

## HƯỚNG DẪN THỰC HÀNH 3

### Môn: Trí Tuệ Nhân Tạo (503030)

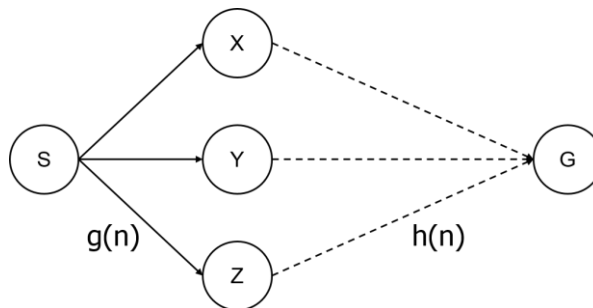
#### I. Mục Tiêu

- Ôn tập lý thuyết về các thành phần và thuộc tính của bài toán AI.
- Ôn tập các giải thuật tìm kiếm trong không gian trạng thái đã học.
- Giới thiệu giải thuật **Greedy best-first search**.
- Giới thiệu bài toán 8-puzzle để minh họa giải thuật Greedy best-first search.

#### II. Giải Thuật Greedy Best-First Search

##### 1. Ý tưởng giải thuật

Nhóm giải thuật *Best-First Search* sử dụng hàm đánh giá (evaluation function)  $f(n)$  để lựa chọn trạng thái thỏa điều kiện bài toán (cực tiểu hóa chi phí hoặc cực đại hóa lợi ích) để mở rộng không gian trạng thái. Đối với giải thuật Greedy best-first search, hàm đánh giá  $f(n) = h(n)$  và giải thuật Greedy BFS chọn node tiếp theo là node gần nhất với trạng thái đích (goal state).



Hình 1. Mô hình ý tưởng hàm đánh giá  $g(n)$  và  $h(n)$

Trong hình: hàm  $h(n)$  ước lượng chi phí (cost) từ trạng thái hiện tại đến trạng thái đích, hàm  $g(n)$  tính toán chi phí từ trạng thái khởi đầu (initial state) đến trạng thái hiện tại.

##### 2. Các thuộc tính (properties) của giải thuật Greedy BFS

- Complete: No.
- Time:  $O(b^m)$ .
- Space:  $O(b^m)$  do Greedy BFS lưu lại tất cả trạng thái trong bộ nhớ.
- Optimal: No.

### 3. Mã giải thuật Greedy BFS

```

function TREE-SEARCH (problem) returns a solution, or failure
  fringe  $\leftarrow$  INSERT(MAKE-NODE(INITIAL-STATE[problem]), fringe)
  loop do
    if fringe is empty then return failure
    node  $\leftarrow$  REMOVE-FRONT(frontier)
    if GOAL-TEST[problem](STATE[node]) then return SOLUTION(node)
    fringe  $\leftarrow$  INSERTALL(EXPAND(node, problem), fringe)
  
```

Figure 1. Mã giả giải thuật Tree Search

## III. Bài Toán 8-Puzzle

### 1. Phát biểu bài toán

Cho một bàn cờ có kích thước  $3 \times 3$ , bàn cờ được xếp ngẫu nhiên bằng tám chữ số (từ 1 đến 8) và một ô trống để dùng làm ô hoán đổi vị trí với ô chứa chữ số.

*Yêu cầu:* Bằng cách dịch chuyển ô trống từ trạng thái khởi đầu, sau một số hữu hạn các bước, ta thu được bàn cờ với các ô số giống với trạng thái đích.

1	2	3
4		6
7	5	8

Initial State

1	2	3
4	5	6
7	8	

Goal State

Hình 2. Ví dụ mẫu không gian trạng thái của bài toán

### 2. Hàm đánh giá - Heuristic

Ứng dụng khoảng cách *Manhattan*<sup>1</sup> để hiện thực hàm đánh giá  $h(n)$ . Trong đó,  $h(n)$  giữa hai lần dịch chuyển trạng thái được tính bằng tổng khoảng cách Manhattan các vị trí của từng chữ số ở trạng thái thứ  $n$  với vị trí của nó ở trạng thái đích.

Xét ví dụ trong Hình 2, giá trị của hàm  $h(\text{InitialState})$  được tính theo bảng sau:

Chữ số	Vị trí ở trạng thái Initial State	Vị trí ở trạng thái Goal State	Khoảng cách Manhattan
1	(1, 1)	(1, 1)	0

<sup>1</sup> Khoảng cách Manhattan giữa hai điểm  $P_1(x_1, y_1)$  và  $P_2(x_2, y_2)$ :  $\text{Manhattan}(P_1, P_2) = |x_1 - x_2| + |y_1 - y_2|$

2	(1, 2)	(1, 2)	0
3	(1, 3)	(1, 3)	0
4	(2, 1)	(2, 1)	0
5	(3, 2)	(2, 2)	1
6	(2, 3)	(2, 3)	0
7	(3, 1)	(3, 1)	0
8	(3, 3)	(3, 2)	1
$f(InitialState) = h(InitialState)$			<b>2</b>

### 3. Ý tưởng giải bài toán

Từ trạng thái thứ  $n$ , bằng cách dịch chuyển ô trống (theo 4 hướng *left, right, up, down*) để sinh ra các trạng thái tiếp theo. Áp dụng hàm đánh giá  $h(n)$  để tính giá trị heuristic và chọn trạng thái có giá trị heuristic nhỏ nhất để dịch chuyển trạng thái, tiếp tục lặp lại đến khi  $h(n) = 0$ .

## IV. Bài Tập

1. Dựa vào mã nguồn được cung cấp, anh (chị) hãy giải bài toán 8-puzzle bằng cách hiện thực các phương thức để trống bằng giải thuật Greedy BFS.
2. Dựa vào giải thuật đã hiện thực ở *câu 1*, anh (chị) hãy thay đổi hàm heuristic và chạy lại giải thuật Greedy BFS.

--- HẾT ---