Phương pháp nhánh cận

Ngày 25 tháng 4 năm 2025

Nội dung

1. Giới thiệu
2. Mô hình
3. Bài tập

Nội dung

1. Giới thiệu
2. Mô hình
3. Bài tập

Giới thiệu

Hạn chế của phương pháp quay lui, vét cạn

- Nhiều bài toán có không gian lời giải lớn, không thể giải được bằng phương pháp quay lui, vét cạn.
- Nhánh cận là phương pháp cải tiến của quay lui.

Giới thiệu

Ý tưởng và nguyên lý đánh giá

- Lần lượt xây dựng lời giải, khi hoàn tất lời giải, ta so sánh chi phí của nó với chi phí tốt nhất hiện có. Nếu chi phí mới là tốt hơn thì ta cập nhật thành chi phí tốt nhất hiện có.
- Mỗi lời giải khi xây dựng các thành phần nghiệm thì luôn kiểm tra điều kiện nếu đi tiếp theo hướng này thì có thể có khả năng nhận được lời giải tốt hơn lời giải hiện có hay không? Nếu không thì không đi theo hướng này nữa.

Nguyên lý đánh giá: Lợi dụng các thông tin đã tìm được trong lời giải của bài toán để loại bỏ sớm phương án không tối ưu.

Nội dung

Mô hình

- Giả sử ta có $D = \{x|x = (x_1, x_2, ..., x_n), x_i \in D_i\}$ là miền nghiệm của bài toán,
- $f : D \to \mathcal{R}$ gọi là hàm chi phí của nghiệm x,
- $(x_1, x_2, ..., x_i)$ gọi là lời giải bộ phận cấp i. X_i là tập các lời giải bộ phận cấp $i, \forall i = \overline{1, n}$,
- Đánh giá cận là tìm một hàm g xác định trên X_i sao cho:

$$g(x_1, ..., x_i) \leq Min\{f(a) : a = (a_1, ..., a_n) \in X, x_i = a_i, \forall i = \overline{1, n}\} \tag{1}$$

(Có nghĩa là: giá trị $g(x_1, ..., x_i)$ không lớn hơn giá trị của các phương án mở rộng từ lời giải bộ phận $(x_1, ..., x_i)$).

Mô hình

- x^* là lời giải tốt nhất hiện có (phương án mẫu), f^* là giá trị tốt nhất tương ứng $f^* = f(x^*)$.
- Nếu $g(x_1, ..., x_i) > f^*$ thì lời giải mở rộng từ $(x_1, x_2, ..., x_i)$ sẽ không tốt hơn phương án mẫu. Loại bỏ lời giải mở rộng phát triển từ $(x_1, ..., x_i)$

Mô hình

Sơ đồ kỹ thuật nhánh cận

```
branch_bound (i)
  for (j thuộc tập khả năng Dᵢ) {
      if (j chấp nhận được ){
         Xác định xᵢ theo j;
         Ghi nhận trạng thái mới (nếu có);
         if (i==n) {
            if (f ≤ f*)
                      Cập nhật lời giải;
         }else {
            Xác định cận g(x₁, x₂, ..., xᵢ);
            if (g(x₁, x₂, ..., xᵢ) ≤ f*)
                 branch_bound(i+1);
         }
         Bỏ ghi nhận (nếu có);
      }
  }
```

Nội dung

Bài tập

Bài toán người bán hàng

Bài toán: Một người bán hàng muốn tham quan n thành phố $T_1, ..., T_n$. Xuất phát từ một thành phố nào đó, người bán hàng muốn đi qua tất cả các thành phố còn lại. Mỗi thành phố đi qua đúng 1 lần rồi quay trở lại phành phố xuất phát. Gọi C_{ij} là chi phí đi từ thành phố T_i đến thành phố T_j.

Yêu cầu: Tìm một hành trình thỏa mãn yêu cầu bài toán sao cho tổng chi phí là nhỏ nhất

Bài tập

Bài toán người bán hàng

Ý tưởng:

- Gọi D là tập hoán vị của $\{1, 2, ..., n\}$
- Có $|D| = n!$.
- Nếu cho trước thành phố xuất phát, thì có $(n-1)!$ hành trình có thể có.

Định nghĩa hàm $f : D \rightarrow \mathcal{R}^+$, trong đó:
$f(x_1, x_2, ..., x_n) = \sum_{i=1}^{n-1} C_{x_i x_{i+1}}$
Tìm x^* để $f(x^*) \rightarrow min$

Bài tập

Bài toán người bán hàng

Thiết kế

- Đánh giá nhánh cận
  - Đặt $CMin = Min\{C_{ij} : i, j \in \{1, ..., n\}\}$
  - Giả sử tại bước i ta tìm được lời giải bộ phận cấp i là $(x_1, x_2, ..., x_i)$, tức là đã đi qua đoạn đường $T_1 \to T_2, ..., T_i$, tương ứng với chi phí: $sum_i = C_{1,x_2} + C_{x_2 x_3} + ... + C_{x_{i-1} x_i}$
  - Cần đi qua đoạn $n - i + 1$ đoạn đường nữa, gồm $n - i$ thành phố còn lại và đoạn quay lại T_1.
  - Do chi phí mỗi một trong $n - i + 1$ không nhỏ hơn $CMin$, nên hàm đánh giá nhánh cận được xác định như sau: $g(x_1, ..., x_i) = sum_i + (n - i + 1) * CMin$
  - Điều kiện chấp nhận được của j là thành phố T_j chưa đi qua. Dùng một mảng để biểu diễn tình trạng này.

$$flag_j = \begin{cases} 1 & \textit{nếu } T_j \textit{ đã được đi qua} \\ 0 & \textit{nếu } T_j \textit{ chưa được đi qua} \end{cases} \quad (2)$$

Bài tập

Bài toán người bán hàng

Thiết kế

- Đánh giá nhánh cận
  - Xác định x_i theo j bằng câu lệnh gán: $x_i = j$,
  - Cập nhận trạng thái $flag[j] = 1$,
  - Cập nhật lại chi phí sau khi tìm được $x_i = S + C_{x_{i-1}x_i}$,
- Cập nhật lời giải tối ưu:
  - Tính chi phí hành trình vừa tìm được,

    `cost = sum +` $C_{x_n 1}$`,`
    ```
        Nếu (cost < f*) thì
               f* = cost
    ```
  - `f*` là chi phí tốt nhất hiện có
- Thao tác hủy bỏ trạng thái:
  - $flag_j = 0$,
  - Trả lại chi phí cũ: $sum = sum - C_{x_{i-1}x_i}$,

Bài tập

Bài toán người bán hàng – Thủ tục nhánh cận

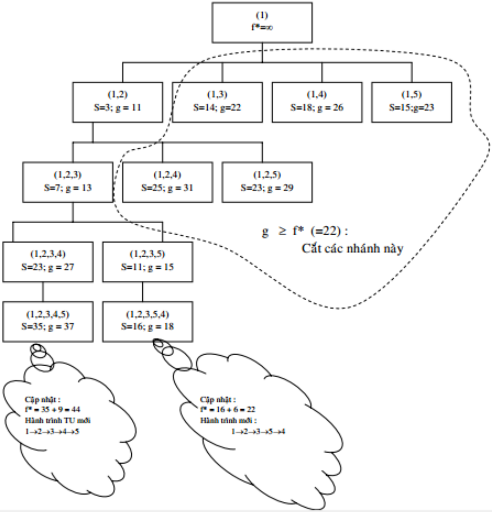
```
tsp (i)
  for (j = 2 → n) {
      if (flag_j == 0 ){
          x_i = j; flag_j = 1; sum = sum + C_{x_{i-1}x_i};
          g = sum + (n − i + 1) ∗ CMin;
        if (i==n) {
            cost = sum + C_{x_n 1};
            if (cost ≤ f*)
                  Cập nhật lời giải;
        }else {
            if (g ≤ f)
                tsp(i+1);
        }
        sum = sum − C_{x_{i-1}x_i}; flag_j = 0;
  } }
```

Bài tập

Bài toán người bán hàng

$$C = \begin{bmatrix} \infty & 3 & 14 & 18 & 15 \\ 3 & \infty & 4 & 22 & 20 \\ 17 & 9 & \infty & 16 & 4 \\ 6 & 2 & 7 & \infty & 12 \\ 9 & 15 & 11 & 5 & \infty \end{bmatrix}$$

Hình 1: Ma trận chi phí



Bài tập

Bài toán cái túi

Bài toán: Có n loại đồ vật, mỗi loại có số lượng không hạn chế. Đồ vật loại i có trọng lượng w_i và giá trị sử dụng v_i, $i \in \{1, ..., n\}$.
Yêu cầu: Chọn các đồ vật để đặt vào túi có giới hạn trọng lượng m, sao cho tổng giá trị lựa chọn là lớn nhất.

Bài tập

Bài toán cái túi

Ý tưởng

- Chọn những vật có giá trị cao, nhưng trọng lượng nhỏ, tức là (*đơn giá/giá trị* phải cao)
- Tính đơn giá cho các loại đồ vật
- Xét các loại đồ vật theo thứ tự giảm dần của đơn giá
- Với mỗi loại, lấy số lượng có thể có của các loại đồ vật.
- Xác định trọng lượng còn lại của túi và xét tiếp các loại đồ vật khác

Bài tập

Bài toán cái túi

Mô hình hóa

- Đặt $D = \{u = (x_1, ..., x_n) : \sum_{i=1}^{n} x_i w_i \leq m\}$.
- $f : D \to R^+, f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} x_i v_i$
- Tìm $x^* \in D : f^* = f(x^*) \to max$

Cách chọn vật

- Xét mảng đơn giá: $Dg = (\frac{v_1}{w_1}, ..., \frac{v_n}{w_n})$. Chọn vật theo đơn giá giảm dần

Đánh giá cận trên: Giả sử đã tìm được lời giải bộ phận: $(x_1, ..., x_i)$. Khi đó

- Tổng giá trị hiện tại của túi là: $sum = \sum_{j=1}^{i} x_j v_j$
- Giới hạn trọng lượng còn lại của túi là: $m_i = m - \sum_{j=1}^{i} x_j w_j$, do đó : $\sum_{j=i+1}^{n} x_j w_j \leq m_i$
- Ở bước i: Tối đa chọn được $\frac{m_i}{w_{i+1}}$ đồ vật
- Cận trên $sum + v_{i+1} * \frac{m_i}{w_{i+1}}$

Bài tập

Bài toán cái túi

- Các giá trị có thể chấp nhận được cho x_{i+1} là:
  $t = 0 \rightarrow (\frac{m_i}{w_{i+1}})$
- Ghi nhận trạng thái mới:
  - Cập nhật giá trị: $sum = sum + x_i * v_i$,
  - Cập nhật trọng lượng: $weight = weight + x_i * w_i$
- Trả lại trạng thái cũ:
  - Cập nhật giá trị: $sum = sum - x_i * v_i$,
  - Cập nhật trọng lượng: $weight = weight - x_i * w_i$
- Cập nhật lời giải tối ưu: Khi tìm được một lời giải, ta so sánh lời giải này với lời giải tốt nhất vào thời điểm hiện tại để chọn ra lời giải tối ưu

Bài tập

Bài toán cái túi – Thủ tục nhánh cận

```
knapsack (i)
   for (j = (m−weight)/w_i → 0) {
         x_i = j; weight = weight + w_i * x_i; sum = sum + v_i * x_i;
          if (i==n || m - weight ≤ 0) {
             if (f ≤ sum)
                    Cập nhật lời giải;
          }else {
             g = sum + v_{i+1} * (m − weight)/w_{i+1}
             if (g > f)
                 knapsack(i+1);
          }
          weight = weight − w_i * x_i; sum = sum − v_i * x_i;
      }
}
```

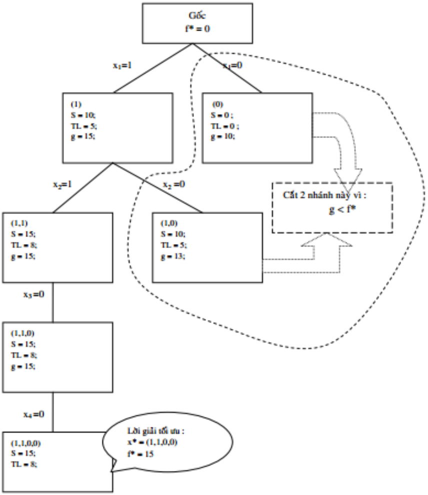
Bài tập

Bài toán cái túi

i	1	2	3	4
w	5	3	2	4
v	10	5	3	6

m = 8

Hình 3: Trọng lượng và giá trị của các vật



Bài tập về nhà

1. Cài đặt tất cả các bài tập trên lớp
2. Có n đồ vật, mỗi vật i đặc trưng bởi trọng lượng w_i và giá trị sử dụng v_i. Cần chọn các vật đặt vào một chiếc túi xách có giới hạn trọng lượng m, sao cho tổng giá trị sử dụng các vật được chọn là lớn nhất.