1. **GIỚI THIỆU CHUNG VỀ THUẬT TOÁN THAM LAM**
2. **Ý tưởng**

Phương pháp tham lam là kỹ thuật thiết kế thường được dung để giải các

bài toán tối ưu. Phương pháp được tiến hành trong nhiều bước. Tại mỗi bước, theo một lựa chọn nào đó ( xác định bằng một hàm chọn), sẽ tìm ra một lời giải tối ưu cho bài toán nhỏ tương ứng. Lời giải của bài toán được bổ sung dần từng bước từ lời giải của các bài toán con.

Lời giải được xây dựng như thế có chắc là lời giải tối ưu của bài toán?

Các lời giải tìm được bằng phương pháp tham lam thường là chấp nhận

được theo điều kiện nào đó, chưa chắc là tối ưu.

Cho trước một tập A gồm n đối tượng, ta cần phải chọn một tập con S của A. Với một tập con S được chọn ra thỏa mãn các yêu cầu của bài toán, ta gọi là một nghiệm chấp nhận được. Một hàm mục tiêu gắn mỗi nghiệm chấp nhận được với một giá trị. Nghiệm tối ưu là nghiệm chấp nhận được với giá trị nhỏ nhất ( lớn nhất).

Đặc trưng tham lam của phương pháp thể hiện bởi: trong mối bước việc xử lí sẽ tuân theo một sự lựa chọn trước, không kể đến tình trạng không tốt có thể xảy ra khi thực hiện lựa chọn lúc đầu.

1. **Mô hình**

Chọn S từ tập A.

Tính chất tham lam của thuật toán định hướng bởi hàm Chọn.

* Khởi động S = ∅;
* Trong khi A ≠ ∅:
  + Chọn phần tử tốt nhất của A gán vào x: x= Chọn(A);
  + Cập nhật các đối tượng để chọn: A = A – {x};
  + Nếu S∪{x} thỏa mãn yêu cầu bài toán thì
    - Cập nhật lời giải: S = S∪{x};

Thủ tục thuật toán tham lam có thể cài đặt như sau:

Input A[1..n]

Output S//lời giải

Greedy (A,n) ≡ S = ∅;

while ( A ≠ ∅)

{

x = Chọn(A);

A = A-{x};

If(S∪{x}chấp nhận được)

S = S∪{x};

}

Return S;

1. **THUẬT TOÁN PRIM – TÌM CÂY BAO TRÙM NHỎ NHẤT**
2. **Bài toán**

G = (V,E) là đơn đồ thị vô hướng lien thông, có trọng số.

V = {1,..,n} là tập các đỉnh. E là tập các cạnh (edge).

Một cây T gọi là cây bao trùm của G nếu T là đồ thị con của G và chứa mọi đỉnh của G.

Vấn đề là tìm cây bao trùm có trọng số nhỏ nhất: MST (Minimal Spanning Tree) của G.

Các thuật toán cơ bản giải bài toán trên là các thuật toán Prim và Ksuskal. Trong phần này, ta giới thiệu thuật toán Prim.

1. **Ý tưởng**

Thuật toán Prim xây dựng một đồ thị con T của G như sau:

* Đầu tiên chọn tùy ý 1 đỉnh của G đặt vào T.
* Quá trình sau còn thực hiện trong khi T chưa chứa hết các đỉnh của G:
* Mỗi bước, tìm một cạnh có trọng số nhỏ nhất nối 1 đỉnh trong T với 1 đỉnh ngoài T. Thêm cạnh này vào T.
* Kết thúc thuật toán Prim cho ta một cây bao trùm có trọng số nhỏ nhất (MST) của đồ thị G.

Tính tham lam của thuật toán Prim là tại mỗi bước them vào T một cạnh cá trọng số nhỏ nhất nối một đỉnh trong T và một đỉnh ngoài T.

1. **Mô tả thuật toán**

Input G = (V,E)

Output T =(V,.) là MST của G.

Mô tả:

* Gọi U là tập con của V.
* Khởi động T = (U,.) = ∅; //đồ thị con rỗng.
* Khởi động U = {1}; //chọn đỉnh 1 đặt vào T.
* Trong khi U ≠ V

Tìm cạnh (u,v) có trọng số nhỏ nhất, với u ∈ U và v ∉ U. Thêm đỉnh v này vào U. Thêm (u,v ) vào T.

Lời giải của bài toán là lời giải tối ưu.

*Minh họa:*

Xét đồ thị sau:

3

6

2

8

10

2

1

5

4

6

5

3

4

1

2

Áp dụng thuật toán Prim, bắt đầu tứ đỉnh 1, ta xây dựng được 1 MST của đồ thì trên:

5

4

2

2

6

1

1

3

5

2

3

Hoạt động của thuật toán:

|  |  |  |
| --- | --- | --- |
| Bước | Các cạnh chọn | Tập U |
| 0 | - | {1} |
| 1 | (3,1) | {1,3} |
| 2 | (2,3) | {1,3,2} |
| 3 | (4,2) | {1,3,2,4} |
| 4 | (5,4) | {1,3,2,4,5} |
| 5 | (6,5) | {1,3,2,4,5,6} |

**Độ phức tạp của thuật toán:**

*Độ phức tạp của thuật toán theo thực nghiệm:*