A* Search and associated terminology

(Project1)

Nguyen Quoc Huy

23020082

CS 188

Method

Research Design

Objective: Understand the functionality of A star Search and associated terminology.

Results

Fundamental Terminology

Heuristics

- Heuristics are the driving force that allow estimation of distance to goal states.

- Heuristic functions take in a state as input and output a corresponding estimate.

- Heuristics are typically solutions to relaxed problems.

Greedy Search

- Greedy search always selects the frontier node with the lowest heuristic value for expansion.

- Frontier representation: A priority queue is used as UCS, but instead of the computed backward cost, it uses the estimated forward cost.

- Completeness and Optimality: Greedy search is not guaranteed to find a goal state if one exists, nor is it optimal.

Admissibility

- g(n) - Total backwards cost

- h(n) - Estimated forward cost

- f(n) - Estimated total cost: $f(n) = g(n) + h(n)$

- A* is reduced to BFS if: $h(n) = 1 - g(n)$. Hence, we have $f(n) = 1$ for all edges which means not optimal.

- The admissibility constraint: $\forall n, 0 \leq h(n) \leq h*(n)$ which $h*(n)$ is the true optimal forward cost to reach a goal state.

- Theorem: If the admissibility constraint is satisfied by a heuristic function h, using A* search with h on a search problem will yield an optimal solution.

Consistency

- A stronger property than admissibility to maintain optimality under A* graph search.

- Enforcing not only that a heuristic underestimates the total distance to a goal from any given node, but also the cost of each edge in the graph.

- The consistency constraint: $\forall A,C\ h(A)-h(C) \leq cost(A,C)$

- Theorem: If the consistency constraint is satisfied by a heuristic function h, using A* graph search with h on a search problem will yield an optimal solution

Dominance

- To know if a heuristic is better than another, we have a standard metric: dominance.

- If heuristic a is dominant over heuristic b, then the estimated goal distance for a is greater than the estimated goal distance for b for every node in the state space graph: $\forall n: h_a(n) \geq h_b(n)$

- Trivial heuristic: $h(n) = 0$, and using it reduces A* search to UCS.

- Computing the max over values output by multiple admissible/consistent heuristics generates a heuristic that dominates (and hence is better than) all of them individually.

A* Search

Overview

- A* Search selects the frontier node with the lowest estimated total cost for expansion

- Frontier Representation: Using a priority queue to represent its frontier. A* combines the total backward cost (sum of edge weights in the path to the state) with the estimated forward cost (heuristic value).

- Completeness and Optimality: A* search is both complete and optimal

Project1

I implement A* search with a heuristic function in Project 1. It passes all test successfully.

Discussion

About how to find a good heuristic:

- Heuristic is a solution to a relaxed problem where some constraints of the original problem have been removed. It is usually both admissible and consistent.

- There are some trade-offs between computing for heuristic functions and the number of expanded nodes.

- Inconsistency can be detected by verifying that for each node you expand, its successor nodes are equal or higher in f-value.

- If UCS and A* with a heuristic return path of different lengths, that heuristic is inconsistent (because the trivial heuristic h(n) = 0 reduces A* to UCS, and UCS is both complete and optimal)