**Heuristic và A\* Search cho Sokoban**



**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**







**TRÍ TUỆ NHÂN TẠO**

**GVHD: Lương Ngọc Hoàng**

**Lớp: CS106.O21**

**Tên: Lê Minh Nhựt**

**MSSV: 22521060**



1. **Hàm heuristics**

Hàm heuristics trong Sokoban dùng để ước lượng khoảng cách còn lại để hoàn thành mục tiêu dựa vào các yếu tố như vị trí của các hộp và mục tiêu nhằm đánh giá mức độ “tốt” khi chuyển sang 1 trạng thái khác.

* Heuristics có sẵn trong code mẫu dùng khoảng cách Manhattan để tính tổng khoảng cách giữa các hộp và các ô mục tiêu:
  + Đầu tiên, nó xác định các ô mục tiêu đã chứa hộp.
  + Sau đó, nó tạo danh sách vị trí của các hộp chưa được đặt vào ô mục tiêu và vị trí của các ô mục tiêu chưa có hộp đặt vào.
  + Với mỗi hộp chưa được đặt vào ô mục tiêu, nó tính khoảng cách Manhattan giữa vị trí của hộp và ô mục tiêu tương ứng.
  + Cuối cùng, nó tính tổng khoảng cách Manhattan này cho tất các các cặp hộp và ô mục tiêu còn trống. Tổng này chính là giá trị của hàm heuristics.

=> Hàm heuristics trên sẽ ưu tiên cho các trạng thái mà các hộp càng gần vị trí đích của chúng.

* Hàm heuristic của em cũng tận dụng cách hoạt động tương tự nhưng thay bước tính bằng khoảng cách Euclide giữa các hộp và ô mục tiêu.

1. **So sánh các hàm heuristics**

**Bảng so sánh 2 hàm heuristics dùng khoảng cách Manhattan và Euclide**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Thời gian chạy** | | **Số nút** | | **Số bước đi** | |
|  | **Manhattan** | **Euclide** | **Manhattan** | **Euclide** | **Manhattan** | **Euclide** |
| **1** | 0.01 | 0.02 | 208 | 367 | 13 | 12 |
| **2** | 0.00 | 0.00 | 81 | 81 | 9 | 9 |
| **3** | 0.01 | 0.01 | 99 | 91 | 15 | 15 |
| **4** | 0.00 | 0.00 | 52 | 52 | 7 | 7 |
| **5** | 0.09 | 0.07 | 873 | 478 | 22 | 20 |
| **6** | 0.01 | 0.01 | 473 | 505 | 19 | 19 |
| **7** | 0.06 | 0.11 | 1224 | 1904 | 21 | 21 |
| **8** | 0.23 | 0.25 | 5128 | 5170 | 97 | 97 |
| **9** | 0.00 | 0.01 | 61 | 61 | 8 | 8 |
| **10** | 0.02 | 0.02 | 458 | 458 | 33 | 33 |
| **11** | 0.02 | 0.02 | 627 | 627 | 34 | 34 |
| **12** | 0.04 | 0.06 | 1253 | 1368 | 23 | 23 |
| **13** | 0.14 | 0.17 | 3923 | 4233 | 31 | 31 |
| **14** | 0.92 | 1.35 | 16748 | 20896 | 23 | 23 |
| **15** | 0.27 | 0.31 | 5366 | 5546 | 105 | 105 |
| **16** | 0.28 | 0.46 | 2693 | 3858 | 42 | 36 |
| **17** | 25.68 | 27.74 | 169158 | 169158 | No solution | No sulution |
| **18** | x | x | x | x | x | x |

**Bảng đánh giá độ tối ưu đường đi của thuật toán A\***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Manhattan heuristics** | | **Euclide heuristics** | |
| **Màn** | **Đường đi tối ưu của bài toán** | **Số bước đi** | **Tối ưu** | **Số bước đi** | **Tối ưu** |
| **1** | 12 | 13 | Không | 12 | Có |
| **2** | 9 | 9 | Có | 9 | Có |
| **3** | 15 | 15 | Có | 15 | Có |
| **4** | 7 | 7 | Có | 7 | Có |
| **5** | 20 | 22 | Không | 20 | Có |
| **6** | 19 | 19 | Có | 19 | Có |
| **7** | 21 | 21 | Có | 21 | Có |
| **8** | 97 | 97 | Có | 97 | Có |
| **9** | 8 | 8 | Có | 8 | Có |
| **10** | 33 | 33 | Có | 33 | Có |
| **11** | 34 | 34 | Có | 34 | Có |
| **12** | 23 | 23 | Có | 23 | Có |
| **13** | 31 | 31 | Có | 31 | Có |
| **14** | 23 | 23 | Có | 23 | Có |
| **15** | 105 | 105 | Có | 105 | Có |
| **16** | 34 | 42 | Không | 36 | Không |
| **17** | No solution | | | | |
| **18** | x | x | x | x | x |

**Nhận xét:**

* Về thời gian nhìn chung cả 2 đều cho lời giải rất nhanh nhưng Manhattan heuristics lại có phần nhỉnh hơn Euclide ở hầu hết các màn.
* Về số nút mở ra thì ta có thể thấy rõ Manhattan có số nút cần mở ra ít hơn Euclide, cũng vì vậy nên Manhattan mới cho tốc độ tốt hơn.
* Về số bước đi Euclide có phần tối ưu hơn Manhattan khi cho lời giải tối ưu ở gần như tất cả màn trừ màn 16, trong khi Manhattan vẫn còn các màn 1, 5, 16 chưa được tối ưu.

=> 2 thuật toán đều có những ưu nhược điểm khác nhau, nếu cần một lời giải có tốc độ tốt hơn ta có thể cân nhắc chọn Manhattan, còn nếu cần 1 lời giải tối ưu hơn và chấp nhận chậm hơn đôi chút ta có thể chọn Euclide.

1. **So sánh UCS và A\***

**Bảng so sánh thời gian chạy và số nút của thuật toán UCS và A\* dùng Euclide heuristics ở từng màn.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Thời gian chạy** | | **Số nút** | |
| **Màn** | **UCS** | **A\*** | **UCS** | **A\*** |
| **1** | 0.06 | 0.02 | 1280 | 367 |
| **2** | 0.01 | 0.00 | 131 | 81 |
| **3** | 0.07 | 0.01 | 792 | 91 |
| **4** | 0.00 | 0.00 | 96 | 52 |
| **5** | 69.81 | 0.07 | 562460 | 478 |
| **6** | 0.01 | 0.01 | 587 | 505 |
| **7** | 0.50 | 0.11 | 13640 | 1904 |
| **8** | 0.21 | 0.25 | 5254 | 5170 |
| **9** | 0.01 | 0.01 | 107 | 61 |
| **10** | 0.01 | 0.02 | 499 | 458 |
| **11** | 0.02 | 0.02 | 649 | 627 |
| **12** | 0.08 | 0.06 | 2806 | 1368 |
| **13** | 0.17 | 0.17 | 582 | 4233 |
| **14** | 2.81 | 1.35 | 65683 | 20896 |
| **15** | 0.29 | 0.31 | 6283 | 5546 |
| **16** | 15.73 | 0.46 | 108391 | 3858 |
| **17** | 24.51 | 27.74 | 169158 | 169158 |
| **18** | x | x | x | x |

**Nhận xét:**

* Nhìn vào thống kê ta có thể thấy UCS cần phải mở toàn bộ các nút để tìm được lời giải, trong khi A\* chỉ cần mở ra số nút ít hơn rất nhiều lần nhờ vào hoạt động của hàm heuristics tốt. Điều này cũng là lí do A\* nhanh hơn UCS về mặt thời gian, giúp tiết kiệm bộ nhớ hơn, đặc biệt là ở các màn phức tạp có thể cho tốc độ gấp hơn mấy chục lần UCS.
* Lời giải của A\* có thể không phải lời giải tối ưu vì nó không duyệt qua tất cả các nút nhưng lời giải này là một lời giải ở mức tốt (không quá chênh lệch so với lời giải tối ưu) và chấp nhận được so với những ưu điểm mà nó mang lại.

File code: [BT2\_22521060.zip](https://drive.google.com/file/d/1S-twWsBKi6fO1MhzHpOSzf5S2HYOYwGd/view?usp=sharing)