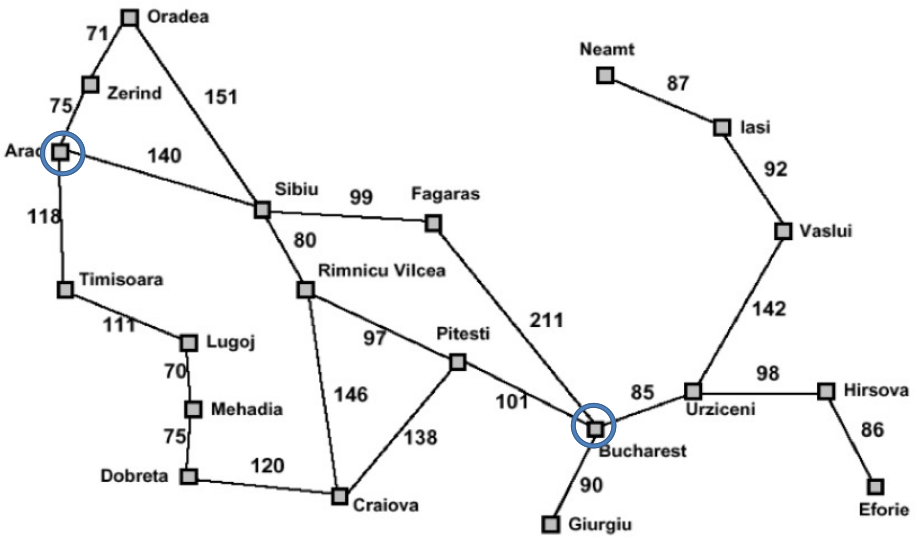
Tìm kiếm heuristic

- Bài toán tìm đường:
  - *Thành phố xuất phát: Arad*
  - *Thành phố đích: Bucharest*
  - *Các cạnh biểu diễn đường nối trực tiếp giữa hai thành phố, các con số ghi trên các cạnh là chi phí đi giữa hai thành phố.*
  - *Cột bên phải là khoảng cách Euclic từ các thành phố đến thành phố đích Bucharest.*
- Sử dụng phương pháp tìm kiếm A* (hàm ước lượng f(n) = g(n) + h(n), với g(n) là chi phí từ thành phố xuất phát đến n và h(n) là khoảng cách Euclic từ n đến đích)

116

Tìm kiếm heuristic



Straight–line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

117

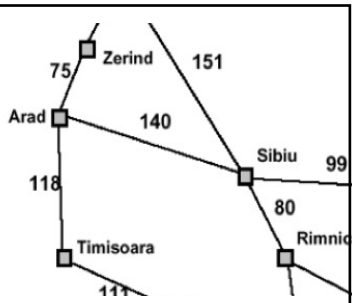
Mã giả giải thuật A*

```
g(n_o)=0; f(n_o)=h(n_o);
open:=[n_o]; closed:=[];
while open<>[] do
    loại n bên trái của open và đưa n vào closed;
    if (n là một đích) then thành công, thoát
    else
        Sinh các con m của n;
        For m thuộc con(n) do
            g(m)=g(n)+c[n,m];
            If m không thuộc open hay closed then
                f(m)=g(m)+h(m); cha(m)=n;  Bỏ m vào open;
            If m thuộc open (tồn tại m' thuộc open, sao cho m=m')  then
                If g(m)<g(m') then   g(m')=g(m); f(m')=g(m')+h(m');  Cha(m')=n;
            If m thuộc closed (tồn tại m' thuộc closed, sao cho m=m') then
                If g(m)<g(m') then  f(m)=g(m)+h(m); cha(m)=n;
                                    Đưa m vào open; loại m' khỏi closed;
        Sắp xếp open để t.thái tốt nhất nằm bên trái;
```

118

118

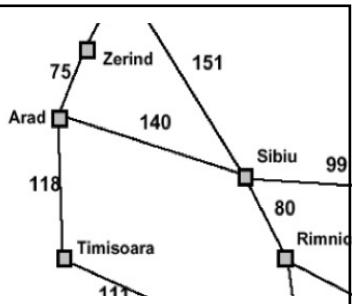
- Mỗi trạng thái n tùy ý sẽ gồm: (g(n), h(n), f(n), cha(n))
- **Bước 1**:
  - *Open={ Arad (0,366,366,-)}; close={}*
- **Bước 2:**
  - *Các con của Arad: Timisoara, Sibu, Zerind*
  - ***Xét Timisoara***
    - **g(Timisoara) = g(Arad) + c[Arad, Timisoara] = 0+ 118= 118 (do đề bài cung cấp) (g(m) = g(n) + c[m,n])**
    - **Timisoara không thuộc Open; Close**
      - ***Tính giá trị f(Timisoara) = g(Timisoara) + h(Timisoara)***
        = 118 + 329 = 447
      - ***Cập nhật cha của Timisoara : Arad***
      - ***Đưa Timisoara vào open: Timisoara(118,329,447,Arad)***



119

119

- **Bước 2:**
  - …..
  - *Xét Sibiu*
    - g(Sibiu) = g(Arad) + c[Arad, Sibiu]
      = 0+ 140= 140 (do đề bài cung cấp)
      (g(m) = g(n) + c[m,n])
    - Sibu không thuộc Open; Close
      - *Tính giá trị f(Sibiu) = g(Sibiu) + h(Sibiu) = 140 + 253 = 393*
      - *Cập nhật cha của Sibiu : Arad*
      - *Đưa Sibiu vào open: Sibiu (140,253,393,Arad)*

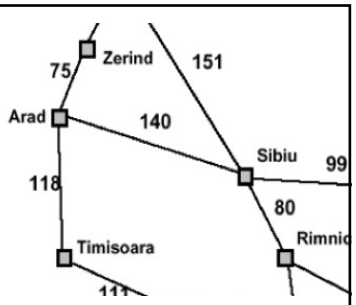


120

120

- **Bước 2:**
  - …..
  - *Xét Zerind*
    - g(Zerind) = g(Arad) + c[Arad, Zerind]
      = 0+ 75= 75 (do đề bài cung cấp)
      (g(m) = g(n) + c[m,n])
    - Zerind không thuộc Open; Close
      - *Tính giá trị f(Zerind) = g(Zerind) + h(Zerind) = 75 + 374 = 449*
      - *Cập nhật cha của Zerind : Arad*
      - *Đưa Zerind vào open: Zerind(75,374,449,Arad)*
  - *Sắp xếp các phần tử trong open để trạng thái tốt nhất bên trái*

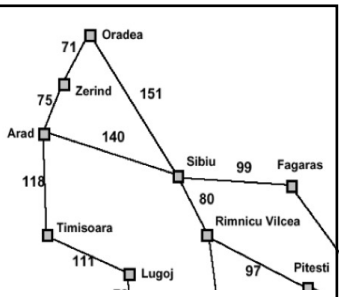
*Open{**Sibiu (140,253,393,Arad), Timisoara(118,329,447,Arad),** Zerind**(75,374,449,Arad)**};*
Closed = { Arad (0,366,366,-) }



121

121

- **Bước 3**
  - *Quay lại đầu vòng lặp while*
  - *Lấy phần tử **Sibiu** ra khỏi Open đưa vào Closed*



*Open{**Timisoara(118,329,447,Arad)**, Zerind**(75,374,449,Arad)**};*

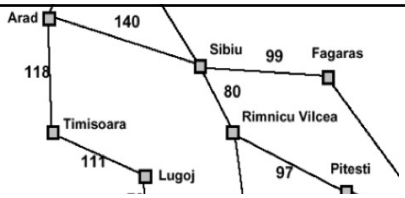
*Closed = { Arad (0,366,366,-), **Sibiu (140,253,393,Arad)** }*

  - *Các con của **Sibiu** : Rimnicu Vilces, Fagaras, Arad, Oradea*
  - ***Xét Rimnicu***
    - **g(Rimnicu) = g(Sibiu) + c[Sibiu, Rimnicu] = 140+ 80= 220**
      (g(m) = g(n) + c[m,n])
    - **Rimnicu không thuộc Open; Close**
      - ***Tính giá trị** f(Rimnicu) = g(Rimnicu) + h(Rimnicu)*
        = 220 + 193 = 413
      - ***Cập nhật cha của** Rimnicu : Sibiu*
      - ***Đưa** Rimnicu **vào open:** Rimnicu(220,193,413,Sibiu)*

122

122

- **Bước 3**
  - …..
  - *Các con của **Sibiu** : Rimnicu Vilces, Fagaras, Arad, O*
  - ***Xét Fagaras***
    - **g(Fagaras) = g(Sibiu) + c[Sibiu, Fagaras] = 140+ 99= 239 (g(m) = g(n) + c[m,n])**
    - **Fagaras không thuộc Open; Close**
      - ***Tính giá trị** f(Fagaras) = g(Fagaras) + h(Fagaras) = 239 + 178 = 417*
      - ***Cập nhật cha của** Fagaras : Sibiu*
      - ***Đưa** Fagaras **vào open:** Fagaras(239,178,417,Sibiu)*
  - *Sắp xếp các phần tử trong open để trạng thái tốt nhất bên trái*



*Open{**Rimnicu(220,193,413,Sibiu)**, **Fagaras(239,178,417,Sibiu)**,*
*Timisoara(118,329,447,Arad)**, Zerind**(75,374,449,Arad)**};*

*Closed = { Arad (0,366,366,-), **Sibiu (140,253,393,Arad)** }*

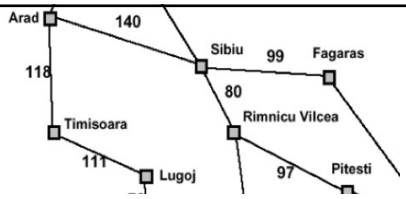
123

123

- **Bước 3**
  - …..
  - *Các con của* ***Sibiu*** *: Rimnicu, Fagaras, Arad, Oradea*
  - ***Xét Oradea***
    - **g(Oradea) = g(Sibiu) + c[Sibiu, Oradea] = 140+ 151= 291 (g(m) = g(n) + c[m,n])**
    - **Oradea không thuộc Open; Close**
      - ***Tính giá trị f(Oradea) = g(Oradea) + h(Oradea) = 291 + 380 = 671***
      - ***Cập nhật cha của Oradea : Sibiu***
      - ***Đưa Oradea vào open: Oradea(291,380,671,Sibiu)***
  - *Sắp xếp các phần tử trong open để trạng thái tốt nhất bên trái*

*Open{**Rimnicu(220,193,413,Sibiu), Fagaras(239,178,417,Sibiu),***
***Timisoara(118,329,447,Arad),** Zerind**(75,374,449,Arad), Oradea(291,380,671,Sibiu)};*

Closed = { Arad (0,366,366,-), ***Sibiu (140,253,393,Arad)*** *}*



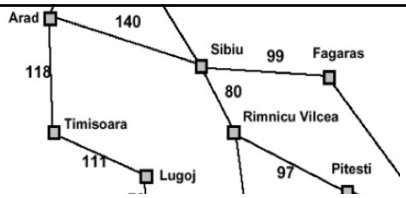
124

124

- **Bước 3**
  - …..
  - *Các con của* ***Sibiu*** *: Rimnicu, Fagaras, Arad, Oradea*
  - ***Xét*** *Arad*
    - **g(**Arad**) = g(Sibiu) + c[Sibiu,** Arad**] = 140+ 140= 280 (g(m) = g(n) + c[m,n])**
    - Arad **thuộc Closed**
      - ***G(Arad)=280> G(Arad') => ko làm gì cả (ko đưa vào open hay closed)***
  - *Sắp xếp các phần tử trong open để trạng thái tốt nhất bên trái*

*Open{**Rimnicu(220,193,413,Sibiu), Fagaras(239,178,417,Sibiu),***
***Timisoara(118,329,447,Arad),** Zerind**(75,374,449,Arad), Oradea(291,380,671,Sibiu)};*

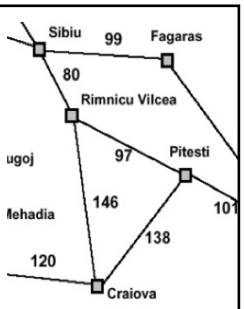
Closed = { Arad (0,366,366,-), ***Sibiu (140,253,393,Arad)*** *}*



125

125

- **Bước 4**
  - *Quay lại đầu vòng lặp while*
  - *Lấy phần tử* ***Rimnicu*** *ra khỏi Open đưa vào Closed*



*Open{**Fagaras(239,178,417,Sibiu),Timisoara(118,329,447,Arad),***
*Zerind**(75,374,449,Arad), Oradea(291,380,671,Sibiu)};***

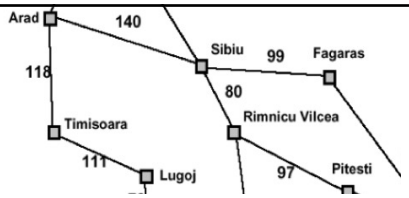
Closed = { Arad (0,366,366,-), ***Sibiu (140,253,393,Arad), Rimnicu(220,193,413,Sibiu*** *}*

- *Các con của* ***Rimnicu*** *: Craiova, Pitesti,* ***Sibiu***
- ***Xét Craiova***
  - **g(Craiova) = g(Rimnicu) + c[Rimnicu, Craiova] = 220 + 146 = 366**
  - **Craiova không thuộc Open; Close**
    - ***Tính giá trị f(Craiova) = g(Craiova) + h(Craiova) = 366 + 160 = 526***
    - ***Cập nhật cha của Craiova là Rimnicu***
    - ***Đưa Craiova vào open: Craiova(366,160,526, Rimnicu)***

126

126

- **Bước 4**
  - ….
  - *Các con của* ***Rimnicu*** *: Craiova, Pitesti,* ***Sibiu***
  - ***Xét Pitesti***
    - **g(Pitesti) = g(Rimnicu) + c[Rimnicu, Pitesti] = 220 + 97 = 317**
    - **Pitesti không thuộc Open; Close**
      - ***Tính giá trị f(Pitesti) = g(Pitesti) + h(Pitesti) = 317 + 98 = 415***
      - ***Cập nhật cha của Pitesti là Rimnicu***
      - ***Đưa Pitesti vào open: Pitesti (317,98,415, Rimnicu)***
  - *Sắp xếp các phần tử trong open để trạng thái tốt nhất bên trái*

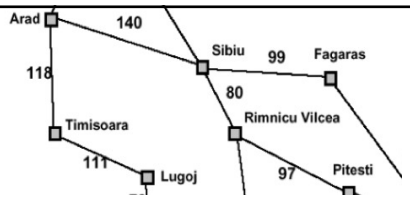


*Open{**Pitesti (317,98,415, Rimnicu), Fagaras(239,178,417,Sibiu),Timisoara(118,329,447,Arad),***
*Zerind**(75,374,449,Arad), Craiova(366,160,526, Rimnicu), Oradea(291,380,671,Sibiu)};***

Closed = { Arad (0,366,366,-), ***Sibiu (140,253,393,Arad), Rimnicu(220,193,413,Sibiu*** *}*

127

127



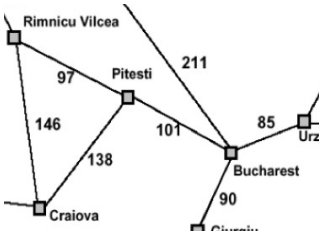
- **Bước 4**
  - ….
  - *Các con của **Rimnicu** : Craiova, Pitesti, **Sibiu***
  - ***Xét Sibiu***
    - **g(Sibiu) = g(Rimnicu) + c[Rimnicu, Sibiu] = 220 + 80 = 400**
    - **Sibiu thuộc Close**
      - ***Tính giá trị g(Sibiu) =400 > g(Sibiu')=140***
      - ***=> không làm gì cả (ko đưa vào open hay closed)***
      -
  - *Sắp xếp các phần tử trong open để trạng thái tốt nhất bên trái*

*Open{**Pitesti (317,98,415, Rimnicu), Fagaras(239,178,417,Sibiu),Timisoara(118,329,447,Arad),***
*Zerind**(75,374,449,Arad), Craiova(366,160,526, Rimnicu), Oradea(291,380,671,Sibiu)};***

*Closed = { Arad (0,366,366,-), **Sibiu (140,253,393,Arad), Rimnicu(220,193,413,Sibiu** }*

128

128



- **Bước 5**
  - *Quay lại đầu vòng lặp while*
  - *Lấy phần tử **Pitesti** ra khỏi Open đưa vào Closed*

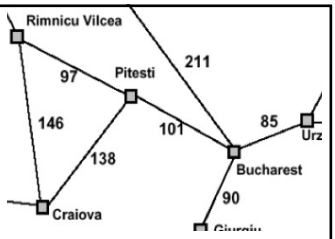
*Open{**Fagaras(239,178,417,Sibiu),Timisoara(118,329,447,Arad),***
*Zerind**(75,374,449,Arad), Craiova(366,160,526, Rimnicu), Oradea(291,380,671,Sibiu)};***

Closed = { Arad (0,366,366,-), Sibiu (140,253,393,Arad), Rimnicu(220,193,413,Sibiu),
Pitesti (317,98,415, Rimnicu) }

  - *Các con của **Pitesti** : Craiova, Bucharest, Rimnicu*
  - ***Xét Craiova***
    - **g(Craiova) = g(Pitesti) + c[Pitesti ,Craiova] = 317 + 138 = 455**
    - **Craiova thuộc Open;**
    - **g(Craiova)=455 > g(Craiova')=** 368 => Không làm gì cả

129

129



- **Bước 5**
  - *Quay lại đầu vòng lặp while*
  - *Lấy phần tử **Pitesti** ra khỏi Open đưa vào Closed*

*Open{**Fagaras(239,178,417,Sibiu),Timisoara(118,329,447,Arad),***
*Zerind**(75,374,449,Arad), Craiova(366,160,526, Rimnicu), Oradea(291,380,671,Sibiu)**};*

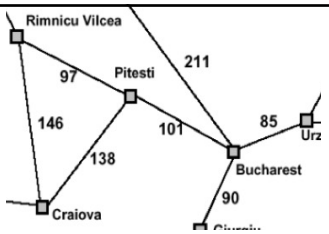
***Closed** = { Arad (0,366,366,-), Sibiu (140,253,393,Arad), Rimnicu(220,193,413,Sibiu),*
***Pitesti** (317,98,415, Rimnicu) }*

*Các con của **Pitesti** : Craiova, Bucharest, Rimnicu*

- ***Xét** Bucharest*
  - **g(**Bucharest **) = g(Pitesti) + c[Pitesti,** Bucharest **] = 317 + 101 = 418**
  - Bucharest **không thuộc Open; Closed**
    - ***Tính giá trị f(**Bucharest **) = g(**Bucharest **) + h(**Bucharest **) = 418 +0 = 418***
    - ***Cập nhật cha của** Bucharest **: Pitesti***
    - ***Đưa** Bucharest **vào open:** Bucharest **(418,0,418,Pitesti)***

130

130



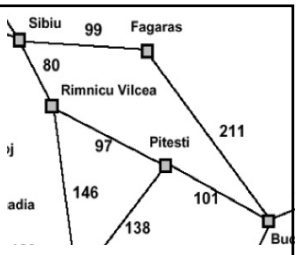
- **Bước 5**
  - …
  - *Các con của **Pitesti** : Craiova, Bucharest, **Rimnicu***
  - ***Xét Rimnicu***
    - **g(Rimnicu) = g(Pitesti) + c[Pitesti , Rimnicu] = 317 + 138 = 455**
    - **Rimnicu thuộc Closed**
    - **Xét g(Rimnicu)=455> g(Rimnicu ')=220**
      - ***=> Không làm gì cả***

***Open**{Fagaras(239,178,417,Sibiu),Bucharest(418,0,418,Pitesti),Timisoara(118,329,447,Arad),*
Zerind(75,374,449,Arad), Craiova(368,160,528, Rimnicu), Oradea(291,380,671,Sibiu)};

***Closed** = { Arad (0,366,366,-), Sibiu (140,253,393,Arad), Rimnicu(220,193,413,Sibiu), Pitesti (317,98,415, Rimnicu) }*

131

131



- **Bước 6**
  - *Quay lại đầu vòng lặp while*
  - *Lấy phần tử **Fagaras** ra khỏi Open đưa vào Closed*

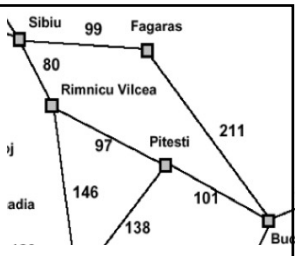
***Open**{Bucharest(418,0,418,Pitesti),Timisoara(118,329,447,Arad), Zerind(75,374,449,Arad), Craiova(368,160,528, Rimnicu),Oradea(291,380,671,Sibiu)};*

Closed** = { Arad (0,366,366,-), Sibiu (140,253,393,Arad), Rimnicu(220,193,413,Sibiu), Pitesti (317,98,415, Rimnicu), **Fagaras(239,178,417,Sibiu)) }

  - *Các con của **Fagaras** : Bucharest, Sibiu*
  - ***Xét Sibiu***
    - **g(Sibiu) = g(Fagaras) + c[Fagaras , Sibiu] = 239 + 99 = 338**
    - **Sibiu thuộc Closed;**
    - **g(Sibiu)=338 > g(Sibiu ')=** 140 => Không làm gì cả

132

132



- **Bước 6**
  - *Quay lại đầu vòng lặp while*
  - *Lấy phần tử **Fagaras** ra khỏi Open đưa vào Closed*

***Open**{Bucharest(418,0,418,Pitesti),Timisoara(118,329,447,Arad), Zerind(75,374,449,Arad), Craiova(368,160,528, Rimnicu),Oradea(291,380,671,Sibiu)};*

Closed** = { Arad (0,366,366,-), Sibiu (140,253,393,Arad), Rimnicu(220,193,413,Sibiu), Pitesti (317,98,415, Rimnicu), **Fagaras(239,178,417,Sibiu)) }

  - *Các con của **Fagaras** : Bucharest, Sibiu*
  - ***Xét Bucharest***
    - **g(Bucharest) = g(Fagaras) + c[Fagaras , Bucharest] = 239 + 211 = 450**
    - **Bucharest thuộc Open;**
    - **g(Bucharest)=450> g(Bucharest ')= 418** => Không làm gì cả

133

133

- **Bước 7**
  - *Quay lại đầu vòng lặp while*
  - *Lấy phần tử **Bucharest** ra khỏi Open đưa vào Closed*

***Open**{Timisoara(118,329,447,Arad),*
Zerind(75,374,449,Arad), Craiova(368,160,528, Rimnicu),Oradea(291,380,671,Sibiu)};

Closed** = { Arad (0,366,366,-), Sibiu (140,253,393,Arad), Rimnicu(220,193,413,Sibiu), Pitesti (317,98,415, Rimnicu), Fagaras(239,178,417,Sibiu), **Bucharest(418,0,418,Pitesti)}

- ***Xét Bucharest là trạng thái đích, giải thuật dừng lại.***
- ***Đường đi được tìm bằng cách truy ngược cha của nút gốc và các nút kế tiếp:***

***Bucharest(418,0,418,Pitesti) =>** Pitesti (317,98,415, Rimnicu) => Rimnicu(220,193,413,Sibiu) => Sibiu (140,253,393,Arad) => Arad (0,366,366,-)*

***Lưu ý:** Các trạng thái trong closed là trạng thái đã xét qua, tuy nhiên **không phải tất cả** các trạng thái trong closed đều góp phần trong đường đi lời giải (Ví dụ đỉnh Fagaras trong bài toán này không thuộc đường đi lời giải)*

134

134

Leo đồi (Hill climbing)

- **Ý tưởng:** Tìm kiếm trạng thái đích bằng cách hướng tới trạng thái tốt hơn trạng thái hiện tại (Leo lên đỉnh của một ngọn đồi)
- **Đặc điểm của giải thuật leo đồi:**
  - *Trạng thái con tốt nhất sẽ được chọn cho bước tiếp theo*
  - *Không lưu giữ bất kỳ thông tin về các nút cha và anh em.*
  - *Quá trình tìm kiếm sẽ dừng lại khi gặp trạng thái đích hoặc trạng thái kế tiếp "xấu" hơn trạng thái đang xét ($f_{\text{đang xét}} < f_{\text{trang thái kế tiếp}}$)*
  - *Sử dụng hàm đánh giá để đo tính tốt hơn của một trạng thái so với trạng thái khác*

135

135

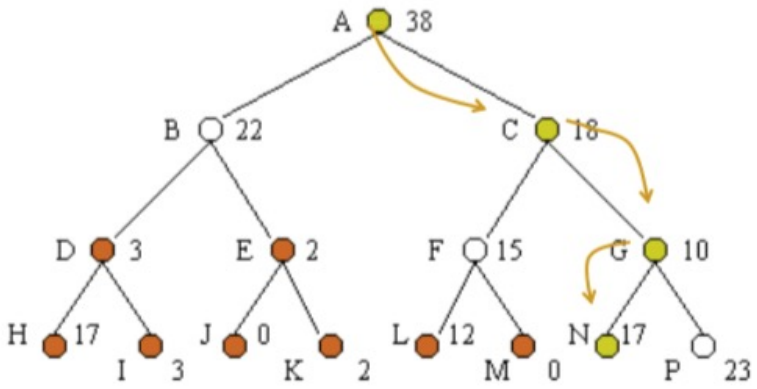
Leo đồi (Hill climbing)

1. **Đánh giá trạng thái khởi đầu**. Nếu nó là trạng thái đích, thoát, nếu không xét nó như trạng thái hiện hành
2. **Lặp lại đến khi tìm thấy một lời giải hoặc đến khi không tìm thấy « toán tử » mới nào có thể áp dụng lên trạng thái hiện hành**:
   a) *Chọn một toán tử chưa được áp dụng đối với trạng thái hiện hành, áp dụng nó để sinh ra một trạng thái mới NS*
   b) *Đánh giá trạng thái mới NS*
      i. Nếu NS là một trạng thái đích, return NS và thoát
      ii. Nếu NS không là đích nhưng « tốt hơn » trạng thái hiện hành, lấy NS làm trạng thái hiện hành
      iii. Nếu NS không tốt hơn trạng thái hiện hành, tiếp tục vòng lặp

136

136

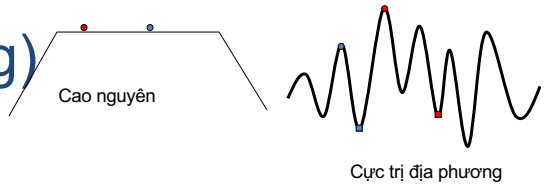
Tìm kiếm leo đồi (tt)



137

137

Leo đồi (Hill climbing)



- Hạn chế của tìm kiếm leo đồi:
  - *Không thể phục hồi lại từ những thất bại trong chiến lược của nó*
  - *Hiệu quả hoạt động chỉ có thể được cải thiện trong một phạm vi giới hạn nào đó*
  - *Lời giải tìm được không tối ưu hoặc không tìm được lời giải mặc dù có tồn tại lời giải do:*
    - Có khuynh hướng sa lầy ở cực đại cục bộ
    - Cao nguyên
    - Chỉ chỉ với một phép toán, không cho ra trạng thái « tốt hơn », nhưng với một vài phép toán có thể chuyển đến trạng thái « tốt hơn »

138

THUẬT TOÁN LEO ĐỒI (3)
(Hill-Climbing)

- Một vài giải pháp xử lý các vấn đề này:
  - ***Quay lui*** *« một vài bước » trước đó và thử đi theo một hướng khác. Để thực thi chiến lược này, duy trì một danh sách các bước đã trải qua. Giải pháp này đặc biệt phù hợp để xử lý tình huống « Local Optima »*
  - ***Tạo ra một « bước nhảy đột phá » theo một hướng****: để chuyển sang một « vùng » mới trong không gian tìm kiếm. Phù hợp để xử lý tình huống « Plateau »*
  - ***Áp dụng nhiều hơn một toán tử*** *để nhận được một trạng thái sau đó mới kiểm thử. Phù hợp để xử lý tình huống « Ridge »*

139

139