**BÁO CÁO ĐỒ ÁN 8 PUZZLE**

Sinh viên thực hiện:

**Võ Phú Đức 18110101**

**Hà Đức Anh 18110075**

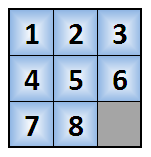
*Trường đại học Sư phạm Kỹ thuật TP.HCM*

**TÓM TẮT**

Hiện nay , sau những giờ học tập vất vả , cùng với nhu cầu giải trí của giới trẻ hiện nay , trò chơi điện tử là một phần không thể thiếu *.*Nhóm chúng em quyết định làm ra một game đơn giản và quen thuộc để đáp ứng nhu cầu đó*.*Cấu trúc của game dựa trên 3 thuật toán Hill climbing , Hill climbing with random restart , simulated annealing*.* Sau khi nghiên cứu nhiều tài liệu cũng như kết hợp những kiến thức mà thầy Huỳnh Xuân Phụng đã giảng,chúng em đã làm ra được một chương trình hoàn thiện cũng như đúng yêu cầu của đồ án trí tuệ nhân tạo; cụ thể đã tạo ra được tập dữ liệu có số lượng lớn trạng thái ban đầu của game 8-puzzle, 1 trạng thái đích. Giải game bằng giải thuật hill climbing (steepest-ascent and first-choice variant), hill climbing with random restart, simulated annealing, tính được chi phí tìm kiếm (số bước giải), số trường hợp có thể tìm được lời giải và so sánh 3 thuật toán này.

1. **Giới thiệu**

* Bài toán (hay game) **n-puzzle** có lẽ rất quen thuộc với chúng ta cũng như những người mới bắt đầu tiếp cận với môn trí tuệ nhân tạo. Nó được biết đến với nhiều phiên bản và tên gọi khác nhau như 8-puzzle, 15-puzzle, Gem puzzle, Boss puzzle, Game of Fifteen, Mystic Square
* Bài toán gồm một bảng 3×3 với các ô số được đánh từ 1->8 và một ô trống. Ở trạng thái bắt đầu, các ô được sắp đặt ngẫu nhiên, và nhiệm vụ của người giải là tìm cách đưa chúng về đúng thứ tự kiểu lợp ngói (lớn dần từ trái qua phải và từ trên xuống)



**Hình 1.1**

**II . Định nghĩa các giải thuật có trong bài**

1. **Hill Climbing [1]**

Phương pháp tìm kiếm leo đồi chú trọng tìm hướng đi dễ dẫn đến trạng thái đích nhất. Cách đó được đưa ra nhằm làm giảm công sức tìm kiếm. Thuật toán tìm kiếm leo đồi thực chất là thuật toán tìm kiếm theo chiều sâu, song tại mỗi bước ta sẽ ưu tiên chọn một trạng thái có hứa hẹn nhanh tới đich nhất để phát triển trước. Vấn đề quan trọng là biết khai thác kheo léo thông tin phản hồi để xác định hướng đi tiếp và đẩy nhanh quá trình tìm kiếm. Thông thường ta gán mỗi trạng thái của bài toán với một số đo (hàm đánh giá) nào đó nhằm đánh giá mức độ gần đích của nó. Điều đó có nghĩa là nếu trạng thái hiện thời là u thì trạng thái v sẽ được phát triển tiếp theo nếu v kề với u và hàm đanh giá của v đạt giá trị max (hoặc min).

**Thuật toán Hill Climbing được sử dụng trong bài**

procedure HillClimbing

begin

kiểm tra trạng thái đã đạt đích ;

Nếu đạt đích trả về giá trị là 0;

Gán current= trạng thái hiẹn tại;

for mỗi trạng thai so với đích nhưng giới hạn bước leo đồi là 2500 bước là dừng;

Trạng thái xung quang được chỉ định với cost nhỏ hơn so với trạng thái đích;

nếu giá trị trạng thái tiếp theo > hiện tại ;

trả về giá trị trạng thái là tiếp theo ;

end;

1. **Hilll Climbing with random restart [2]**

Đây là một cách khác để giải quyết vấn đề liên quan đến việc local maximum cứ lặp đi lặp lại.Random restart hill climbing tiến hành thuật Hill Climbing có , ngẫu nhiên tạo ra các trạng thái ban đầu, chạy từng trạng thái cho đến khi nó dừng lại hoặc không có tiến bộ rõ rệt.Tiến trình sẽ cho phép so sánh mọi kết quả đạt được để chọn ra vị trị tốt nhất.

procedure RandomRestart\_HillClimbing

begin;

Tạo một Mảng chứa : Temp

loop do

Nếu ko phải trạng thái đích;

Random giá trị 10 lần;

For mỗi trạng thái được random ngẫu nhiên;

ngược lại nếu radom không đạt được đích;

Trờ lại bài toán leo đồi;

Hành động này diễn ra trong khi cost vẫn khác 0 nhưng có giới hạn số lần di chuyển sẽ bé hơn 1000;

trả về Temp;

end;

Hình 2.2

1. **Sitmutaled Annealing (SA)**

SA là một thuật toán tìm kiếm xác suất di truyền, là phương pháp tối ưu hoá có thể áp dụng để tìm kiếm tối ưu hoá toàn cục của hàm chi phí và tránh tối ưu hoá địa phương bằng việc chấp nhận một lời giải tồi hơn với một xác suất phụ thuộc nhiệt độ T

procedure Simulatedannealing

begin;

Khởi tạo trạnh thai ban đầu

Nung với nhiệt độ là 100 độ

giới hạn lầ di chuyển là 300

Nếu tt hiện tại = đích

trả về tt hiện tại

Ngược lại

giảm nhiệt độ mỗi lần leo xuống với chỉ số giảm là 0.96

Random Trạng thái khi hạ nhiệt xuống mỗi lần với cost< curent

Delta= Cost của trạng thái sau khi random - cost của hiện tại

Nếu delta >0

curent =next;

Ngược lại

Khởi tạo biến e tính xác suát để tìm tiếp đường xuống e=pow(exp,deltaE / Temperature)

random xác suất với giá trị (0.0, 1.0)

Nếu e> rand

current=next;

Trả về giá trị current

end;

**III . Báo Cáo**

**A.Chi phí tìm kiếm,số trường hợp có thể tìm được lời giải**

Nhóm chúng em cho mặc định ngẫu nhiên 20 trường hợp bắt đầu để giải thuật và đây là KQ:

* **Hill Climbing**

Hình 3.1

Step: chi phí tìm kiếm

Số trường hợp có thể giải được : 13

Thời gian trung bình : 0.001s

* **Hill climbing with random restart**

Hình 3.2

Step: chi phí tìm kiếm , số 1000 là do bị giới hạn khi thuật toán không giải được, để tránh bị lặp, tối ưu bộ nhớ

Số trường hợp có thể giải được : 17

Thời gian trung bình : 0.005s

* **Simulated annealing**

Hình 3.3

Step: chi phí tìm kiếm

Số trường hợp có thể giải được : 18

Thời gian trung bình : 0.001s

**B. So sánh 3 thuật toán**

|  |  |  |  |
| --- | --- | --- | --- |
| Base on 20 random states | Hill Climbing | HillClimbing with random restart | Simulated Annealing |
| State có thể giải | 13 | 17 | 18 |
| Avg time | 0.001350s | 0.033450s | 0.002250s |

Đây là số liệu do radom trạng thái ngẫu nhiên

1. **Kết Luận**

**Hill Climbing** : Đây là thuật toán yếu nhất , nhưng thời gian thực thi nhanh , cho kết quả không chính xác

**Hill Climbing with random restart**: Chậm hơn nhiều lần so với Hill Climbing ,nhưng lại cho kết quả khả quan hơn

**Simulated Annealing**:Đây là thuật toán toàn diện nhất ,cho kết quả nhiều nhất, thời gian tối ưu, như vậy ta có thể thấy SA là thuật toán tốt nhất

**IV. Đánh Giá**

Để hoàn thành bài báo cáo này trước hết nhóm chúng em xin gửi đến Thầy Huỳnh Xuân Phụng lời cảm ơn sâu sắc nhất vì đã tạo cơ hội và hướng dẫn nhóm trong quá trình thực hiện bài bào cáo này.

Nhìn chung thì nhóm chúng em đã hoàn thiện khá tốt và đầy đủ với yêu cầu của đề tài tuy còn thiếu sót ở vài điểm , biết cách vận dụng những kiên thức đã học trên lớp để áp dụng vào đồ án lớn .

Hiểu được lí thuyết , cách vận hành của mỗi thuật toán

* **ƯU ĐIỂM**

- Hoàn thành gần như yêu cầu của đồ án.

- Giao diện trực quan dễ so sánh.

Nhược điểm:

- Chưa tối ưu ở một số bước của thuật toán nên còn sai sót.

- Giao diện còn sơ sài và đơn giản.

**\*Hướng phát triển**

* Chúng em sẽ tiếp tục nghiên cứu để tìm ra cách tối ưu thuật toán tốt nhất để có thể giải được nhanh hơn và đạt được nhiều kết quả hơn.
* Cũng như có thể áp dụng được các thuật toán này lên nhiều đồ án sau này cũng như các dự án , phần mềm cần giải thuật đó.

**Tài liệu tham khảo**

[1] tài liệu về thuật toán hill climbing tác giả : thao ho ( 7.8.2019)

<https://thanhthao94blog.wordpress.com/2016/08/07/giai-thuat-leo-doi-hill-climbing/>

1. tài liệu về thuật toán Hill climbing restart (19 August 2018, at 14:28)

[https://en.wikibooks.org/wiki/Artificial\_Intelligence/Search/Iterative\_Improvement/Hill\_Climbing#:~:text=%E2%80%9CRandom%2Drestart%20hill%2Dclimbing,Russell%20%26%20Norvig%2C%202003).](https://en.wikibooks.org/wiki/Artificial_Intelligence/Search/Iterative_Improvement/Hill_Climbing" \l ":~:text=%E2%80%9CRandom%2Drestart%20hill%2Dclimbing,Russell%20%26%20Norvig%2C%202003).) Tác giả wk

[3]<https://www.sciencedirect.com/topics/engineering/simulated-annealing-algorithm>

Tác giả : Yavuz Eren (2017)

[4]<https://yinyangit.wordpress.com/2010/12/16/algorithm-phan-tich-va-gi%E1%BA%A3i-bai-toan-n-puzzle/>

Tác giả: YinYang