COMS W4701: Artificial Intelligence

Lecture 2b: Informed Search

Tony Dear, Ph.D.

Department of Computer Science

School of Engineering and Applied Sciences

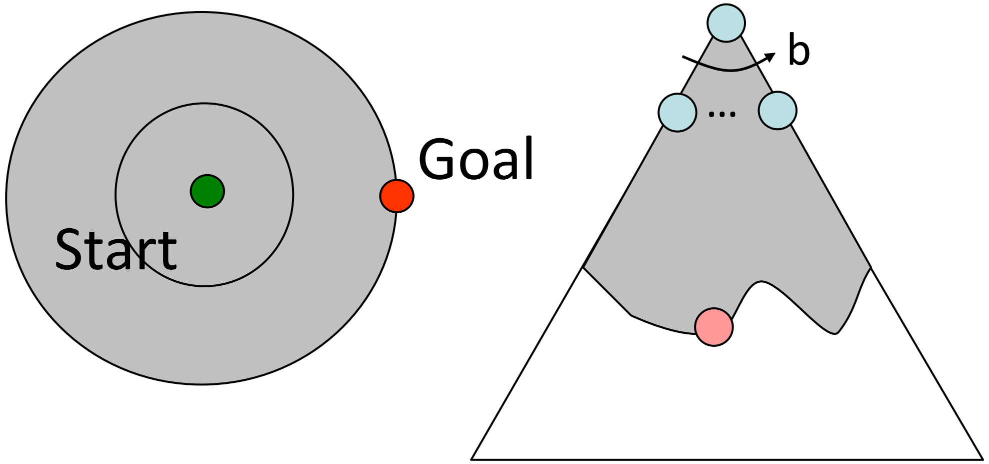
Today

- A* search
- Heuristic function properties
- A* search variations

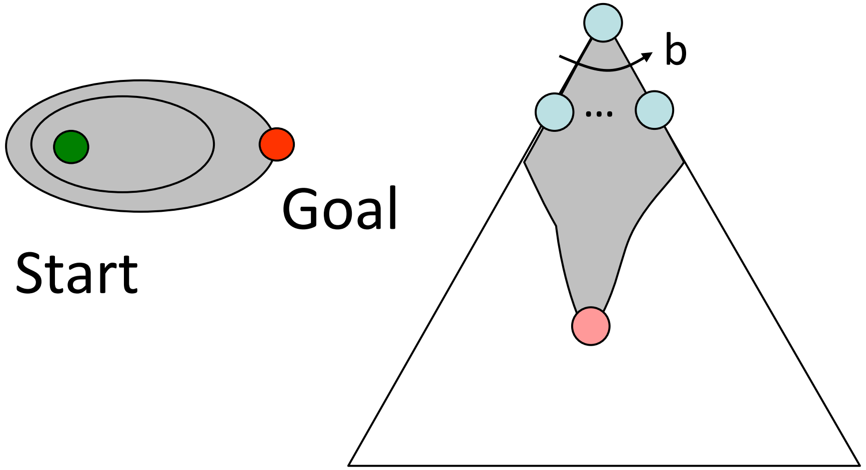
Informed (Heuristic) Search

- DFS, BFS, UCS were **uninformed** about goal state

- Now suppose we have additional, *domain-specific heuristics* that estimate state cost to goal
- **Heuristic function** $h(n)$: Estimated cost of cheapest path from state at node n to a goal state

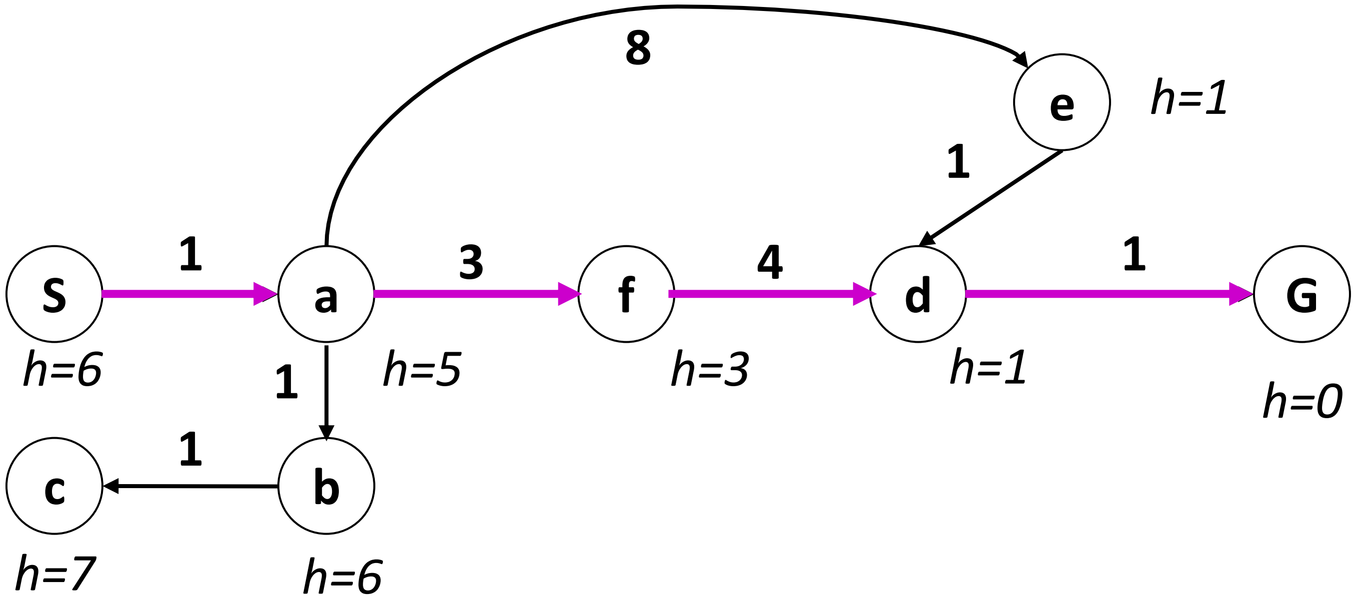


- We can combine heuristics with cumulative path costs
- Can still follow UCS strategy of expanding cheapest paths first, but prune or delay less promising paths



A* Search

- **A* search**: Evaluate each node based on both cumulative cost and heuristic value
- $f(n) = g(n) + h(n)$: Sum of path cost to n and estimated cost from n to goal
- Implementation: Priority queue using $f(n)$ as before



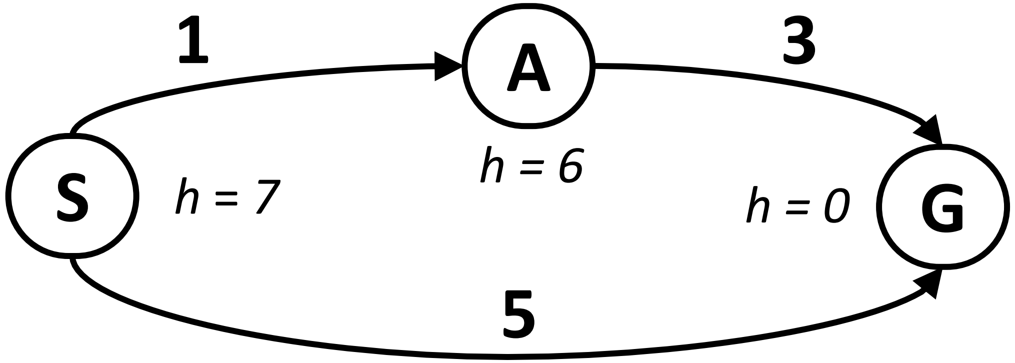
Frontier (node n, $f = g + h$):

1. $[(S, 6)]$
2. $[(a, 6)]$
3. $[(f, 7), (b, 8), (e, 10)]$
4. $[(b, 8), (d, 9), (e, 10)]$
5. $[(d, 9), (c, 10), (e, 10)]$
6. $[(G, 9), (c, 10), (e, 10)]$

Solution: S, a, f, d, G; cost: 9

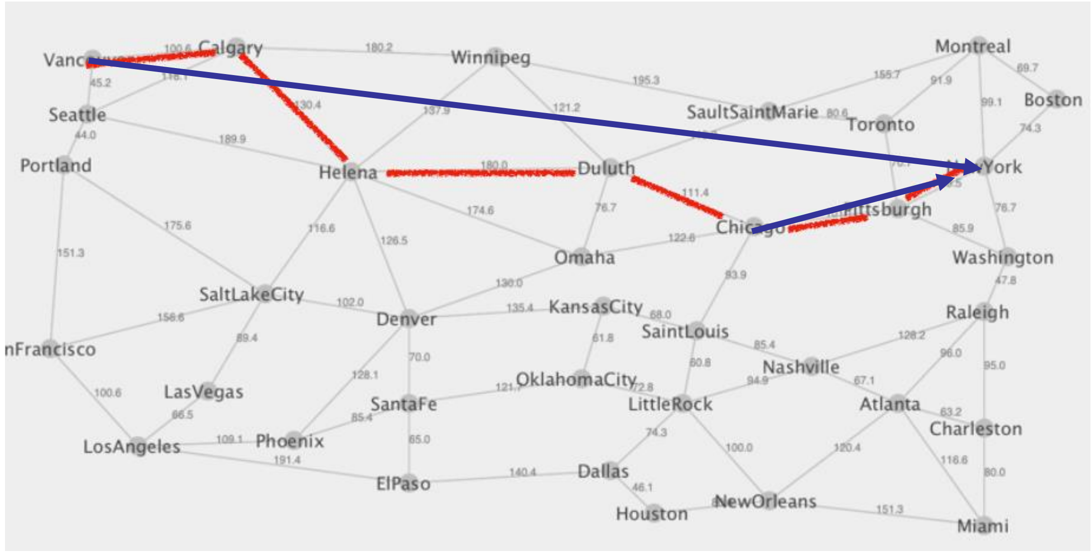
Admissibility

- In this graph, A* returns suboptimal solution S->G rather than optimal solution S->A->G
- $h(A)$ overestimated the cost from A!



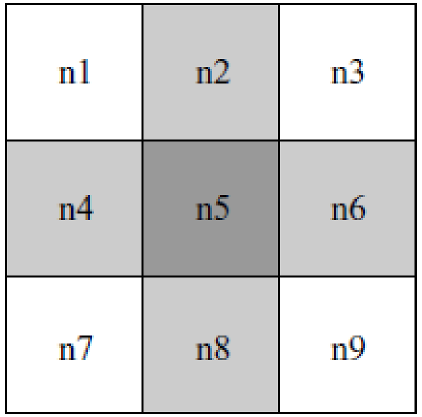
- A heuristic h is **admissible** if $h(n) \leq h^*(n)$ where $h^*(n)$ is true cost from n to goal
- In practice, we usually do not know $h^*(n)$

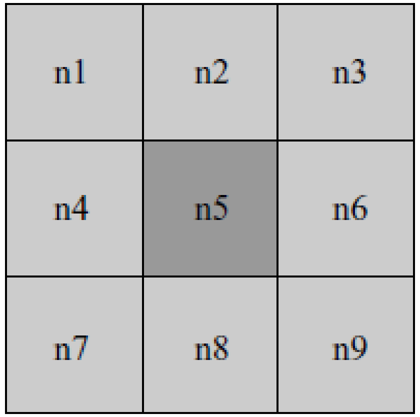
- One strategy to derive admissible heuristics: *relax* problem specifications by removing constraints, making them easier



Example: Grid Distances

- Grid navigation with goal $g = (x_g, y_g)$ and all transitions having cost 1
- **Manhattan distance** (L^1 norm): $h_1(x, y) = |x_g - x| + |y_g - y|$
- **Euclidean distance** (L^2 norm): $h_2(x, y) = \sqrt{(x_g - x)^2 + (y_g - y)^2}$
- If we have *4-point connectivity* (actions = {up, down, left, right}), both heuristics are admissible (h_2 *underestimates* true costs)
- If we have *8-point connectivity* (actions = above + 4 diagonal actions), *neither* is admissible, but $\frac{1}{\sqrt{2}} h_2$ is!





Heuristic Domination

- In a 4-point connected grid, L^1 norm always $\geq L^2$ norm between two cells
- h_1 **dominates** h_2 if $h_1(n) \geq h_2(n)$ for all n

- A* using h_1 will be more efficient and never expand more nodes than h_2
- h_1 reflects true costs more accurately

- Suppose we have collection of admissible heuristics $h_1, h_2, \dots, h_m$
- The composite heuristic $h(n) = \max\{h_1(n), \dots, h_m(n)\}$ is admissible and dominates all other heuristics!

Completeness and Optimality of A*

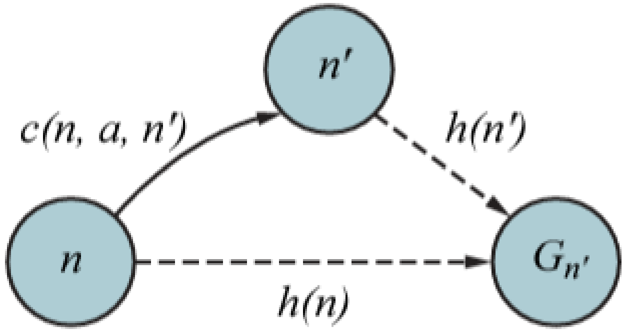
- Let $g^*(n)$ be cheapest path cost from start to node n
- If optimal goal has cost C^*, all nodes along path satisfy $g^*(n) + h^*(n) = C^*$

- Now suppose that heuristic function is admissible: $h(n) \leq h^*(n)\ \forall n$
- Then all nodes along optimal path satisfy $f(n) = g^*(n) + h(n) \leq C^*$
- All optimal solution nodes are expanded before any suboptimal goal with cost $C > C^*$

- A* is **complete**: If it exists, a solution will eventually be found and returned
- A* is **optimal**: Optimal solution will be returned before others with $C > C^*$
- A* *improves* upon UCS by skipping "useless" nodes that have $g^*(n) + h(n) > C^*$

Consistency

- A stronger heuristic property is **consistency** (triangle inequality)
- $h(n) - h(n') \leq c(n, a, n')$ for all n'
- Parent heuristic – child heuristic ≤ true cost



- All consistent heuristics are admissible (but not vice versa)
- *Most* admissible heuristics are also consistent in practice

- Consistency ensures that the first expansion of a node is along cheapest path
- Heuristic consistency ensures that A* is **optimally efficient**—it expands the fewest nodes compared to any other optimal algorithm with the same heuristic

Satisficing Solutions

- Like BFS or UCS, A* may suffer computationally intractable memory requirements
- Idea: Trade off admissibility for more accurate heuristics to reduce computation
- Return **satisficing solutions**—suboptimal, but "good enough"

- **Weighted A* search**: $f(n) = g(n) + \alpha h(n)$
- $\alpha > 1$ focuses the contour of reached states closer to the goal
- Generalizes A* ($\alpha = 1$), UCS ($\alpha = 0$), and greedy best-first ($\alpha \to \infty$)

- Fewer states expanded than A*, but may miss the optimal solution
- Suboptimality: If optimal solution has cost C^*, weighted A* solution may cost up to αC^*

Memory-Bounded Search

- We can also consider A* variants that are more memory-efficient

- **Beam search**: Fixed frontier size, only keep k best nodes at any iteration
- Or set threshold for discarding frontier nodes relative to current lowest f-value

- **Iterative-deepening A*** (IDA*): Depth-first iterative deepening search, only considering nodes with f-value not exceeding current cutoff value
- In each iteration, increment cutoff to *smallest* f-value of the skipped nodes

- IDA* yields linear spatial complexity of DFS; each iteration can progress steadily down the tree if f-values tend to increase consistently along paths

Summary

- Domain-specific heuristics can guide search toward goal
- A* search combines true costs and heuristics to evaluate frontier nodes
- Admissible heuristics do not overestimate true costs -> A* is optimal
- Consistent heuristics satisfy triangle inequality -> A* is optimally efficient
- Many other variations of A* to deal with suboptimality, memory limits