

THUẬT TOÁN TABU SEARCH VÀ TRAVELLING SALESMAN PROBLEM

TỐI ƯU RỜI RẠC MAT3323

Giảng viên: Hoàng Nam Dũng

Nhóm 5

Mai Thanh Tùng K66A2

Nguyễn Long Vũ K66A2

Phạm Ngọc Uy K66A2

Hà Nội, 2023

1. Giới thiệu thuật toán Tabu Search
 - a. Local Search
 - b. Các thành phần của thuật toán Tabu Search
 - c. Pseudo-code
 - d. Ưu và nhược điểm
2. Bài toán người bán hàng du lịch (Travelling Salesman Problem - TSP)
 - a. Giới thiệu bài toán
 - b. Code
 - c. Độ phức tạp tính toán
 - d. Kết luận

1. Giới thiệu thuật toán Tabu Search



Fred W. Glover

- Tabu Search (TS) được đề xuất bởi Fred Glover vào năm 1988.
- Tabu Search được giới thiệu như một trong những phương pháp tìm kiếm cục bộ hiệu quả nhất cho việc giải quyết các bài toán tối ưu hóa tổ hợp trong nhiều lĩnh vực như bài toán người du lịch (TSP).
- Trong Tabu Search mỗi bước di chuyển được ghi lại để tránh quay lại các giải pháp đã thăm và coi chúng là "tabu" (không được sử dụng).

1. Giới thiệu thuật toán Tabu Search

a. Local Search

b. Các thành phần của thuật toán Tabu Search

c. Pseudo-code

d. Ưu và nhược điểm

2. Bài toán người bán hàng du lịch (Travelling Salesman Problem - TSP)

a. Giới thiệu bài toán

b. Code

c. Độ phức tạp tính toán

d. Kết luận

1. Giới thiệu thuật toán Tabu Search

a. Local Search

Phương pháp tìm kiếm cục bộ (Local search) bắt đầu với một giải pháp tiềm năng và kiểm tra khu vực láng giềng của nó để tìm một giải pháp láng giềng tốt hơn. Sau đó thực hiện điều tương tự với giải pháp láng giềng mới đó. Quá trình này được lặp lại cho đến khi chúng ta tìm thấy giải pháp tối ưu hoặc đáp ứng điều kiện dừng được xác định trước bởi người dùng.

1. Giới thiệu thuật toán Tabu Search

a. Local Search

b. Các thành phần của thuật toán Tabu Search

c. Pseudo-code

d. Ưu và nhược điểm

2. Bài toán người bán hàng du lịch (Travelling Salesman Problem - TSP)

a. Giới thiệu bài toán

b. Code

c. Độ phức tạp tính toán

d. Kết luận

1. Giới thiệu thuật toán Tabu Search

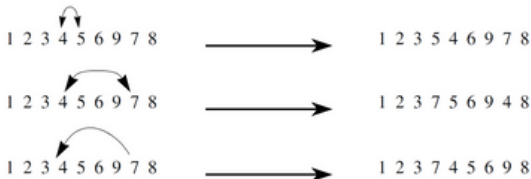
b. Các thành phần của thuật toán Tabu Search

- **Short term memory:** Loại bộ nhớ có hạn đối với thời gian và dung lượng lưu trữ. Trong thuật toán tìm kiếm cục bộ (TS), danh sách tabu có thể coi là một bộ nhớ ngắn hạn.
- **Long term memory:** Bộ nhớ này giữ lại các đặc điểm của các giải pháp tốt hơn, sẽ được sử dụng trong:
 - Tăng cường: ưu tiên đặc điểm của một nhóm các giải pháp triển vọng hơn.
 - Đa dạng hóa: làm giảm thiểu đặc điểm của các giải pháp đã được chọn để đa dạng hóa tìm kiếm đến các khu vực khác của không gian tìm kiếm.
- **Danh sách Tabu:** Để ngăn chặn việc quay lại các giải pháp đã thấy trước đó, TS sử dụng một danh sách tabu, trong đó liệt kê các bước cấm hoặc đặc điểm.
- **Moves:** Di chuyển từ một giải pháp ban đầu đến các giải pháp hàng xóm, tìm giải pháp tốt hơn và lặp lại đến khi dừng.

1. Giới thiệu thuật toán Tabu Search

b. Các thành phần của thuật toán Tabu Search

- **Tiêu chí khát vọng (aspiration criteria):** Trong tình huống mà một bước di chuyển bị cấm có một đánh giá đủ tốt, quy tắc cho phép sự ngoại lệ này tồn tại.
- **Điều kiện dừng:**
 - Số lần lặp cố định đã cho.
 - Một khoảng thời gian xử lý cụ thể đã cho.
 - Không có bước di chuyển khả thi nào vào khu vực lân cận của giải pháp hiện tại đã tìm thấy.
- **Neighborhood:** Lân cận $N(s, k)$ của nghiệm s trong lần lặp thứ k là tập hợp các nghiệm liền kề với s trong không gian tìm kiếm S trong lần lặp k đó.



1. Giới thiệu thuật toán Tabu Search

a. Local Search

b. Các thành phần của thuật toán Tabu Search

c. Pseudo-code

d. Ưu và nhược điểm

2. Bài toán người bán hàng du lịch (Travelling Salesman Problem - TSP)

a. Giới thiệu bài toán

b. Code

c. Độ phức tạp tính toán

d. Kết luận

1. Giới thiệu thuật toán Tabu Search

b. Pseudo code

Algorithm 1: Tabu Search

Data: S - the search space, $maxIter$ - the maximal number of iterations, f - the objective function, the definition of neighborhoods, and the aspiration criteria.

Result: the best solution found

Choose the initial candidate solution $s \in S$

$s^* \leftarrow s$ // Initialize the best-so-far solution.

$k \leftarrow 1$

while $k \leq MaxIter$ **do**

 /* Sample the allowed neighbors of s */

 Generate a sample $V(s, k)$ of the allowed solutions in $N(s, k)$

 // $s' \in V(s, k) \iff (s' \notin T) \vee (a(k, s') = true)$

 Set s to the best solution in $V(s, k)$

 /* Update the best-so-far if necessary */

if $f(s) < f(s^*)$ **then**

 | $s^* \leftarrow s$

end

 Update T and a

 /* Start another iteration */

$k \leftarrow k + 1$

end

1. Giới thiệu thuật toán Tabu Search

- a. Local Search
- b. Các thành phần của thuật toán Tabu Search
- c. Pseudo-code
- d. Ưu và nhược điểm

2. Bài toán người bán hàng du lịch (Travelling Salesman Problem - TSP)

- a. Giới thiệu bài toán
- b. Code
- c. Độ phức tạp tính toán
- d. Kết luận

1. Giới thiệu thuật toán Tabu Search

d. Ưu điểm và nhược điểm

- Ưu điểm:
 - Có thể áp dụng cho cả các giải pháp rời rạc và liên tục.
 - Danh sách Tabu có thể được sử dụng để chống lại các chu kỳ và quay lại các giải pháp cũ.
 - Tabu Search cải thiện hiệu suất của một phương pháp tìm kiếm cục bộ bằng cách sử dụng các cấu trúc bộ nhớ mô tả các giải pháp đã truy cập.
- Nhược điểm:
 - Độ phức tạp của tham số:
 - Kích thước tabu list.
 - Thời gian giữa các cập nhật, và các tham số khác.
 - Hiệu suất phụ thuộc vào lựa chọn các bước di chuyển (Moves)
 - Không đảm bảo tìm kiếm toàn cầu (No Global Optimality Guarantee)

1. Giới thiệu thuật toán Tabu Search

- a. Local Search
- b. Các thành phần của thuật toán Tabu Search
- c. Pseudo-code
- d. Ưu và nhược điểm

2. Bài toán người bán hàng du lịch (Travelling Salesman Problem - TSP)

- a. Giới thiệu bài toán
- b. Code
- c. Độ phức tạp tính toán
- d. Kết luận

1. Giới thiệu thuật toán Tabu Search

- a. Local Search
- b. Các thành phần của thuật toán Tabu Search
- c. Pseudo-code
- d. Ưu và nhược điểm

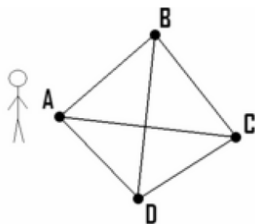
2. Bài toán người bán hàng du lịch (Travelling Salesman Problem - TSP)

- a. Giới thiệu bài toán
- b. Code
- c. Độ phức tạp tính toán
- d. Kết luận

2. Travelling Salesman Problem

b. Giới thiệu bài toán:

- Phát biểu: “Cho một danh sách các thành phố và khoảng cách giữa mỗi cặp thành phố, con đường ngắn nhất có thể đi qua mỗi thành phố đúng một lần và trở về thành phố gốc là gì?”
- Bài toán travelling salesman là một bài toán NP-hard (Non Polynomial Hard)



Phát biểu dưới dạng đồ thị

Cho đồ thị đầy đủ n đỉnh vô hướng, có trọng số $G = (V, E)$. Tìm chu trình $v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_n \rightarrow v_1$ với $v_i \in V, i = \overline{1, n}$ sao cho tổng trọng số hành trình trên các cạnh (v_i, v_{i+1}) và (v_n, v_1) là nhỏ nhất.

\Rightarrow Có thể coi mục tiêu của bài toán là tìm chu trình Hamilton có độ dài ngắn nhất của đồ thị trên.

2. Travelling Salesman Problem

b. Giới thiệu bài toán:

Bộ dữ liệu chuẩn: <http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/tsp/>

```
NAME : att48
COMMENT : 48 capitals of the US (Padberg/Rinaldi)
TYPE : TSP
DIMENSION : 48
EDGE_WEIGHT_TYPE : ATT
NODE_COORD_SECTION
1 6734 1453
2 2233 10
3 5530 1424
4 401 841
5 3082 1644
.....
```

1. Giới thiệu thuật toán Tabu Search

- a. Local Search
- b. Các thành phần của thuật toán Tabu Search
- c. Pseudo-code
- d. Ưu và nhược điểm

2. Bài toán người bán hàng du lịch (Travelling Salesman Problem - TSP)

- a. Giới thiệu bài toán
- b. Code
- c. Độ phức tạp tính toán
- d. Kết luận

1. Giới thiệu thuật toán Tabu Search

- a. Local Search
- b. Các thành phần của thuật toán Tabu Search
- c. Pseudo-code
- d. Ưu và nhược điểm

2. Bài toán người bán hàng du lịch (Travelling Salesman Problem - TSP)

- a. Giới thiệu bài toán
- b. Code
- c. Độ phức tạp tính toán
- d. Kết luận

2. Travelling Salesman Problem

Độ phức tạp tính toán của phương thức này phụ thuộc vào một số tham số như:

- **maxIterations**: số vòng lặp tối đa của thuật toán. (maxITER)
- **cities**: số thành phố trong bài toán TSP. (n)
- **currentTabuSize**: kích thước của danh sách tabu. (tabuSize)
- **neighbors**: số thành phố lân cận được tạo qua mỗi lần lặp. $n \cdot (n-1) / 2$

Worst Case: $O(\text{maxITER} * n^3)$

1. <http://i-rep.emu.edu.tr:8080/xmlui/bitstream/handle/11129/2933/alizeran.pdf>
2. https://en.wikipedia.org/wiki/Tabu_search
3. <https://www.baeldung.com/cs/tabu-search>
4. Johann Dréo, Alain Pétrowski, Patrick Siarry, Eric Taillard, A. Chatterjee - Metaheuristics for Hard Optimization Simulated Annealing Tabu Search E-Spr
5. Edited by_ Wassim Jaziri - Tabu Search

CẢM ƠN THẦY VÀ CÁC BẠN ĐÃ LẮNG NGHE