

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KHOA HỌC MÁY TÍNH

**PHƯƠNG PHÁP THIẾT KẾ THUẬT TOÁN GẦN ĐÚNG
BÀI TẬP NHÓM 4**

Sinh viên:

NGUYỄN HUY PHƯỚC - 23521234

PHAN NHẬT TÂN - 23521405

Giảng viên:

Nguyễn Thanh Sơn



UIT

TP. HỒ CHÍ MINH, năm 2024

Mục lục

1 Bài 1: Set cover	2
1.1 Đề bài	2
1.2 Hướng giải quyết gần đúng	2
1.2.1 Ngây thơ	2
1.2.2 Cải tiến	2
1.3 Nhận xét:	3
2 Bài 2: Traveling Salesman Problem	3
2.1 Đề bài	3
2.2 Hướng giải quyết gần đúng	3
2.2.1 Tham lam	3
2.2.2 Cải tiến	4
2.3 Nhận xét	4
3 Đường dẫn cài đặt các thuật toán	4
3.1 Bài 1	4
3.2 Bài 2	4

1 Bài 1: Set cover

1.1 Đề bài

Cho:

$$U = \{u_1, u_2, \dots, u_n\}, \quad S = \{S_1, S_2, \dots, S_m\} \quad \text{với} \quad S_i \subseteq U \quad \text{cho mọi} \quad i.$$

Mục tiêu là tìm một tập con các tập con $S' \subseteq S$ sao cho:

$$\bigcup_{S_i \in S'} S_i = U$$

và tối thiểu hóa số lượng các tập con được chọn, tức là:

$$\min |S'|.$$

1.2 Hướng giải quyết gần đúng

1.2.1 Ngây thơ

- Sắp xếp lại tập S giảm dần theo kích thước của mỗi tập
- Lấy các tập lần lượt từ các tập từ trái sang phải
- Kết thúc thuật toán khi đã phủ đủ n phần tử.

Giải pháp có thời gian thực thi khá nhanh nhưng độ hiệu quả tỉ lệ nghịch với độ lớn của n và độ lớn của tập S .

Yếu điểm: gặp hiện tượng bốc phải các tập mà toàn bộ phần tử bên trong đều đã bị bao phủ bởi các tập trước đó.

1.2.2 Cải tiến

Duy trì một bảng điểm $score_i$ là số lượng phần tử trong tập mà chưa bị các tập được chọn bao phủ

- Chọn tập có score lớn nhất cho vào đáp án.
- cập nhật lại score của các tập chưa được chọn.
- Duy trì cho đến khi không còn tập để chọn hoặc n phần tử đều bị phủ

Giải pháp có thời gian thực thi vẫn đủ tốt không thua nhiều so với ngây thơ (do có sự cải tiến để cập nhật score nhanh hơn).

Tránh được hiện tượng gặp phải ở thuật toán ngây thơ và đạt độ hiệu quả tốt hơn.

1.3 Nhận xét:

Sau đây là bảng thống kê điểm số chênh lệch (so sánh độ chính xác của 2 giải thuật trên với giải thuật hiện đại rất mạnh trong bài toán set cover là giải thuật di truyền với bộ test cá nhân tự tạo).

Thuật toán	Tỉ lệ sắp xỉ
Ngây thơ	0.000214025
Cải tiến	0.2173913043

Số liệu cho thấy sự cải tiến rõ rệt trong thuật toán cải tiến. Thuật toán ngây thơ nhìn chung vẫn có lợi thế về tốc độ thực thi nhưng bù lại sẽ không mang lại độ hiệu quả kết quả không chấp nhận được. Khi chỉnh sửa thành cải tiến có vẻ mọi thứ đã tốt hơn khi giá trị chỉ thua 5 lần so với kết quả tối ưu.

2 Bài 2: Traveling Salesman Problem

2.1 Đề bài

Cho một đồ thị $G = (V, E)$, trong đó:

- $V = \{v_1, v_2, \dots, v_n\}$ là tập hợp các đỉnh, mỗi đỉnh tương ứng với một thành phố.
- E là tập hợp các cạnh nối các đỉnh.
- Mỗi cạnh $(v_i, v_j) \in E$ có một trọng số $c(v_i, v_j)$, có thể là chi phí, thời gian hoặc khoảng cách di chuyển giữa các thành phố v_i và v_j .

Mục tiêu là tìm một chu trình Hamiltonian (một chuỗi các đỉnh mà bắt đầu từ một thành phố và quay lại thành phố ban đầu) sao cho tổng trọng số của các cạnh trong chu trình này là nhỏ nhất.

2.2 Hướng giải quyết gần đúng

2.2.1 Tham lam

Các bước của giải pháp tham lam:

1. Khởi tạo: Chọn một đỉnh bất kỳ làm điểm bắt đầu của chu trình.
2. Lặp lại các bước sau cho đến khi tất cả các đỉnh đều được chọn:
 - (a) Chọn đỉnh kề có cạnh có trọng số nhỏ nhất mà chưa được chọn vào chu trình.
 - (b) Thêm đỉnh vừa chọn vào chu trình.
3. Khi tất cả các đỉnh đã được chọn, quay lại đỉnh bắt đầu để hoàn thành chu trình.

Kết quả: Chu trình Hamiltonian với tổng trọng số nhỏ nhất.

2.2.2 Cải tiến

Thực hiện giải thuật tham lam như trên để tạo ra một khởi đầu lộ trình tốt.

Thực hiện giải thuật local search:

- Duyệt từng cặp đỉnh, swap hai đỉnh trong lộ trình
- Tính lại trọng số, nếu không tốt thì swap hai đỉnh về lại như cũ
- lặp lại đến khi không còn 2 cặp tốt hơn.

2.3 Nhận xét

Tham lam cho thời gian nhanh hơn đáng kể so với cải tiến tuy nhiên về kết quả tối ưu thì thuật cải tiến vẫn tốt hơn nhiều do rằng khởi đầu của cải tiến chính là tham lam và có sự chỉnh sửa trên chúng điều này có thể giúp ta nhanh đạt tới các điểm đến tốt hơn.

3 Đường dẫn cài đặt các thuật toán

3.1 Bài 1

- Thuật toán ngẫu nhiên
- Thuật toán cải tiến

3.2 Bài 2

- Thuật toán tham lam
- Thuật toán cải tiến