



## Áp dụng thuật toán UCS, A\* cho game Sokoban

Tăng Gia Hân – 22520394

### 1. Uniform cost search

UCS là một biến thể của BFS, nhưng thay vì sử dụng chi phí cố định cho mỗi bước di chuyển, UCS sử dụng chi phí thực tế của từng bước. Trong trò chơi Sokoban, chi phí có thể được định nghĩa là số bước đi để di chuyển một hộp mà không gây ra việc đẩy. Sử dụng một hàng đợi ưu tiên để lưu trữ các trạng thái của trò chơi. Mỗi phần tử trong hàng đợi ưu tiên là một danh sách, đại diện cho một trạng thái của trò chơi. Độ ưu tiên của mỗi phần tử được xác định bằng hàm chi phí. Tại mỗi bước, thuật toán UCS lấy ra trạng thái đầu tiên từ hàng đợi ưu tiên và mở rộng nó bằng cách xem xét các hành động có thể thực hiện được từ trạng thái hiện tại. Sau mỗi bước mở rộng, UCS kiểm tra xem trạng thái hiện tại có phải là trạng thái mục tiêu hay không. Nếu là trạng thái mục tiêu, thuật toán dừng lại và trả về đường đi ngắn nhất đã tìm thấy.

### 2. A-star (A\*) Search

Thuật toán A\* là một trong những thuật toán tìm đường đi trong đồ thị phổ biến nhất do những đặc điểm của nó bao gồm tính “đầy đủ” và “tối ưu”. Thuật toán A\* kết hợp giữa chi phí thực tế từ một đỉnh đến đích (được đo bằng hàm cost) và một hàm ước lượng khoảng cách từ đỉnh hiện tại đến đích (được đo bằng hàm heuristic). A\* đảm bảo tìm ra đường đi ngắn nhất từ điểm bắt đầu đến điểm kết thúc.

Hàm cost được minh họa như sau:

$$CostFunction = \sum cost_{iteration}$$

Hàm heuristic tính tổng khoảng cách Manhattan giữa các hộp chưa nằm trên mục tiêu và các mục tiêu chưa được hoàn thành. Công thức:

$$d(box, goal) = |goal.x - box.x| + |goal.y - box.y|$$

### 3. So sánh giữa hai thuật toán UCS và A\*

Ta có thể so sánh hai thuật toán dựa trên bảng thống kê thời gian thực thi (số giây), số bước đi và tổng số nút đã được mở ra cho mỗi level.

Level	UCS			A*		
	Time(s)	Số bước đi	Số nút mở ra	Time(s)	Số bước đi	Số nút mở ra
1	0.16	12	1099	0.02	13	171
2	0.02	9	129	0.01	9	84
3	0.23	15	786	0.02	15	108
4	0.02	7	78	0.01	7	55
5	133.11	20	395066	0.22	22	901
6	0.04	19	567	0.04	19	492
7	1.59	21	13657	0.22	21	1430
8	0.53	99	5234	0.59	97	5230
9	0.02	8	94	0.01	8	61
10	0.05	33	487	0.05	33	493
11	0.05	34	646	0.07	34	663
12	0.26	23	2479	0.14	23	1297
13	0.52	31	5463	0.44	31	3994
14	8.09	23	59785	2.80	23	17345
15	0.73	105	6243	0.76	105	5546
16	31.15	34	90384	0.73	42	2817
17	-	-	-	-	-	-
18	-	-	-	-	-	-

Table 1: Thống kê thời gian chạy (số giây), số bước và số nút đã được mở ra của hai thuật toán UCS và A\* cho mỗi level

Ta có thể thấy trong trường hợp của level 17, thuật toán UCS và A\* không đưa ra được lời giải. Điều này có thể được giải thích bởi sự phức tạp của bản đồ, khi có nhiều hộp cần di chuyển và các đường đi có thể bị chặn.

Trong trường hợp của level 18, thời gian thực thi của UCS và A\* trở nên rất lớn và có thể gây ra vấn đề về hiệu suất hoặc thậm chí không thể hoàn thành trong thời gian hợp lý. Việc áp dụng các thuật toán vào Sokoban ở level này có thể dễ dàng dẫn đến việc sử dụng quá nhiều bộ nhớ và vượt quá giới hạn bộ nhớ có sẵn.

Ta có thể quan sát qua đồ thị dưới đây để có cái nhìn trực quan hơn:

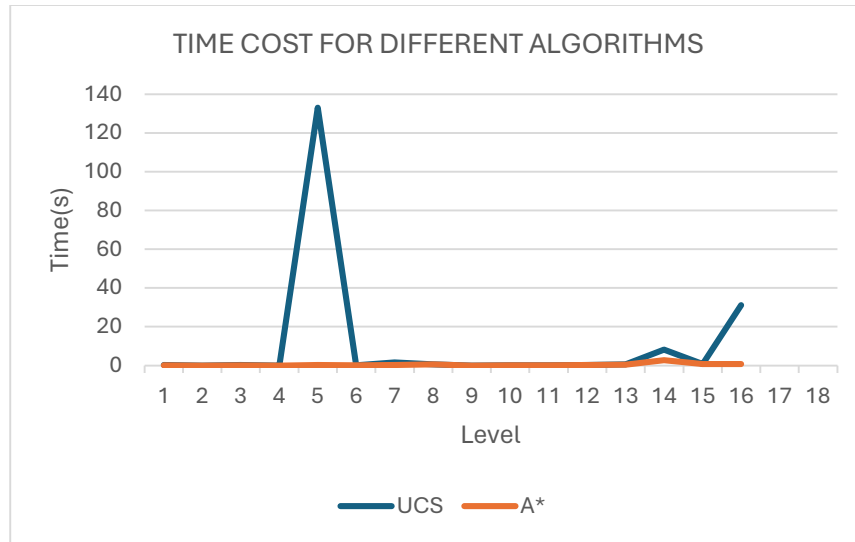


Figure 1: Biểu đồ chi phí thời gian cho các thuật toán UCS, A\*

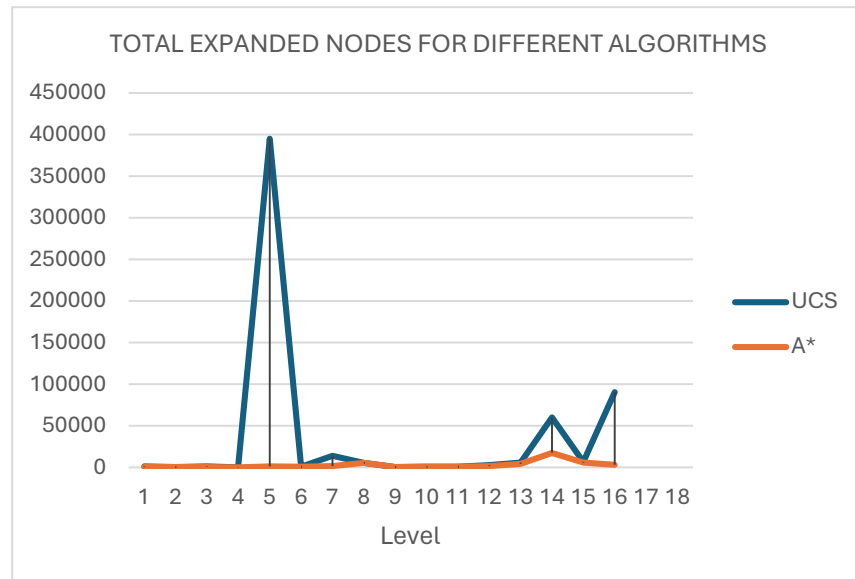


Figure 2: Biểu đồ tổng số nút đã được mở ra của UCS, A\*

### Nhận xét:

1. Thời gian thực thi (Time):
  - A\* thường có thời gian thực thi thấp hơn so với UCS ở hầu hết các level (13/18).
  - A\* có thể nhanh hơn UCS nhiều lần đặc biệt là ở các mức độ phức tạp cao như level 5, 7, 14, 16.
  - UCS chỉ nhanh hơn A\* trong một số trường hợp nhất định (level 8, 11, 15).
  - Điều này cho thấy A\* thường có khả năng tìm kiếm lời giải nhanh hơn so với UCS.
2. Số bước đi:
  - A\* thường có số bước đi ít hơn hoặc bằng với UCS ở hầu hết các level (13/18)

- Điều này cho thấy cả hai thuật toán đều tìm được một lời giải gần với nhau về mặt chiều dài của đường đi.
3. Số nút đã được mở ra:
    - A\* thường mở rộng ít nút hơn UCS ở hầu hết các level (13/18).
    - Tuy nhiên, trong một số level, số nút mở ra của hai thuật toán gần như bằng nhau (level 2, 4, 8, 9, 10, 11).
    - Điều này chỉ ra rằng A\* thường tìm được đường đi tối ưu hơn mà không cần phải mở ra quá nhiều nút trong quá trình tìm kiếm.
  4. Thuật toán cho lời giải tối ưu cho các Level:
    - Các Level 2, 4, 6, 8, 9, 10 và 13: A\* và UCS có hiệu suất tương đương nhau, với thời gian thực thi, số bước đi và tổng số nút mở ra không có sự khác biệt đáng kể giữa hai thuật toán.
    - Các Level 1, 3, 5, 7, 12, 14, 15 và 16: A\* nhanh hơn UCS rất nhiều. Thường thì A\* có hiệu quả tối ưu hơn trong các trường hợp này vì nó có khả năng tìm kiếm đường đi tối ưu hơn mà không cần mở ra quá nhiều nút.
    - Các Level 10 và 11: UCS nhanh hơn A\* một chút. Trong các trường hợp này, UCS có thể hoạt động tốt hơn do có ít nút cần mở ra hơn so với A\*, dù với một số level, sự chênh lệch không đáng kể.

#### 4. Kết luận

Trong quá trình nghiên cứu và áp dụng các thuật toán UCS và A\* vào trò chơi Sokoban, chúng ta đã thấy rằng mỗi thuật toán có những ưu và nhược điểm riêng. Tuy nhiên, khi đánh giá hiệu suất của từng thuật toán trên các level của trò chơi, có thể thấy rằng A\* thường cho hiệu quả tối ưu hơn so với UCS trên các level phức tạp. Tuy nhiên, UCS có thể là lựa chọn trong một số trường hợp hiếm hoi khi heuristic của A\* không hiệu quả.