Solving Problems by Searching

ARTIFICIAL INTELLIGENCE

JUCHEOL MOON

1

Problem-solving agents

- On holiday in Romania; currently in Arad.
- Flight leaves tomorrow from Bucharest

2

2

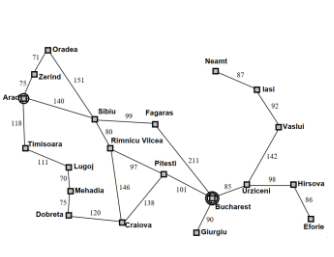
Holiday in Romania

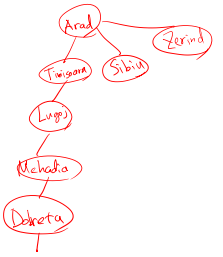
- Formulate goal:
  - be in Bucharest
- Formulate problem:
  - states: cities (nodes)
  - actions: move to the next cities
- Find solution:
  - Path from Arad to Bucharest

3

3

Tree search example





4

4

Uninformed search strategies

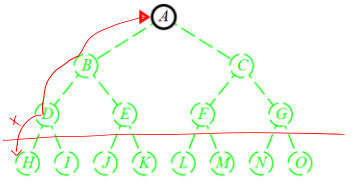
- Uninformed strategies use only the information available in the problem definition
  - Breadth-first search
  - Depth-first search
  - Uniform-cost search (Dijkstra's algorithm)
  - Depth-limited search
  - Iterative deepening search

5

5

Depth-limited search

- depth-first search with depth limit l,
  - i.e., nodes at depth l have no successors

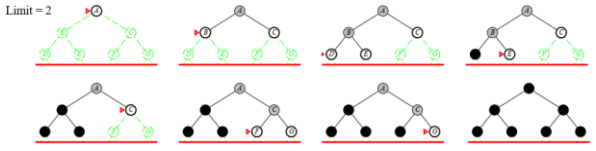


6

6

Iterative deepening search

```
function Iterative-Deepening-Search(problem)
    for depth 0 to ∞ do
        depth-Limited-Search(problem, depth)
```



7

7

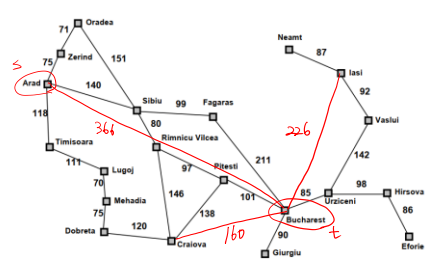
Informed search strategy (Best-first search)

- Idea: use an evaluation function for each node
  - estimate of desirability
- Expand most desirable unexpanded node
- Implementation:
  - fringe is a queue sorted in decreasing order of desirability

8

8

Romania with step costs in km



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	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

9

9

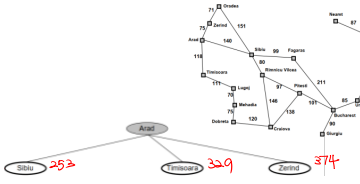
Greedy search

- Evaluation function $h(n)$ (heuristic)
  - estimate of cost from n to the closest goal
  - $h(n)$ = straight-line distance from n to Bucharest
- Greedy search expands the node that appears to be closest to goal

10

10

Greedy search example

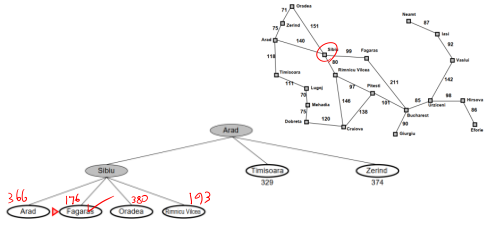


Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

11

11

Greedy search example

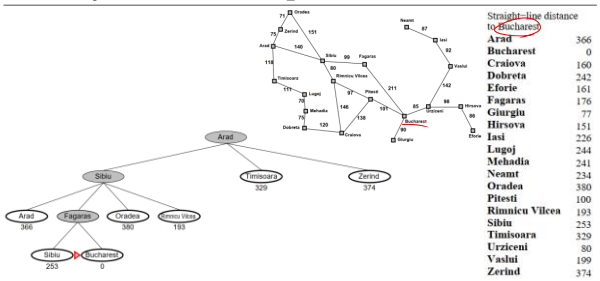


Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

12

12

Greedy search example



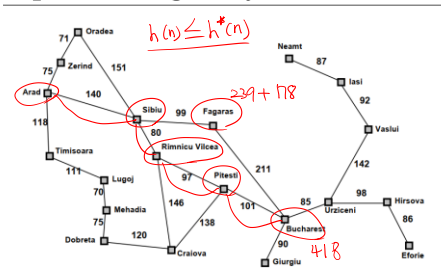
Straight-line distance to Bucharest

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	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

13

13

Update the greedy 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

14

14

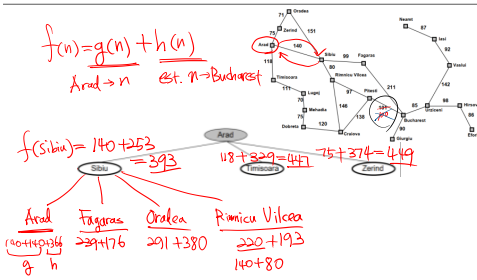
A∗ search

- Idea: avoid expanding paths that are already expensive
- Evaluation function $f(n) = g(n) + h(n)$
  - $g(n)$ = cost so far to reach n
  - $h(n)$ = estimated cost to goal from n
  - $f(n)$ = estimated total cost of path through n to goal

15

15

A∗ search example



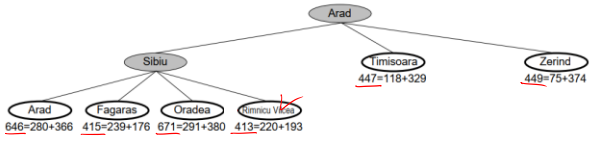
Straight-line distance to Bucharest

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	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

16

16

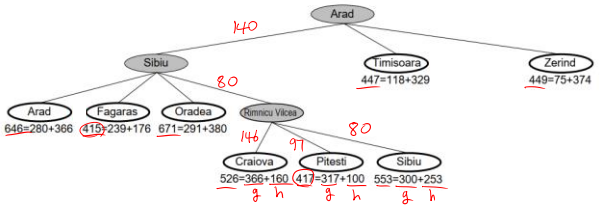
A∗ search example



17

17

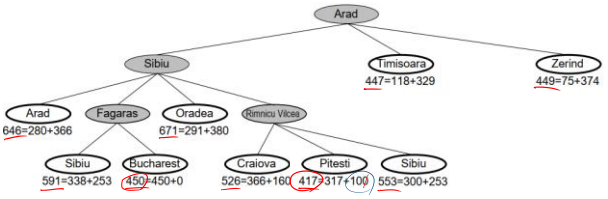
A∗ search example



18

18

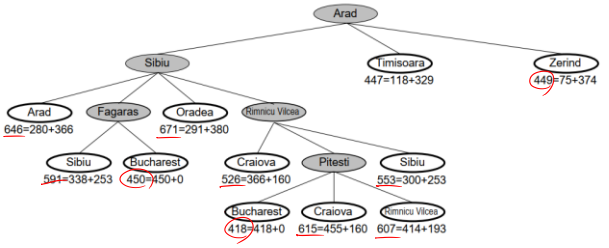
A∗ search example



19

19

A∗ search example

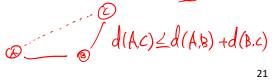


20

20

A∗ search

- $h(n)$ = straight-line distance from n to Bucharest
- Can $h(n)$ over estimate the actual road distance?
  - Yes / No
- $h(n) \leq h^*(n)$ where $h^*(n)$ is the true cost from n?
  - A∗ search uses an admissible heuristic
  - Admissible heuristics are by nature optimistic because they think the cost of solving the problem is (less) / more) than it actually is.



21

21