

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:
Tô Anh Phát - 21520085

Mục lục

1	Giới thiệu tổng quan	2
1.1	Yêu cầu	2
1.2	Source code	2
2	Trình bày ý tưởng heuristics	2
3	Thống kê thời gian chạy và số nút mở ra của các thuật toán	3
4	Nhận xét	5

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.

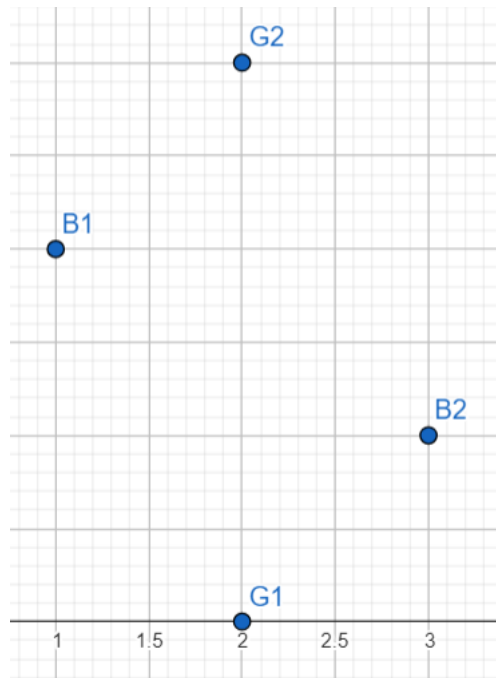
1.2 Source code

Link Github: <https://github.com/pattan99/CS106.021.KHTN/tree/main/sokoban>

2 Trình bày ý tưởng heuristics

Hàm heuristics tính tổng khoảng cách giữa các cặp box và goal chưa được ghép. Các box và goal được sắp xếp theo thứ tự tăng dần theo tọa độ x, nếu tọa độ x như nhau thì tăng dần theo tọa độ y. Sau đó, tính tổng khoảng cách hamming giữa các cặp box và goal tương ứng sau khi sắp xếp.

Tuy nhiên, hàm heuristic này chưa lạc quan. Ví dụ như hình dưới đây:



Hình 1: Một trường hợp hàm heuristic không lạc quan

Theo như hàm heuristic mặc định thì sẽ tính khoảng cách giữa cặp điểm B1 (box 1) đến G1 (goal 1) và B2 (box 2) đến G2 (goal 2). Tuy nhiên, thực tế thì ta sẽ tính khoảng

cách từ B1 đến G2 và từ B2 đến G1 sẽ cho ra kết quả ngắn hơn. Từ đó cho thấy vi phạm điều kiện lạc quan: $h(n) \leq h^*(n)$ nên thuật toán sử dụng hàm heuristic này có thể đưa ra lời giải không tối ưu.

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

Màn chơi	UCS	A*
1	0.050	0.041
2	0.003	0.003
3	0.072	0.011
4	0.002	0.002
5	67.555	0.651
6	0.009	0.013
7	0.500	0.440
8	0.195	0.284
9	0.006	0.004
10	0.012	0.019
11	0.014	0.022
12	0.078	0.071
13	0.165	0.189
14	2.730	1.597
15	0.260	0.410
16	15.155	0.903
17	21.508	40.234
18	Tràn RAM	Tràn RAM

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	Trần RAM	Trần RAM

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	Không có lời giải	Không có lời giải
18	Tràn RAM	Tràn RAM

Bảng 3: Bảng thống kê về độ dài đường đi

4 Nhận xét

Qua các bảng 1, 2, 3, ta có thể thấy thuật toán A* luôn **tối ưu về bộ nhớ** (ít số nút được mở ra) hơn so với UCS, ngoài ra ta có nhận xét về cả hai thuật toán ở mỗi màn chơi như sau:

- Level 1: A* chạy nhanh hơn UCS và độ dài đường đi của hai thuật toán đưa ra bằng nhau. Vì thế, thuật toán A* đưa ra lời giải tối ưu và ít thời gian hơn.
- Level 2: Thời gian và độ dài đường đi của cả hai thuật toán bằng nhau. Do đó, A* đưa ra lời giải tối ưu với thời tương đương với UCS.
- Level 3: Độ dài đường đi của cả hai bằng nhau và UCS chạy nhanh hơn. Từ đó cho thấy A* tối ưu hơn UCS.
- Level 4: Thời gian của 2 thuật toán tương đương nhau. Do đó, A* tối ưu hơn.
- Level 5: A* chạy nhanh hơn UCS rất nhiều. Vì thế, trong trường hợp này A* tối ưu hơn UCS.
- Level 6: UCS chạy nhanh hơn A*, nhưng không đáng kể. Từ đó, A* không tối ưu.
- Level 7: Thời gian của A* ít hơn UCS nên A* tối ưu hơn UCS.
- Level 8: A* chạy chậm hơn UCS. Do đó, A* chưa tối ưu bằng UCS.

- Level 9: A* tối ưu hơn UCS do A* chạy nhanh hơn.
- Level 10: Thời gian của UCS ít hơn A*. Do đó, A* không tối ưu.
- Level 11: A* không tối ưu bằng UCS do chạy lâu hơn UCS.
- Level 12: A* chạy nhanh hơn UCS. Trong trường hợp này, A* tối ưu hơn.
- Level 13: UCS chạy nhanh hơn A* nên A* không tối ưu ở màn chơi này.
- Level 14: Thời gian của A* ít hơn đáng hơn so với UCS nên A* tối ưu hơn UCS.
- Level 15: Thời gian của A* tốn nhiều hơn UCS khá nhiều. Vì thế A* không tối ưu.
- Level 16: Độ dài đường đi của A* nhiều hơn UCS 2 đơn vị (do hàm heuristic không lạc quan), tuy nhiên nếu so về thời gian chạy thì A* chạy nhanh hơn UCS rất nhiều. Do đó, ở màn chơi này, nhìn chung thì A* vẫn tối ưu hơn UCS.
- Level 17: Ở màn chơi không tồn tại lời giải này, A* chạy lâu hơn UCS rất nhiều (do phải tính thêm hàm heuristic khi xét 1 nút và có nhiều nút ở màn chơi này). Vì thế, A* không tối ưu.
- Level 18: Khi chạy màn chơi này, máy em bị tràn RAM, do đó em không có nhận xét về 2 thuật toán ở màn chơi này.

Tổng kết lại thì A* tối ưu hơn UCS về thời gian chạy, lời giải và số nút mở ra. Tuy có vài trường hợp A* chạy lâu hơn nhưng không đáng kể. Ngoài ra, đối với những màn chơi không có lời giải như Level 17 thì A* chạy lâu hơn UCS rất nhiều do phải duyệt hết các nút và phải tốn thời gian tính hàm heuristic.