

Exercise walk-through

13.1

Search

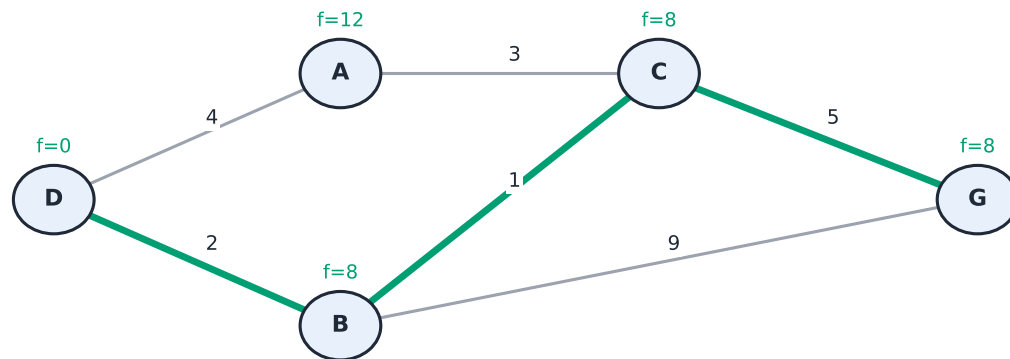
University Computer Science

Search: $f = g + h$, and the estimate is the contract

- 1 **Tree** Nodes are states, edges are moves: BFS shortest when steps equal, DFS frugal; A^* picks $\min g + h$.
- 2 **Course** D-A 4, D-B 2, A-C 3, B-C 1, C-G 5, B-G 9 with $h = A\ 8, B\ 6, C\ 5, G\ 0$: pops D, B, C, G - optimal 8 via D-B-C-G, four expansions.
- 3 **Lie** Overestimate $h(C)$ to 12: C carries $f\ 15$ vs Goal 11 - pops Goal at 11, suboptimal: the algorithm trusts a lie.

Search tree

A* expands four nodes, BFS visits all five



pop order:

D, B, C, G

optimal 8 via

D-B-C-G

overestimating
 $h(C)=12$ pops G
 at 11: suboptimal

$f = g + h$ with admissible h : the first goal pop is optimal

A* with an admissible h expands four nodes; BFS visits all five.

Exercises 2.1-2.3

2.1 What does admissible mean, in symbols? [3]

2.2 What is A^* 's f at the moment it pops the goal? [3]

2.3 Why is A^* with an admissible h optimal at the FIRST goal pop? [3]

Exercises 2.1-2.3

2.1 What does admissible mean, in symbols? [3]

2.1

- $h(n) \leq h^*(n)$ for every node: never overestimates the true remaining cost ✓.
-

2.2 What is A*'s f at the moment it pops the goal? [3]

2.2

- $f(goal) = g(goal)$, since $h(goal) = 0$ ✓.
-

2.3 Why is A* with an admissible h optimal at the FIRST goal pop? [3]

2.3

- That f is a lower bound for every open node: no unexpanded path could be cheaper ✓.

Exercise 3.1

3.1 Course graph above. Animate A^* from D.

Give every pop with its (g, h, f) at pop time. Give the expansions count and the optimal path with its cost. State which edge A^* never uses and why it still cannot be missed. Contrast the BFS visit order. [8]

Exercise 3.1

3.1 Course graph above. Animate A^* from D.

Give every pop with its (g, h, f) at pop time. Give the expansions count and the optimal path with its cost. State which edge A^* never uses and why it still cannot be missed. Contrast the BFS visit order. [8]

3.1

- Pops: D $(g0, h0, f0)$; B $(g2, h6, f8)$; C $(g3, h5, f8)$; G $(g8, h0, f8)$ ✓.
- Four expansions; path $D \rightarrow B \rightarrow C \rightarrow G$ at cost $2 + 1 + 5 = 8$ ✓.
- A ($f = 12$) is never popped: its subtree cannot beat the goal's bound – and the unused $A \rightarrow C$ edge is exactly what that bound excludes ✓.
- BFS order: D, A, B, C, G (level by level) – five visits, no cost awareness ✓.

Exercise 3.2

3.2 Replace h with $h'(C) = 12$ (everything else equal).

Re-derive the pop order and the returned cost. State why optimality fails, naming the violated property. Give the smallest change to the graph that would restore optimality despite h' (the direct B→G edge). State the general lesson about trusting estimates. [7]

Exercise 3.2

3.2 Replace h with $h'(C) = 12$ (everything else equal).

Re-derive the pop order and the returned cost. State why optimality fails, naming the violated property. Give the smallest change to the graph that would restore optimality despite h' (the direct $B \rightarrow G$ edge). State the general lesson about trusting estimates. [7]

3.2

- With $h'(C) = 12$: C carries $f = 3 + 12 = 15$; pops become D, B, then G (through B, $f = 11$) – returned cost 11 ✓.
- Admissibility is violated at C ($12 > 5$): the goal pops while a cheaper open path exists ✓.
- Restoring fix: remove the direct $B \rightarrow G$ edge – then G is reachable only through C, and even the lying h' must pop C first ✓.
- Lesson: an estimate is a contract; optimality is only as honest as the tightest overestimate in the graph ✓.