Introduction to Artificial Intelligence

Lecture: Informed Search

Outline

- Informed Search Strategies
- Best-first Search
- Greedy Best-first Search
- A* Search
- Heuristic Dominance

Informed search strategies

- Use problem-specific knowledge beyond the definition of the problem itself
- Find solutions more efficiently
- Provide significant speed-up in practice

What are heuristics?

- Additional knowledge of the problem is imparted to the search algorithm using heuristics.
- A heuristic is any practical approach to problem solving sufficient for reaching an immediate goal where an optimal solution is usually impossible.
  - Not guaranteed to be optimal, perfect, logical, or rational
  - Speed up the process of finding a satisfactory solution
  - Ease the cognitive load of making a decision

What are heuristics?

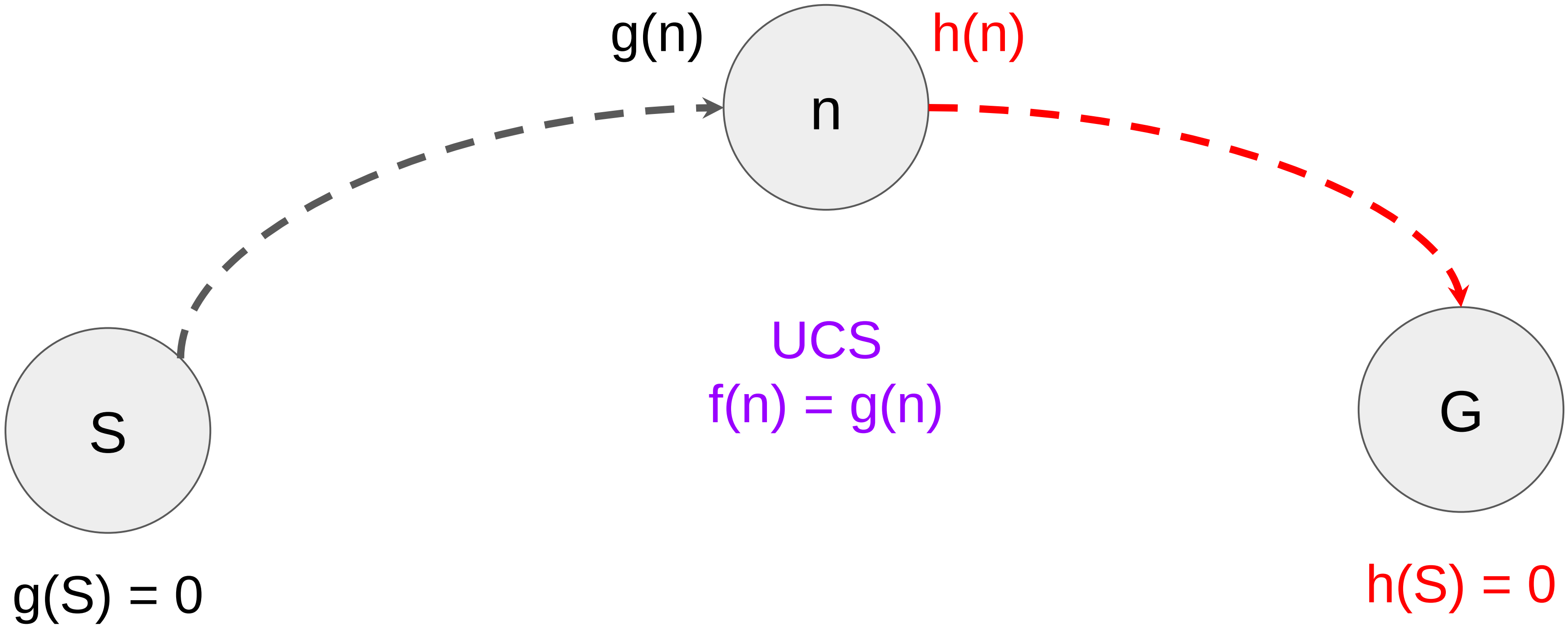
Best-first search

- An instance of the general TREE-SEARCH or GRAPH-SEARCH algorithm
- A node is selected for expansion based on an evaluation function, $\boldsymbol{f(n)}$.
- Node with the lowest $f(n)$ is expanded first
- The choice of f determines the search strategy.

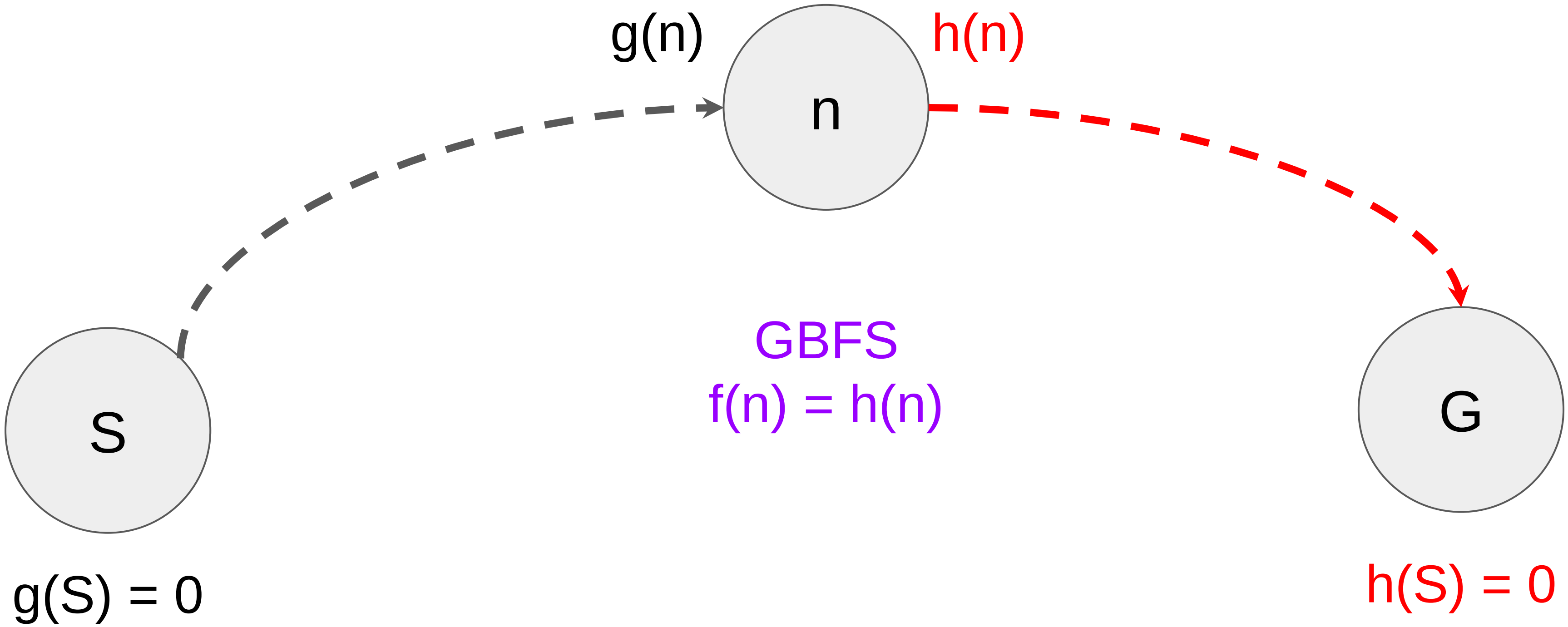
Heuristic function

- h(n) $\rightarrow$ estimated cost of the cheapest path from the state at node n to a goal.
- Unlike $g(n)$, h(n) depends only on the state at that node
- Assumption of h(n)
  - Arbitrary, nonnegative, problem-specific functions
  - Constraint: if n is a goal node, then $\boldsymbol{h}(\boldsymbol{n}) = \mathbf{0}$

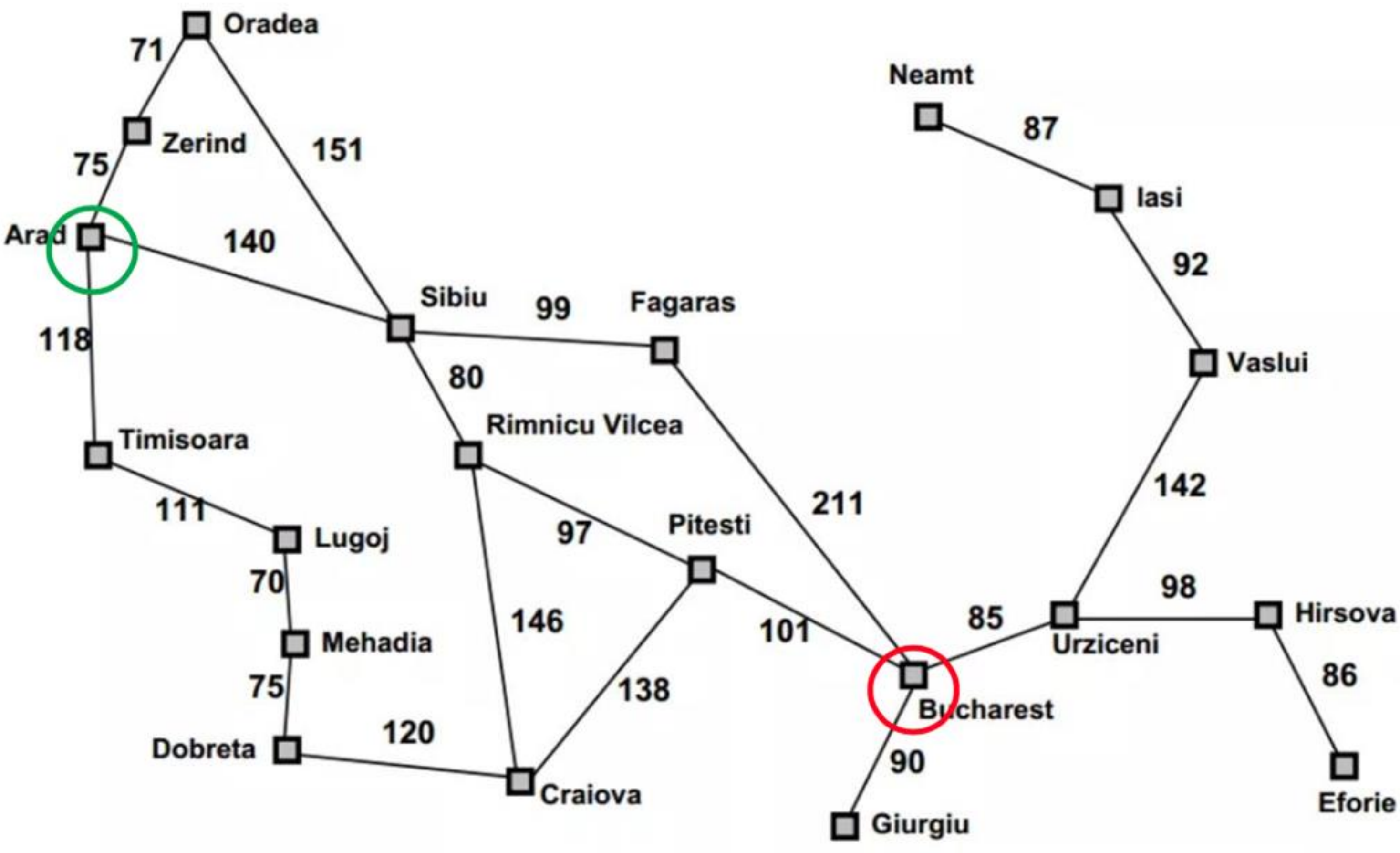
g(n) vs h(n)



Greedy best-first search



Straight-line distance heuristic



Straight–line distance to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Greedy best-first search

- Draw the search tree
- Determine the path
- Compute the path-cost

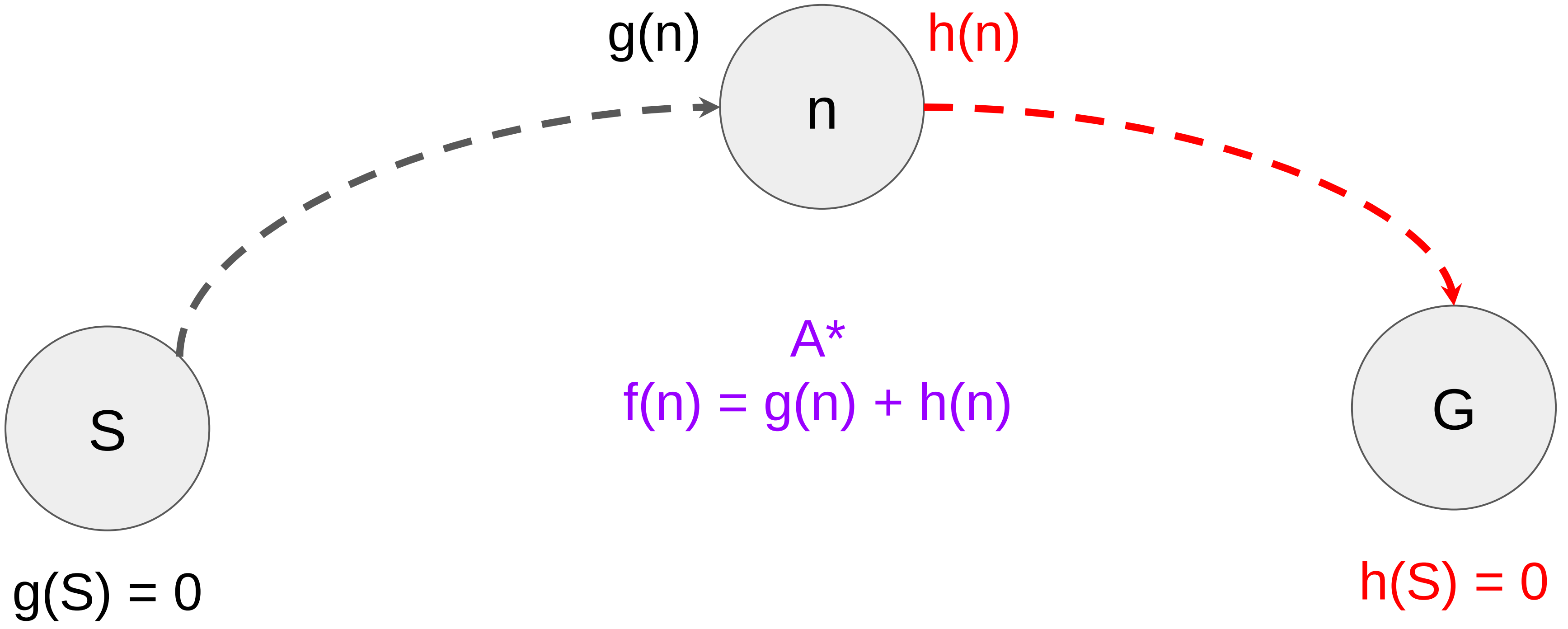
Greedy best-first search: Evaluation

- Completeness:
  - NO – may get stuck forever
- Time complexity
  - $O(b^m) \rightarrow$ reduced substantially with a good heuristic
- Space complexity
  - $O(b^m)$ – keeps all nodes in memory
- Optimality
  - NO

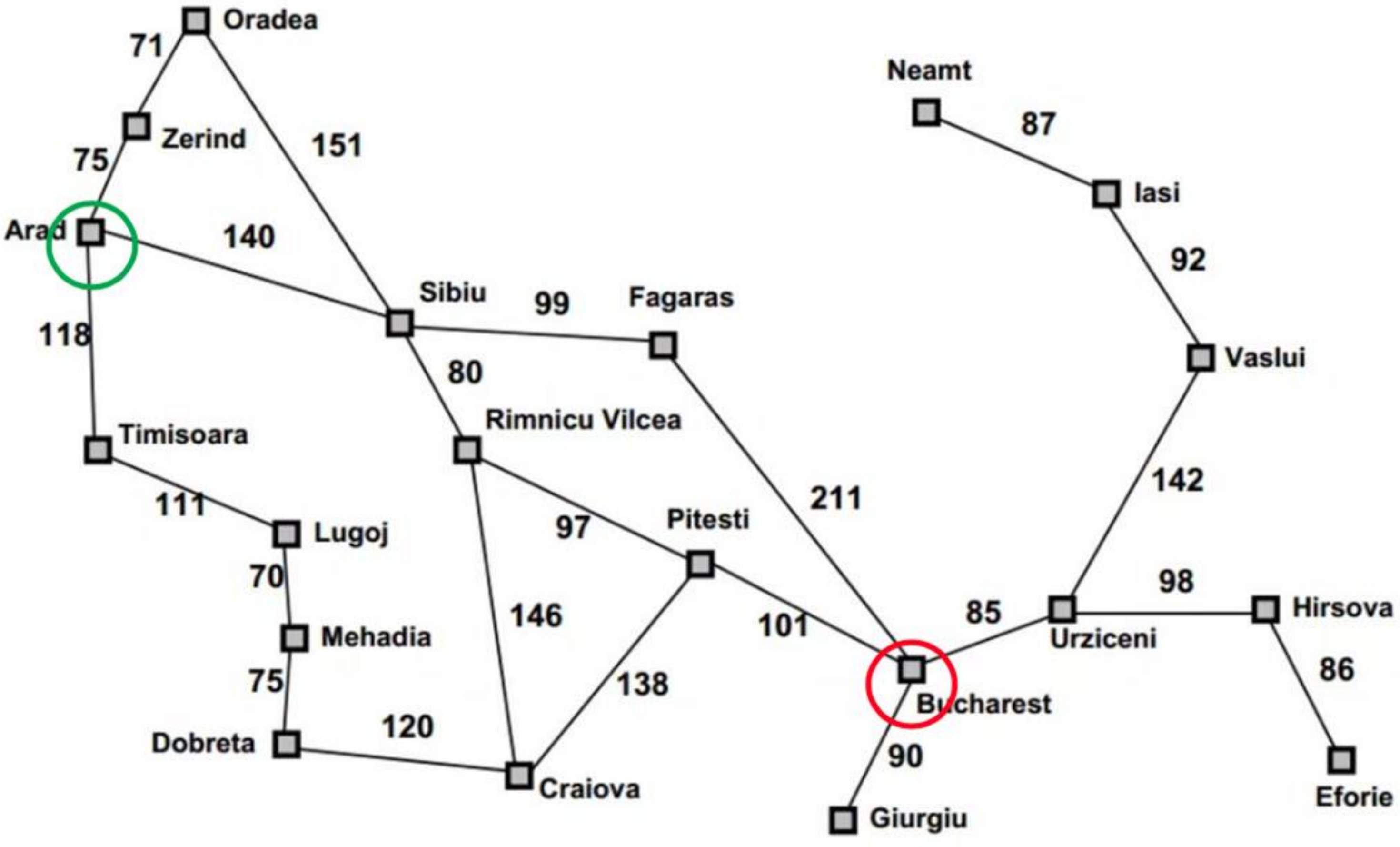
A* search

- The most widely known form of best-first search
- Evaluate nodes by $\boldsymbol{f(n)}$ = $\boldsymbol{g(n)}$ + $\boldsymbol{h(n)}$
    - $g(n)$ is the cost to reach the node n
    - h(n) is the cost to get from n to the goal
    - $f(n)$ = estimated cost of the cheapest solution through n

A* search



A* search



Straight–line distance to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

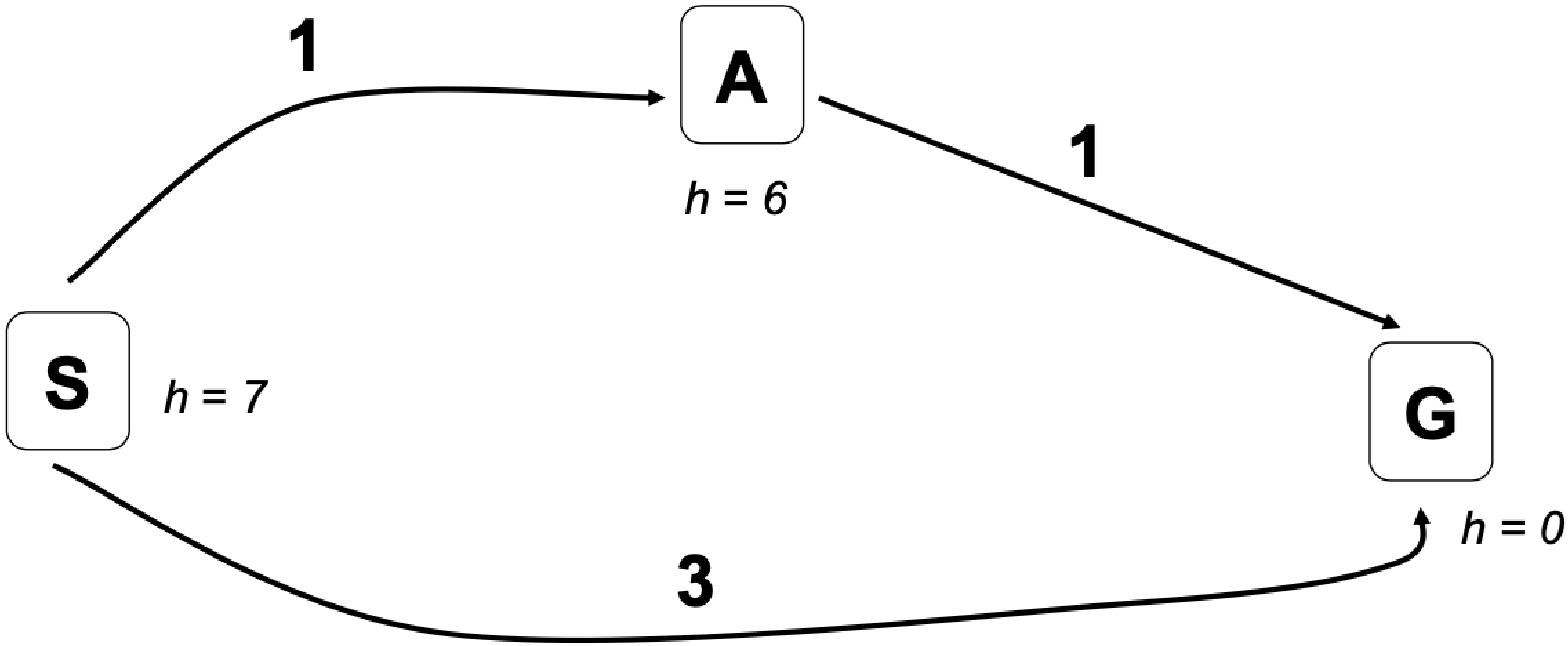
A* search

- Draw the search tree
- Determine the path
- Compute the path-cost

A* search: Evaluation

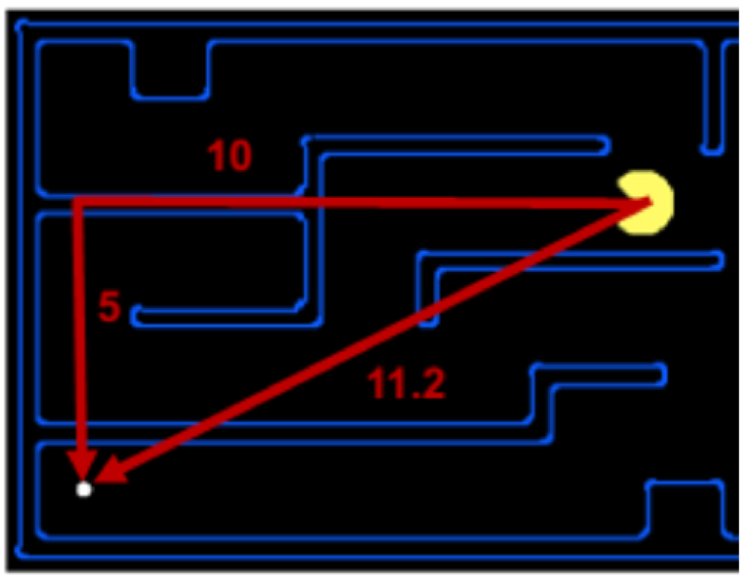
- Completeness
  - YES if all step costs exceed some finite ϵ and if b is finite
  - (review the condition for completeness of UCS)
- Optimality
  - YES – with conditions on heuristic being used
- Time complexity
  - Exponential
- Space complexity
  - Exponential (keep all nodes in memory)

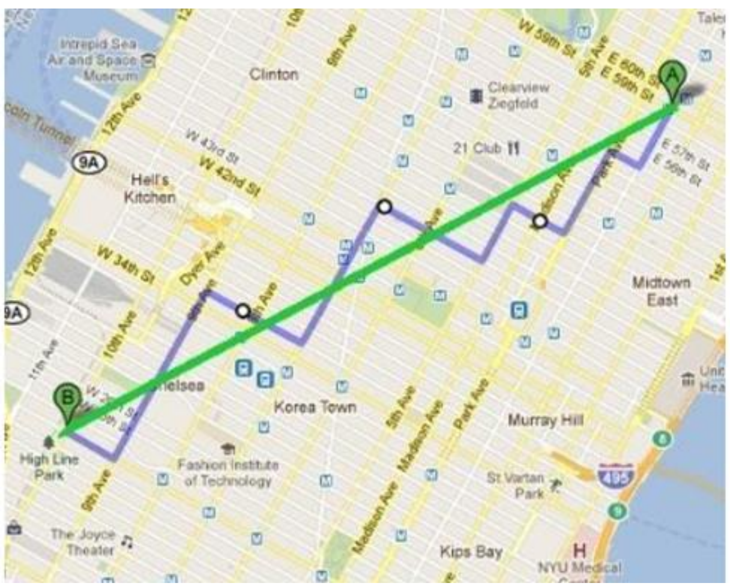
A* search is not always optimal



Heuristic: Admissibility

- h(n) must be an admissible heuristic
  - Never overestimate the cost to reach the goal → optimistic
  - E.g: Euclidean distance, Manhattan distance, straight-line distance, etc.



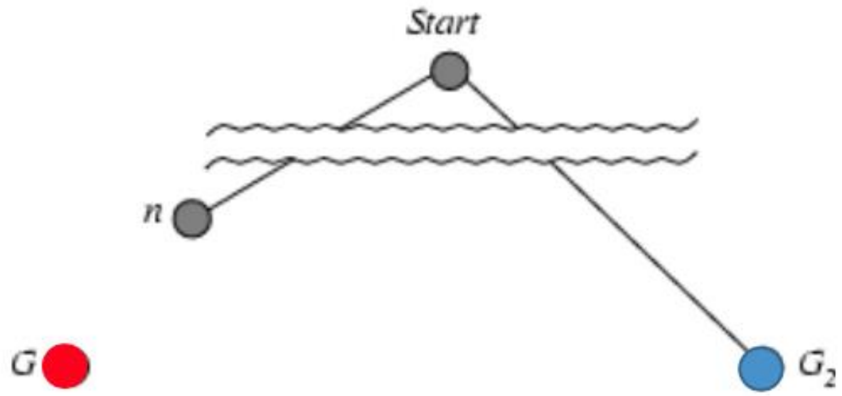


Heuristic: Admissibility

- If h(n) is admissible, A* using TREE-SEARCH is optimal

Suppose some suboptimal goal $G2$ has been generated and is in the frontier.

Let n be an unexpanded node in the frontier such that n is on a shortest path to an optimal goal G.



$f(G2) = g(G2)$ since $h(G2) = 0$

$g(G2) > g(G)$ since G2 is suboptimal

$f(G) = g(G)$ since $h(G) = 0$

$\Rightarrow f(G2) > f(G)$ (1)

$h(n) \le h^*(n)$ since h is admissible

$g(n) + h(n) \le g(n) + h^*(n) \Rightarrow f(n) \le f(G)$ (2)

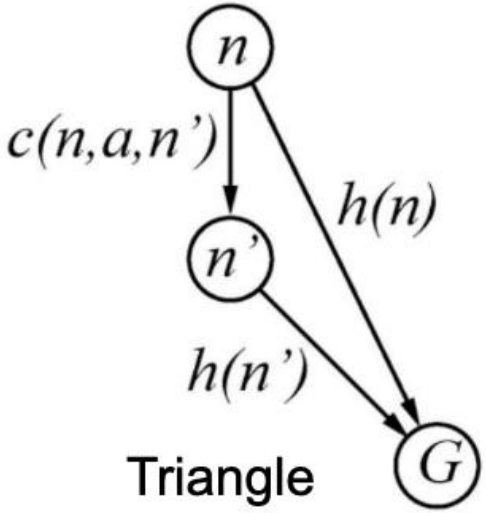
From (1), (2): $f(G2) > f(n) \rightarrow$ A will never select $G2$ for expansion*

Heuristic: Consistency

- Admissibility is insufficient for graph search.
  - The optimal path to a repeated state could be discard if it is not the first one selected.
- h(n) is consistent if for every node n, every successor n' of n generated by any action a,

$$\boldsymbol{h(n) \leq c(n, a, n') + h(n')}$$

- Every consistent heuristic is also admissible.



Triangle inequality

Heuristic: Consistency

- If h(n) is consistent, A* using GRAPH-SEARCH is optimal

If h(n) is consistent, the values of $f(n)$ along any path are non-decreasing.

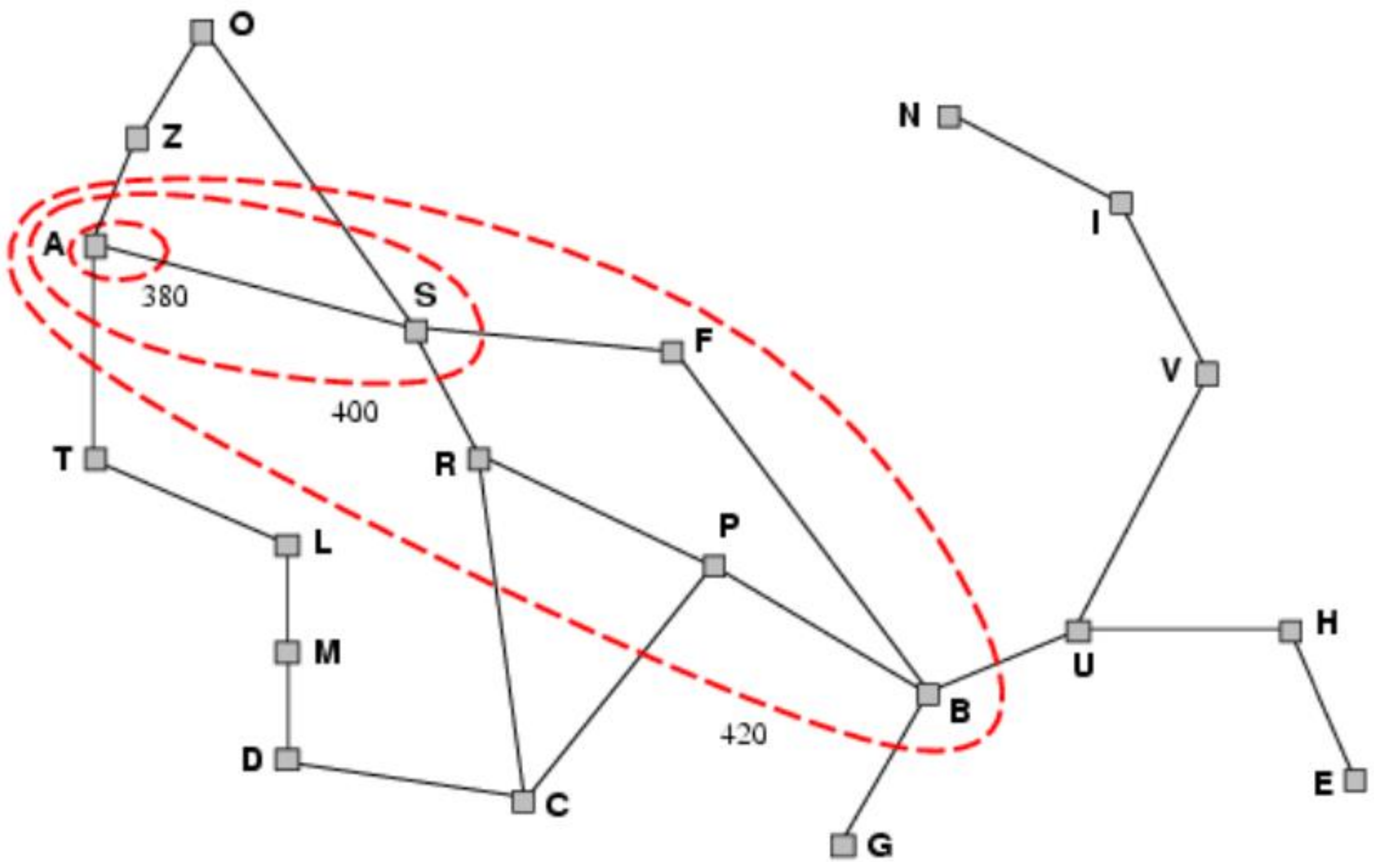
Suppose n' is a successor of $n \rightarrow g(n') = g(n) + c(n,a,n')$

$$f(n') = g(n') + h(n') = g(n) + c(n,a,n') + h(n') \geq g(n) + h(n) = f(n)$$

Whenever A selects a node n for expansion, the optimal path to that node has been found.*

Contours of A* search

- A* expands nodes in order of increasing f-value
- A* will expand all nodes with costs $f(n) < C^*$



Heuristic dominance

- Given two admissible heuristics, h1 and h2
- If h2(n) ≥ h1(n), for all n, then h2 dominates h1
- A* using h2 will never expand more nodes than A* using h1
- Better to use a heuristic function with higher values, provided it is consistent and its computation time is not too long.

Homework

- Conduct homework in the given notebook.

References

- Stuart Russell and Peter Norvig. 2009. Artificial Intelligence: A Modern Approach (3rd ed.). Prentice Hall Press, Upper Saddle River, NJ, USA.
- Lê Hoài Bắc, Tô Hoài Việt. 2014. Giáo trình Cơ sở Trí tuệ nhân tạo. Khoa Công nghệ Thông tin. Trường ĐH Khoa học Tự nhiên, ĐHQG-HCM.
- Nguyễn Ngọc Thảo, Nguyễn Hải Minh. 2020. Bài giảng Cơ sở Trí tuệ Nhân tạo. Khoa Công nghệ Thông tin. Trường ĐH Khoa học Tự nhiên, ĐHQG-HCM.