức là tổng trọng lượng của các đồ vật đã xếp vào túi và trọng lượng của đồ vật đang xét là không vượt quá b)

Bài toán cái túi: Tham lam 3

Phản ví dụ với bộ dữ liệu sau

- Số lượng đồ vật $n = 3$
- Trọng lượng cái túi $b \geq 2$

Đồ vật	1	2
Giá trị C_i	1	$10b - 1$
Trọng lượng W_i	5	b

- Rõ ràng

$$\frac{C_1}{W_1} = \frac{10}{1} \geq \frac{10b-1}{b} = \frac{C_2}{W_2}$$

- Greedy3 cho lời giải: $I_3 = \{1\}$ với giá trị 10
- Trong khi đó, ta có lời giải tốt hơn là $I^* = \{2\}$ với giá trị $10b - 1$

Bài toán cái túi: Tham lam 4

Greedy4:

Gọi I_j là lời giải thu được theo thuật toán Greedy$j$, $j = 1, 2, 3$. Gọi I_4 là lời giải đạt

$$\max\{\sum_{i \in I_1} C_i, \sum_{i \in I_2} C_i, \sum_{i \in I_3} C_i\}$$

- Định lí: Lời giải I_4 thoả mãn bất đẳng thức

$$\sum_{i \in I_4} C_i \geq \frac{1}{2} OPT,$$

với OPT là đáp số tối ưu của bài toán

Bài toán cái túi: Tham lam 4

Phản ví dụ với bộ dữ liệu sau

- Số lượng đồ vật $n = 4$
- Trọng lượng cái túi $b = 11$

Đồ vật	1	2	3	4
Giá trị C_i	9	10	18	27
Trọng lượng W_i	4	5	6	10
C_i/W_i	2.25	2	3	2.7
Greedy$i$	27	19	27	27

- Greedy4 cho lời giải với giá trị 27
- Trong khi đó, ta có lời giải tốt hơn là $I^* = \{2, 3\}$ với giá trị 28

Bài toán thực hành

Money Changing

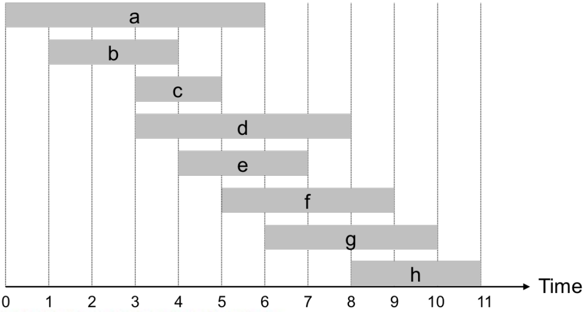
ATM Withdrawal

1. Sơ đồ thuật toán tham lam
2. Bài toán đổi tiền
3. Bài toán cái túi
4. Tập đoạn thẳng không giao nhau
   - Bài toán
   - Các ý tưởng tham lam
   - Chứng minh tính tối ưu

Tập đoạn thẳng không giao nhau

Phát biểu bài toán

- Có n công việc, công việc j bắt đầu tại S_j và kết thúc tại F_j
- Hai công việc là phù hợp với nhau nếu chúng không chồng lên nhau
- Yêu cầu: Tìm tập con nhiều nhất các công việc đôi một phù hợp với nhau



Tập đoạn thẳng không giao nhau: Các ý tưởng tham lam

Xét các công việc theo một thứ tự ưu tiên nào đó. Tại mỗi bước chọn lần lượt công việc theo thứ tự ưu tiên mà phù hợp với tất cả các công việc đã chọn

1. [Bắt đầu sớm xét trước] Xét các công việc theo thứ tự ưu tiên tăng dần của thời gian bắt đầu s_j
2. [Kết thúc sớm xét trước] Xét các công việc theo thứ tự ưu tiên tăng dần của thời gian kết thúc f_j
3. [Ngắn hơn xét trước] Xét các công việc theo thứ tự ưu tiên tăng dần của tổng thời gian thực hiện công việc $f_j - s_j$
4. [Nhiều mâu thuẫn hơn xét trước] Với mỗi công việc j, tính c_j là số lượng công việc không tương thích với j. Xét các công việc theo thứ tự ưu tiên giảm dần của c_j

Tập đoạn thẳng không giao nhau: Các ý tưởng tham lam

Các ý tưởng tham lam

Xét các công việc theo một thứ tự ưu tiên nào đó. Tại mỗi bước chọn lần lượt công việc theo thứ tự ưu tiên mà phù hợp với tất cả các công việc đã chọn

[Bắt đầu sớm xét trước]

[Ngắn hơn xét trước]

[Nhiều mâu thuẫn hơn xét trước]

Tập đoạn thẳng không giao nhau: Kết-Thúc-Sớm-Xét-Trước *Demo*

EARLIEST-FINISH-TIME-FIRST($n, S_1, \ldots, S_n, F_1, \ldots, F_n$)

```
SORT(các công việc theo thời gian kết thúc: F_1 ≤ ... ≤ F_n)
A ← ∅     // tập các công việc đã được chọn
for j=1 to n
    if (công việc j phù hợp với tập A)
        A ← A ∪ {j}
return A
```

Độ phức tạp: $\mathcal{O}(n \log n)$

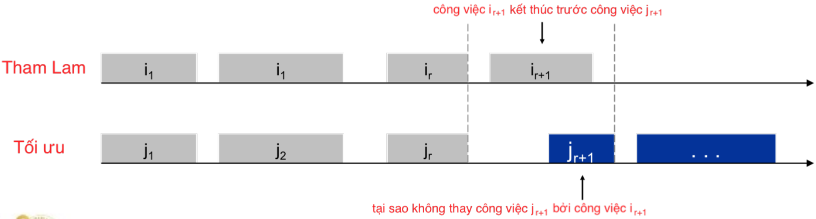
Tập đoạn thẳng không giao nhau: Chứng minh tính tối ưu thuật toán Kết-Thúc-Sớm-Xét-Trước

Định lí

Thuật toán Kết-Thúc-Sớm-Xét-Trước cho kết quả tối ưu!

CM: [bằng phản chứng] Giả sử thuật toán tham lam không cho kết quả tối ưu:

- Gọi $\{i_1, i_2, \ldots, i_k\}$ là tập các công việc được chọn bởi thuật toán tham lam;
- Gọi $\{j_1, j_2, \ldots, j_m\}$ là tập các công việc được chọn của lời giải tối ưu với $i_1 = j_1, i_2 = j_2, \ldots, i_r = j_r$ với giá trị r lớn nhất có thể



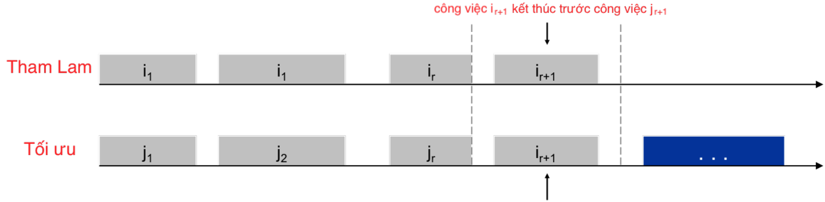
Tập đoạn thẳng không giao nhau: Chứng minh tính tối ưu thuật toán Kết-Thúc-Sớm-Xét-Trước

Định lí

Thuật toán Kết-Thúc-Sớm-Xét-Trước cho kết quả tối ưu!

CM: [bằng phản chứng] Giả sử thuật toán tham lam không cho kết quả tối ưu:

- Gọi $\{i_1, i_2, \ldots, i_k\}$ là tập các công việc được chọn bởi thuật toán tham lam;
- Gọi $\{j_1, j_2, \ldots, j_m\}$ là tập các công việc được chọn của lời giải tối ưu với $i_1 = j_1, i_2 = j_2, \ldots, i_r = j_r$ với giá trị r lớn nhất có thể



Bài toán thực hành

Planting Trees

Postman

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for your attentions!

solct.hust.edu.vn/ fb.com/groups/solct