**HIỆN THỰC THUẬT TOÁN CHO BÀI TOÁN 8\_PUZZLE**

**Nguyễn Thị Như Quỳnh1, Nguyễn Hữu Tường1**

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

1. **Tóm tắt:**

Bài toán 8-puzzle có lẽ rất quen thuộc với chúng ta cũng như những người mới bắt đầu tiếp cận với môn trí tuệ nhân tạo. Bài toán gồm một bảng 3x3 với các ô được đánh số từ 1 đến 8 và một ô trống. Cho sẵn một trạng thái đích, ở trạng thái ban đầu các ô được sắp xếp ngẫu nhiên, nhiệm vụ là tìm cách đưa chúng về đúng thứ tự như đích đã cho. Câu hỏi đặt ra ở đây là thuật toán nào tối ưu nhất để giải bài toán 8-puzzle này. Thực thi 3 thuật toán local search: Hill-climbing, First-choice Hill-climbing, Simuated Annealing bằng ngôn ngữ C/C++ để giải bài toán 8-puzzle với tập các trạng thái ban đầu được sinh ngẫu nhiên, so sánh các kết quả: số trường hợp giải được, số trường hợp không giải được, chi phí tìm kiếm để đưa ra kết luận là thuật toán nào là tốt nhất.

1. **Nội dung:**
2. **Cơ sở lý thuyết:**
   1. **Thuật toán Hill Climbing:**

Thuật toán Hill-Climbing là một thuật toán “leo đồi” tìm kiếm trạng thái tốt hơn trạng thái hiện tại để mở rộng, Để biết trạng thái nào tốt hơn, thuật toán dùng một hàm H để xác định trạng thái nào là tốt nhất. Thuật toán phù hợp cho các bài toán tìm kiếm.

Mã giả thuật toán Hill-Climbing:

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

current MAKE-NODE(problem.INITIAL-STATE)

**loop do**

neighbor a highest-valued successor of current

**if** neighbor.VALUE ≤ current.VALUE **then return** current.STATE

current neighbor

*Giải thuật Hill-climbing*

* 1. **Thuật toán First-choice Hill-Climbing:**

Thuật toán First-choice Hill-Climbing là một biến thể của thuật toán Hill-climbing. Thuật toán này sinh ngẫu nhiên các hàm biến đổi trạng thái đến khi nào có trạng thái tốt hơn. Trong bài toán 8-puzzle thuật toán First-choice Hill-Climbing sinh ngẫu nhiên các hàm action: left, right, up, down.

Mã giả thuật toán First-choice Hill-climbing:

**function** FIRST-CHOICE-HILL-CLIMBING(problem) **returns** state that is better than current state

current MAKE-NODE(problem.INITIAL-STATE)

**loop do**

Neighbor a random-successor of current

**if** Neighbor.VALUE current.VALUE **then return** current.STATE

current Neighbor

*Giải thuật First-choice Hill-climbing*

* 1. **Thuật toán Simulated Annealing**

Thuật toán này chọn ngẫu nhiên trạng thái *v* nào đó trong vùng lân cận của trạng thái hiện tại *u*. Nếu trạng thái *v* tốt hơn trạng thái hiện tại, thì chuyển sang trạng thái đó. Ngược lại nếu trạng thái *v* không tốt hơn trạng thái hiện tại thì chỉ chuyển sang trạng thái *v* với một xác suất nào đó. Xác suất này giảm theo hàm mũ, phụ thuộc vào tham số T, T càng lớn khả năng chuyển sang trạng thái không tốt càng cao. Xác suất được xác định theo công thức:

e∆/T (1)

*với ∆ = cost(v)-cost(u)*

**function** SIMULATED-ANNEALING(problem, schedule) **returns** a solution state

**inputs:** problem, a problem

schedule, a mapping from time to “temperature”

current MAKE-NODE(problem.INITIAL-STATE)

**for** t = 1 **to do**

T schedule(t)

**if** T = 0 **then return** current

next a randomly selected successor of current

next.VALUE – current.VALUE

**if**  < 0 **then**  current next

**else** current next only with probability

*Giải thuật Simulated-Annealing*

1. **Đánh giá và so sánh thuật toán:**

Để thực thi và đánh giá các thuật toán một các đồng bộ. Ban đầu tạo dựng một tập lớn đầu vào bằng cách sinh ngẫu nhiên ngược từ trạng thái đích đã được cho sẵn. Sau một lần random và các thuật toán đều thực thi từ trạng thái ban đầu vừa được sinh. Theo bảng dưới đây, sẽ có 10 lần chạy chương trình, 1 lần chạy sẽ có 100 tập đầu vào được sinh mới ngẫu nhiên. Đầu ra là số trường hợp giải được, số trường hợp không giải được và số bước đi cho trường hợp giải không được tương ứng với mỗi thuật toán.

***Bảng 1.*** *Thống kê các thuật toán*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Hill-climbing | First-choice Hill-climbing | Simulated Annealing |
| Lần 1 | Số trường hợp giải được | 67 | 53 | 3 |
| Số trường hợp không giải được | 33 | 47 | 97 |
| Số bước giải | 77 | 140 | 151 |
| Lần 2 | Số trường hợp giải được | 50 | 31 | 3 |
| Số trường hợp không giải được | 50 | 69 | 97 |
| Số bước giải | 96 | 164 | 146 |
| Lần 3 | Số trường hợp giải được | 64 | 46 | 3 |
| Số trường hợp không giải được | 36 | 54 | 97 |
| Số bước giải | 68 | 124 | 141 |
| Lần 4 | Số trường hợp giải được | 58 | 55 | 1 |
| Số trường hợp không giải được | 42 | 45 | 99 |
| Số bước giải | 107 | 133 | 161 |
| Lần 5 | Số trường hợp giải được | 47 | 47 | 3 |
| Số trường hợp không giải được | 53 | 53 | 97 |
| Số bước giải | 134 | 138 | 162 |
| Lần 6 | Số trường hợp giải được | 59 | 63 | 3 |
| Số trường hợp không giải được | 41 | 37 | 97 |
| Số bước giải | 103 | 86 | 146 |
| Lần 7 | Số trường hợp giải được | 55 | 50 | 1 |
| Số trường hợp không giải được | 45 | 50 | 99 |
| Số bước giải | 142 | 130 | 168 |
| Lần 8 | Số trường hợp giải được | 53 | 55 | 2 |
| Số trường hợp không giải được | 47 | 45 | 98 |
| Số bước giải | 113 | 115 | 135 |
| Lần 9 | Số trường hợp giải được | 60 | 60 | 0 |
| Số trường hợp không giải được | 40 | 40 | 100 |
| Số bước giải | 99 | 105 | 155 |
| Lần 10 | Số trường hợp giải được | 59 | 62 | 3 |
| Số trường hợp không giải được | 41 | 38 | 97 |
| Số bước giải | 117 | 85 | 156 |
| Trung bình | Số trường hợp giải được | 57.2 | 52.2 | 2.2 |
| Số trường hợp không giải được | 42.8 | 47.8 | 97.8 |
| Số bước giải | 105.6 | 122 | 152.1 |

***Bảng 2.*** *Cấu hình máy để hiện thực các thuật toán 8-puzzle*

|  |  |
| --- | --- |
|  | Thông số |
| Cấu hình máy tính | -Phần cứng: CPU Intel Core-i7, RAM: 8GB, Win10 (64-bit)  -Phần mềm: Visual Studio 2019 |

Đánh giá chung:

Sau khi phân tích so sánh bằng biểu đồ: Hill-climbing có số trường hợp giải được cao nhất. First-choice Hill-climbing sử dùng hàm sinh ngẫu nhiên, có số bước đi để tìm ra trường hợp sai và độ chính xác thấp hơn Hill-climbing. Thuật toán Simulated Annealing cho số trường hợp không giải được và số bước giải cao nhất. Vì vậy, thuật toán Hill-climbing vẫn là thuật toán tối ưu nhất cho bài toán 8-puzzle.

1. **KẾT LUẬN:**

Hill-climbing tìm thấy trạng thái đích nhanh, đơn giản và hiệu quả trong không gian tìm kiếm nhỏ, ít lồi lõm, khó tìm thấy trạng thái đích nếu không gian tìm kiếm lớn, có nhiều điểm tối ưu cục bộ. First-choice thường dùng cho các bài toán có không gian tìm kiếm rộng lớn. Simulated Annealing luôn cho ra trạng thái tốt, mặc dù chưa phải trạng thái tốt nhưng vẫn có thể chấp nhận nếu T đủ lớn. Tuy nhiên, hàm chi phí và hàm schedule của nhóm em chưa được tốt nên sự đánh giá giữa các thuật toán chưa được khách quan. Nhóm em sẽ cố gắng cải tiến các hàm để thuật toán có thể chạy tối ưu nhất và đánh giá các thuật toán có độ chính xác cao hơn.

**TÀI LIỆU THAM KHẢO**

1. <https://yinyangit.wordpress.com/2010/12/11/algorithm-tim-hi%E1%BB%83u-v%E1%BB%81-bai-toan-n-puzzle-updated/>
2. <https://www.geeksforgeeks.org/simulated-annealing/>
3. <https://www.geeksforgeeks.org/introduction-hill-climbing-artificial-intelligence/>
4. Artificial intelligence: A Modern Approach,2020 by Stuart Russell and Peter Norving