COMP90054 – Week 3 tutorial

Prepared by Zijie Xu. Last updated: 9 Aug 2023

Evaluating search strategies

Guarantees

- Completeness: Guaranteed to find a solution where there is one
- Optimality: Solution found guaranteed to be optimal

Complexity

- Time complexity
- Space complexity
- Features governing complexity:
  - b Branching factor
  - d Goal depth

Blind search algorithms

	BrFS	DFS	ID
Feature	FIFO frontier	LIFO frontier	Use depth-limited search
Complete?	Yes	No^	Yes
Optimal?	Yes*	No	Yes*
Time complexity	$O(b^d)^\dagger$	$O(b^m)^\ddagger$	$O(b^d)$
Space complexity	$O(b^d)$	$O(bm)^\ddagger$	$O(bd)$

* Optimal if action costs are uniform

† $O(b^{d+1})$ if goal check is performed at node-expansion time

^ Complete if state space is acyclic, or if state space is finite and we check for cycles

‡ m stands for the maximum depth reached by DFS

Informed/Heuristic search

Heuristic function

- Heuristic function $h: S \to \mathbb{R}_0^+ \cup \{\infty\}$
- h estimates the cost of an optimal path to the goal
- h^* is the perfect heuristic/actual cost of an optimal path
- $h(s)$ is the heuristic value/h-value given a state $s \in S$

Properties of heuristic functions

Safe

- $\forall s \in S$ with $h(s) = \infty, h^*(s) = \infty \Rightarrow h$ is **safe**
- h never tells me infinite when there is a solution

Goal-aware

- $\forall s \in S^G, h(s) = 0 \Rightarrow h$ is **goal-aware**
- h knows if I have reached any goal

Admissible

- $\forall s \in S, h(s) \leq h^*(s) \Rightarrow h$ is **admissible**
- h never yields cost larger than actual optimal cost

Consistent

- $\forall$transitions $s \xrightarrow{a} s', h(s) \le h(s') + c(a) \Rightarrow h$ is **consistent**
- Difference in heuristic values of old state s and new state s' never over-estimates the action costs of relevant transitions

Relationship

- consistent + goal-aware → admissible
- admissible → goal-aware
- admissible → safe

Dominance

- If h_1 and h_2 are both admissible and $h_2(s) \ge h_1(s)\ \forall s \in S$, then h_2 dominates h_1

Heuristics search algorithms

- Use a priority queue/min-heap with key f to determine the order of search node expansion
- The following algorithms are all global searches

Greedy Best-first Search

- $f(s) = h(s)$
- Complete for safe heuristics and duplicate detection
- Not optimal, even for perfect heuristics

A* (with re-opening)

- $f(s) = g(s) + h(s)$
- Complete for safe heuristics
- Optimal for admissible heuristics

Weighted A*

- $f(s) = g(s) + W \times h(s)$
- $W \in \mathbb{R}_0^+$ is an algorithm parameter
    - $W = 0 \Rightarrow$ Uniform-cost search
    - $W = 1 \Rightarrow$ A* search
    - $W \to \infty \Rightarrow$ Greedy best-first search