Thuật toán Quay Lui

Nguyễn Văn Huy

huyite.vn@gmail.com

Ngày 21 tháng 10 năm 2021

Outline

Định nghĩa

Tìm nghiệm của bài toán bằng cách xem xét tất cả các phương án có thể

Ưu điểm

- Cài đặt đơn giản
- Luôn tìm được nghiệm chính xác

Nhược điểm

- Thời gian thực thi khá lớn
- Độ phức tạp thuật toán lớn

Phù hợp với các bài toán kích thước nhỏ.

Định nghĩa

Bài toán có dạng tìm đối tượng x có dạng là một vector thỏa mãn một số điều kiện sau:

- Đối tượng x gồm n phần tử: $x = (x_1, x_2, ..., x_n)$ vector nghiem
- x thỏa mãn bởi hàm ràng buộc $g(x)$

Mot trong nhung cach quet het kgian nghiem: thuat toan quay lui - tu tuong vet can

De quy la mot th cua thuat toan quay lui (backing) - tuy nhien nhuoc diem la ton bo nho.

i: chi so tren vector nghiem

```
1  attempt(i): (try) (co gang thu den khi nao dung thi thoi)
2     for v in ⟨các giá trị có thể của xᵢ⟩ :
3         gán xᵢ= v
4         if ⟨xᵢ là phần tử cuối cùng trong cấu hình⟩
5             {xuất ra cấu hình tìm được}
6         else:
7             {đánh dấu đã gán v cho xᵢ (nếu cần)}
8             attempt(i+1)
9             {bỏ đánh dấu đã gán v cho xᵢ (nếu cần)}
```

Sinh nhị phân

Sinh dãy nhị phân độ dài n Ví dụ với n = 3, các dãy nhị phân cần sinh ra sẽ là:

- 000
- 001
- 010
- 011
- 100
- 101
- 110
- 111

Định nghĩa

Hãy xác định tất cả các tổ hợp chập k của tập n phần tử

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \tag{1}$$

Bài toán tổ hợp

Nghiệm cần tìm của bài toán tìm các tổ hợp chập k của n phần tử phải thoả mãn các điều kiện sau:

- Là một vector $x = (x_1, x_2, \dots x_k)$
- x_i lấy giá trị trong tập $\{1, 2, \dots n\}$
- Ràng buộc: $x_i < x_{i+1}$ với mọi giá trị i từ 1 đến $k-1$.

Có ràng buộc 3 là vì tập hợp không phân biệt thứ tự phần tử nên ta sắp xếp các phần tử theo thứ tự tăng dần.

Phương pháp quay lui

Algorithm 1 Quay lui

```
1: procedure TRY(i)
2:     for j = x[i − 1] + 1 to n − k + i do
3:         x[i] = j
4:         if i = k then print(x)
5:         else
6:             Try(i + 1)
```

Sinh chỉnh hợp không lặp chập k của n

Cho tập hợp AA gồm nn phần tử; $n \geq 1$. Một chỉnh hợp chập k các phần tử của A là một cách sắp xếp k phần tử khác nhau của A

$$A_n^k = \frac{n!}{(n-k)!} \tag{2}$$

Phương pháp quay lui

Algorithm 2 Quay lui

```
1: procedure TRY(i)
2:     for j = 1 to n do
3:         if d[i]! = 1 then
4:             x[i] = j
5:             d[i] = 1
6:             if i = k then print(x)
7:             else
8:                 Try(i + 1)
9:             d[i] = 0
```

Bài toán tối ưu tổ hợp

Định nghĩa

Bài toán tối ưu tổ hợp là bài toán tìm phương án tối ưu trên tập các cấu hình tổ hợp. Nghiệm của bài toán cũng là một vector x gồm n thành phần sao cho:

- $x = (x_1, x_2, \ldots x_n)$
- x_i lấy giá trị trong tập $a_1, a_2, \ldots a_m$
- x thoả mãn các ràng buộc cho bởi hàm $G(x)$.
- $f(x) \longrightarrow min/max$.

Bài toán Knapsack

Có n đồ vật, mỗi đồ vật có giá trị v_i, trọng lượng w_i. Một người có khả năng mang được trọng lượng W, hãy lựa chọn các đồ vật sao cho tổng trọng lượng các đồ vật nhỏ hơn W và tổng giá trị các đồ vật lớn nhất có thể.

Math Model

- Biến quyết định

$$x_i = \begin{cases} 1 & \text{Chọn đồ vật } i \\ \\ 0 & \text{otherwise} \end{cases}$$

- Tham số: v_i giá trị đồ vật thứ i, w_i trọng lượng đồ vật thứ i
- Ràng buộc

$$\sum_{i=1}^{n} w_i \times x_i \leq W \tag{3}$$

- Hàm mục tiêu

$$f(x_{i,j}) = \sum_{i=1}^{n} v_i \times x_i \longrightarrow max \tag{4}$$

Nghiệm của bài toán cũng là một vector x gồm n thành phần sao cho:

- $x = (x_1, x_2, \ldots x_n)$
- x_i lấy giá trị trong tập $\{0, 1\}$
- x thoả mãn các ràng buộc

$$\sum_{i=1}^{n} w_i \times x_i \leq W \tag{5}$$

- Hàm mục tiêu:

$$f(x_{i,j}) = \sum_{i=1}^{n} v_i \times x_i \longrightarrow max \tag{6}$$

Phương pháp quay lui

Algorithm 3 Quay lui

1: **procedure** TRY(i)
2: **for** $j = 0$ to 1 **do**
3: $x[i] = j$
4: **if** $i = n$ and $\sum_{i=1}^{n} w[i] \times x[i] \leq W$ and $Max \leq \sum_{i=1}^{n} v_i \times x_i$ **then**
5: Update Best solution is x
6: **else**
7: $Try(i + 1)$

Nhánh cận

Algorithm 4 Quay lui

```
1: procedure TRY(i)
2:     for j = 0 to 1 do
3:         x[i] = j
4:         if ∑_{i=1}^{n} w[i] × x[i] ≤ W then
5:             if i = n and Max ≤ ∑_{i=1}^{n} v_i × x_i then
6:                 Update Best solution is x
7:             else
8:                 Try(i + 1)
```

Lựa chọn cân

Cho n quả cân có khối lượng khác nhau. Viết chương trình đưa ra tất cả các cách có thể đưa các quả cân lên 2 đĩa cân sao cho cân cân bằng!

Lựa chọn cân

Có 1 rôbốt có thể di chuyển 1 đến 2 mét 1 lần, hãy viết 1 chương trình đưa ra các cách mà 1 robot có thể đi trên quảng đường n mét. Đưa ra tất cả các cách đi đó!

- vd: n= 3 mét
- Số cách đi là : 3
- Các cách đi:
- 1 1 1
- 2 1
- 1 2

Phân tích số

Liệt kê tất cả các cách phân tích số nguyên dương n thành tổng các số nguyên dương, hai cách phân tích là hoán vị của nhau chỉ tính là một cách. Ví dụ với n = 5, các cách phân tích cần liệt kê sẽ là:

Bài toán người du lịch

Có n thành phố, $d[i, j]$ là chi phí để di chuyển từ thành phố i đến thành phố j. (Nếu không có đường đi thì $d[i, j] = \infty$). Một người muốn đi du lịch qua tất cả các thành phố, mỗi thành phố một lần rồi trở về nơi xuất phát sao cho tổng chi phí là nhỏ nhất. Hãy xác định một đường đi như vậy.