NHẬP MÔN TRÍ TUỆ NHÂN TẠO

Họ và tên: Đinh Anh Huy

Mã số sinh viên: 18110103 **Bài tập:** Tuần 4

Problem 1 **Read book and define the analysis of the completeness, time and space complexity, optimization:**

(a) Greedy best-first search.

- Completeness: Greedy best-first search không đầy đủ, tức là luôn có rủi ro trên đường đi đến nút mục tiêu.
- Optimazation: Greedy best-first search không tối ưu vì đường đi đến nút mục tiêu có được tìm thấy có thể không tối ưu.
- Time complexity: $O(b^m)$
- Space complexity: $O(b^m)$

Trong đó b là yếu tố phân nhánh, m là độ sâu tối đa của không gian tìm kiếm.

(b) A^* search.

- Completeness: Nếu một đồ thị tìm kiếm có hệ số phân nhánh hữu hạn và tất cả các trọng số đều lớn hơn một số $\epsilon > 0$ thì A^* search đầy đủ.
- Optimazation: A^* search tối ưu nếu heuristic function (h) là admissible hoặc consistent.
- Time complexity: $O(b^m)$
- Space complexity: $O(b^m)$

Trong đó b là yếu tố phân nhánh, m là độ sâu tối đa của không gian tìm kiếm.

Problem 2 **Theorem: if $h(n)$ is admissible, A^* using Tree-Search is optimal.**

Giả sử có 2 *goal states* trên cây tìm kiếm (search tree) là optimal goal A và suboptimal goal B. Gọi n là một nút cha bất kỳ của A (có thể bao gồm cả A).

A là điểm tối ưu và B là điểm dưới tối ưu nên chi phí đi ngược về nút đầu từ A sẽ ít hơn đi từ B, do đó

$$g(A) < g(B) \tag{1}$$

Cả A và B đều là goal state nên chi phí tối ưu thực sự đến goal state từ A hay B là

$$h^*(A) = h^*(B) = 0$$

Do đó

$$\begin{aligned} &0 \leq h(A) \leq h^*(A) \wedge 0 \leq h(B) \leq h^*(B) \\ \Rightarrow &0 \leq h(A) \leq 0 \wedge 0 \leq h(B) \leq 0 \\ \Rightarrow &h(A) = h(B) = 0 \end{aligned} \tag{2}$$

Theo tính admissible của h, ta có

$$f(n) = g(n) + h(n) \leq g(n) + h^*(n) = g(A) = f(A) \tag{3}$$

Từ (1) và (2), ta có

$$f(A) = g(A) + h(A) = g(A) < g(B) = g(B) + h(B) = f(B) \tag{4}$$

Từ (3) và (4), ta có

$$f(n) \leq f(A) < f(B) \Rightarrow f(n) < f(B)$$

Như vậy, n luôn được mở rộng trước B. Hay A^* sử dụng cây tìm kiếm là tối ưu.

Problem 3 (Bonus) **Theorem: if $h(n)$ is consistent, A^* using Graph-Search is optimal.**
Xét bổ đề: Nếu h consistent thì các giá trị của f dọc theo bất kỳ đường đi nào đều không giảm. Thật vậy, gọi n' là nút kề dưới của n, khi đó

$$g(n') = g(n) + c(n, a, n')$$

Ta có

$$f(n') = g(n') + h(n') = g(n) + c(n, a, n') + h(n') \geq g(n) + h(n) = f(n)$$

Chứng minh 2 ý sau để giải bài toán:

(1) Khi thuật toán A^* chọn một node n để mở rộng, ta luôn tìm thấy đường đi tối ưu đến nút đó. Thật vậy, giả sử điều trên là sai, tức là với n được chọn từ fringe thì đường đi đến n sẽ không tối ưu. Điều này đồng nghĩa với việc tồn tại n'' là một nút cha của n trên fringe mà không được mở rộng nhưng lại nằm trên đường đi tối ưu đến n. Mẫu thuẫn với bổ đề vì f có giá trị không giảm dọc theo đường đi đến n thì n'' phải được loại bỏ trước n để mở rộng.

(2) Optimal goal A sẽ luôn được loại bỏ để mở rộng trước bất kỳ suboptimal goal B nào. Thật vậy, vì A và B đều là goal state nên $h(A) = h(B) = 0$; A là optimal goal và B là suboptimal goal nên $g(A) < g(B)$, do đó ta có

$$f(A) = g(A) < g(B) = f(B).$$

Từ (1) và (2), ta suy ra được A^* sử dụng đồ thị tìm kiếm là tối ưu.