Elective Course
Course Code: CS4103
Autumn 2025-26

Lecture #10

Artificial Intelligence for Data Science

Week-3: PROBLEM SOLVING BY SEARCH
Introduction to Informed Search [Part-II]
(A* Search and More on Heuristics)

Course Instructor:
Dr. Monidipa Das
Assistant Professor
Department of Computational and Data Sciences
Indian Institute of Science Education and Research Kolkata, India 741246

A* — A Better Best-First Strategy

2

Greedy Best-first search
- minimizes estimated cost $h(n)$ from current node n to goal
- is informed but almost always suboptimal and incomplete

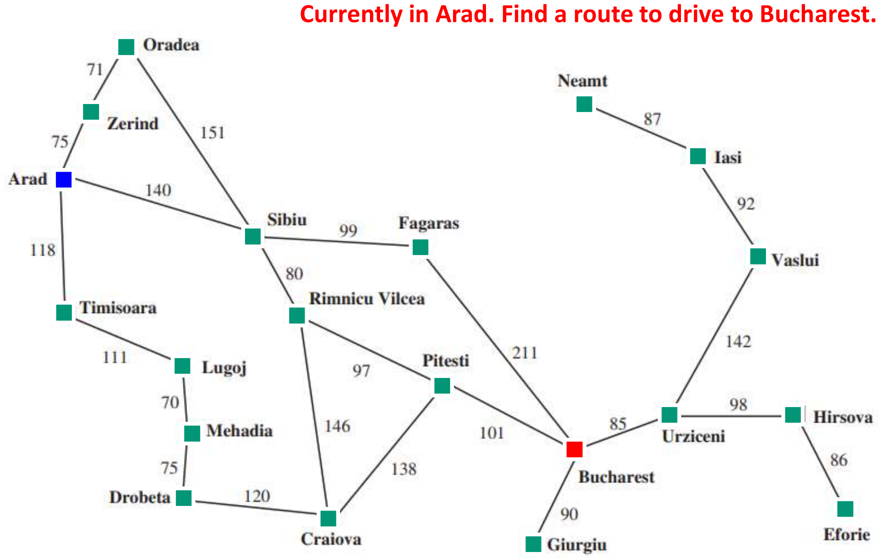
Uniform cost search
- minimizes actual cost $g(n)$ to current node n
- is, in most cases, optimal and complete but uninformed

A* search
- combines the two by minimizing $f(n) = g(n) + h(n)$
- is, *under reasonable assumptions*, optimal and complete, and also informed

Romania with Step Costs in Km

3

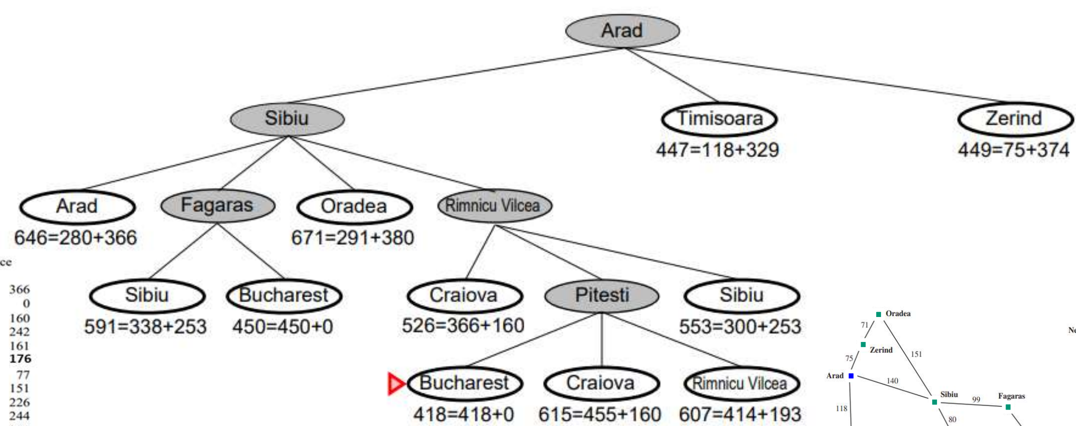
Currently in Arad. Find a route to drive to Bucharest.



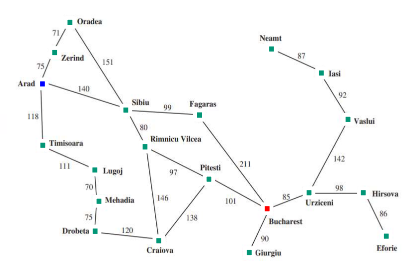
Straight−line distance to Bucharest	h_{SLD}
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	**176**
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

A* Search Example

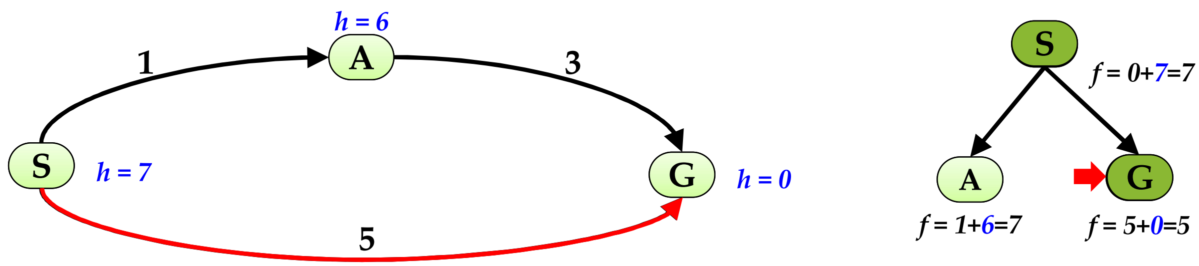
4



Straight−line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	**176**
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



Is A* Optimal?



Not optimal!

What went wrong?

- Actual cost to reach the goal < estimated cost of reaching the goal
- We need estimates to be less than actual costs!

A* with TREE-SEARCH is optimal if $h(n)$ is an **admissible heuristic**

Admissible Heuristics

- A heuristic $h(n)$ is *admissible* **if it never overestimates the actual/true cost to reach the goal**:

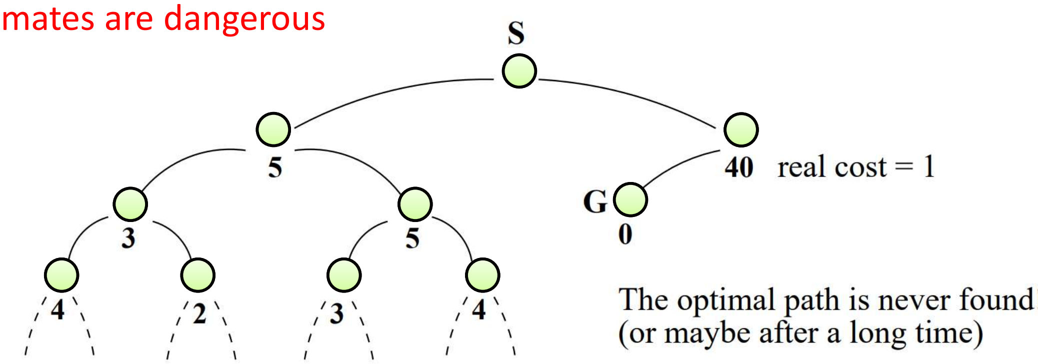
$$0 \leq h(n) \leq h^*(n)$$

where $h^*(n)$ is the true cost to a nearest goal

- Admissible heuristics are by nature optimistic
- E.g., $h_{SLD}(n)$ never overestimates the actual road distance
- Finding good, cheap, admissible heuristics is the key to success

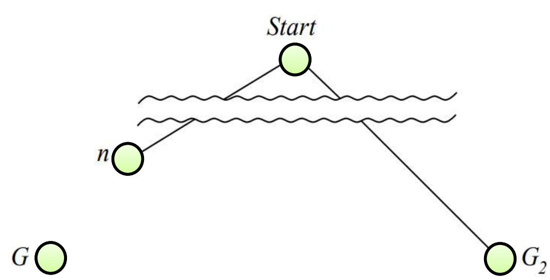
A* Search: Why an Admissible Heuristic

- If h is admissible, $f(n)$ never overestimates the actual cost of the best solution through n
- Overestimates are dangerous



Optimality of A* Tree Search

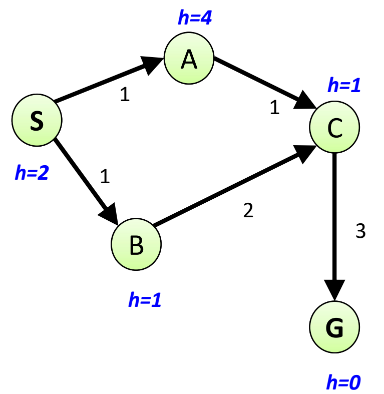
- Suppose some suboptimal goal G_2 has been generated and is in the queue. Let n be an unexpanded node on a least-cost path to an optimal goal G
- $f(G_2) \ = \ g(G_2)$ since $h(G_2) \ = \ 0$
  $> C^*$ since G_2 is suboptimal
  $\geq g(n) + h(n)$ since **h is admissible**
- So, $f(G_2) \geq f(n)$
- Since $f(G_2) \geq f\ (n\ )$,
  A* will never select G_2 for expansion



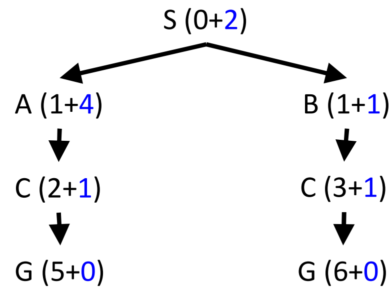
Optimality of A* Graph Search

9

State space graph



Search tree



A* Graph Search Gone Wrong?

Simple check against expanded set blocks C
Fancy check allows new C if cheaper than old
but requires recalculating C's descendants

The problem can be fixed by imposing requirement of **consistency** on h

Consistent Heuristics

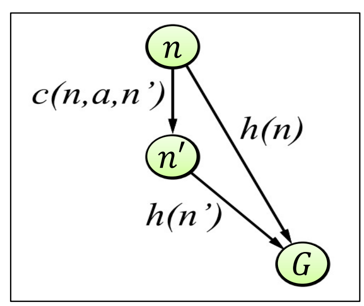
10

- **A heuristic is consistent if**

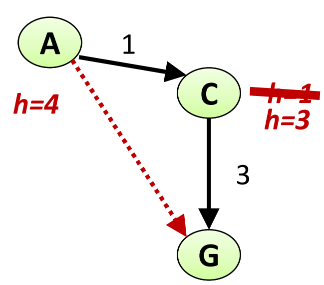
$$h(n) \leq c(n,a,n') + h(n')$$

- If f is consistent, we have

$$\begin{aligned} f(n') &= g(n') + h(n') \\ &= g(n) + c(n,a,n') + h(n') \\ &\geq g(n) + h(n) = f(n) \end{aligned}$$



i.e., $f(n)$ is nondecreasing along any path



Consequences of consistency:

- The f value along a path never decreases
- A* graph search is optimal

Optimality of A*

11

- **Tree search:**
  - A* is optimal if heuristic is admissible
- **Graph search:**
  - A* optimal if heuristic is consistent
- Consistency implies admissibility
- Most natural admissible heuristics tend to be consistent, especially if from relaxed problems
- A* is *optimally efficient* for h:
  - no other optimal strategy using h expands fewer nodes than A*
  - Any algorithm that does not expand all nodes with $f(n) < C^*$ runs the risk of missing the optimal solution.

Homework:
Prove that if a heuristic is consistent, it must be admissible.

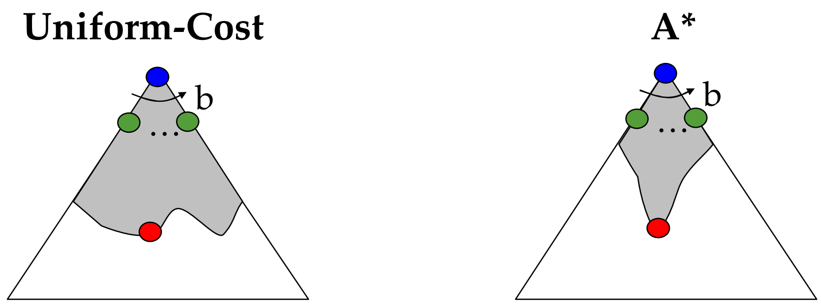
Properties of A*

12

- **Complete?**
  - Yes, unless there are infinitely many nodes n with $f(n) \leq f(G)$
- **Time complexity?**
  - $O(b^{\epsilon d})$
  - where $\epsilon = |h(n_0) - h^*(n_0)| / h^*$
    - n_0 = start state
    - h^* = actual cost to goal state
- **Space complexity?**
  - $O(b^m)$, as in Greedy Best-First — may end up with all nodes in memory
- **Optimal?**
  - Yes if h is admissible

UCS vs. A*

13



Uniform-cost expands equally in all "directions"

A* expands mainly toward the goal

Dr. Monidipa Das, Department of CDS, IISER Kolkata

14

More on Heuristics

Dr. Monidipa Das, Department of CDS, IISER Kolkata

Heuristics

15

- **All domain knowledge** used in the search is encoded in the heuristic function $h()$.
- Heuristic search is an example of a **"weak method"** because of the limited way that domain-specific information is used to solve the problem
- **Examples:**
  - Missionaries and Cannibals: Number of people on starting river bank
  - 8-Puzzle: Number of tiles out of place
  - 8-Puzzle: Sum of distances of each tile from its goal position
- A* search is optimal with an **admissible heuristic** function h

Heuristics [contd.]

16

- In general:
  - $h(n) \geq 0$ for all nodes n
  - $h(n) = 0$ implies that the node n is a goal node
  - $h(n) = \infty$ implies that n is a dead-end that can never lead to a goal

Examples of Admissible Heuristics

17

- **8-puzzle problem:**
  - $h_1(n)$ = number of tiles in the wrong position at state n
  - $h_2(n)$ = sum of the Manhattan distances of each tile from its goal position

5	4	
6	1	8
7	3	2

Start State

1	4	7
5	6	2
8	3	

Goal State

  - $h_1(Start)$ = 6
  - $h_2(Start)$= 2+1+0+0+1+1+4+3 = 12

Effectiveness of Heuristic Functions

18

- Let
  - h be a heuristic function for A*
  - N be the total number of nodes expanded by one A* search with h
  - d is the depth of the found solution

- The effective branching Factor (EBF) of h is the value b* that solves the equation

$$x^d + x^{d-1} + \cdots + x^2 + x + 1 - N = 0$$

(the branching factor of a uniform tree with N nodes and depth d)

A heuristics h for A* is effective in practice if its average EBF is close to 1

Dominance

19

- A heuristic function h_2 dominates a heuristic function h_1 for a problem P if $h_2(n) \geq h_1(n)$ for all nodes n in P's space
- **Ex.:** the 8-puzzle
  - h_2 = total Manhattan distance dominates
  - h_1 = number of misplaced tiles
- With A*, if h_2 is admissible and dominates h_1, then it is always better for search: A* will never expand more nodes with h_2 than with h_1
- **Note:** If h_2 dominates h_1, then $EFB(h_2) \leq EFB(h_1)$
- **What if neither of h_1, h_2 dominates the other?**
  - If both h_1, h_2 are admissible, use $h(n) = \max(h_1(n), h_2(n))$

Example

20

Comparison of the **search costs** and **effective branching factors** for the ITERATIVE-DEEPENING-SEARCH and A∗ algorithms with h1, h2

	Search Cost (nodes generated)			Effective Branching Factor		
d	IDS	$A^*(h_1)$	$A^*(h_2)$	IDS	$A^*(h_1)$	$A^*(h_2)$
2	10	6	6	2.45	1.79	1.79
4	112	13	12	2.87	1.48	1.45
6	680	20	18	2.73	1.34	1.30
8	6384	39	25	2.80	1.33	1.24
10	47127	93	39	2.79	1.38	1.22
12	3644035	227	73	2.78	1.42	1.24
14	–	539	113	–	1.44	1.23
16	–	1301	211	–	1.45	1.25
18	–	3056	363	–	1.46	1.26
20	–	7276	676	–	1.47	1.27
22	–	18094	1219	–	1.48	1.28
24	–	39135	1641	–	1.48	1.26

Dr. Monidipa Das, Department of CDS, IISER Kolkata

Devising Heuristic Functions

21

- How do we devise good heuristic functions for a given problem?
- Typically, that depends on the problem domain
- However, there are some general techniques that work reasonably well across several domains

22

Questions?