

THUẬT TOÁN TÌM KIẾM LEO ĐỒI

NHÓM 2

1. Phạm Minh Châu
2. Thân Bửu Châu
3. Nguyễn Mạnh Cường
4. Nguyễn Ngô Cường



01 | Nguồn gốc và Hoàn
cảnh ra đời

02 | Tìm hiểu mã giả
của thuật toán

03 | Ví dụ, thuật toán

04 | Ưu - nhược điểm và
vấn đề phát sinh

05 | Ứng dụng thực tế
của thuật toán

Nguồn gốc và Hoàn cảnh ra đời

01

Lịch sử

Năm ra đời : 1983

- Các thuật toán leo đồi và độ dốc đã có ở trong văn hóa dân gian khoa học máy tính từ lâu
- Giải thuật này được đưa ra bởi Stefanie Gunther và được J.W.L. Glaisher hoàn chỉnh.

Nguồn gốc



**Là một kỹ thuật tối ưu toán học :
thuộc họ tìm kiếm cục bộ**

**Là một thuật toán lặp lại bắt đầu
với một giải pháp tùy ý**

Tìm hiểu mã
giả của
thuật toán

02

Giải thuật

Xét Trạng Thái ban đầu:

- Nếu là đích : DỪNG lại;

- Ngược lại :gán trạng thái ban đầu = trạng thái hiện tại

- Vòng Lặp: đến khi gặp đích hoặc không còn luật nào chưa được áp dụng trên trạng thái hiện tại.

- Lựa chọn 1 luật để áp dụng vào TT Hiện tại để sinh ra TT mới

- Xét TT mới

- Nếu là đích : dừng lại

- Nếu không là đích nhưng tốt hơn TT hiện tại thì TT Mới = TT Hiện tại

- Nếu không tốt hơn thì tiếp tục lặp

Mã giả

function HILL-CLIMBING(*problem*) returns a state that is a local maximum

inputs: *problem*, a problem

local variables: *current*, a node

neighbor, a node

current \leftarrow MAKE-NODE(INITIAL-STATE[*problem*])

loop do

neighbor \leftarrow a highest-valued successor of *current*

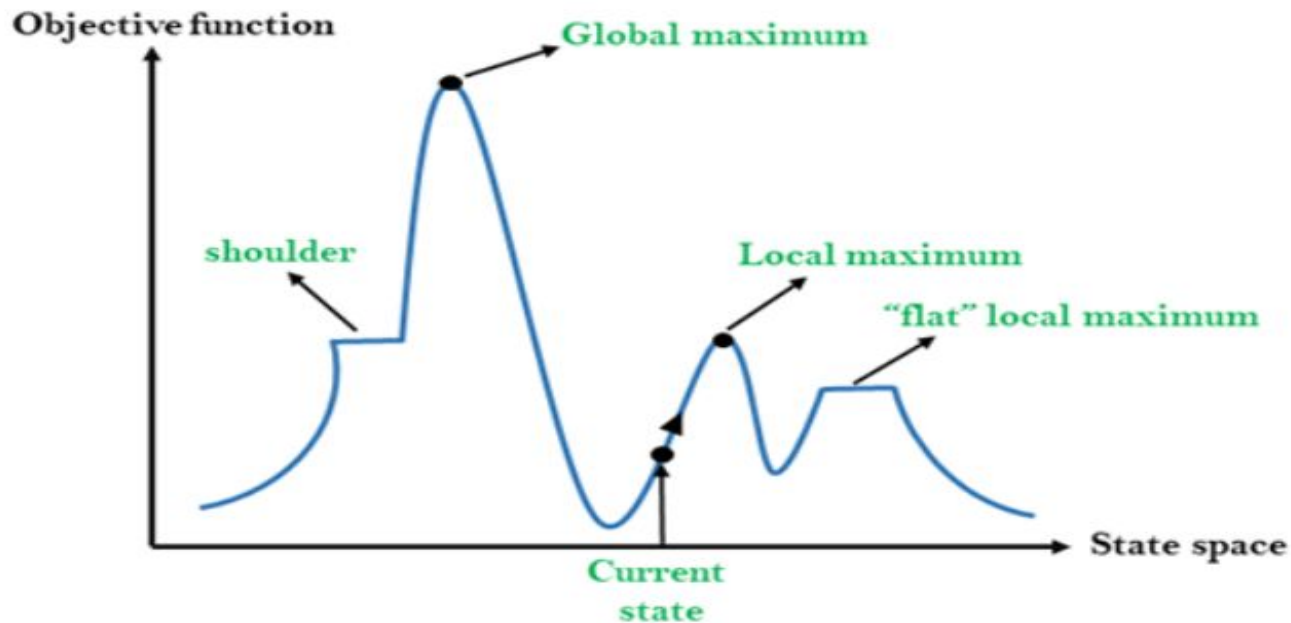
if VALUE[*neighbor*] \leq VALUE[*current*] **then return** STATE[*current*]

current \leftarrow *neighbor*

Ví dụ và giải
thuật toán

03

Sơ đồ không gian trạng thái Hill - Climb



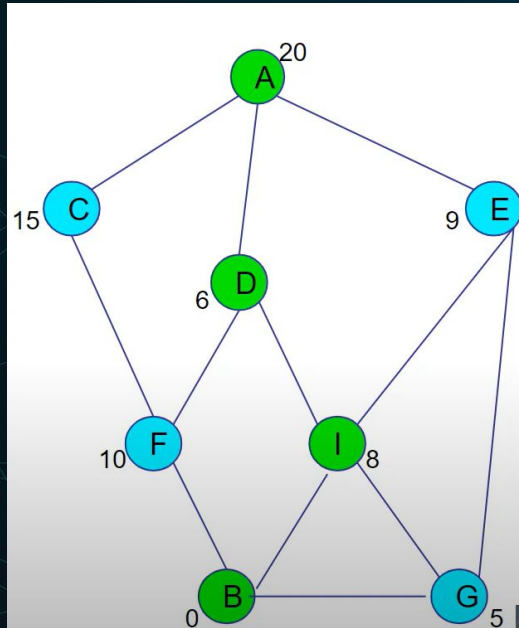
Hill-climbing search: 8-queens problem

18	12	14	13	13	12	14	14
14	16	13	15	12	14	12	16
14	12	18	13	15	12	14	14
15	14	14	♚	13	16	13	16
♚	14	17	15	♚	14	16	16
17	♚	16	18	15	♚	15	♚
18	14	♚	15	15	14	♚	16
14	14	13	17	12	14	12	18

Các bước giải thuật Hill-Climb:

- Bước 1: Đánh giá trạng thái ban đầu, nếu là trạng thái mục tiêu thì trả về thành công và Dừng lại.
- Bước 2: Vòng lặp Cho đến khi tìm ra giải pháp hoặc không còn người vận hành mới để áp dụng.
- Bước 3: Chọn và áp dụng một toán tử cho trạng thái hiện tại.
- Bước 4: Kiểm tra trạng thái mới:
 - Nếu đó là trạng thái mục tiêu, sau đó trả lại thành công và bỏ thuốc lá.
 - Khác nếu nó tốt hơn trạng thái hiện tại thì gán trạng thái mới làm trạng thái hiện tại.
 - Nếu không, nếu không tốt hơn trạng thái hiện tại, sau đó quay lại bước 2.
- Bước 5: Thoát.

Ví dụ : Sử dụng thuật toán tìm đường đi ngắn nhất từ A->B



Lặp	u	kề	L	Open
1				A^{20}
2	A^{20}	C^{15}, D^6, E^9	D^6, E^9, C^{15}	D^6, E^9, C^{15}
3	D^6	F^{10}, I^8	I^8, F^{10}	I^8, F^{10}, E^9, C^{15}
4	I^8	B^0, G^5	B^0, G^5	$B^0, G^5, F^{10}, E^9, C^{15}$
5	B^0			G^5, F^{10}, E^9, C^{15}

Ưu - nhược điểm và vấn đề phát sinh

04

Các vấn đề của thuật toán

Tối ưu toàn cục



Chóp nhọn



Cao nguyên

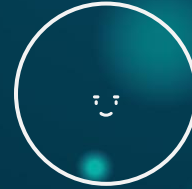


Ưu - Nhược điểm



Ưu điểm

Nhanh, đơn giản, hiệu quả trong không gian tìm kiếm nhỏ, ít lỗi lẫm.

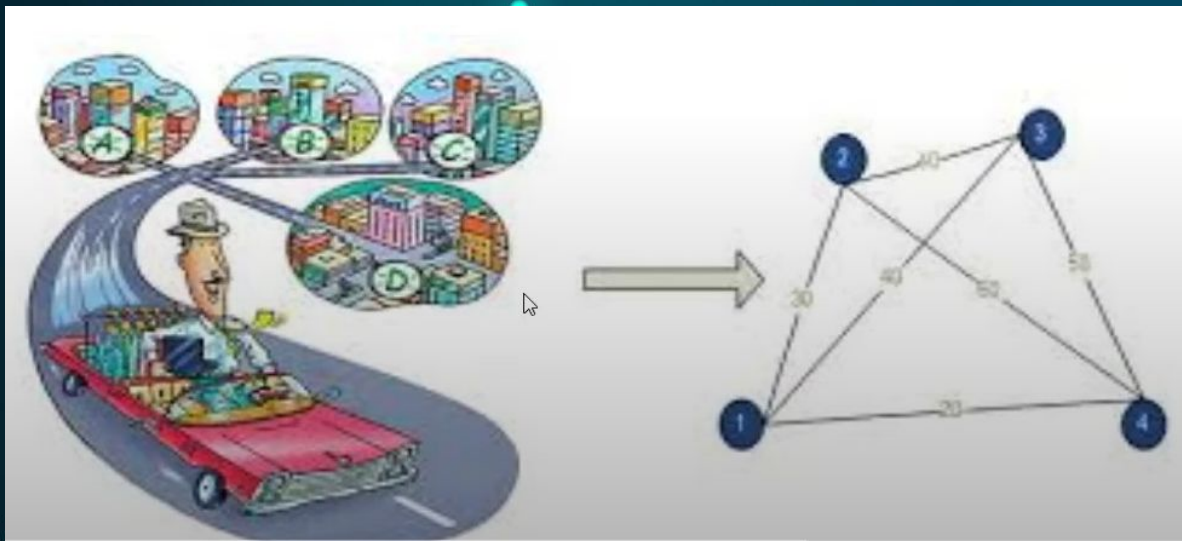


Nhược điểm

Khó tìm thấy trạng thái đích nếu như không gian tìm kiếm lớn, có nhiều điểm tối ưu cục bộ.

Ứng dụng thực tế
của thuật toán

05



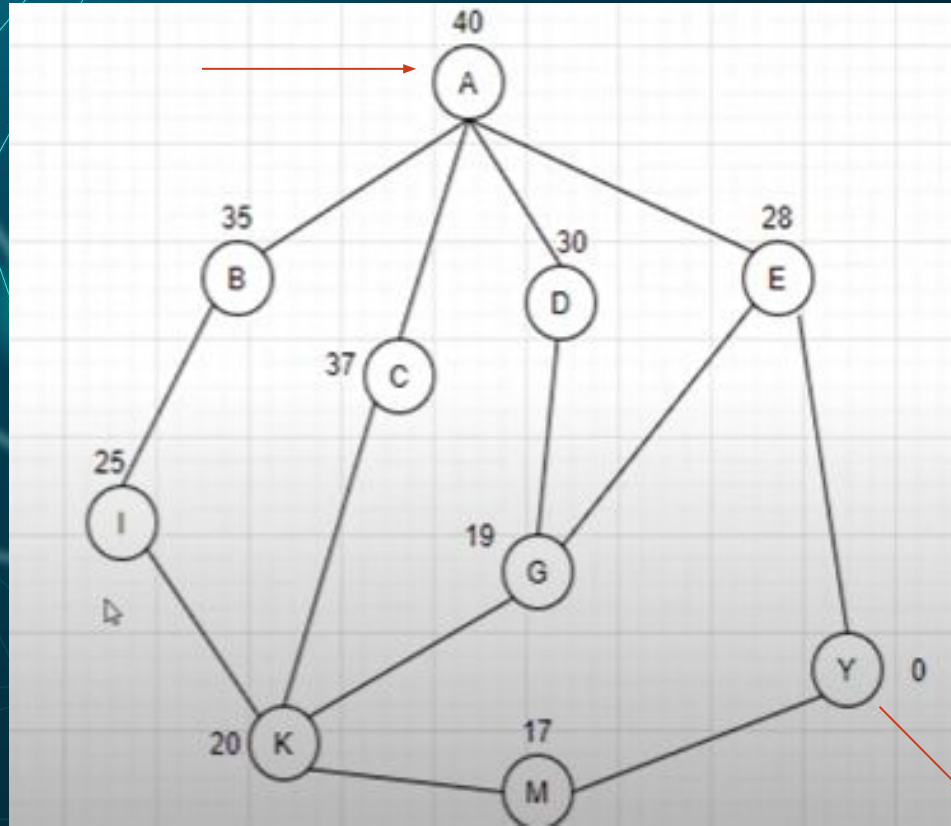
Hill Climb - Leo đồi có thể được áp dụng cho bài toán nhân viên bán hàng du lịch . Có thể dễ dàng tìm thấy một giải pháp ban đầu khi thăm tất cả các thành phố nhưng có thể sẽ rất kém so với giải pháp tối ưu. Thuật toán bắt đầu với một giải pháp như vậy và thực hiện các cải tiến nhỏ cho nó, chẳng hạn như chuyển đổi thứ tự mà hai thành phố được ghé thăm. Cuối cùng, một lộ trình ngắn hơn nhiều có thể đạt được.

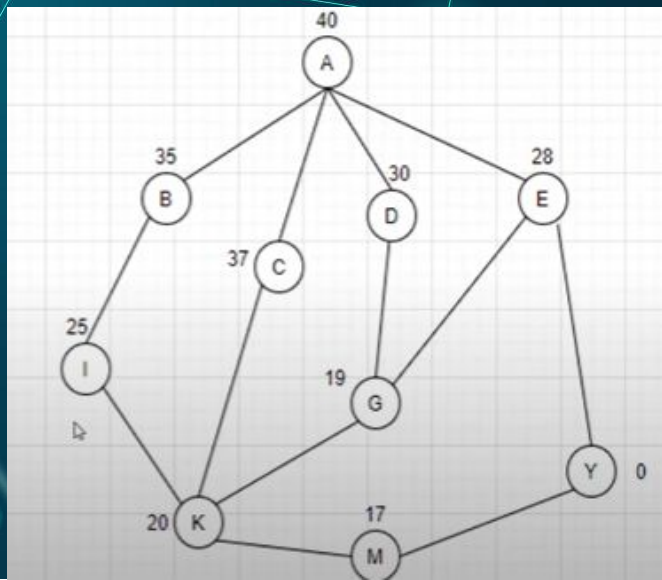
Hill Climbing cũng từng được sử dụng trong lĩnh vực robot để quản lý nhiều nhóm robot. Người ta cũng sử dụng thuật toán Leo đồi thiết kế mạch tích hợp các phương pháp toán học quy nạp cho robot để chúng quyết định khi nào làm việc một mình, khi nào làm việc nhóm.



KẾT LUẬN

- Hill Climbing Algorithm là một dạng thuật toán tìm kiếm được đưa ra xoay quanh mục đích tối ưu hóa những bài toán khó trong lĩnh vực trí tuệ nhân tạo.
- Thuật toán này luôn cố gắng tìm kiếm ra phương pháp giải quyết bài toán nhanh nhất, tối ưu hóa nhất có thể.





Lặp	U	kề U	L	open
1				A^{40}
2	A^{40}	$B^{35}, C^{37}, D^{30}, E^{28}$	$E^{28}, D^{30}, B^{35}, C^{37}$	$E^{28}, D^{30}, B^{35}, C^{37}$
3	E^{28}	G^{19}, Y^0	Y^0, G^{19}	$Y^0, G^{19}, D^{30}, B^{35}, C^{37}$
4	Y^0			$G^{19}, D^{30}, B^{35}, C^{37}$

THANKS FOR LISTENING !