Informed Search

Lecture 3

Outline

Outline

Blind Search

- So far, we have looked at a variety of blind search strategies.
- They are blind because they can only recognize an exact goal match: whether a node is a goal node.
- They have no notion of a node being (somehow) closer to the goal.
- For example, coffee agent in R7 facing east will have no preference for the node resulting from a forward move over one resulting in a rotation (assuming the two actions have the same cost).

R1	R2	R2	R3
R4	R2	R5	R3
R6	R2	R7	R8
R9	R10	R8	R8

Evaluation Function

- Blind search is evidently not efficient.
- The search cost may be cut considerably if search is more goal-directed.
- Goal-directedness is typically provided by an evaluation function that assigns a number to each node indicating how desirable (or not) it is to expand the node.
    - Where desirability is (vaguely) in terms of closeness to the goal.
- Best-first search is a class of search strategies that expands nodes in order of desirability as indicated by some evaluation function.

Best-First Search

function BEST-FIRST-SEARCH(*problem*, EVAL-FUN)
    returns a solution, or failure
  Qing-Fn $\longleftarrow$ a function that inserts nodes in order as per EVAL-FN
  return GENERAL-SEARCH(*problem*, *Qing-Fn*)

- Different definitions of the evaluation function give rise to several best-first strategies.
- We shall consider two obvious ones.

Greedy Search

- Minimizes the estimated cost to reach the goal.
- Uses a heuristic evaluation function $h(n)$:

  $h(n)$ = estimated cost of the cheapest path from the state at node n to a goal state

function GREEDY-SEARCH(*problem*)
returns a solution, or failure
return BEST-FIRST-SEARCH(*problem*, *h*)

Heuristic Functions

- Are there any constraints on h?
- Well, formally, it could be any function

$$h : Nodes \longrightarrow \mathbb{R}$$

- In addition, it should satisfy the centering property:

$$h(n) = 0 \text{ whenever } n \text{ is a goal node.}$$

- A *good* heuristic function should have more properties. (More on this later.)

Coffee-Delivery Heuristic

- What is a suitable heuristic for the coffee-delivery environment?
- A good heuristic is the city block distance (or the Manhattan distance):

$$h(n) = \min_{r \in G}\{|x_r - x_n| + |y_r - y_n|\}$$

where G is the set of goal states and (x_i, y_i) are the indices of the cell in the state of node i.

Non-Standard Coffee-Delivery

R1	R2	R2	R8
R4	R11	R5	R3
R6	R11	R7	R3
R9	R10	R3	R3

- Here, greedy search may increase the search cost.
- If we are not careful with repeated states, we might end up in an infinite loop.

Outline

The Gist of It

- A* is a best-first search algorithm.
- It combines the best of two worlds: the efficiency of greedy search and the "completeness and optimality" of uniform cost search.
- How? It defines the evaluation function as

$$f(n) = g(n) + h(n)$$

  where
  - $g(n)$ is the path cost function used in uniform cost search, and
  - $h(n)$ is an admissible heuristic function.
- A* desirability criterion

  $f(n)$ = estimated cost of the cheapest solution through n.

Admissible Heuristics

A heuristic function h is admissible if it never overestimates the cost to reach the goal.

- That is, if $h^*(n)$ is the *actual* path cost from n to the closest goal node, then
$$h(n) \leq h^*(n)$$
- Is the city block distance heuristic admissible with respect to the coffee-delivery environment?

A* Search

function A*-SEARCH(*problem*)
  returns a solution, or failure
 return BEST-FIRST-SEARCH(*problem*, $g + h$)

Evaluation

- In the worst-case, A* is similar to breadth-first search in both time and space complexity.
  - Some restrictions on h may reduce complexity, though.
- However, A* is both optimal and complete.
  - In fact, it is optimally efficient.
  - That is, no other *optimal* algorithm is guaranteed to expand fewer nodes than A*.

Why Is It Optimal?

- Let G be an optimal goal state, with path cost $g(G) = f^*$.
- Suppose that A^* selects for expansion a sub-optimal goal state, G_2, before selecting G.
- Let n be a currently-leaf node on an optimal path to G.
  - There must be such a node. Right?
- Remember that h is admissible.

Why Is It Optimal?

- It must be that

$$f(n) \leq f^*$$

- Now if n is not chosen for expansion over G_2, it must be that

$$f(G_2) = g(G_2) \leq f(n) \leq f^*$$

- But this means that G_2 is not sub-optimal.
- Hence, a contradiction.

Outline

1. Informed Search
2. A^* Search
3. Heuristic Functions

Comparing Heuristics

- Given two (admissible) heuristics, h_1 and h_2, which one should be used?
- Either will return the optimal solution (if one exists).
- Thus, choose the one which is more efficient—the one that will expand less nodes.

Dominance

- h_1 dominates h_2 if $h_1(n) \geq h_2(n)$ for any node n.
- Given some reasonable assumptions on h_1 and h_2 (namely, monotonicity), A* will expand all nodes n for which $f(n) < f^*$.
- That is, it will expand all nodes satisfying

$$h(n) < f^* - g(n)$$

- Thus, every node expanded by h_1 will also be expanded by h_2.
- h_2 may expand more nodes.
- Conclusion: Opt for a heuristic with higher values (provided it remains admissible).

But Note . . .

- Dominance is only a partial order.
- It is possible to have a set, H, of heuristics over which the dominance relation contains only identical pairs.
- Luckily, you can always introduce a dominating heuristic:

$$h(n) = \max_{h_i \in H}\{h_i(n)\}$$

How to Design Heuristics?

- No definite answer.
- However, an exact cost for a relaxed problem is often a good heuristic.
- A relaxed problem is one with less restrictions on the operators.

Example

Non-Standard Coffee-Delivery

The agent can move from cell a to cell b if

1. a is adjacent to b.
2. A door connects a and b.
3. The agent is facing b.

- By removing any combination of these constraints, we produce a relaxed problem.

Possible Relaxations

- Removing the second constraint, we get standard coffee-delivery.
- Removing the last two, we get the city block distance heuristic.
- What about removing all?
- What about other possibilities?