

Chương 3. HEURISTIC



TRÍ TUỆ NHÂN TẠO *Artificial Intelligence*

Đoàn Vũ Thịnh
Khoa Công nghệ Thông tin
Đại học Nha Trang
Email: thinhdv@ntu.edu.vn

Nha Trang, 06-2023

Chương 3. HEURISTIC

- Heuristic là gì?
 - Heuristic là những tri thức được rút tủa từ những kinh nghiệm, “trực giác” của con người (mẹo).
 - Heuristic có thể là những tri thức “đúng” hay “sai”.
 - Heuristic là những meta knowledge và “thường đúng”.
- Heuristic dùng để làm gì?
 - Khi bài toán có thể không có nghiệm chính xác do các mệnh đề không phát biểu chặt chẽ hay thiếu dữ liệu để khẳng định KQ.
 - Có nghiệm chính xác nhưng chi phí tính toán để tìm ra nghiệm là quá lớn (hệ quả của bùng nổ tổ hợp).
- Heuristic giúp tìm kiếm đạt kết quả với chi phí thấp hơn.

Chương 3. HEURISTIC

- Heuristic Greedy (Giải thuật tham lam)
 - Dạng tổng quát $\min\{f(x): x \in D\}$, D là tập các điểm rời rạc $\in \mathbb{R}^n$
 - Ví dụ, máy ATM có $m=4$ loại tiền (100, 50, 20 và 10). Nếu đưa số tiền cần rút là n . Tìm phương án sao cho số tờ tiền là ít nhất.
 - Gọi $x = (x_1, x_2, x_3, x_4)$ là các phương án rút tiền, với $x_1 = 100$, $x_2 = 50$, $x_3 = 20$ và $x_4 = 10$.
 - $\min\{f(x_1 + x_2 + x_3 + x_4)\}$
 - Điều kiện: $100 * x_1 + 50 * x_2 + 20 * x_3 + 10 * x_4 = n$
 - Thông thường người ta sẽ vét cạn $\min(f(x): x \in D)$

Chương 3. HEURISTIC

■ Heuristic Greedy (Giải thuật tham lam)

- Phương pháp tham lam là đưa ra các quyết định dựa vào ngay thông tin đang có và trong tương lai sẽ không xem xét lại các tác động của các quyết định trong quá khứ.
- Hàm Solution (S): nhận biết tính chấp nhận được của lời giải $S(S = \emptyset)$
- Hàm Select (C): chọn từ tập C ứng viên triển vọng nhất để bổ sung vào lời giải hiện có.
- Hàm Feasible (S+x): Kiểm tra tính chấp nhận được của lời giải bộ phận S+x.
- Ý tưởng: Xuất phát từ lời giải \emptyset , xây dựng lời giải của bài toán theo từng bước, mỗi bước sẽ chọn 1 p.tử để bổ sung vào lời giải hiện có

Chương 3. HEURISTIC

- Heuristic Greedy (Giải thuật tham lam)
- Procedure Greedy;
 - Begin
 - $S = \emptyset$;
 - While($C \neq \emptyset$ && not(Solution(S)):{
 - $x \leftarrow \text{Select}(C)$;
 - $C = C \setminus x$;
 - if(Feasible($S \cup C$)): $S = S \cup x$;
 - }
 - If(Solution(C)): return S;
 - End;

Chương 3. HEURISTIC

▪ Heuristic Greedy (Giải thuật tham lam)

▪ Ví dụ bài toán cái túi: V_{\max} và $w \leq W_{\max}$

▪ Tham lam 1

▪ Sắp xếp đồ vật theo giá trị giảm dần

▪ Chọn đồ từ đầu – cuối

▪ Tham lam 2

▪ Sắp xếp đồ vật theo thứ tự tăng dần

▪ Chọn đồ từ đầu – cuối

▪ Tham lam 3

▪ Sắp xếp đồ vật theo tỷ trọng v/w

▪ $\frac{V_1}{W_1} > \frac{V_2}{W_2} > \frac{V_3}{W_3} > \dots$

| Đồ vật | 1 | 2 | 3 |
|----------------------------------|----|----|----|
| Gía trị | 20 | 16 | 8 |
| T. lượng | 14 | 6 | 10 |
| Wmax = 19 | | | |
| Greedy 1: $i=1$, Value = 20 | | | |
| Tối ưu: $i=\{2,3\}$, Value = 24 | | | |

| Đồ vật | 1 | 2 | 3 |
|------------------------------------|----|----|----|
| Gía trị | 10 | 16 | 28 |
| T. lượng | 5 | 6 | 10 |
| Wmax = 19 | | | |
| Greedy 2: $i=\{1,2\}$, Value = 24 | | | |
| Tối ưu: $i=\{3\}$, Value = 28 | | | |

| Đồ vật | 1 | 2 | 3 |
|-------------------------------------|----|-----|-----|
| Gía trị | 60 | 100 | 120 |
| T. lượng | 10 | 20 | 30 |
| Wmax = 50 | | | |
| Greedy 3: $i=\{1,2\}$, Value = 160 | | | |
| Tối ưu: $i=\{2,3\}$, Value = 220 | | | |

Chương 3. HEURISTIC

▪ Heuristic Greedy (Giải thuật tham lam)

- Cho các tờ tiền có mệnh giá lần lượt là: 500, 200, 100, 100, 50, 50, 20, 20, 20, 20, 10.
- Cho số tiền $m = 390 \Rightarrow$ *Cách đổi với số tờ ít nhất.*

- B1. Sắp xếp tờ tiền theo mệnh giá giảm dần: 500, 200, 200...10
- B2. Duyệt từ đầu danh sách \Rightarrow cuối danh sách ở B1
 - $\text{if}(m \geq A[i]): m = m - A[i];$ //số tiền giảm dần
- B3. Kiểm tra $m < 0$: Không đổi được
- B4. Muốn in ra các tờ tiền đã đổi: dùng mảng $VET[]$ để lưu trữ
 - Mỗi lần m đổi được: gán $VET[i]=1$
 - Duyệt từ đầu \Rightarrow cuối danh sách $VET[]$:
 - $\text{if}(VET[i]==1): \text{in } A[i];$

Chương 3. HEURISTIC

▪ Heuristic Greedy (Giải thuật tham lam)

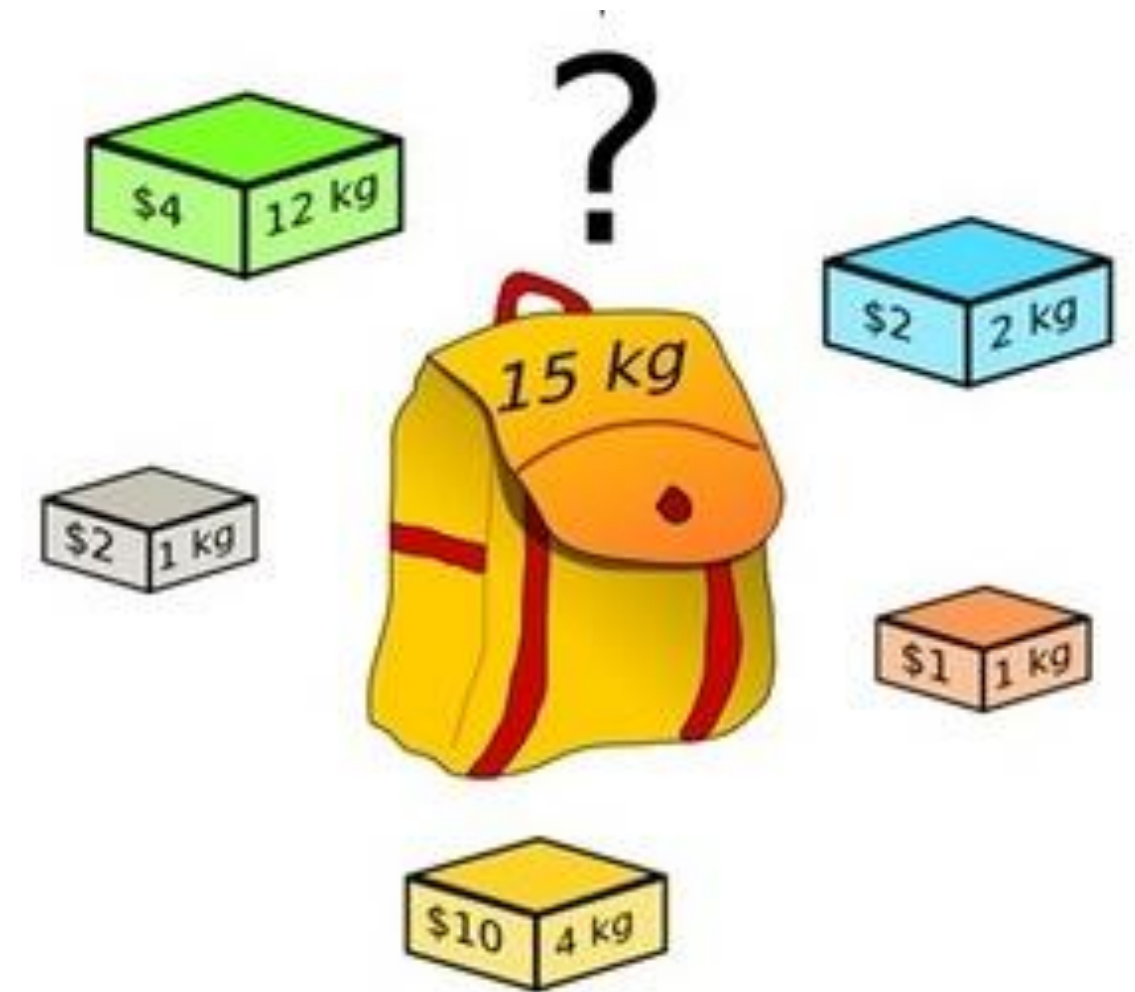
| Test | Input | Output |
|------|--|-------------------------|
| 1 | 390 10 200,100,100,50,50,50,50,20,20,10 | 5 200,100,50,20,20 |
| 2 | 100 11 50,20,20,20,20,20,2,2,2,2,2 | 8 50,20,20,2,2,2,2,2 |
| 3 | 100 6 50,20,20,20,20,20 | -1 |

- Test 1: Nghiệm tối ưu
- Test2: Nghiệm không tối ưu (đổi thành 5 tờ 20 thì sẽ tối ưu hơn)
- Test3: Có nghiệm nhưng trả lời vô nghiệm

Chương 3. HEURISTIC

■ Heuristic Greedy (Giải thuật tham lam)

- Một chiếc ba lô có thể tích là P_{\max} . Có n vật, vật thứ i có thể tích là $P[i]$ và giá trị là $W[i]$.
- Hãy tìm cách sắp các vật vào ba lô sao cho tổng giá trị là lớn nhất. Với trọng lượng tối đa cho trước là P_{\max} .

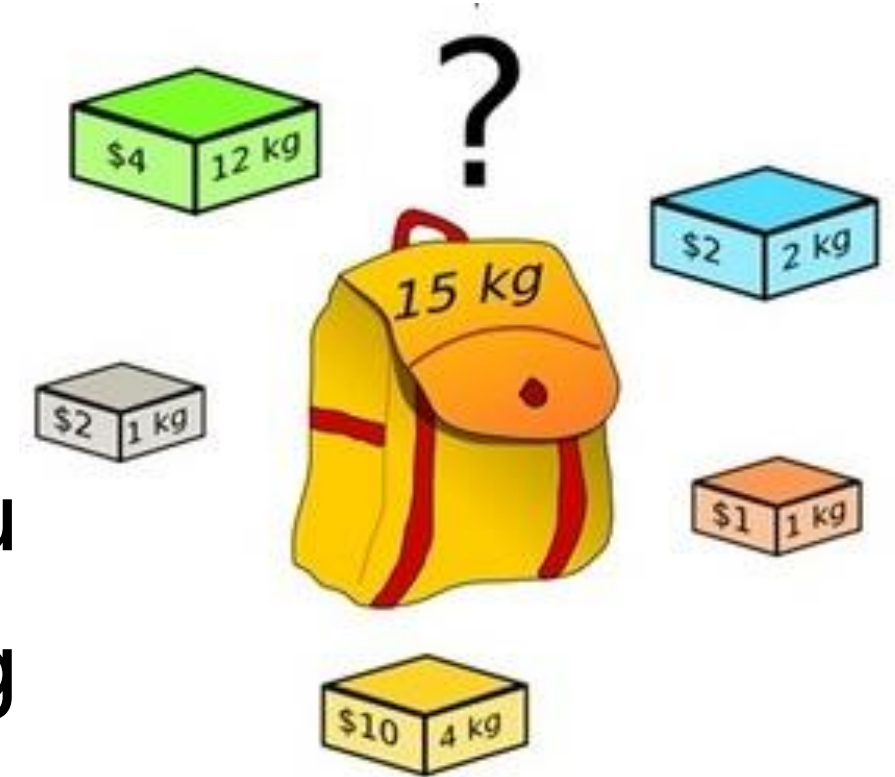


- Tham lam: Vật nào có kích thước bé nhưng giá trị lớn hơn cho vào túi trước.
- Trọng số ưu tiên: $\frac{W_i}{P_i}$ được sắp theo thứ tự giảm dần.

Chương 3. HEURISTIC

■ Heuristic Greedy (Giải thuật tham lam)

- Tổ chức dữ liệu:
- Mảng 2 chiều $P[]$, $W[]$: thể tích và giá trị
- Mảng 1 chiều $cs[]$: thứ tự giảm dần theo trọng số ưu tiên
- Mảng cs chỉ chứa các chỉ số của phần tử có trọng số ưu tiên giảm dần, thứ tự các vật theo đề bài không đổi trong suốt quá trình xử lý.



- B1. Sắp xếp các vật theo thứ tự giảm dần của trọng số ưu tiên
- B2. Xếp các vật vào ba lô.

Chương 3. HEURISTIC

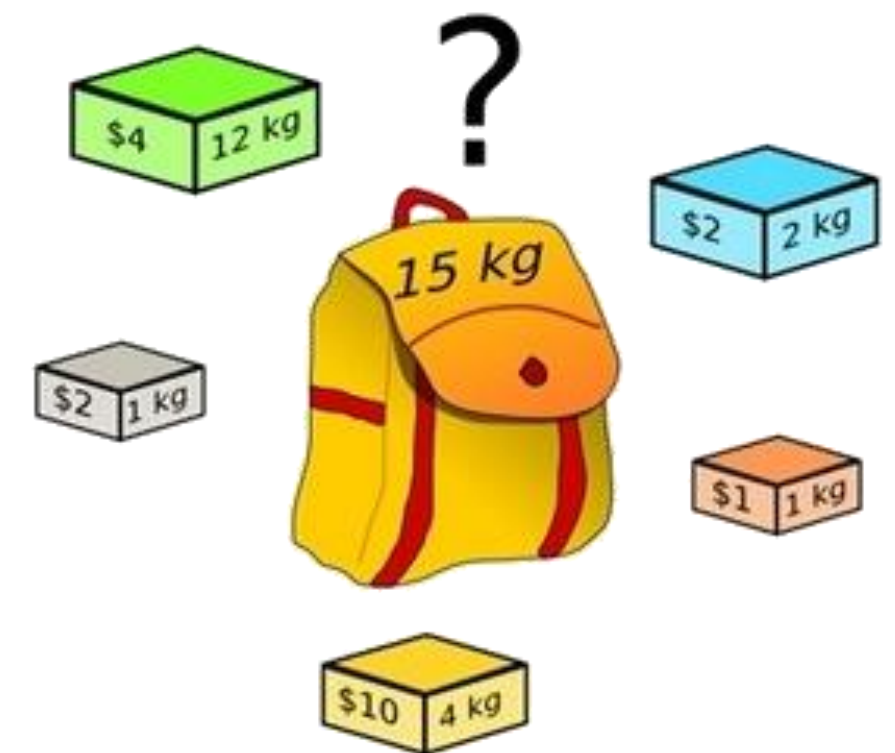
▪ Heuristic Greedy (Giải thuật tham lam)

- B1. Sắp xếp các vật theo thứ tự giảm dần của trọng số ưu tiên

```
for( $i = 1; i < n; i++$ ){  
    for( $j = i + 1; j \leq n; j++$ ){  
        if( $W[cs[i]]/P[cs[i]] < W[cs[j]]/P[cs[j]]$ ):  
            tam = cs[j]; cs[j] = cs[i]; cs[i] = tam;
```

- B2. Xếp các vật vào ba lô.

- $S = 0; i = 1;$
- while($(S < Pmax) \&\& (i \leq n)$):
 - if($S + P[cs[i]] \leq Pmax$):
 - $S += P[cs[i]];$
 - $printf(\$5d, cs[i]);$
 - $i++;$



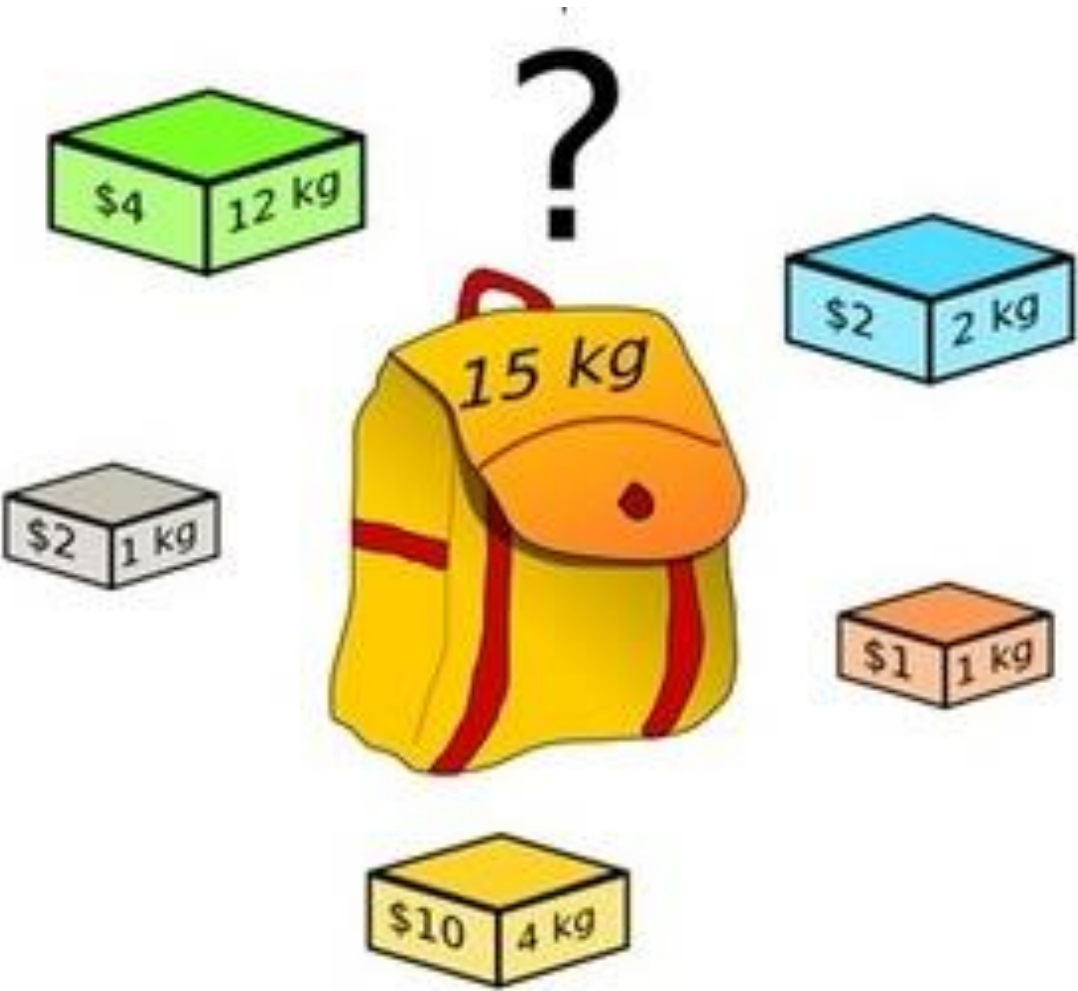
Chương 3. HEURISTIC

▪ Heuristic Greedy (Giải thuật tham lam)

| Đồ vật | 1 | 2 | 3 | 4 | 5 |
|--|------|-----|-----|-----|------|
| Gía trị | 10\$ | 2\$ | 2\$ | 1\$ | 4\$ |
| T. lượng | 4kg | 1kg | 2kg | 1kg | 12kg |
| Wmax = 15kg | | | | | |
| Greedy 3: $i=\{1,2,3,4\}$, Value = 15\$, Weight = 8kg | | | | | |

| Đồ vật | 1 | 5 | 3 | 2 | 4 |
|--|------|------|-----|-----|-----|
| Gía trị | 10\$ | 4\$ | 2\$ | 2\$ | 1\$ |
| T. lượng | 4kg | 12kg | 2kg | 1kg | 1kg |
| Wmax = 15kg | | | | | |
| Greedy 2: $i=\{1,2,3,4\}$, Value = 15\$, Weight = 8kg | | | | | |

| Đồ vật | 4 | 2 | 3 | 5 | 1 |
|--|-----|-----|-----|------|------|
| Gía trị | 1\$ | 2\$ | 2\$ | 4\$ | 10\$ |
| T. lượng | 1kg | 1kg | 2kg | 12kg | 4kg |
| Wmax = 15kg | | | | | |
| Greedy 1: $i=\{1,2,3,4\}$, Value = 15\$, Weight = 8kg | | | | | |



Chương 3. HEURISTIC

■ Heuristic Greedy (Giải thuật tham lam)

- Cho n thành phố $(1,2,...n)$ và khoảng cách giữa chúng $(c_{i,j})$. Hãy tìm hành trình của một người đưa thư, đi qua tất cả các thành phố rồi quay về thành phố xuất phát, sao cho tổng chiều dài đường đi là ngắn nhất.

| | A | B | C | D | E | F | G | H |
|---|------|------|------|------|------|-----|-----|------|
| A | 0 | 730 | 640 | 840 | 800 | 430 | 380 | 1010 |
| B | 730 | 0 | 710 | 1040 | 500 | 300 | 540 | 470 |
| C | 640 | 710 | 0 | 1420 | 1050 | 600 | 920 | 1160 |
| D | 840 | 1040 | 1420 | 0 | 740 | 950 | 570 | 900 |
| E | 800 | 500 | 1050 | 740 | 0 | 520 | 460 | 200 |
| F | 430 | 300 | 600 | 950 | 520 | 0 | 390 | 690 |
| G | 380 | 540 | 920 | 570 | 460 | 390 | 0 | 660 |
| H | 1010 | 470 | 1160 | 900 | 200 | 690 | 660 | 0 |

- Vết cặn: $(n-1)!$. Với n lớn?
- Greedy: Mỗi bước chọn $i \rightarrow j$ sao cho j gần i nhất trong những thành phố chưa đến nối với i .

Chương 3. HEURISTIC

■ Heuristic Greedy (Giải thuật tham lam)

- Cho n thành phố $(1,2,...n)$ và khoảng cách giữa chúng $(c_{i,j})$. Hãy tìm hành trình của một người đưa thư, đi qua tất cả các thành phố rồi quay về thành phố xuất phát, sao cho tổng chiều dài đường đi là ngắn nhất.

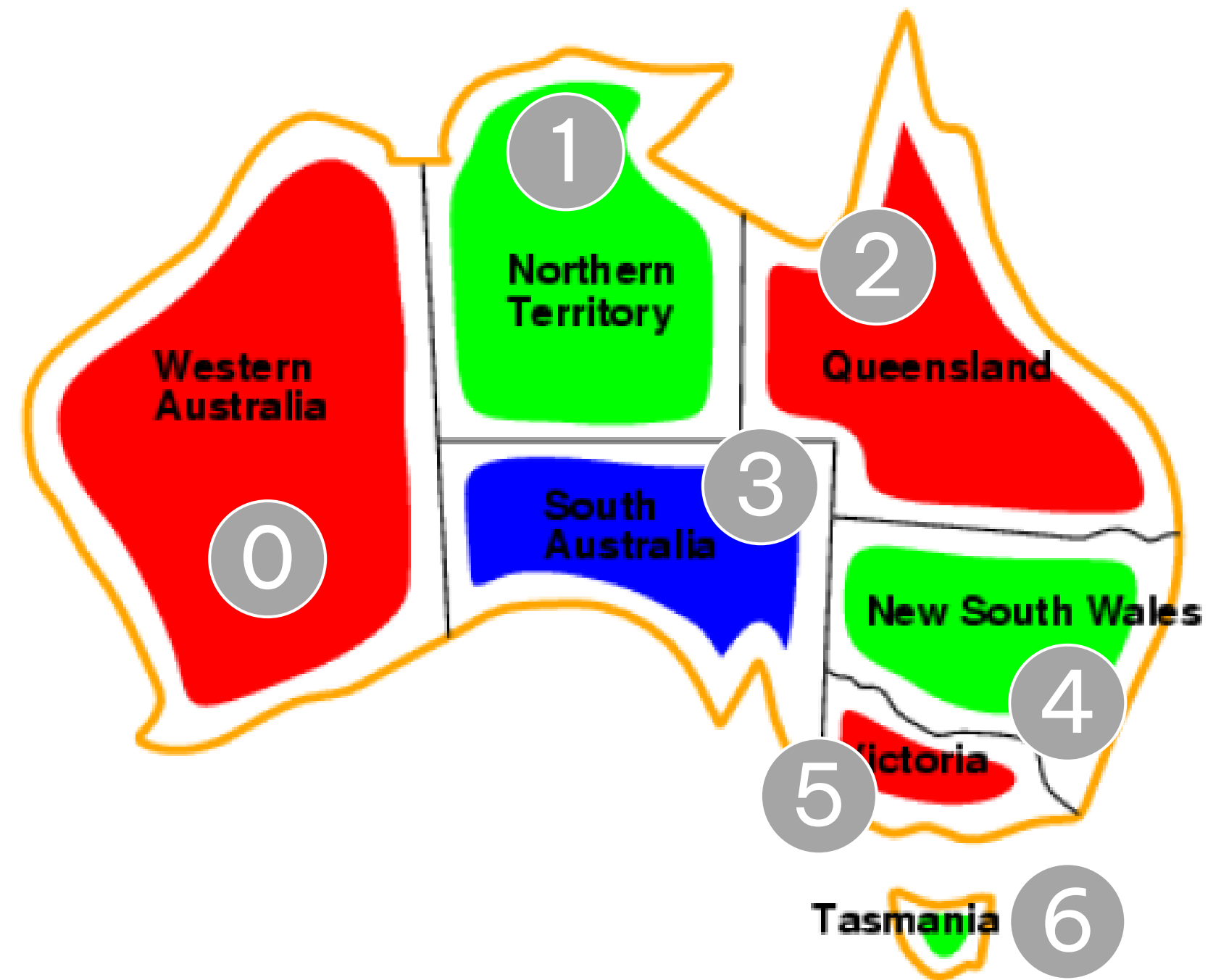
| | A | B | C | D | E | F | G | H |
|---|------|------|------|------|------|-----|-----|------|
| A | 0 | 730 | 640 | 840 | 800 | 430 | 380 | 1010 |
| B | 730 | 0 | 710 | 1040 | 500 | 300 | 540 | 470 |
| C | 640 | 710 | 0 | 1420 | 1050 | 600 | 920 | 1160 |
| D | 840 | 1040 | 1420 | 0 | 740 | 950 | 570 | 900 |
| E | 800 | 500 | 1050 | 740 | 0 | 520 | 460 | 200 |
| F | 430 | 300 | 600 | 950 | 520 | 0 | 390 | 690 |
| G | 380 | 540 | 920 | 570 | 460 | 390 | 0 | 660 |
| H | 1010 | 470 | 1160 | 900 | 200 | 690 | 660 | 0 |

- Greedy 1: $A \rightarrow G \rightarrow F \rightarrow B \rightarrow H \rightarrow E \rightarrow D \rightarrow C \rightarrow A$

Chương 3. HEURISTIC

■ Heuristic Greedy (Giải thuật tham lam)

- Sử dụng ba màu để tô bản đồ các tỉnh của một nước sao cho các tỉnh kề nhau thì có màu khác nhau.
- Ví dụ, nước Australia có 7 bang như hình vẽ, chỉ sử dụng ba màu: đỏ, xanh lơ và xanh da trời để tô màu 7 bang của nước Australia sao cho không có hai bang nào kề nhau lại có màu giống nhau.
- Tập ràng buộc: $WA \neq NT$, $WA \neq SA$, $NT \neq SA$, $NT \neq Q$, $SA \neq Q$, $SA \neq NSW$, $SA \neq V$, $Q \neq NSW$, $NSW \neq V$



Chương 3. HEURISTIC

- **Heuristic Greedy (Giải thuật tham lam)**
 - **Bước 1:** Lập danh sách $V' := [v_1, v_2, \dots, v_n]$ là các đỉnh của đồ thị được sắp xếp theo thứ tự bậc giảm dần: $d(v_1) > d(v_2) > \dots > d(v_n)$. Ban đầu tất cả các đỉnh trong V (V') đều chưa được tô màu. Gán $i := 1$;
 - **Bước 2:** Tô màu i cho đỉnh đầu tiên trong danh sách V' . Duyệt lần lượt các đỉnh khác trong V' (nếu có) và chỉ tô màu i cho các đỉnh không kề đỉnh đã có màu i .
 - **Bước 3:** Kiểm tra nếu tất cả các đỉnh trong V đã được tô màu thì thuật toán kết thúc, đồ thị đã sử dụng i màu để tô. Ngược lại, nếu vẫn còn đỉnh chưa được tô thì chuyển sang bước 4.
 - **Bước 4:** Loại khỏi danh sách V' các đỉnh đã tô màu. Sắp xếp lại các đỉnh trong V' theo thứ tự bậc giảm dần. Gán $i := i + 1$ và quay lại bước 2.

Chương 3. HEURISTIC

- Heuristic Greedy (Giải thuật tham lam)
 - Cho n công việc, công việc i hoàn thành trong thời gian t_i , các công việc được thực hiện trên M máy công suất như nhau, mỗi máy đều có thể thực hiện bất kỳ trong n công việc, mỗi công việc làm liên tục trên 1 máy cho đến khi hoàn thành. Hãy tổ chức máy thực hiện đủ n công việc sao cho thời gian thực hiện càng nhỏ càng tốt.
 - Xét 6 công việc và thời gian hoàn thành các công việc tương ứng:

| CV1 | CV2 | CV3 | CV4 | CV5 | CV6 |
|-----|-----|-----|-----|-----|-----|
| 2 | 5 | 8 | 1 | 5 | 1 |

Chương 3. HEURISTIC

- **Heuristic Greedy (Giải thuật tham lam)**
 - Sắp xếp các công việc giảm dần theo thời gian hoàn thành
 - Phân công M công việc đầu tiên cho M máy. Thời gian lớn nhất để hoàn thành của M công việc này: T_{max}
 - Loop1:
 - Loop2:
 - Chọn máy có thời gian đã làm $< T_{max}$, thêm công việc mới cho máy này (theo thứ tự công việc đã sắp xếp).
 - Lặp lại cho đến khi thêm công việc mới thì không có máy nào có tổng thời gian thực hiện $< T_{max}$
 - Tìm máy có thời gian thực hiện bé nhất, phân công công việc cho máy này và thay đổi giá trị của T_{max}
 - Lặp lại cho đến khi hết việc

Chương 3. HEURISTIC

▪ Heuristic Greedy (Giải thuật tham lam)

| CV1 | CV2 | CV3 | CV4 | CV5 | CV6 | Số máy |
|-----|-----|-----|-----|-----|-----|--------|
| 2 | 5 | 8 | 1 | 5 | 1 | M=3 |

▪ Sắp xếp công việc giảm dần theo thời gian hoàn thành

| CV3 | CV2 | CV5 | CV1 | CV4 | CV6 |
|-----|-----|-----|-----|-----|-----|
| 8 | 5 | 5 | 2 | 1 | 1 |

- Phân công lần 1:
 - $T_{max} = 8$
 - $Min_t = M2 (5)$
- M1: CV3 (8)
- M2: CV2 (5)
- M3: CV5 (5)
- Phân công lần 2:
 - $M2 = M2 + CV1 (2) + CV4 (1) = 8 \geq T_{max}$
 - $M3 = M3 + CV6 (1) = 6$
- Tổng thời gian hoàn thành: 8 đơn vị thời gian

Chương 3. HEURISTIC

▪ Heuristic Greedy (Giải thuật tham lam)

| CV1 | CV2 | CV3 | CV4 | CV5 | Số máy |
|-----|-----|-----|-----|-----|--------|
| 3 | 3 | 2 | 2 | 2 | M=2 |

▪ Sắp xếp công việc giảm dần theo thời gian hoàn thành

| CV1 | CV2 | CV3 | CV4 | CV5 |
|-----|-----|-----|-----|-----|
| 3 | 3 | 2 | 2 | 2 |

- Phân công lần 1:
 - M1: CV1 (3)
 - M2: CV2 (3)
- $T_{max} = 3$
 - $Min_t = M1 (3)$

▪ Phân công lần 2:

- $M1 = M1 + CV2 (2) = 5$

- $\geq T_{max} \Rightarrow T_{max} = 5$

- $M2 = M2 + CV4 (2) = 5$

- $T_{max} = 5$

- $Min_t = M1 (5)$

▪ Phân công lần 3:

- $M1 = M1 + CV3 (2) = 7$

- $\geq T_{max} \Rightarrow T_{max} = 7$

- $M2 = M2 + \emptyset$

▪ Tổng thời gian hoàn thành: 7 đơn vị thời gian

Chương 3. HEURISTIC

▪ Heuristic Greedy (Giải thuật tham lam)

| CV1 | CV2 | CV3 | CV4 | CV5 | CV6 | CV7 | Số máy |
|-----|-----|-----|-----|-----|-----|-----|--------|
| 1 | 8 | 1 | 5 | 4 | 3 | 4 | M=3 |

▪ Sắp xếp công việc giảm dần theo thời gian hoàn thành

| CV2 | CV4 | CV5 | CV7 | CV6 | CV1 | CV3 |
|-----|-----|-----|-----|-----|-----|-----|
| 8 | 5 | 4 | 4 | 3 | 1 | 1 |

▪ Phân công lần 1:

- M1: CV2(8)

- M2: CV4 (5)

- M3: CV5 (4)

- $T_{max} = 8$

- $Min_t = M3 (4)$

▪ Phân công lần 2:

- $M3 = M3 + CV7 (4) = 8$

- $\geq T_{max} \Rightarrow T_{max} = 8$

- $M2 = M2 + CV4 (4) = 8$

Chương 3. HEURISTIC

▪ Heuristic Greedy (Giải thuật tham lam)

| CV1 | CV2 | CV3 | CV4 | CV5 | CV6 | CV7 | Số máy |
|-----|-----|-----|-----|-----|-----|-----|--------|
| 1 | 8 | 1 | 5 | 4 | 3 | 4 | M=3 |

▪ Sắp xếp công việc giảm dần theo thời gian hoàn thành

| CV2 | CV4 | CV5 | CV7 | CV6 | CV1 | CV3 |
|-----|-----|-----|-----|-----|-----|-----|
| 8 | 5 | 4 | 4 | 3 | 1 | 1 |

▪ Phân công lần 2:

▪ $M3 = M3 + CV7 (4) = 8$

$\geq T_{max} \Rightarrow T_{max} = 8$

▪ $M2 = M2 + CV4 (4) = 8$

▪ $T_{max} = 8$

▪ $Min_t = M1 (8)$

▪ Phân công lần 3:

▪ $M1 = M1 + CV1 (1) =$

$9 \geq T_{max} \Rightarrow T_{max} = 9$

▪ $M2 = M2 + CV3 (1) = 9$

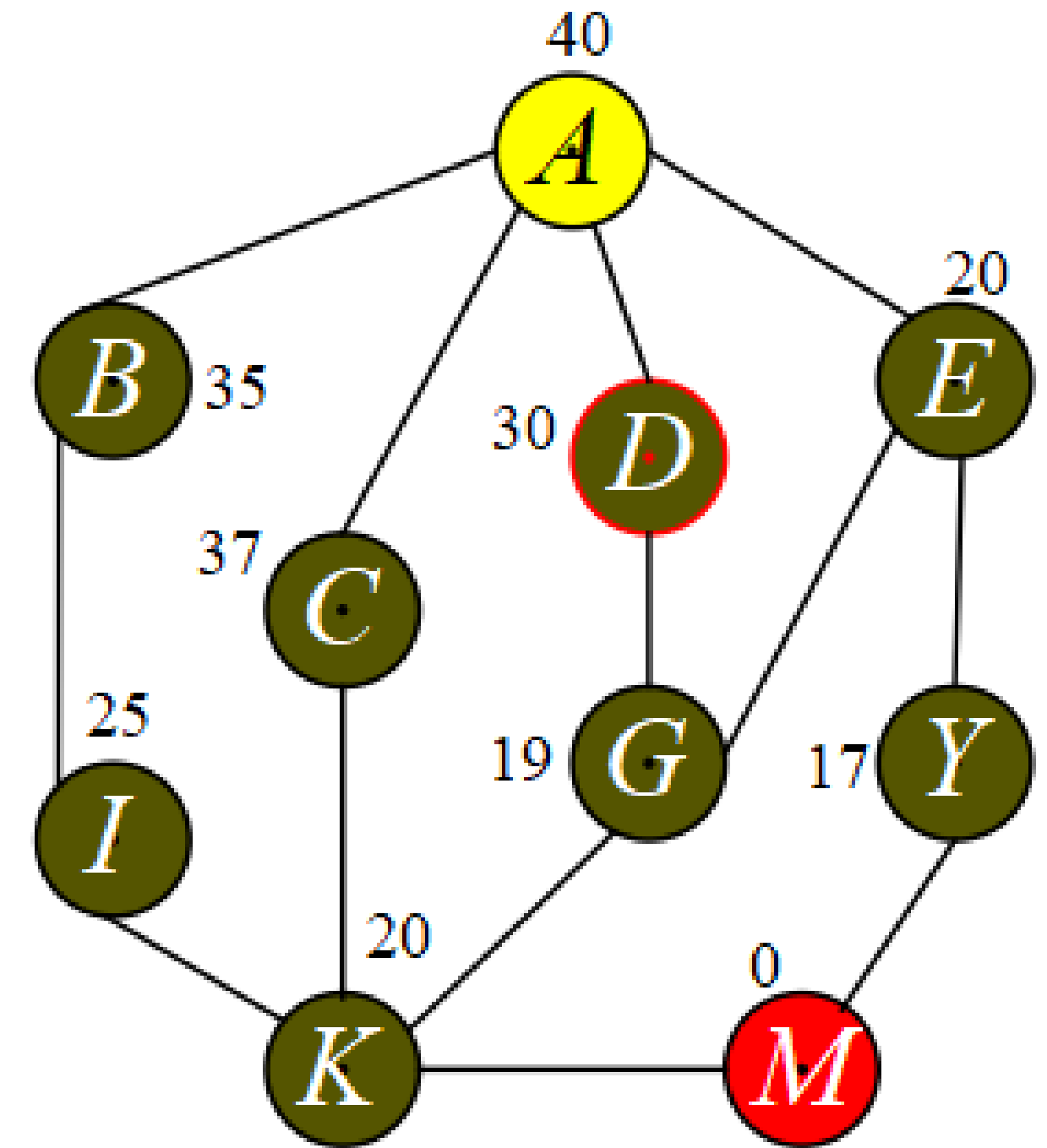
▪ Tổng thời gian hoàn thành: 9 đơn vị thời gian

Chương 3. HEURISTIC

- Tìm kiếm leo đồi (Hill Climbing Search)
 - Là giải thuật tối ưu của tìm kiếm cục bộ (Stefanie Gunther, 1983) nhằm tìm kiếm trạng thái tốt hơn có sử dụng hàm lượng giá h'.
 - Bước 1. Khởi tạo ngăn xếp OPEN = start
 - Bước 2. Loop:
 - if (OPEN=∅): thất bại
 - n = phần tử đầu danh sách OPEN
 - if (n≡goal): thành công (END)
 - for (m: mỗi đỉnh kề $\notin \{\mathbf{OPEN} \cup \mathbf{CLOSE}\}$) $\rightarrow \Gamma(n) // n$ chưa xét đến
 - Sắp xếp danh sách $\Gamma(n)$ theo thứ tự tăng dần của hàm đánh giá h'
 - Chèn $\Gamma(n)$ vào đầu danh sách OPEN

Chương 3. HEURISTIC

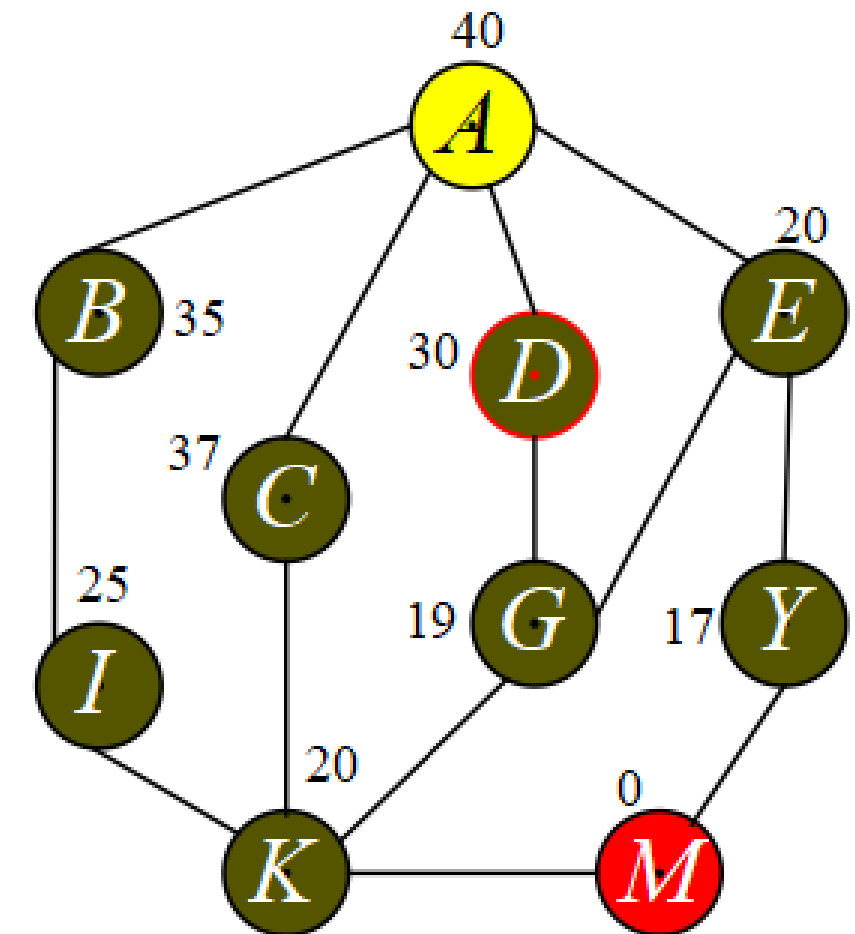
- Tìm kiếm leo đồi (Hill Climbing Search)
- Trình bày thuật toán leo đồi dốc đứng. Áp dụng thuật toán để tìm đường đi ngắn nhất từ đỉnh A đến M trên đồ thị, với các ước lượng heuristic của các trạng thái so với trạng thái đích được cho trong hình.



Chương 3. HEURISTIC

▪ Tìm kiếm leo đồi (Hill Climbing Search)

| Bước | n | $\Gamma(n)$ | OPEN | CLOSE |
|------|---|---|---|-------------|
| 0 | | | \emptyset | \emptyset |
| 1 | A | B ⁽³⁵⁾ , C ⁽³⁷⁾ , D ⁽³⁰⁾ , E ⁽²⁰⁾ | E ⁽²⁰⁾ , D ⁽³⁰⁾ , B ⁽³⁵⁾ , C ⁽³⁷⁾ | A |
| 2 | E | G ⁽¹⁹⁾ , Y ⁽¹⁷⁾ | Y ⁽¹⁷⁾ , G ⁽¹⁹⁾ , D ⁽³⁰⁾ , B ⁽³⁵⁾ , C ⁽³⁷⁾ | E |
| 3 | Y | M ⁽⁰⁾ | M ⁽⁰⁾ , G ⁽¹⁹⁾ , D ⁽³⁰⁾ , B ⁽³⁵⁾ , C ⁽³⁷⁾ | Y |
| 4 | M | TRUE | | |

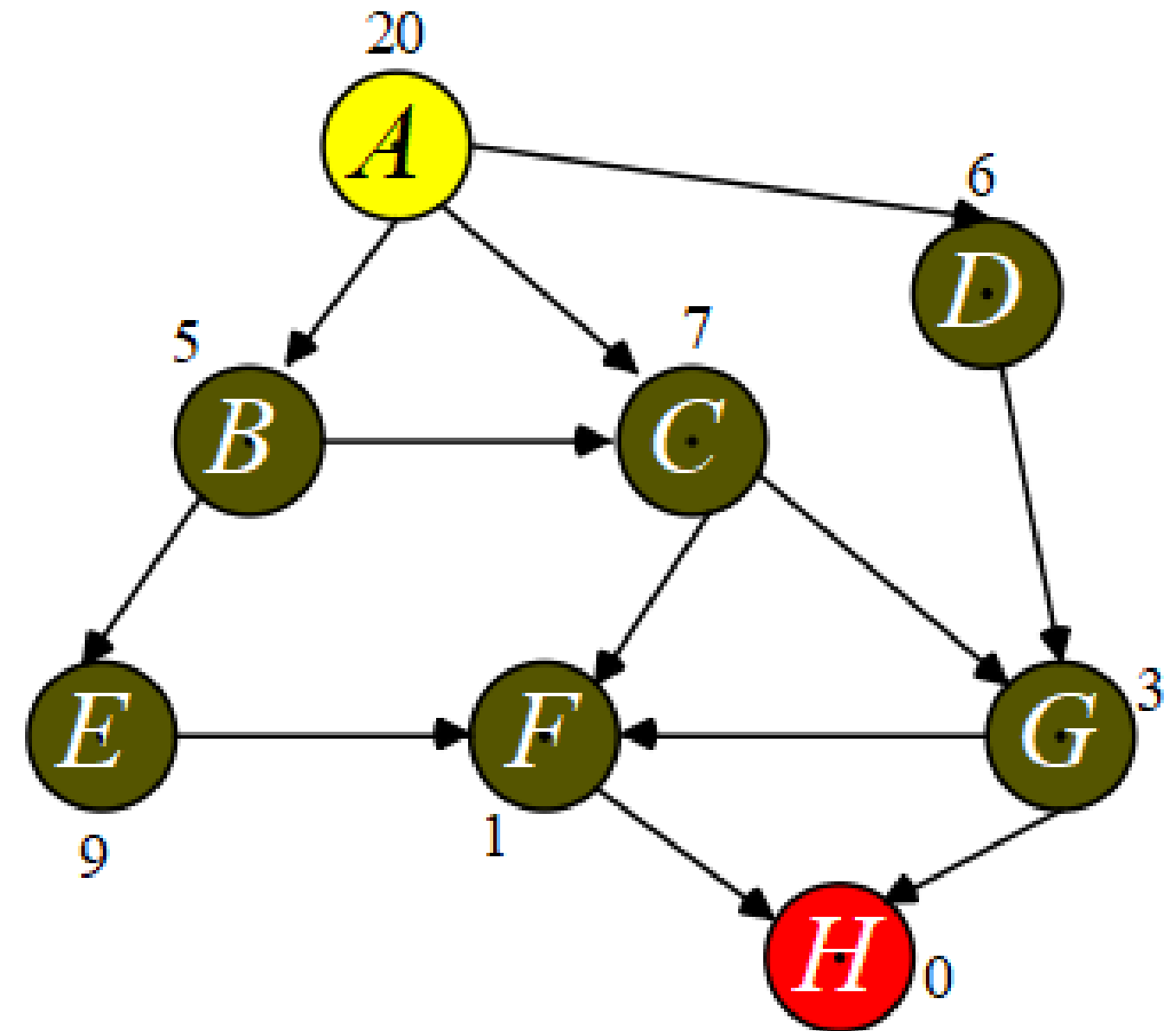


- Giải thuật có khuynh hướng sa lầy ở cực đại cục bộ
- Lời giải tìm được không tối ưu
- Không tìm được lời giải dù rằng có lời giải
- Có thể rơi vào vòng lặp vô hạn vì không lưu giữ trạng thái của các đỉnh đã xét.

▪ A \Rightarrow E \Rightarrow Y \Rightarrow M

Chương 3. HEURISTIC

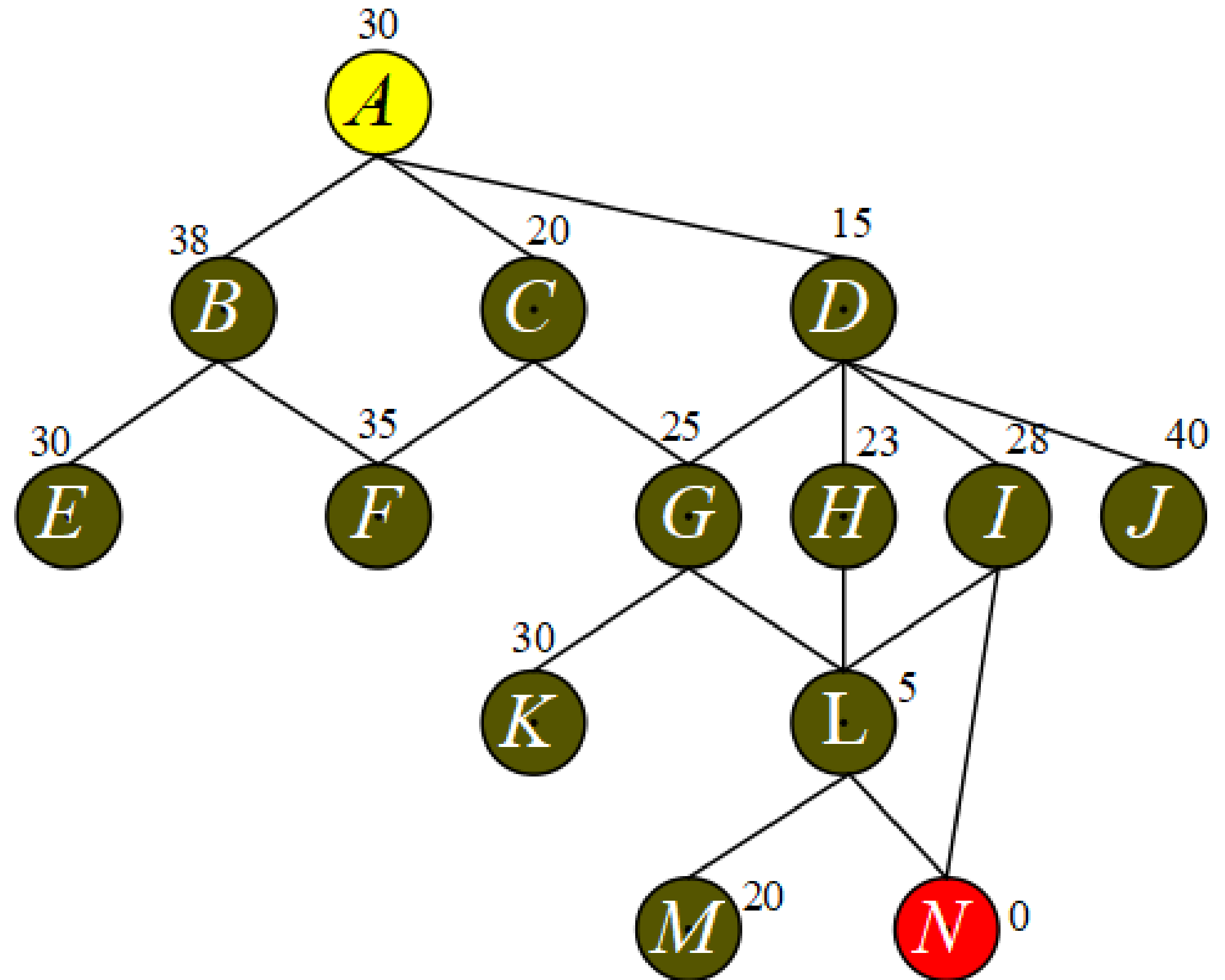
- Tìm kiếm leo đồi (Hill Climbing Search)
- Trình bày thuật toán leo đồi dốc đứng. Áp dụng thuật toán để tìm đường đi ngắn nhất từ đỉnh A đến M trên đồ thị, với các ước lượng heuristic của các trạng thái so với trạng thái đích được cho trong hình.



Chương 3. HEURISTIC

■ Tìm kiếm leo đồi (Hill Climbing Search)

- Trình bày thuật toán leo đồi dốc đứng. Áp dụng thuật toán để tìm đường đi ngắn nhất từ đỉnh A đến N trên đồ thị, với các ước lượng heuristic của các trạng thái so với trạng thái đích được cho trong hình.



HẾT CHƯƠNG 3