

ĐẠI HỌC QUỐC GIA TP.HCM  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



KHOA: KHOA HỌC MÁY TÍNH

MÔN: TRÍ TUỆ NHÂN TẠO - CS106.O21.KHTN

---

## BT2 - Heuristics & A\* search

---

*Sinh viên thực hiện:*  
Trương Thanh Minh - 21520064

# Mục lục

<b>1</b>	<b>Giới thiệu tổng quan</b>	<b>2</b>
1.1	Yêu cầu . . . . .	2
1.2	Source code . . . . .	2
<b>2</b>	<b>Ý tưởng heuristics</b>	<b>2</b>
<b>3</b>	<b>Thống kê thời gian chạy và số nút mở ra của các thuật toán</b>	<b>2</b>
<b>4</b>	<b>Nhận xét</b>	<b>4</b>

# 1 Giới thiệu tổng quan

## 1.1 Yêu cầu

- Trình bày lại heuristics đã sử dụng có ý tưởng là gì? Nếu bạn nào thiết kế và thực nghiệm so sánh thêm với heuristics mới thì sẽ được cộng điểm (Cần phải so sánh với heuristics đang có sẵn trong code mẫu).
- **Bảng thống kê** so sánh thời gian chạy (số giây) và số nút đã được mở ra của hai thuật toán UCS và A\* cho mỗi level. Cần trả lời câu hỏi: Lời giải trả về của A\* có phải lời giải tối ưu hay không cho level tương ứng.

## 1.2 Source code

Link github: <https://github.com/trthminh/CS106.021.KHTN/tree/main/sokoban>

# 2 Ý tưởng heuristics

Ý tưởng của hàm Heuristics ở đây là tính tổng khoảng cách giữa các boxes với các goals. Trong đó, hàm heuristics này sẽ ước lượng khoảng cách giữa boxes với goals bằng cách với mỗi box, chọn goals nào có tọa độ  $x$  gần box đó nhất, nếu có nhiều goals có tọa độ  $x$  gần box đó nhất thì sẽ xét theo tọa độ  $y$ . Theo cách đó, ta sẽ lần lượt tìm ra cặp box - goal tương ứng, sau đó thực hiện tính tổng cách khoảng cách box - goal đó theo công thức tính khoảng cách Hamming.

# 3 Thống kê thời gian chạy và số nút mở ra của các thuật toán

Màn chơi	UCS	A*
1	0.052	0.049
2	0.003	0.008
3	0.074	0.022
4	0.002	0.002
5	67.553	0.905
6	0.009	0.023
7	0.501	0.655
8	0.192	0.326
9	0.006	0.008
10	0.012	0.026
11	0.014	0.034
12	0.078	0.067
13	0.165	0.168
14	2.728	1.653
15	0.260	0.512
16	15.158	0.890
17	21.507	42.739
18	Chạy không nổi	Chạy không nổi

Bảng 1: Bảng thống kê về thời gian chạy (s)

Màn chơi	UCS	A*
1	2115	1014
2	183	157
3	1465	200
4	174	76
5	1355329	9503
6	612	541
7	17283	9733
8	5357	5317
9	181	94
10	526	479
11	668	646
12	3220	1946
13	6202	3800
14	72422	27111
15	6328	6148
16	150971	6182
17	169158	169158
18	Chạy không nổi	Chạy không nổi

Bảng 2: Bảng thống kê về thời gian chạy số nút mở ra

Màn chơi	UCS	A*
1	12	12
2	9	9
3	15	15
4	7	7
5	20	20
6	19	19
7	21	21
8	97	97
9	8	8
10	33	33
11	34	34
12	23	23
13	31	31
14	23	23
15	105	105
16	34	36
17	×	×
18	Chạy không nổi	Chạy không nổi

Bảng 3: Bảng thống kê về độ dài đường đi

## 4 Nhận xét

Đầu tiên, khi quan sát bảng 2, ta thấy rằng thuật toán A\* đều tối ưu hơn thuật toán UCS về mặt bộ nhớ ở tất cả các màn chơi, vì số nút mà A\* cần mở ra ở tất cả các level đều ít hơn số nút mà UCS mở ra.

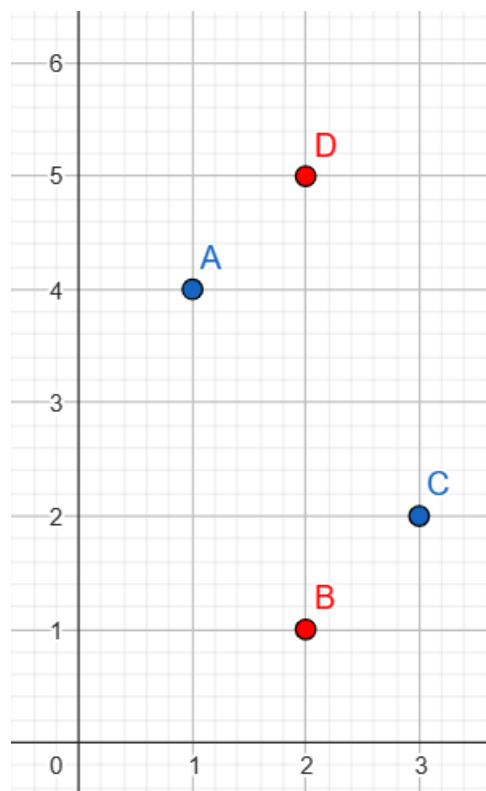
Ngoài ra, trong bảng 3, ta thấy gần như số đường đi mà 2 thuật toán trả về đều giống nhau, trừ level 16, độ dài đường đi mà A\* tìm được dài hơn 2 đơn vị so với độ dài đường đi mà UCS tìm được.

Tiếp theo, ta quan sát thêm bảng 1, 3 ta có một số nhận xét về lời giải của thuật toán A\* ở mỗi màn chơi:

- Level 1: Thời gian thuật toán của A\* ít hơn UCS, số bước để đưa ra kết quả bằng nhau. Do đó, ở màn này, có thể thấy A\* sẽ tối ưu hơn UCS.
- Level 2: Thời gian của A\* nhiều hơn UCS, số bước đưa ra kết quả bằng nhau → A\* **không tối ưu** hơn UCS.
- Level 3: Thời gian của A\* tối ưu hơn đáng kể so với UCS. Do đó A\* tối ưu.
- Level 4: Thời gian của cả 2 thuật toán ngang bằng nhau. Vậy nên A\* tối ưu.
- Level 5: Thời gian của A\* trong việc tìm ra lời giải nhanh hơn 70 lần so với UCS. Vậy nên, trong level này, A\* tối ưu hơn UCS đáng kể.
- Level 6: Trong level này, UCS cho ra kết quả nhanh hơn so với A\*. Do vậy, level này A\* **không tối ưu** so với UCS.

- Level 7: Thời gian  $A^*$  cho ra kết quả lâu hơn UCS, nhưng không đáng kể. Do vậy, có thể nói ở level này  $A^*$  tối ưu.
- Level 8: Thời gian  $A^*$  cho ra kết quả lâu hơn UCS gần 2 lần. Vậy nên, ở level này  $A^*$  **không tối ưu**.
- Level 9: Thời gian  $A^*$  cho ra kết quả lâu hơn UCS, nhưng không đáng kể. Do vậy, có thể nói ở level này  $A^*$  tối ưu.
- Level 10, 11: Thời gian  $A^*$  cho ra kết quả lâu hơn UCS hơn 2 lần. Vậy nên, ở level này  $A^*$  **không tối ưu**.
- Level 12:  $A^*$  cho ra kết quả nhanh hơn UCS, nhưng không đáng kể. Do vậy  $A^*$  vẫn được coi là tối ưu.
- Level 13:  $A^*$  cho ra kết quả chậm hơn UCS, nhưng không đáng kể. Do vậy  $A^*$  vẫn được coi là tối ưu.
- Level 14: Thời gian  $A^*$  trả về kết quả nhanh hơn UCS. Do vậy  $A^*$  tối ưu.
- Level 15:  $A^*$  chạy chậm hơn 2 lần so với UCS  $\rightarrow A^*$  **không tối ưu**.
- Level 16: Ở level này, thời gian cho ra kết quả của  $A^*$  tốt hơn rất nhiều lần so với UCS, tuy nhiên số đường đi mà  $A^*$  trả về lại dài hơn 2 đơn vị so với UCS. Có thể thấy  $A^*$  cho ta sự tối ưu khá tốt về thời gian, nhưng xét về kết quả trả về vẫn chưa thật sự tối ưu (dài hơn 2 đơn vị so với đường đi tối ưu). Tuy vậy, sự sai khác về số bước đi không đáng kể so với sự tối ưu về mặt thời gian  $\rightarrow A^*$  tối ưu.  $A^*$  cho ra đường đi dài hơn UCS là do hàm heuristic không ước lượng tốt, điều này có thể dễ dàng nhận thấy được thông qua hình 1. Trong hình 1, các điểm màu xanh là các boxes, điểm màu đỏ là các goals. Theo hàm heuristic, ta sẽ xét các box theo thứ tự từ box A, sau đó mới tới box C, vì box A có tọa độ  $x$  nhỏ hơn. Mà với box A, hàm heuristic sẽ tính với goal B, bởi vì ý tưởng của hàm heuristic là nếu tọa độ  $x$  bằng nhau thì sẽ chọn điểm nào có tọa độ  $y$  bé hơn. Do đó, hàm heuristic sẽ tính khoảng cách của box A với goals B, box C với goal D. Bằng mắt thường ta có thể thấy khi đó khoảng cách ước lượng sẽ lớn hơn thực tế. Do đó hàm heuristic này không lạc quan. Vậy nên nó không tối ưu.
- Level 17: Trong trường hợp màn chơi không có lời giải như thế này,  $A^*$  mất khá nhiều thời gian mới có thể đưa ra đáp án không có lời giải (lâu gấp đôi UCS)  $\rightarrow A^*$  **không tối ưu**. Sở dĩ thời gian  $A^*$  lâu hơn là do với mỗi node được lấy ra để xét, nó đều phải tính hàm heuristic, mà hàm heuristic có độ phức tạp là  $O(N)$ , với  $N$  là số boxes. Vì số node được mở ra rất lớn, do vậy thời gian sẽ rất lâu.
- Level 18: Vì máy em chạy không nổi vậy nên cũng không thể biết  $A^*$  có tối ưu hơn hay không.

Nhìn chung,  $A^*$  tối ưu hơn UCS về cả thời gian, không gian (số nút được mở ra) và cả số bước đi đến lời giải trong các màn chơi nếu màn chơi đó có lời giải. Tuy nhiên, với các màn chơi không có lời giải,  $A^*$  tốn rất nhiều thời gian mới đưa ra được là màn đó không có lời giải.



Hình 1: Ví dụ minh họa