

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KHOA HỌC MÁY TÍNH

BÀI TẬP MÔN PHÂN TÍCH VÀ THIẾT KẾ THUẬT TOÁN
HOMEWORK #01: DFS/BFS/UCS for Sokoban

GV hướng dẫn: Lương Ngọc Hoàng

Người thực hiện: Lê Quý Nhân. MSSV: 22520999

TPHCM, ngày 20 tháng 3 năm 2024

1 Giới Thiệu:

Sokoban là trò chơi trí tuệ phổ biến, mô phỏng việc di chuyển thùng hàng trong kho. Mục tiêu là di chuyển tất cả các thùng hàng đến vị trí đích đã chỉ định.

2 Các thuật toán tính khoảng cách:

2.1 Euclidean Distance (Khoảng cách Euler hay khoảng cách Euclide).

Trong toán học, khoảng cách **Euclide** (d_E) giữa hai điểm trong không gian *Euclide* là độ dài đoạn thẳng giữa chúng. Nó có thể được tính từ tọa độ *Descartes* của các điểm bằng định lý *Pythagore* và do đó đôi khi được gọi là khoảng cách *Pythagore*.

$$d_E = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Với d_E là khoảng cách Euclide giữa 2 điểm $A(x_1, x_2)$ và $B(y_1, y_2)$

2.2 Mahattan Distance

Khoảng cách **Manhattan**, còn được gọi là khoảng cách L1 hay khoảng cách trong thành phố, là một dạng khoảng cách giữa hai điểm trong không gian *Euclid* với hệ tọa độ *Descartes*. Đại lượng này được tính bằng tổng chiều dài của hình chiếu của đường thẳng nối hai điểm này trong hệ trục tọa độ *Descartes*.

Ví dụ, khoảng cách Manhattan giữa hai điểm A_1 có tọa độ (x_1, y_1) và điểm A_2 có tọa độ (x_2, y_2) là:

$$d_{Manhattan} = |x_1 - x_2| + |y_1 - y_2|$$

3 UCS (Uniform Cost Search):

Duyệt theo chi phí đồng nhất, ưu tiên đường đi có chi phí thấp nhất. Sử dụng cấu trúc dữ liệu Priority Queue để lưu trữ các đỉnh cần khám phá, sắp xếp theo chi phí đường đi.

4 A*:

A* (đọc là A sao) là thuật toán tìm kiếm trong đồ thị. Thuật toán này tìm một đường đi từ một nút khởi đầu tới một nút đích cho trước (hoặc tới một nút thỏa mãn một điều kiện đích). Thuật toán này sử dụng một

"đánh giá *heuristic*" để xếp loại từng nút theo ước lượng về tuyến đường tốt nhất đi qua nút đó. Thuật toán này duyệt các nút theo thứ tự của đánh giá *heuristic* này.

Thuật toán A^* sử dụng hàm *heuristic* để mở rộng các đường dẫn ít tốn kém hơn. Hàm này được định nghĩa như sau:

$$f(n) = g(n) + h(n)$$

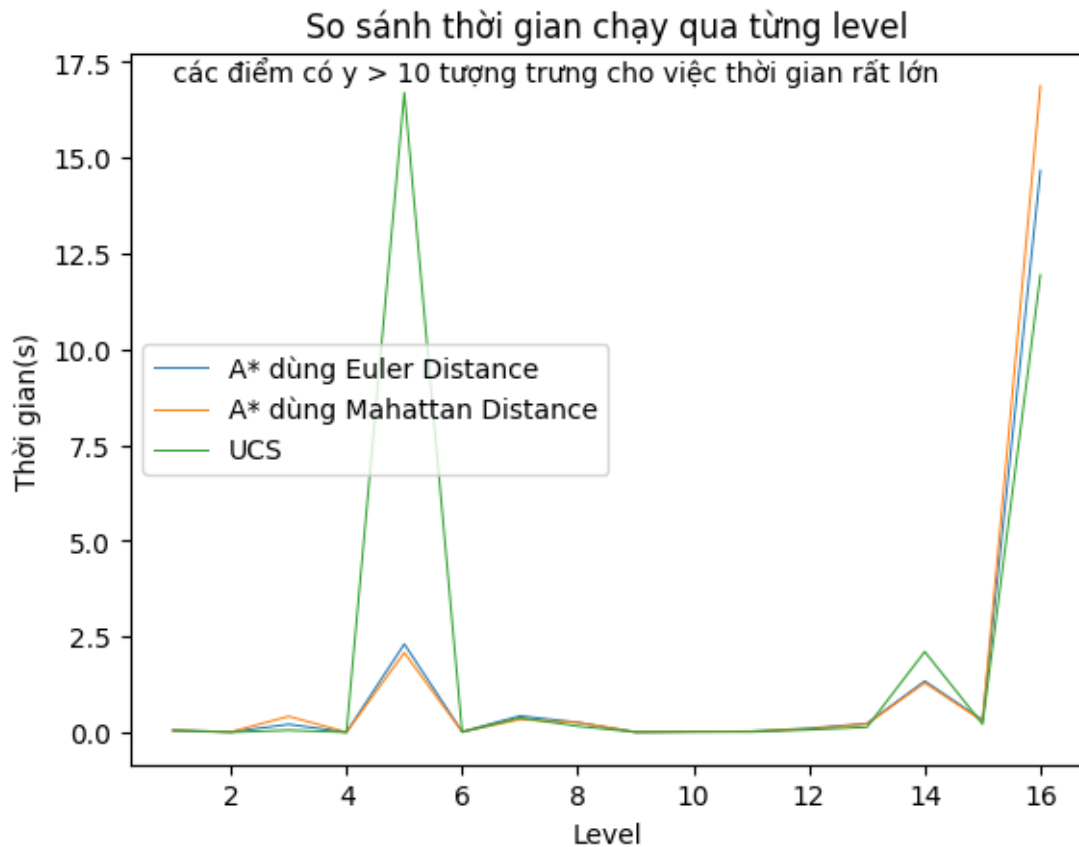
* $f(n)$: Tổng chi phí ước tính của đường đi qua nút n .

* $g(n)$: Chi phí cho đến hiện tại để đến được nút n .

* $h(n)$: Chi phí ước tính từ n đến mục tiêu. Đây là phần *heuristic* của hàm chi phí, đóng vai trò như một dự đoán.

Hàm heuristic giúp thuật toán A^* tập trung vào những đường dẫn có khả năng dẫn đến mục tiêu nhanh nhất, từ đó tăng hiệu quả tìm kiếm. Do đó, thuật toán A^* là một ví dụ của tìm kiếm theo lựa chọn tốt nhất (*best-first search*).

5 Chạy A* và UCS:



Hình 1: So sánh thời gian chạy.

Qua hình ta có thể thấy A* sử dụng *Euler Distance* có thời gian chạy nhanh hơn A* sử dụng *Manhattan Distance* trong hầu hết các level.

Với level 5, UCS chạy rất lâu (hơn 50 giây) trong khi A* với cả 2 hàm *heuristic* đều chạy rất nhanh (< 3 giây). Vì level này có nhiều ngõ cụt, khiến UCS phải khám phá nhiều đường đi không dẫn đến đích. A* sử dụng heuristic để ưu tiên các đường đi có khả năng dẫn đến đích cao hơn, giúp giảm thiểu số lượng nút cần khám phá.

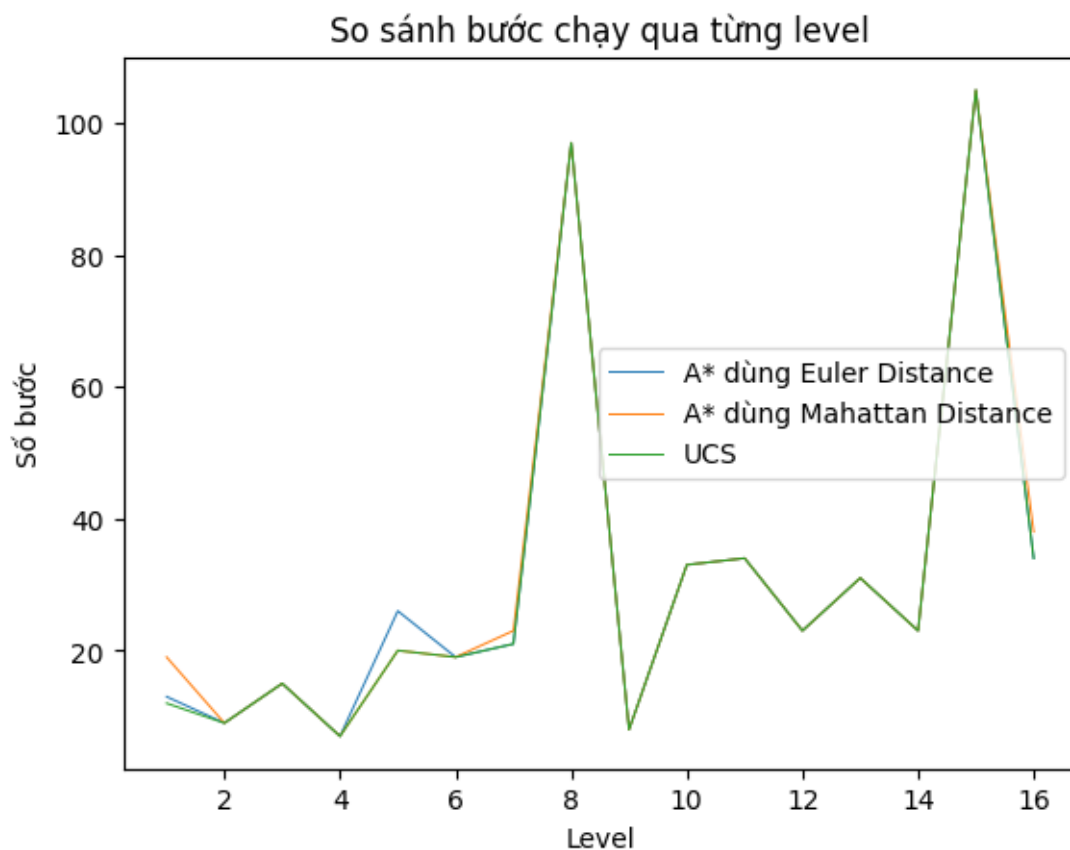
Với level 16, cả 3 thuật toán đều chạy hơn 10s nhưng UCS chạy với 12s trong khi A* với Euler và Mahattan lần lượt là 45s và 81s. Trong level này, heuristic có thể không chính xác do có nhiều chướng ngại vật, khiến A* khám phá nhiều đường đi không dẫn đến đích.

Bảng 1: Bảng so sánh thời gian chạy (giây) - Theo từng level

Level	A*-Euler	A*-Manhattan	UCS
1	0.055	0.051	0.046
2	0.010	0.012	0.003
3	0.209	0.417	0.056
4	0.004	0.006	0.003
5	2.305	2.077	52.696
6	0.016	0.015	0.008
7	0.428	0.334	0.382
8	0.259	0.255	0.160
9	0.008	0.008	0.007
10	0.020	0.019	0.013
11	0.026	0.027	0.013
12	0.106	0.087	0.067
13	0.229	0.217	0.134
14	1.338	1.291	2.106
15	0.322	0.297	0.213
16	44.661	80.868	11.94

Vậy ta có thể thấy UCS có thể nhanh hơn A* trong các level có kích thước nhỏ hoặc heuristic của A* không hiệu quả. A* thường chạy nhanh hơn UCS trong các level có nhiều ngõ cụt hoặc có kích thước lớn.

Ta xem tiếp biểu đồ về bước đi:



Hình 2: So sánh bước chạy.

Ta thấy trừ level đầu và level 5 ra thì các lời giải của 3 thuật toán đều giống nhau, đều là các lời giải tối ưu. Tuy nhiên với level 1 và 5 thì A* cho ra lời giải rất nhanh, nhưng lại không phải là lời giải tối ưu của level đó.

Vậy ta có thể nói **A*** là một thuật toán tìm kiếm hiệu quả, nhưng không đảm bảo tìm được lời giải tối ưu trong mọi trường hợp.

Code vẽ biểu đồ: <https://colab.research.google.com/drive/1Mc41gOikcqSm7kWBcS1-9dZgxcEErRbQ?usp=sharing>