

CS 480

Introduction to Artificial Intelligence

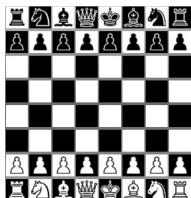
September 9th, 2021

Announcements / Reminders

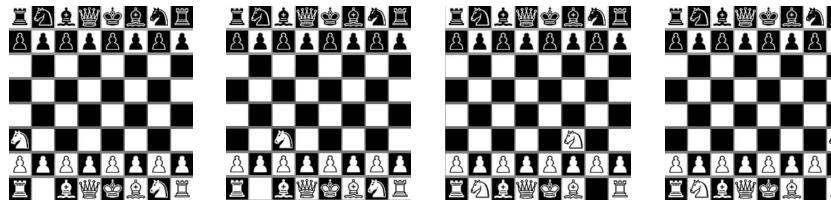
- Contribute to the discussion on Blackboard, please**
- Please follow the Week 03 To Do List instructions**

CORRECTION: Chess: State Expansion

Initial
State



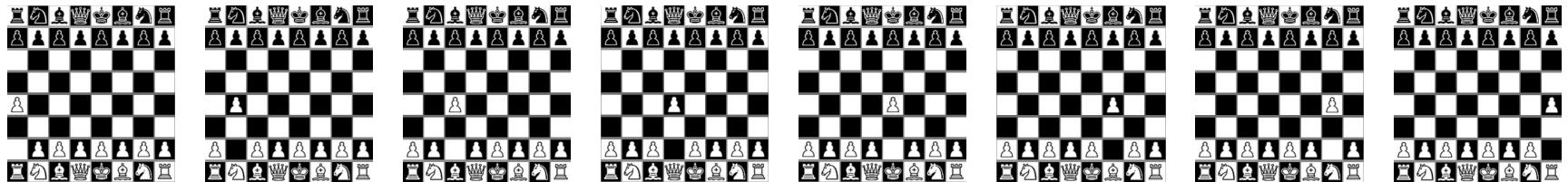
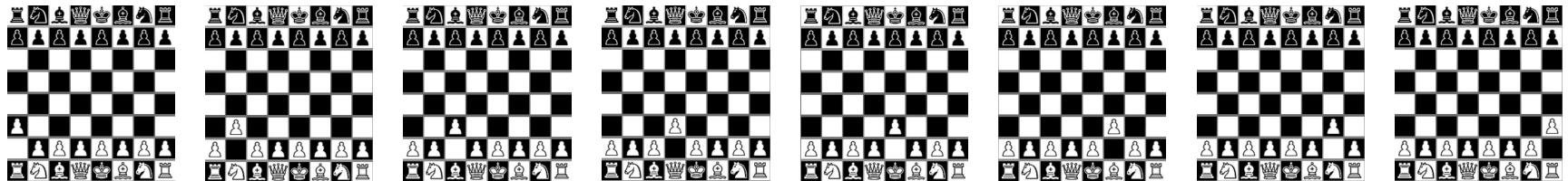
Use game rules to generate subsequent possible game tree states / nodes!



20 Possible **legal** first moves:

16 pawn moves

4 **knight** moves



Plan for Today

- Problem Solving: Informed Searching

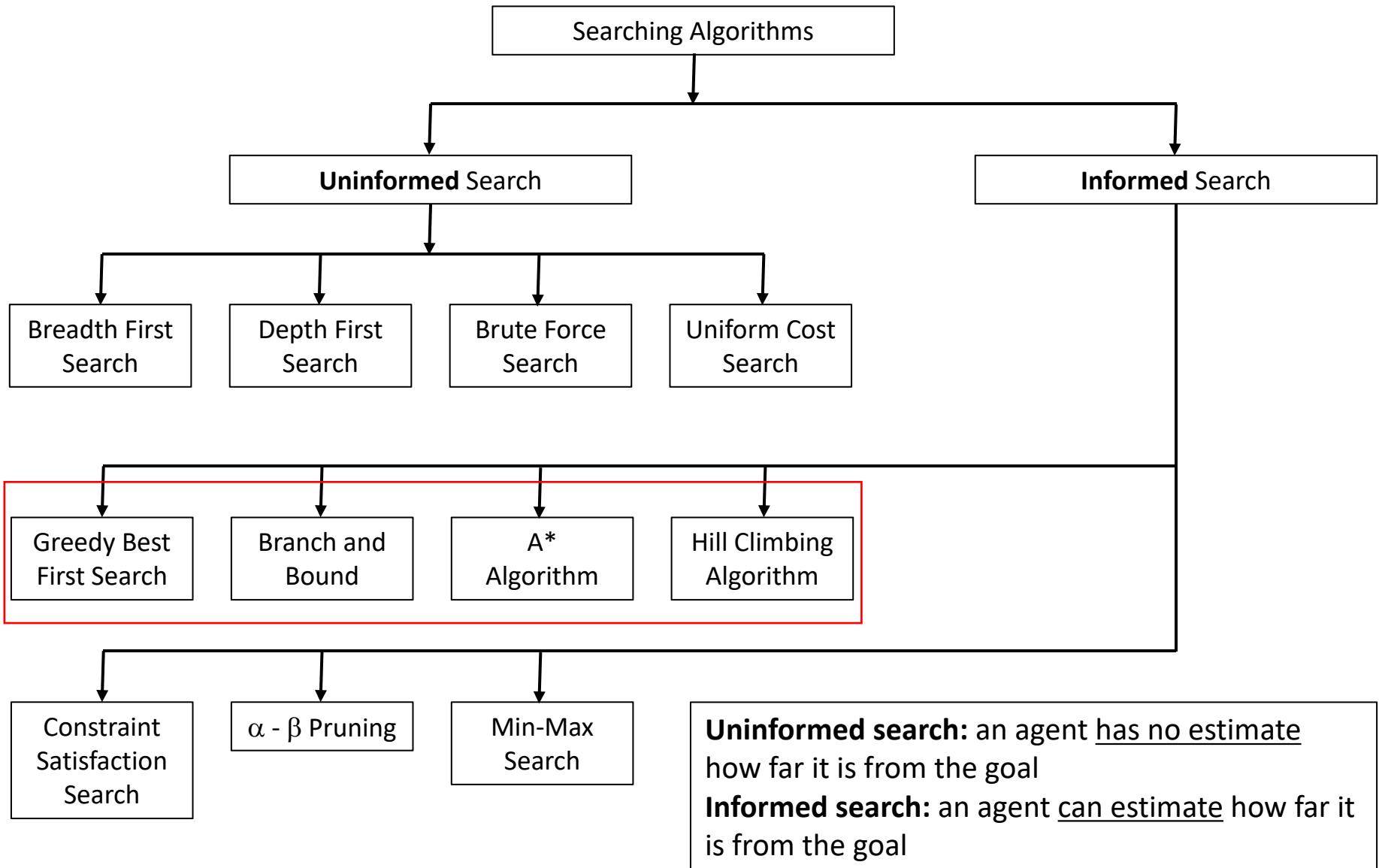
Uninformed Search

- Traverse the search tree, possibly through all legal paths, to find a solution / reach goal state
- Search tree size can be large (or infinite)
 - Use **node expansion / generation** as you traverse
- Avoid **repeated states** (those cause “**loops**”)
 - Keep track of already **visited** states
- Search tree depth can be a challenge
 - Use **Iterative Deepening** or **Depth Limits**
- Extra problem information? → informed search

DFS vs. BFS: When to Use? Tips

- Breadth First Search
 - branching factor **b** is not excessive
 - solution is expected to exist at a reasonable level (depth **d** is reasonable)
 - some search paths can be very deep
- Depth First Search
 - branching factor **b** is relatively large
 - solution is expected to exist at a relatively shallow level (depth **d** is low)
 - search paths are not excessively deep

Selected Searching Algorithms



Informed Search: the Idea

When traversing the search tree **use domain knowledge / heuristics to avoid search paths that are likely to be fruitless**

Evaluation function

Calculate / obtain:

$$f(n) = f(\text{State } n)$$

$$f(n) = f(\text{relevant information about State } n)$$

A state n with minimum (or maximum) $f(n)$
should be chosen for expansion

What about ties?

Best-First Search

```
function BEST-FIRST-SEARCH(problem, f) returns a solution node or failure
    node  $\leftarrow$  NODE(STATE=problem.INITIAL)
    frontier  $\leftarrow$  a priority queue ordered by f, with node as an element
    reached  $\leftarrow$  a lookup table, with one entry with key problem.INITIAL and value node
    while not IS-EMPTY(frontier) do
        node  $\leftarrow$  POP(frontier)
        if problem.IS-GOAL(node.STATE) then return node
        for each child in EXPAND(problem, node) do
            s  $\leftarrow$  child.STATE
            if s is not in reached or child.PATH-COST < reached[s].PATH-COST then
                reached[s]  $\leftarrow$  child
                add child to frontier
    return failure
```

Best-First Search is really a class of search algorithms that:

- Use the **evaluation function $f(n)$** to pick next action
- Keep track of **visited states**
- Keep track of **frontier states**
- **Evaluation function $f(n)$ choice controls their behavior**

Informed Search and Heuristics

Informed search relies on **domain-specific knowledge / hints** that help locate the goal state.

$$h(n) = h(\text{State } n)$$

$h(n)$ = n (relevant information about State n)

$h(n)$: heuristic function - estimated cost of the cheapest path from State n to the goal state

Hill Climbing Search

- The most primitive informed search approach
 - a naive greedy algorithm
 - evaluation function: the cost of next move
 - does not care about the “bigger picture” (for example: total search path cost)
- Practicalities:
 - usually does not keep track of search history:
 - not tracking visited nodes → loops!
 - not tracking frontier nodes → does not look at alternatives

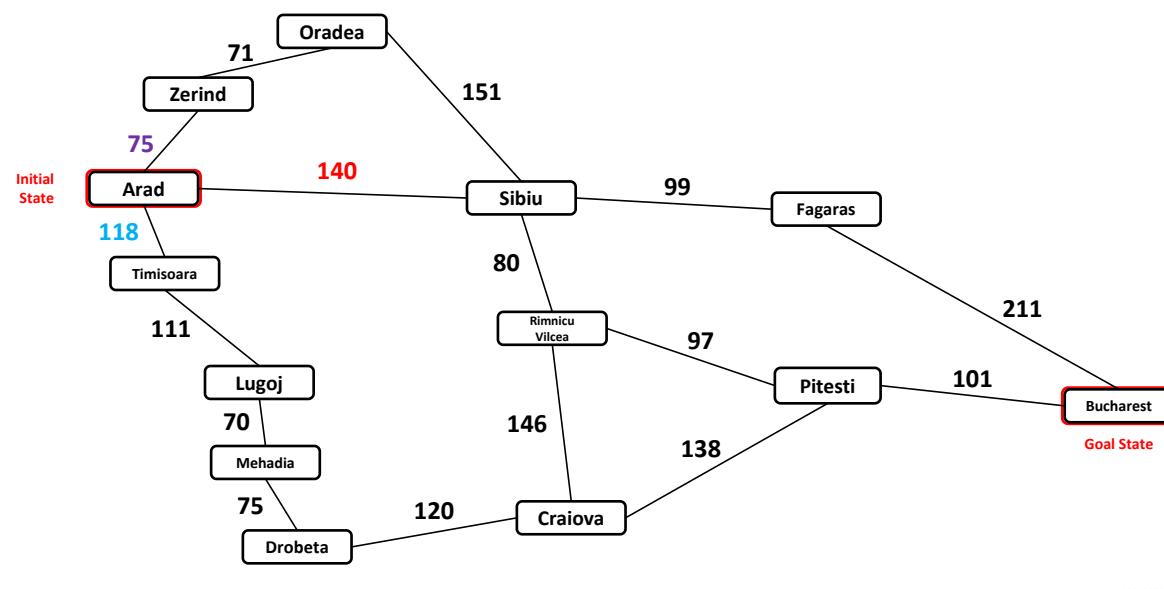
Hill Climbing: Evaluation function

Calculate / obtain:

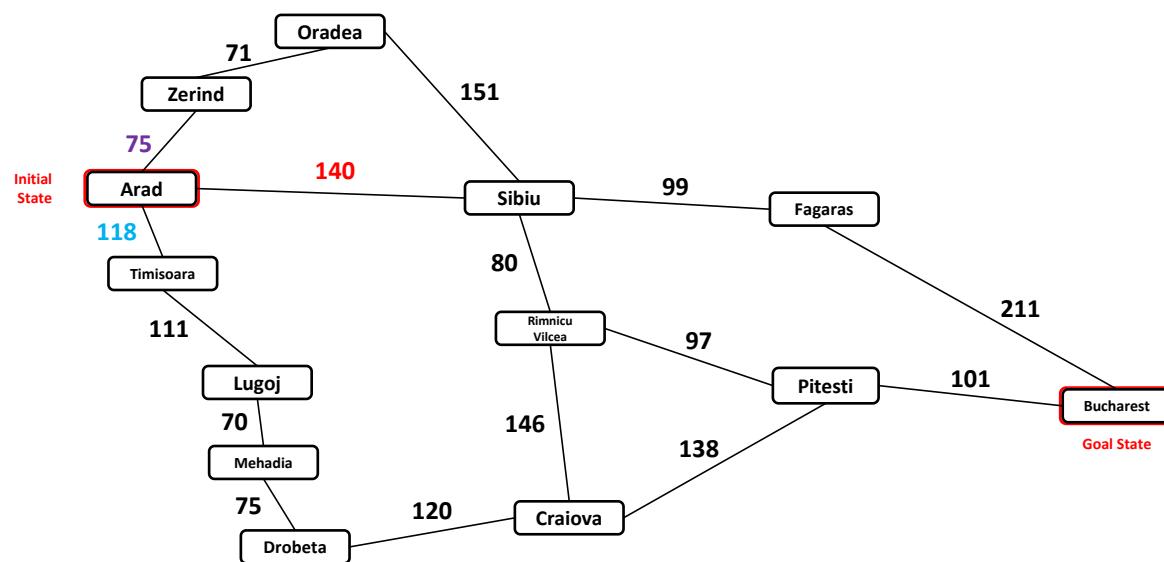
$$f(n) = \text{ACTION-COST}(\text{State}_a, \text{toState}_n, \text{State}_n)$$

A state n with minimum (or maximum) $f(n)$ should be chosen for expansion

Dracula's Roadtrip: Hill Climbing



Dracula's Roadtrip: Hill Climbing



State

Visited state

Arad

Sibiu

Timisoara

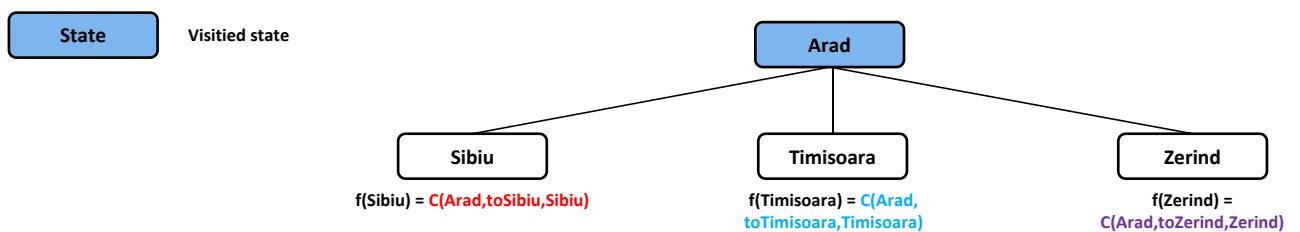
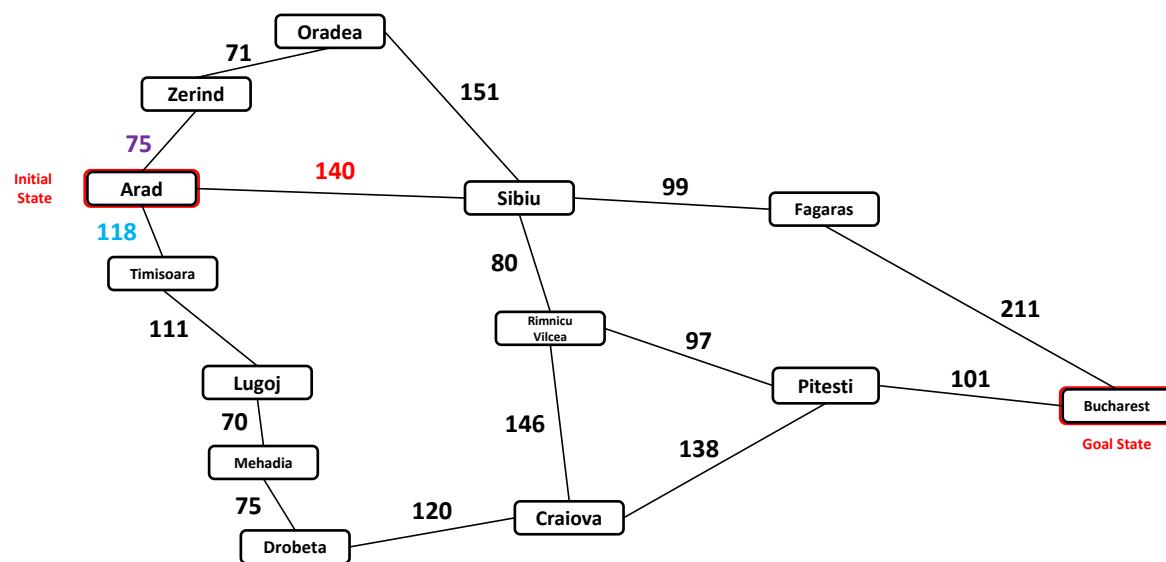
Zerind

$$f(\text{Sibiu}) = g(\text{Sibiu})$$

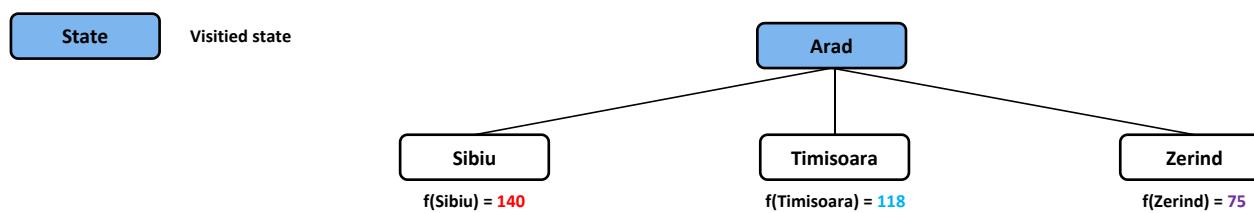
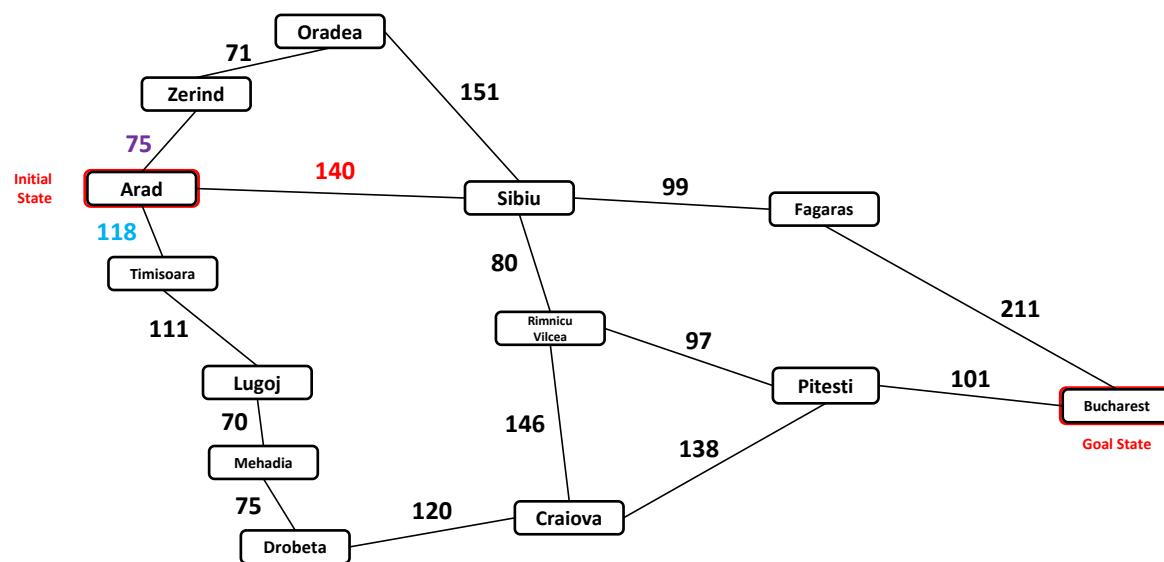
$$f(\text{Timisoara}) = g(\text{Timisoara})$$

$$f(\text{Zerind}) = g(\text{Zerind})$$

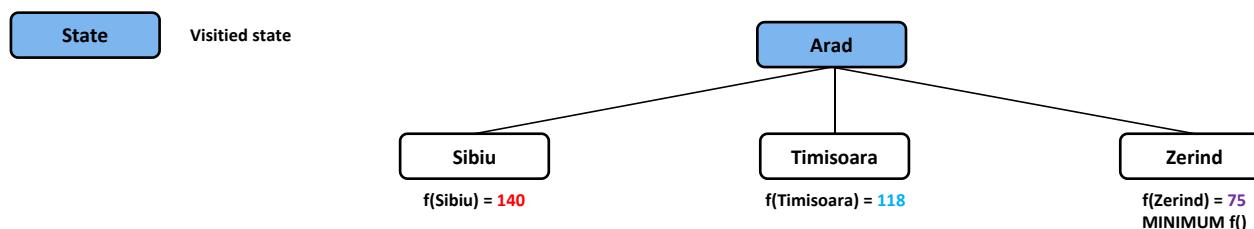
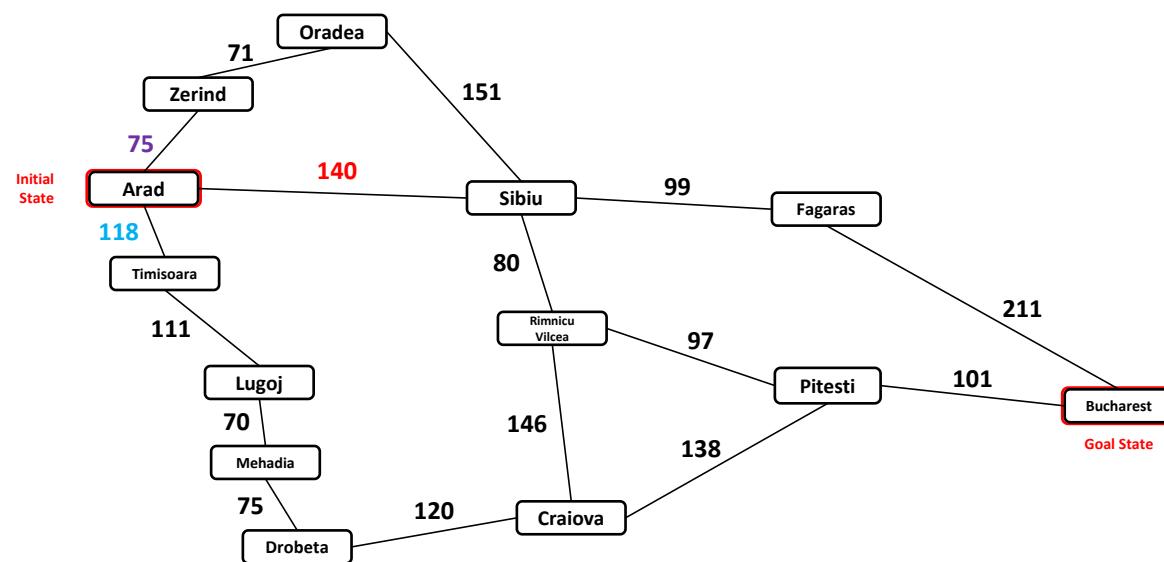
Dracula's Roadtrip: Hill Climbing



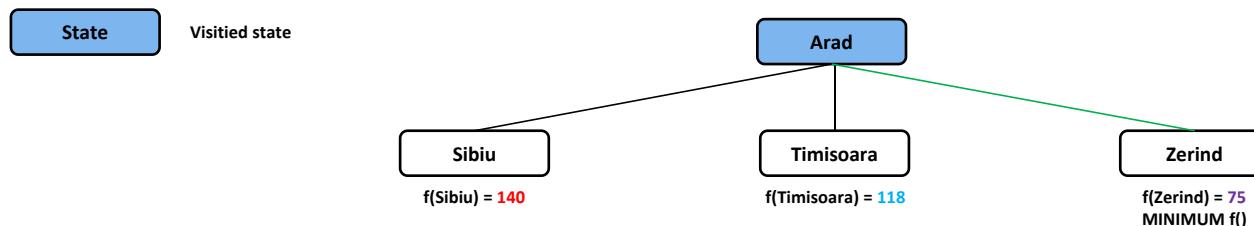
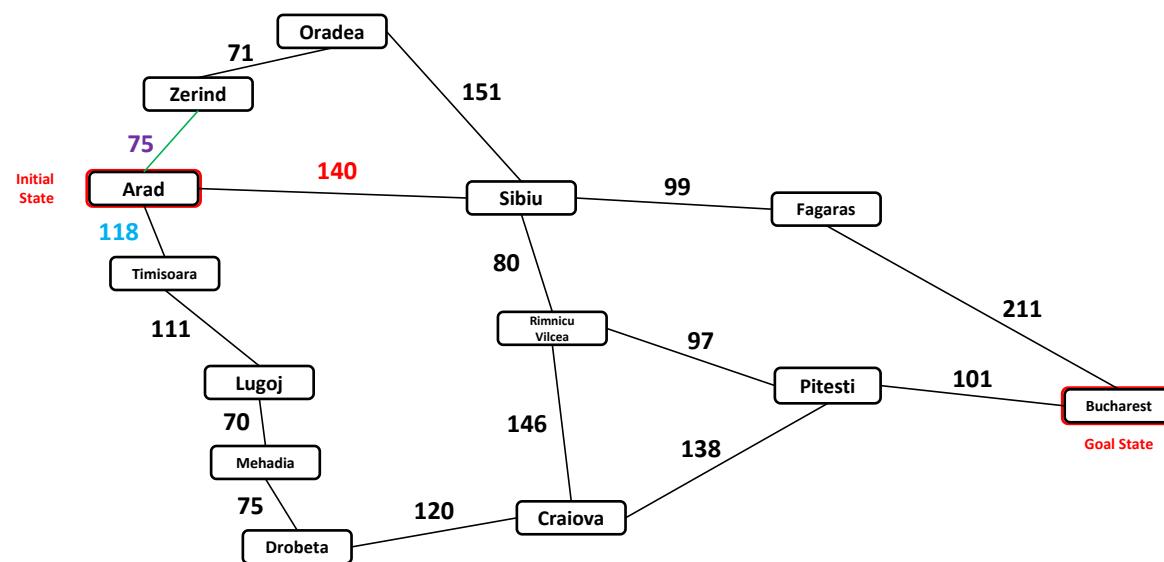
Dracula's Roadtrip: Hill Climbing



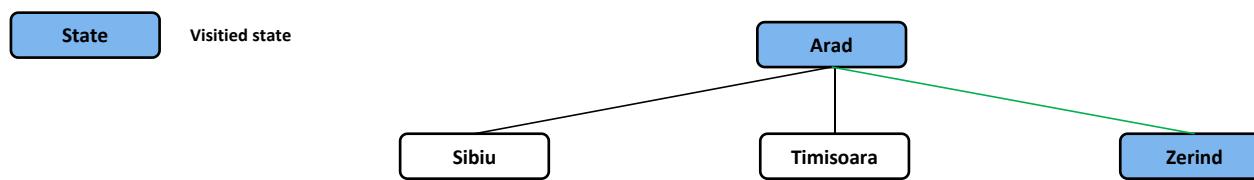
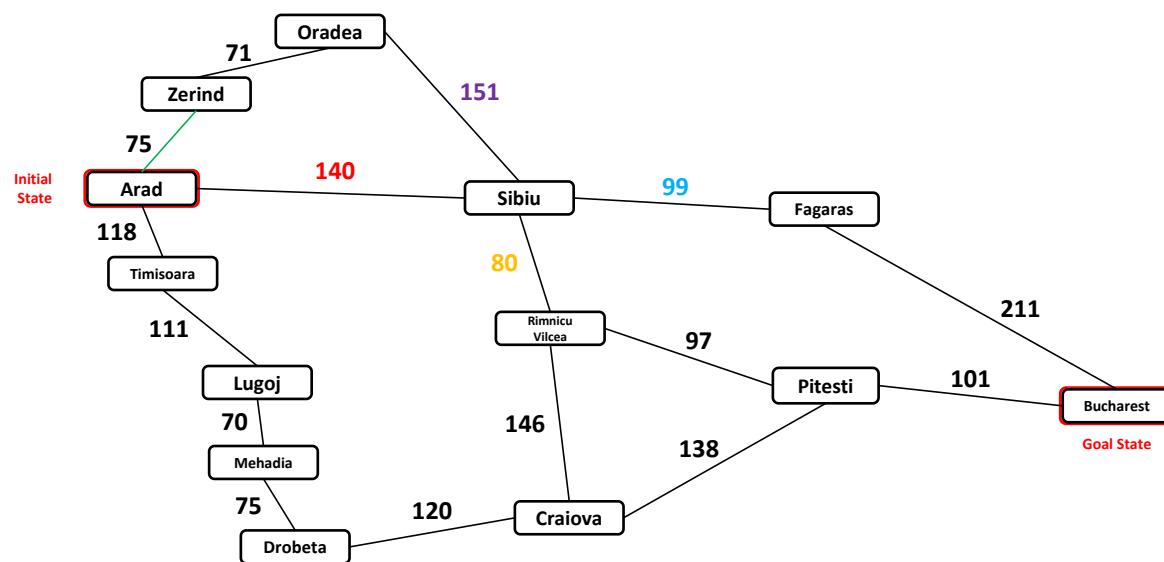
Dracula's Roadtrip: Hill Climbing



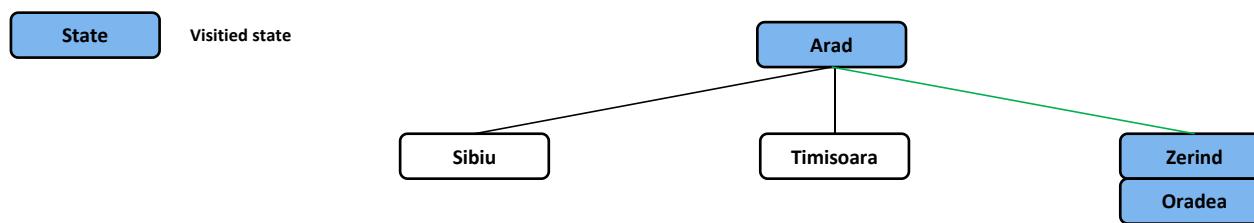
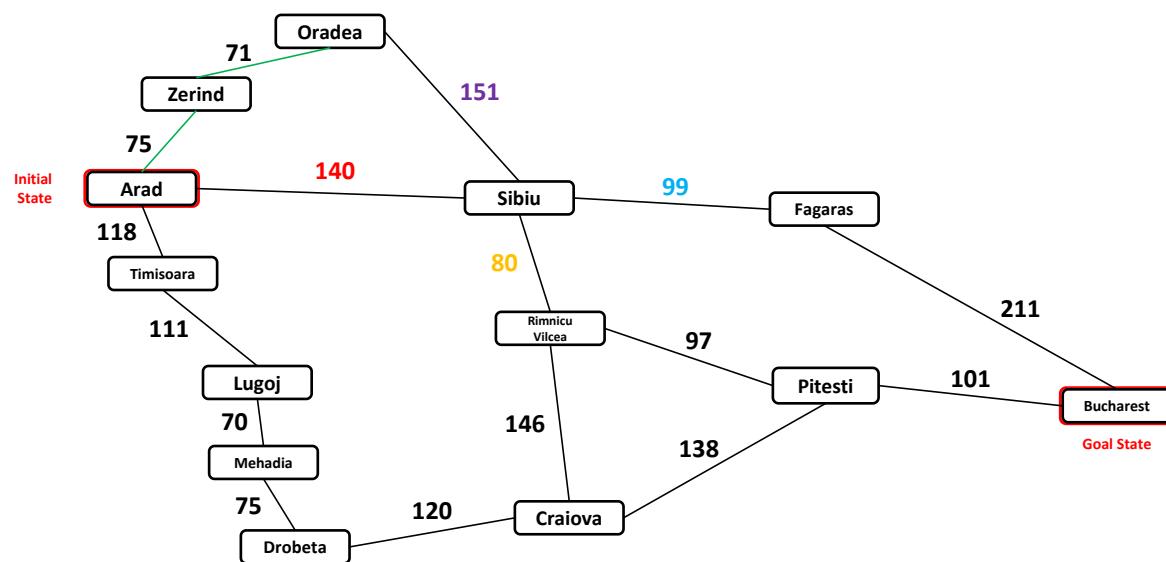
Dracula's Roadtrip: Hill Climbing



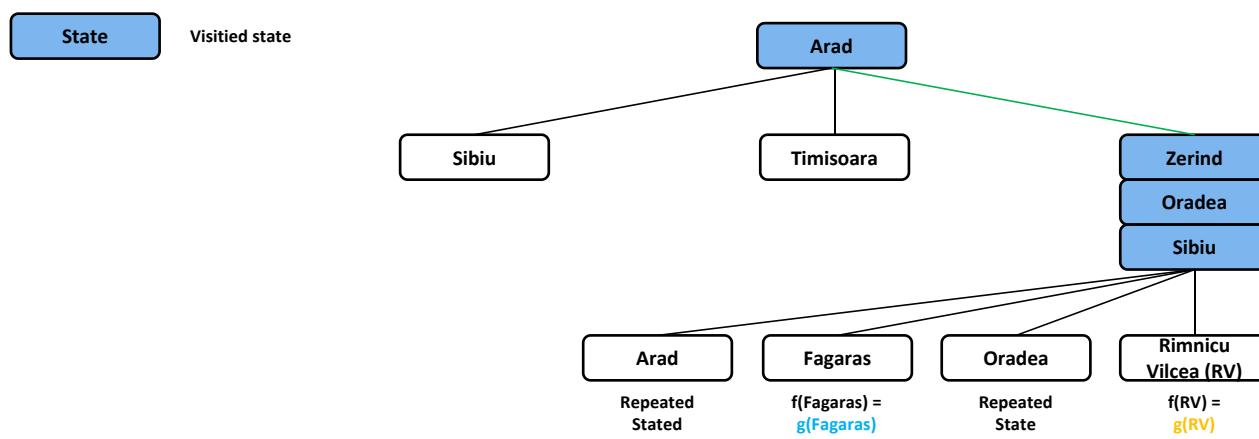
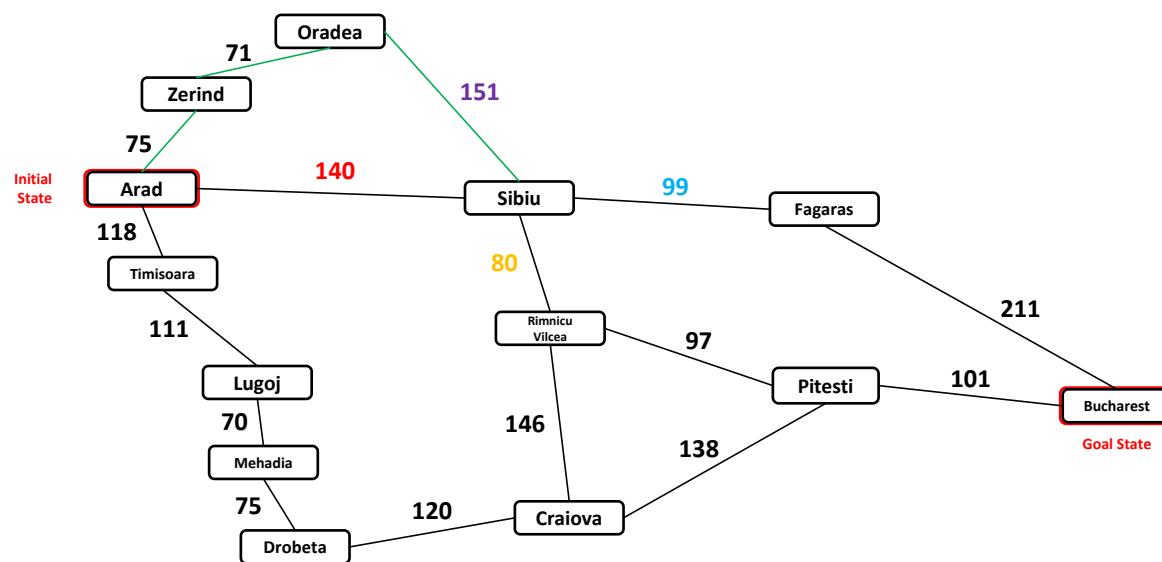
Dracula's Roadtrip: Hill Climbing



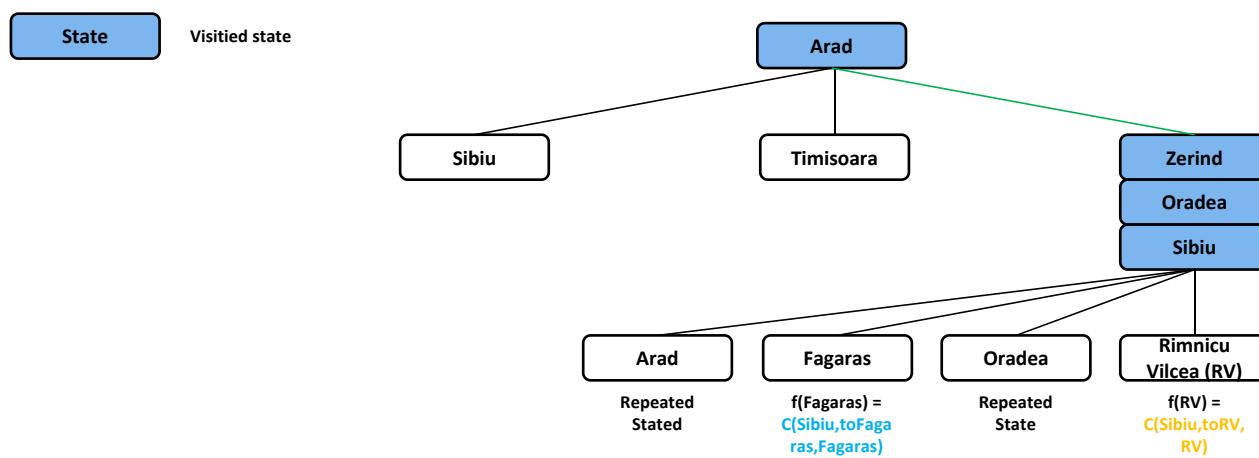
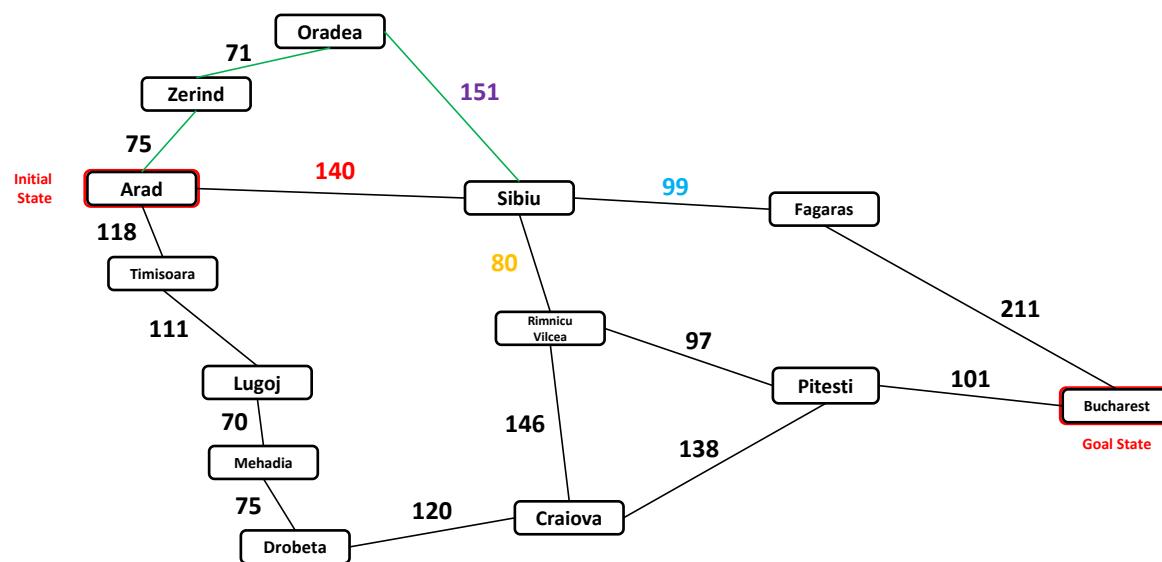
Dracula's Roadtrip: Hill Climbing



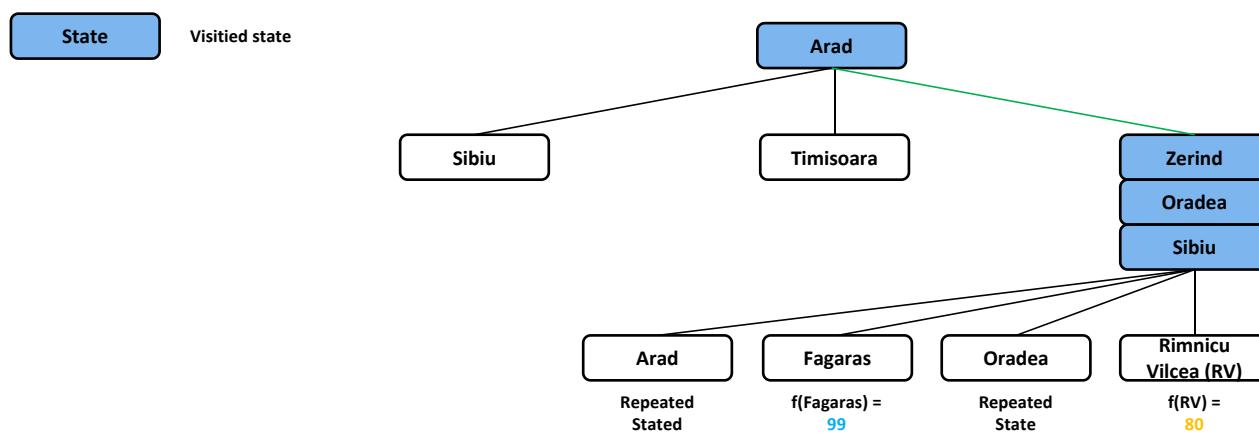
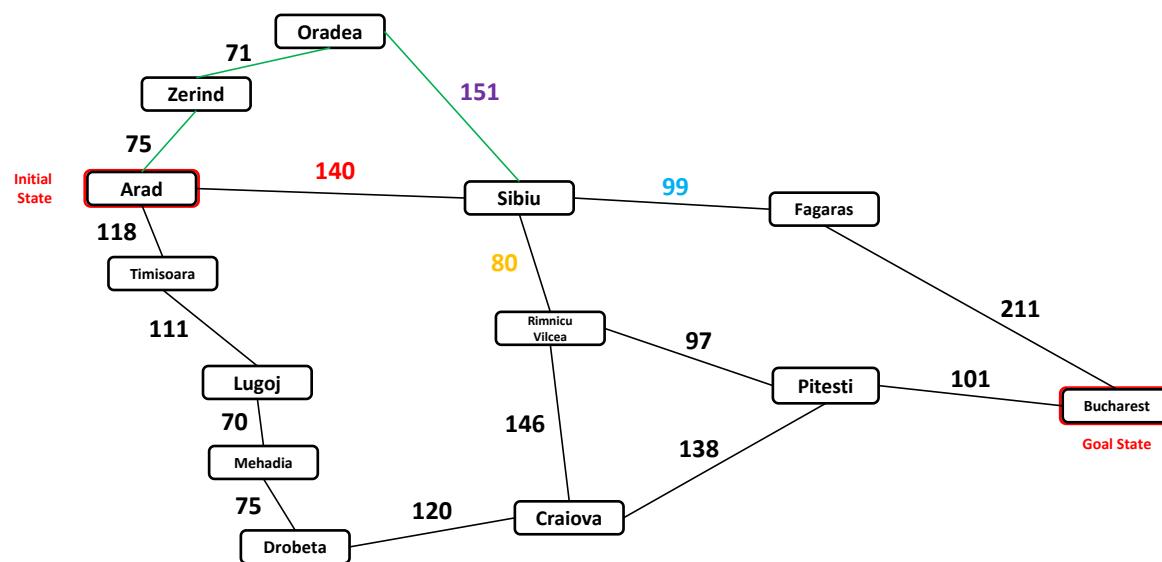
Dracula's Roadtrip: Hill Climbing



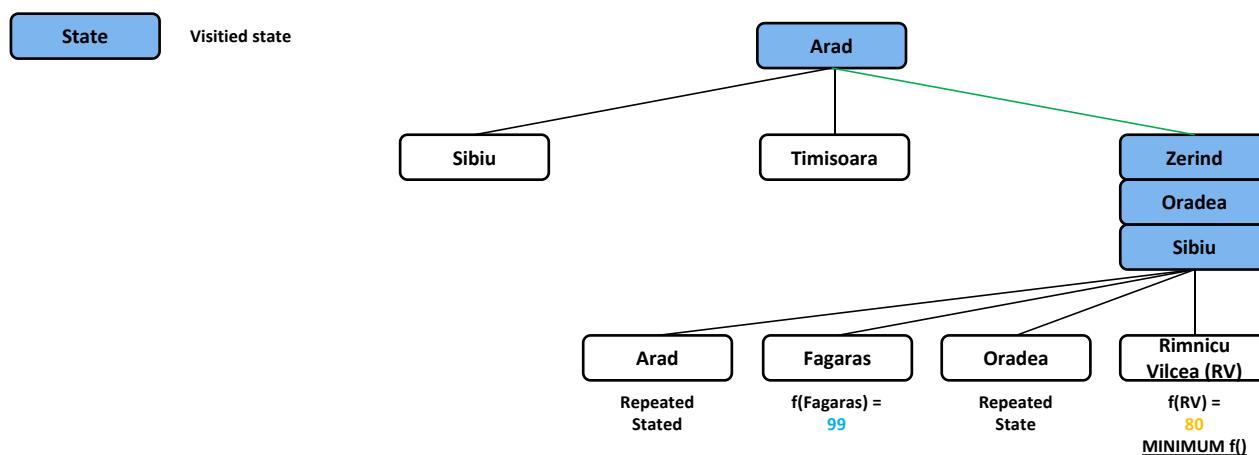
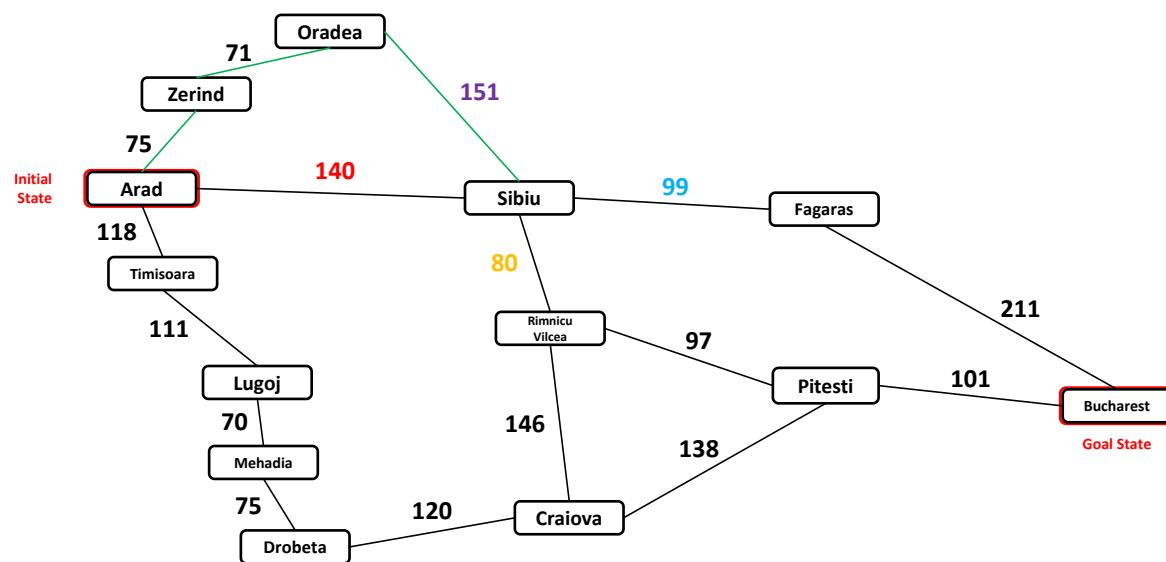
Dracula's Roadtrip: Hill Climbing



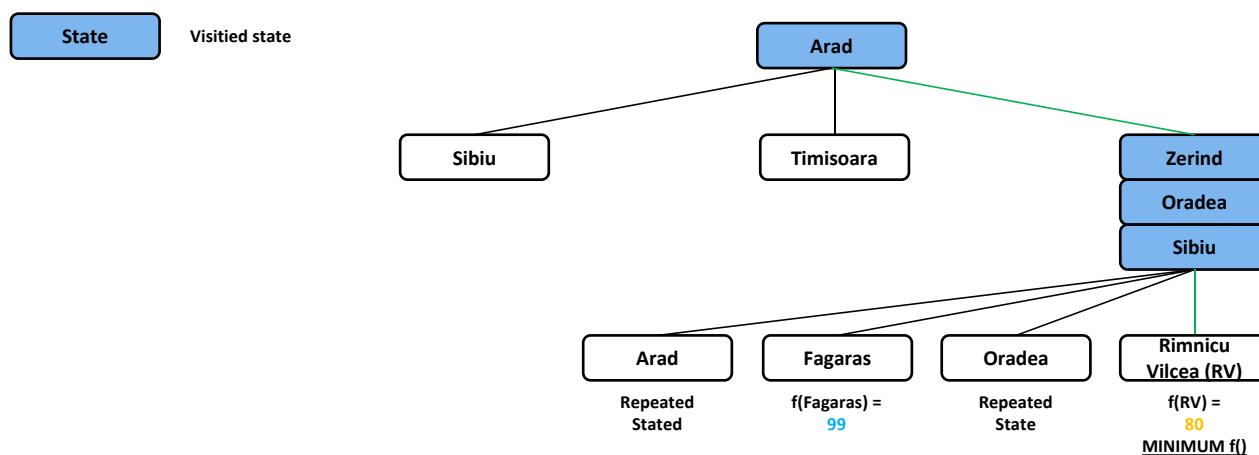
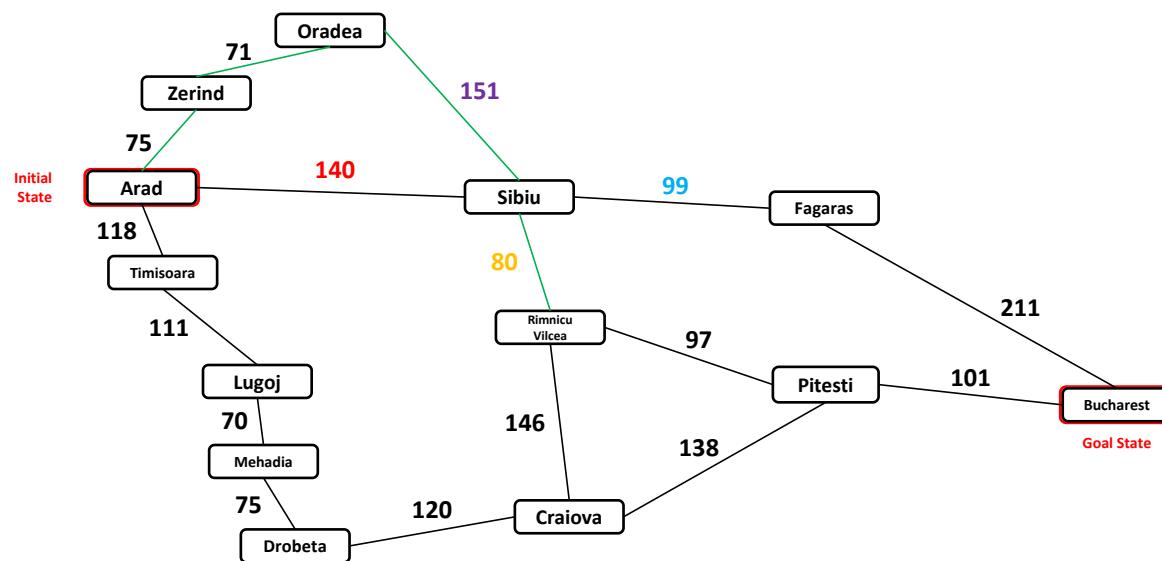
Dracula's Roadtrip: Hill Climbing



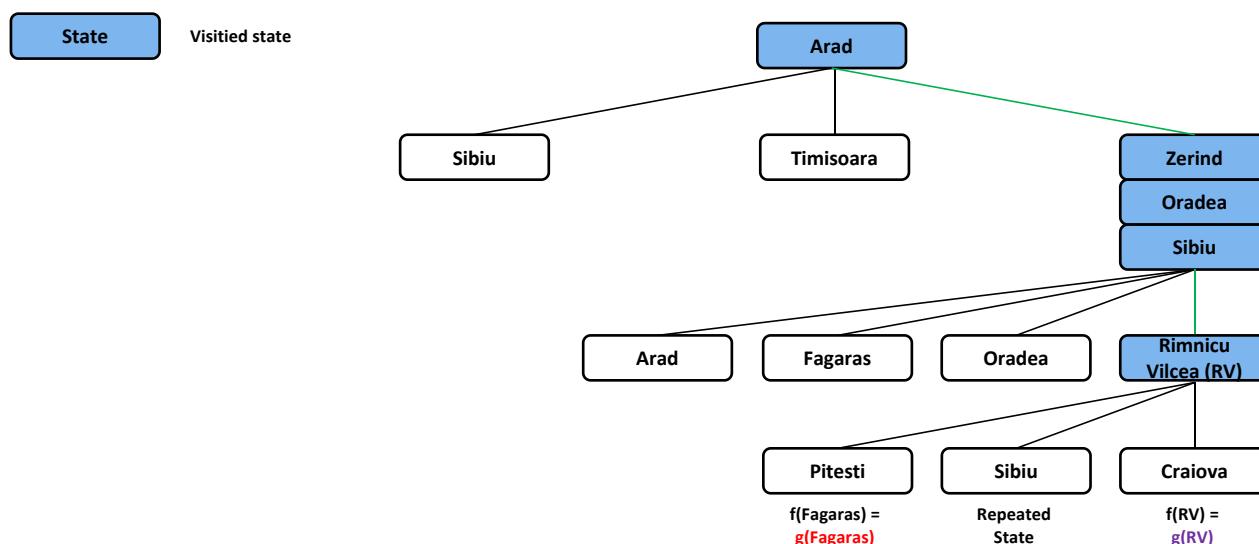
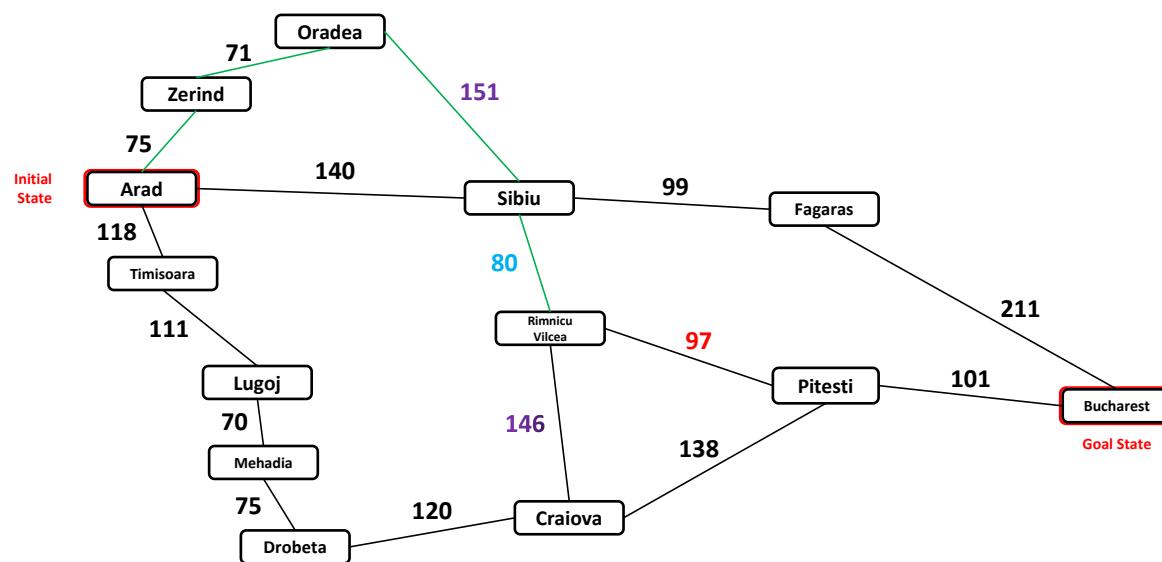
Dracula's Roadtrip: Hill Climbing



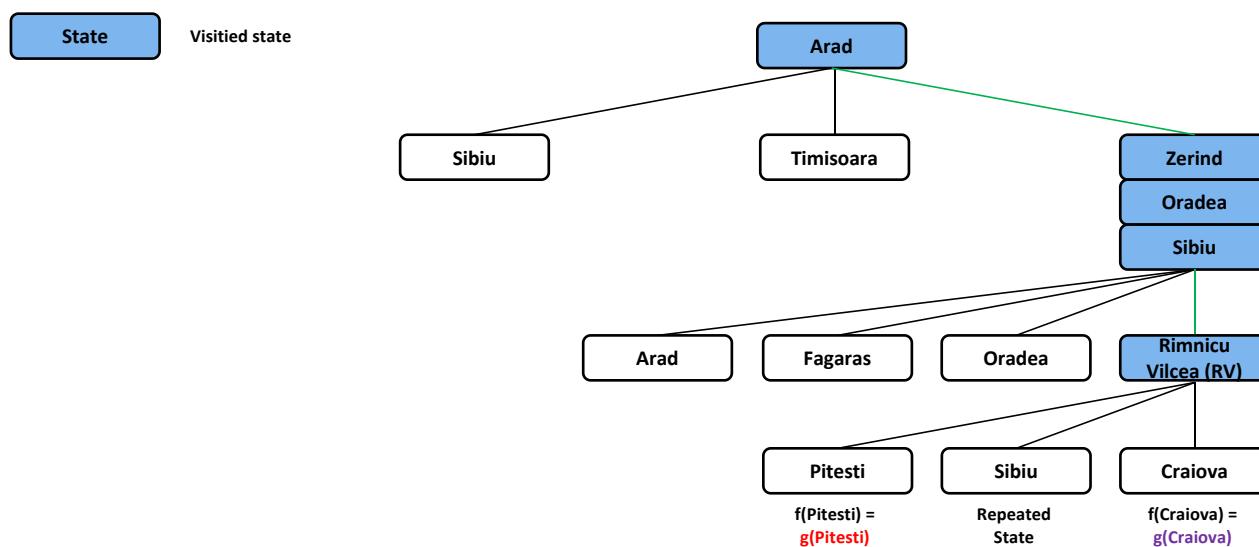
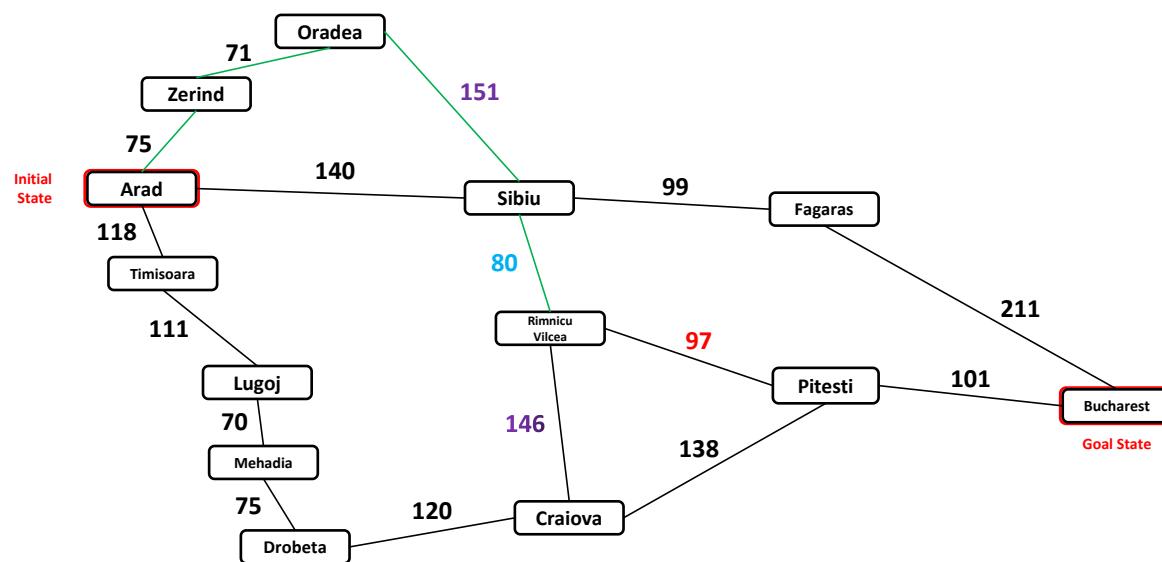
Dracula's Roadtrip: Hill Climbing



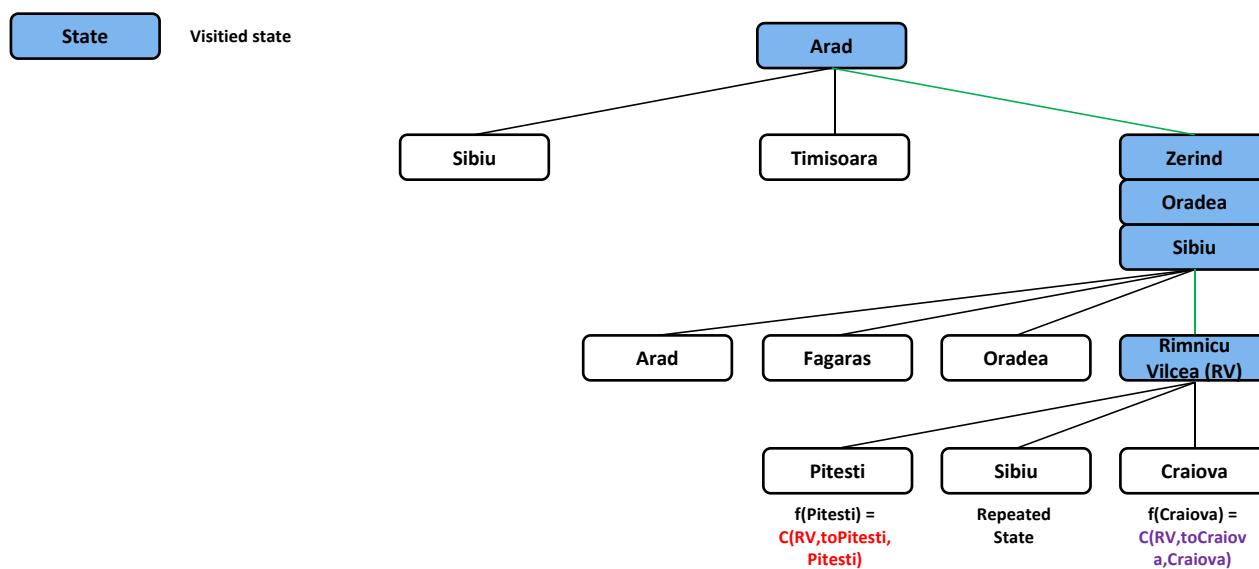
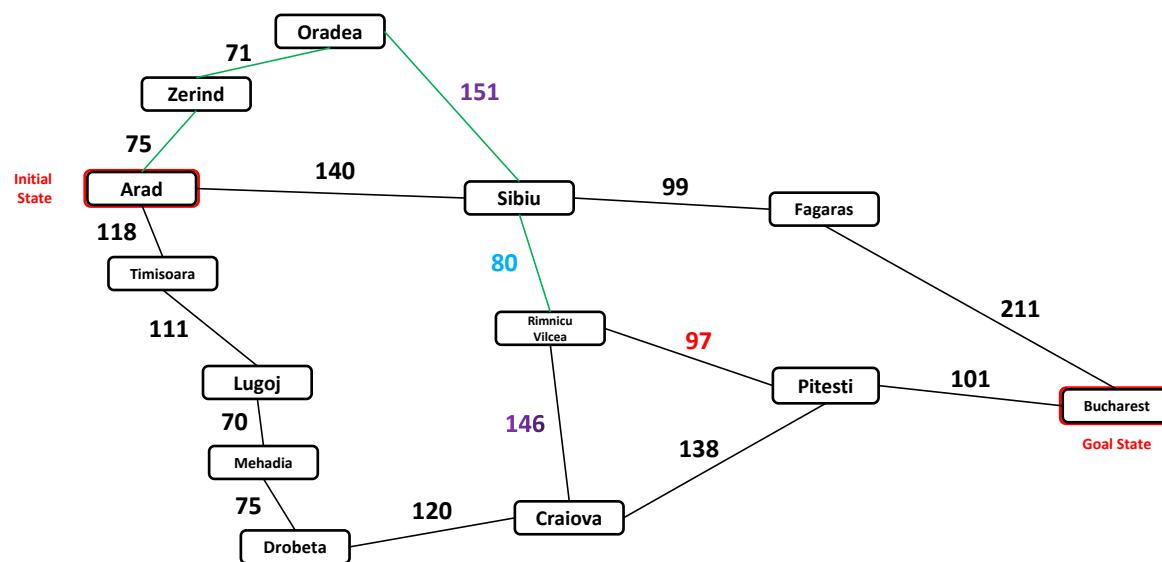
Dracula's Roadtrip: Hill Climbing



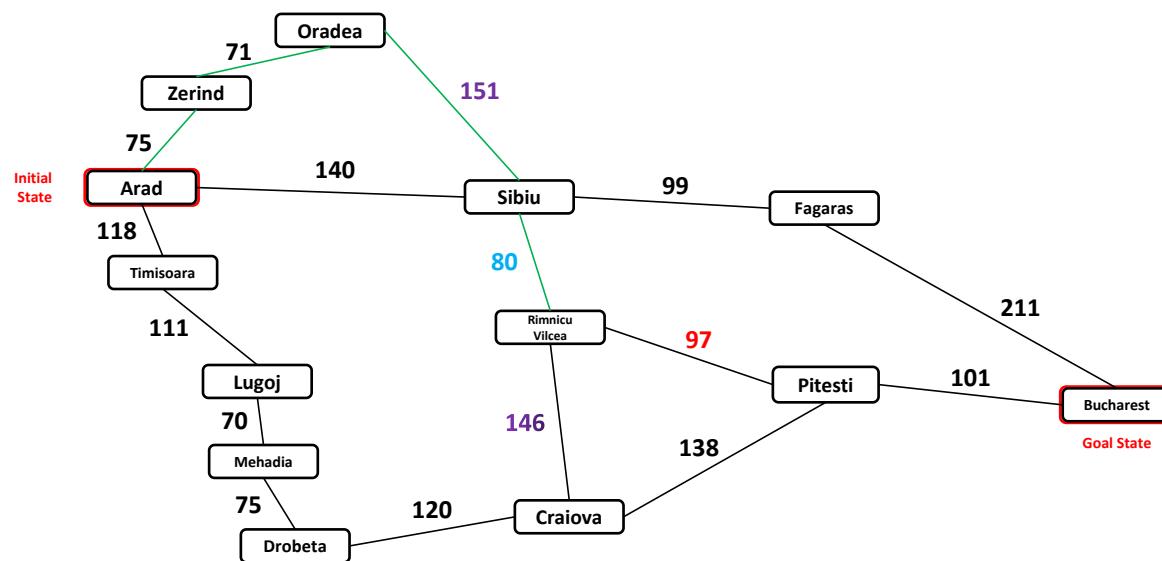
Dracula's Roadtrip: Hill Climbing



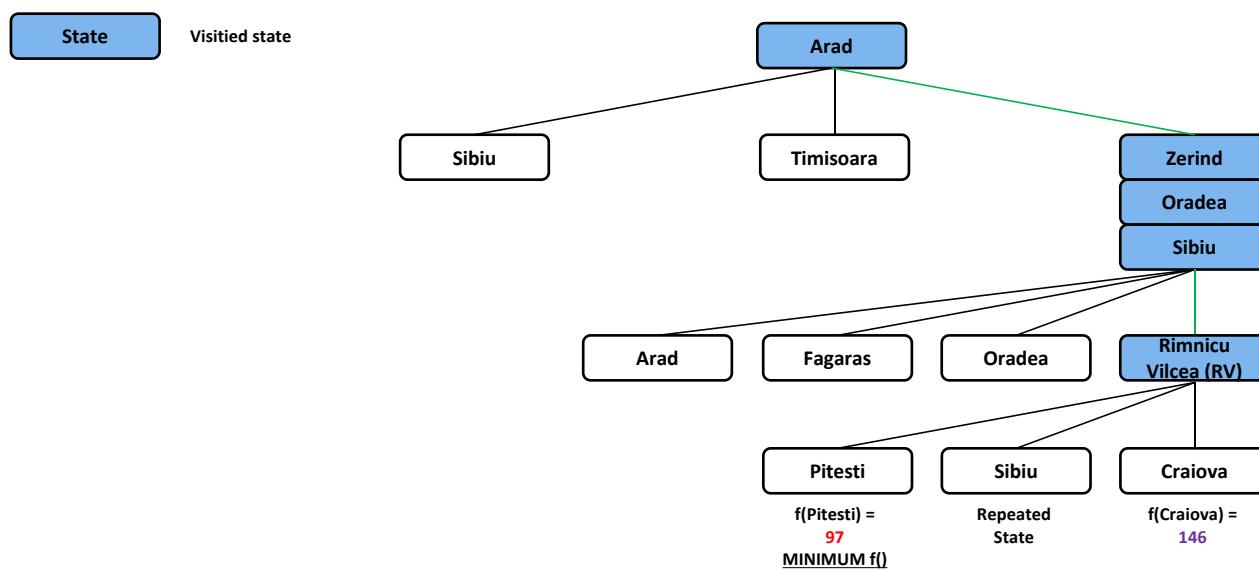
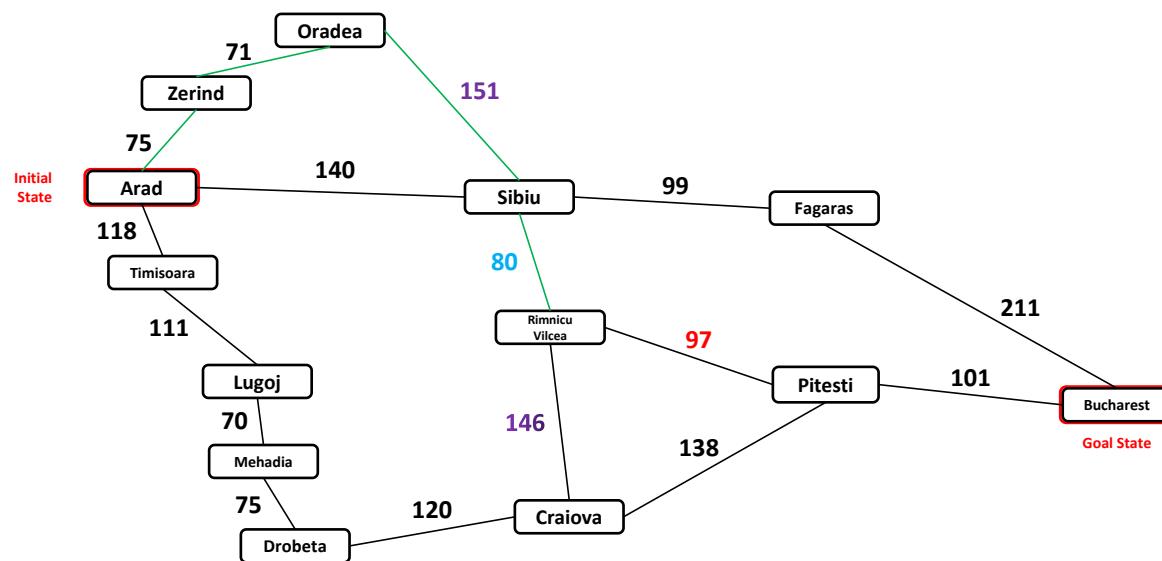
Dracula's Roadtrip: Hill Climbing



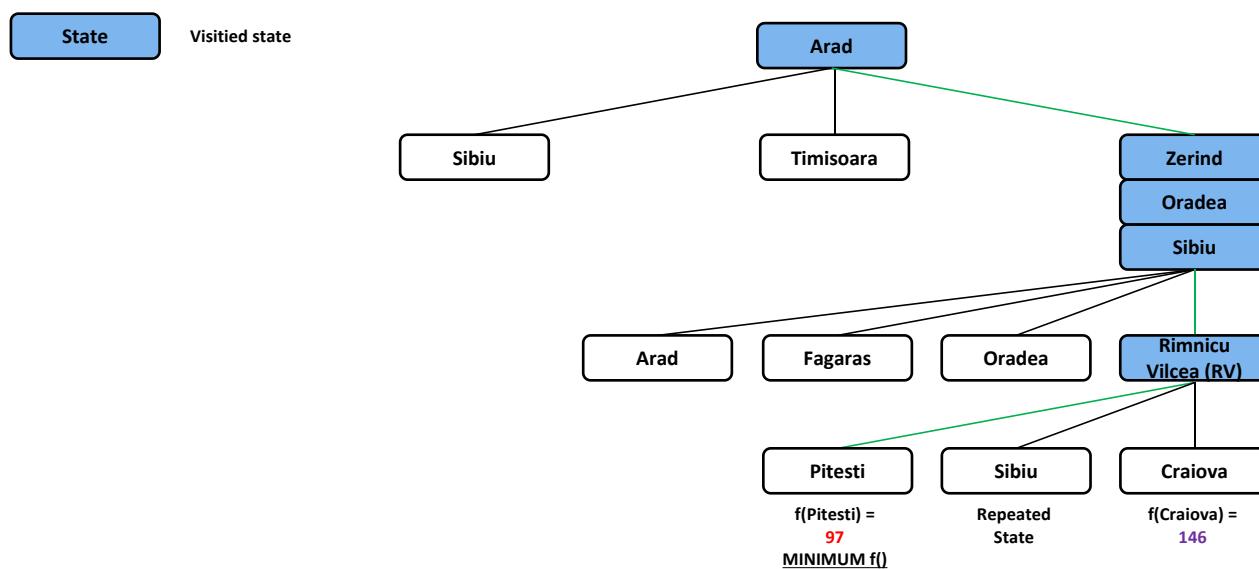
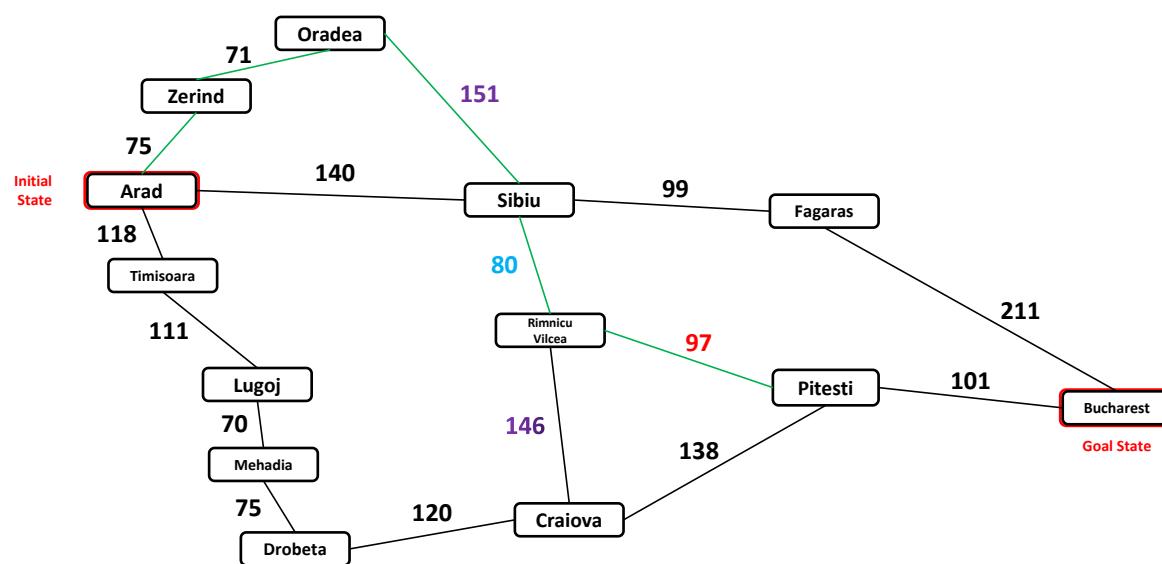
Dracula's Roadtrip: Hill Climbing



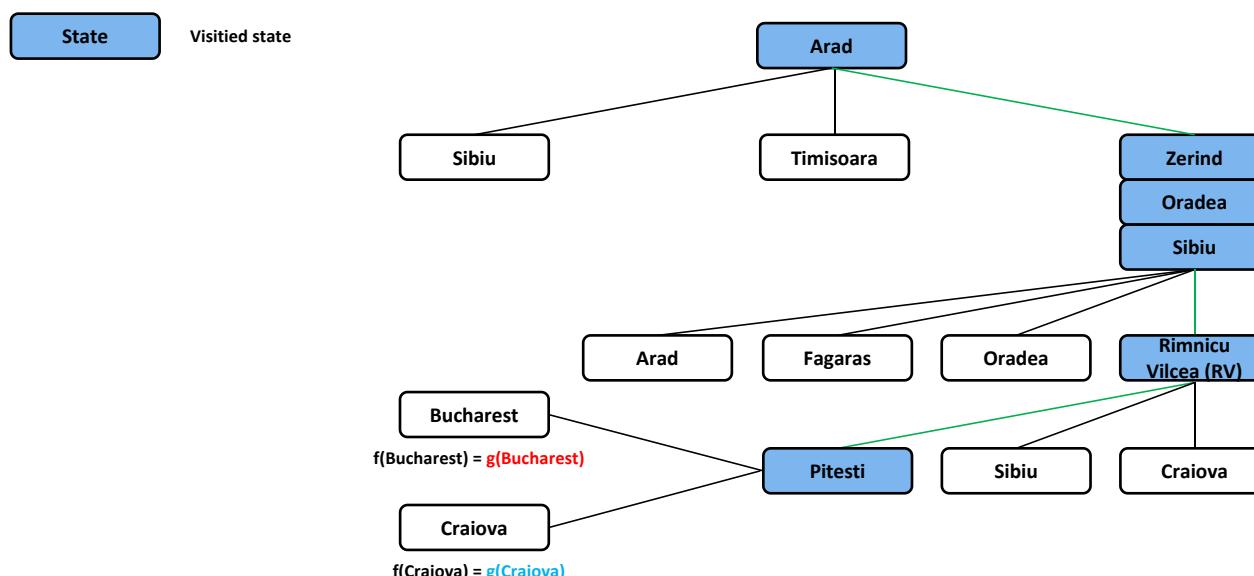
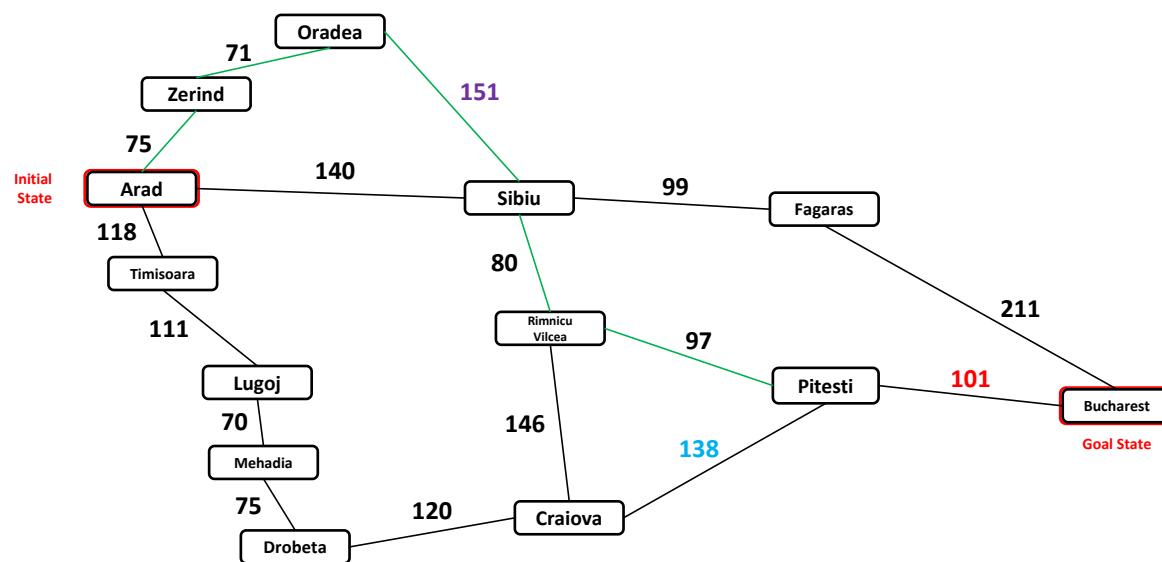
Dracula's Roadtrip: Hill Climbing



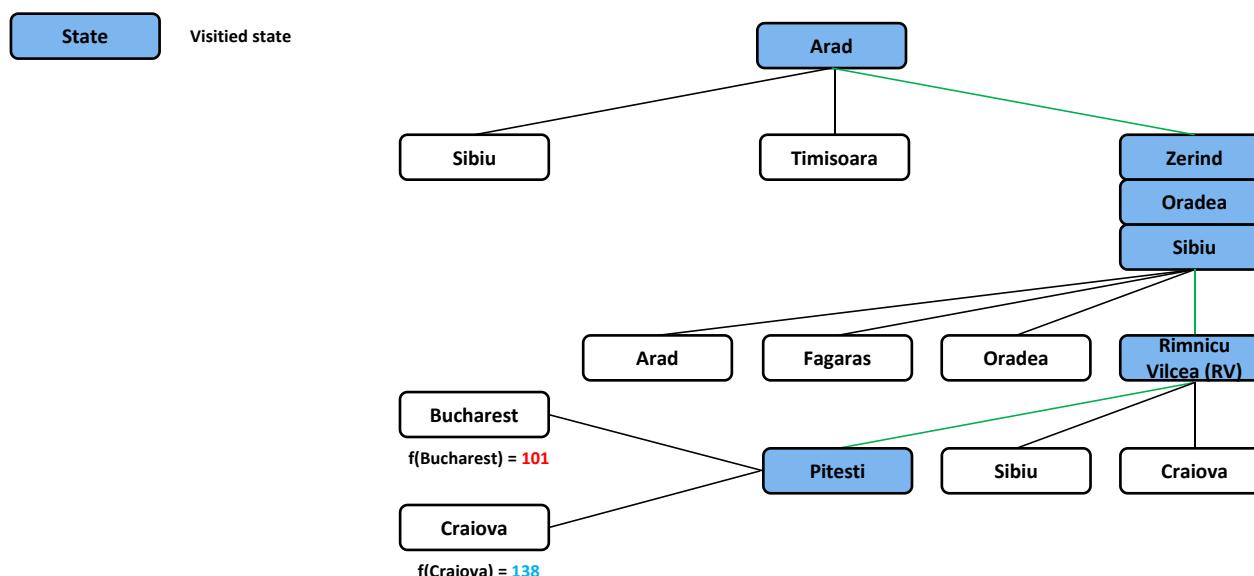
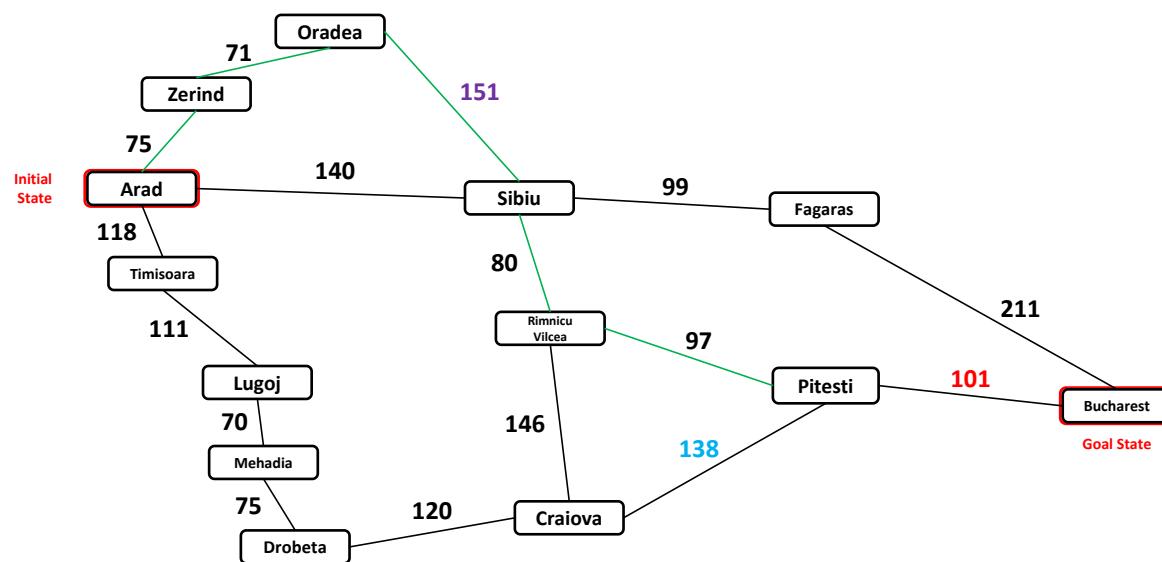
Dracula's Roadtrip: Hill Climbing



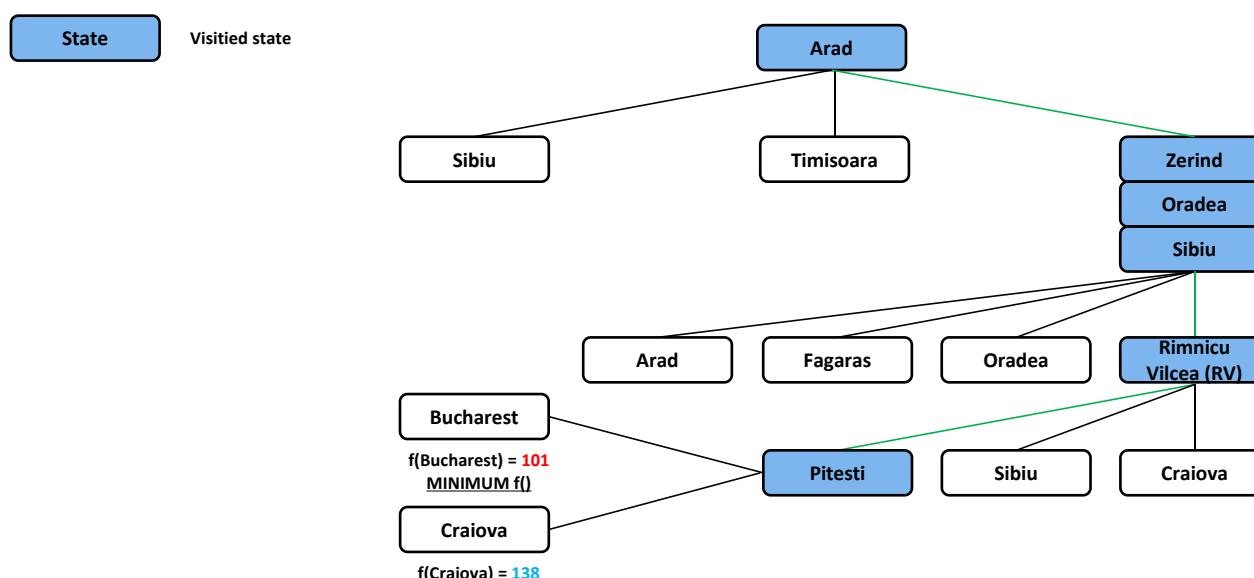
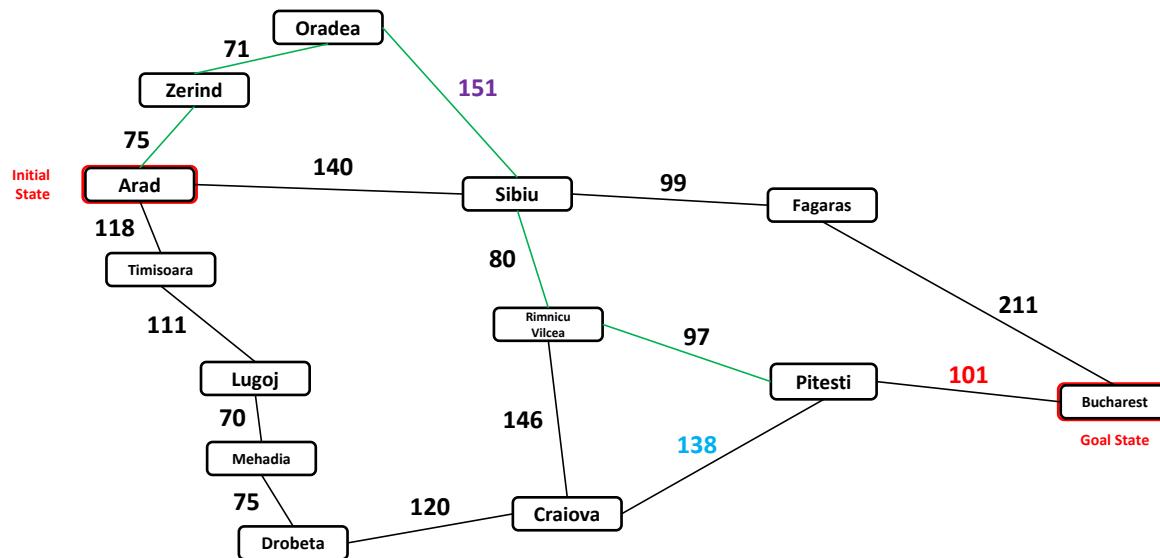
Dracula's Roadtrip: Hill Climbing



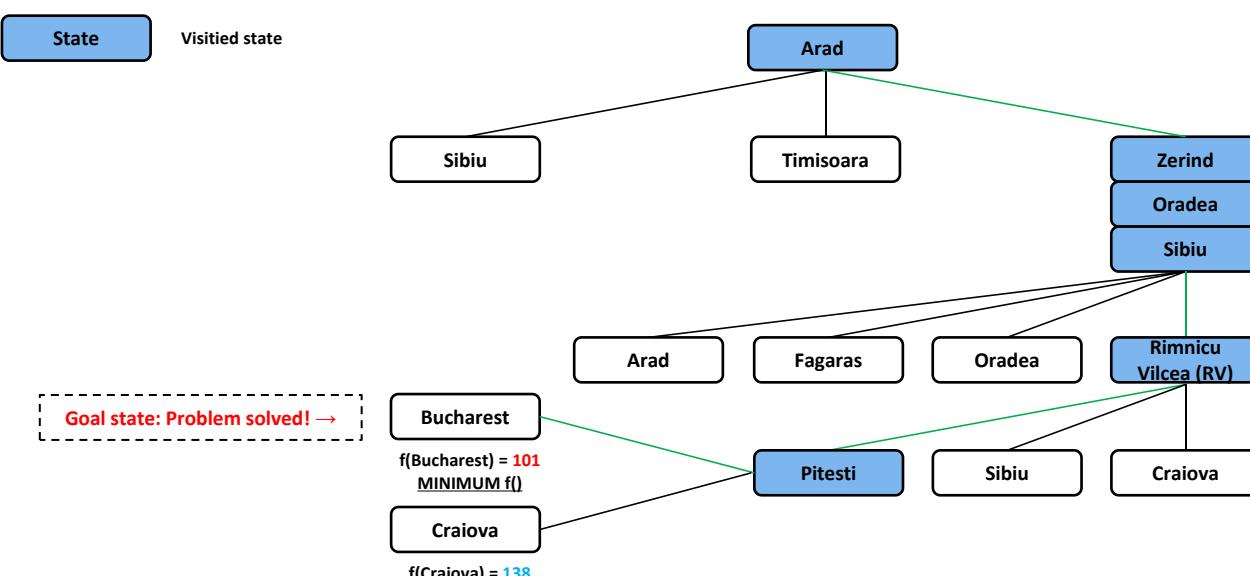
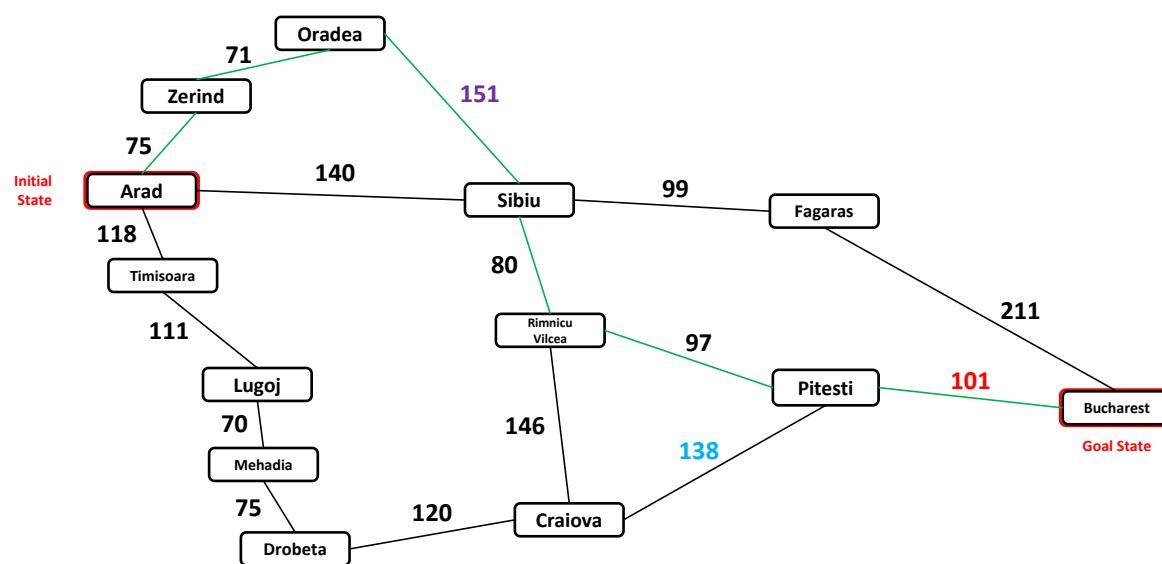
Dracula's Roadtrip: Hill Climbing



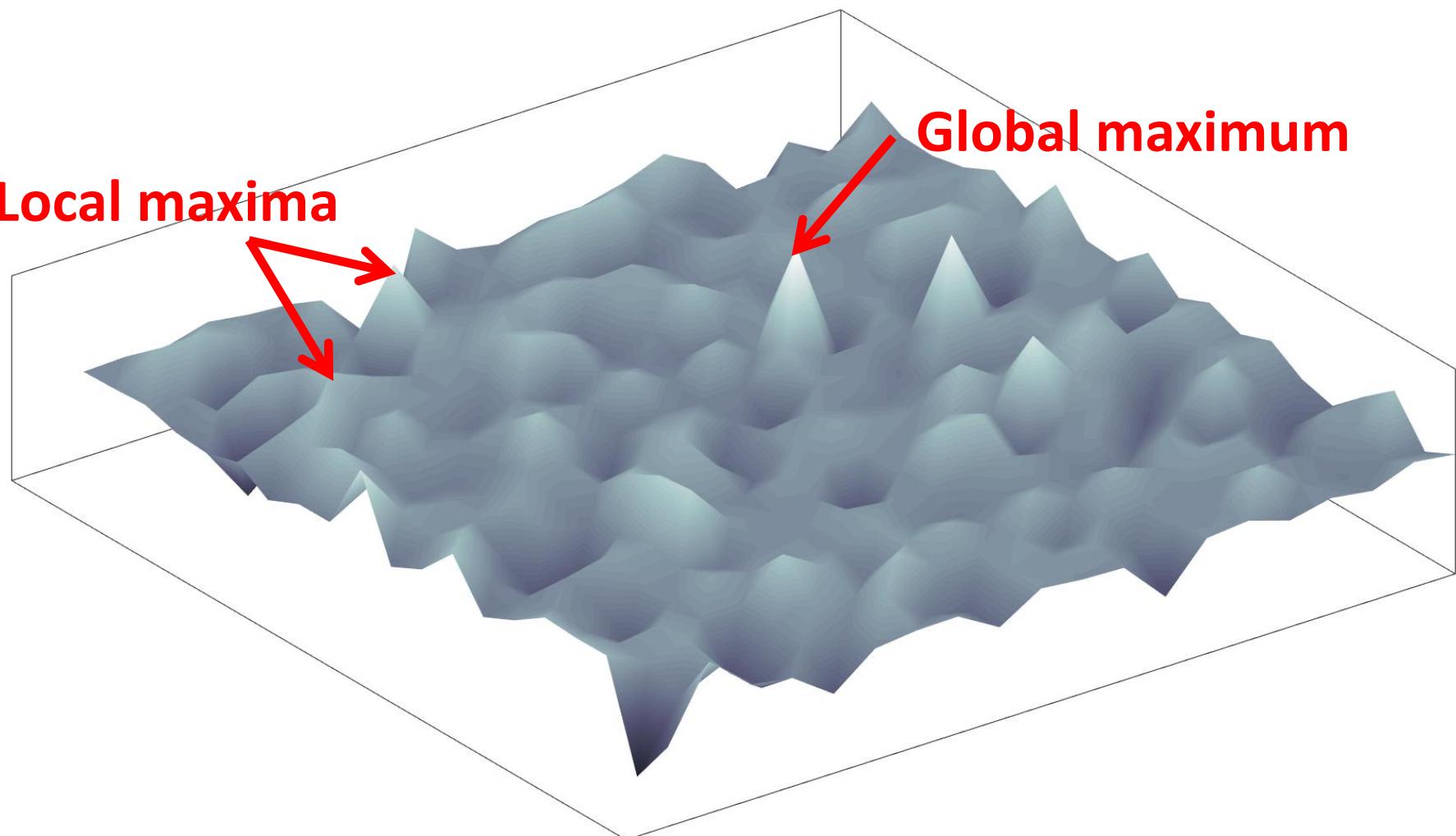
Dracula's Roadtrip: Hill Climbing



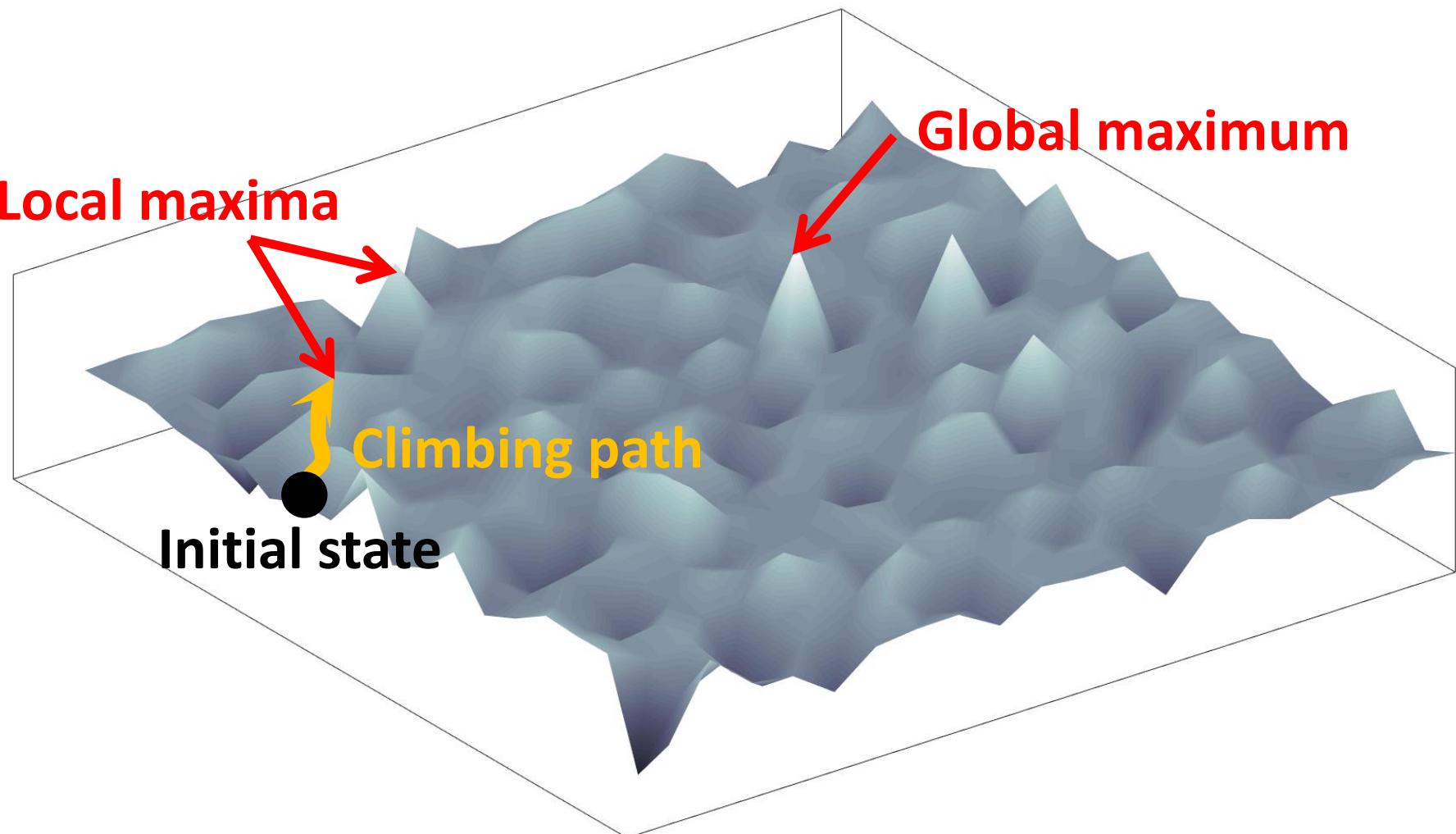
Dracula's Roadtrip: Hill Climbing



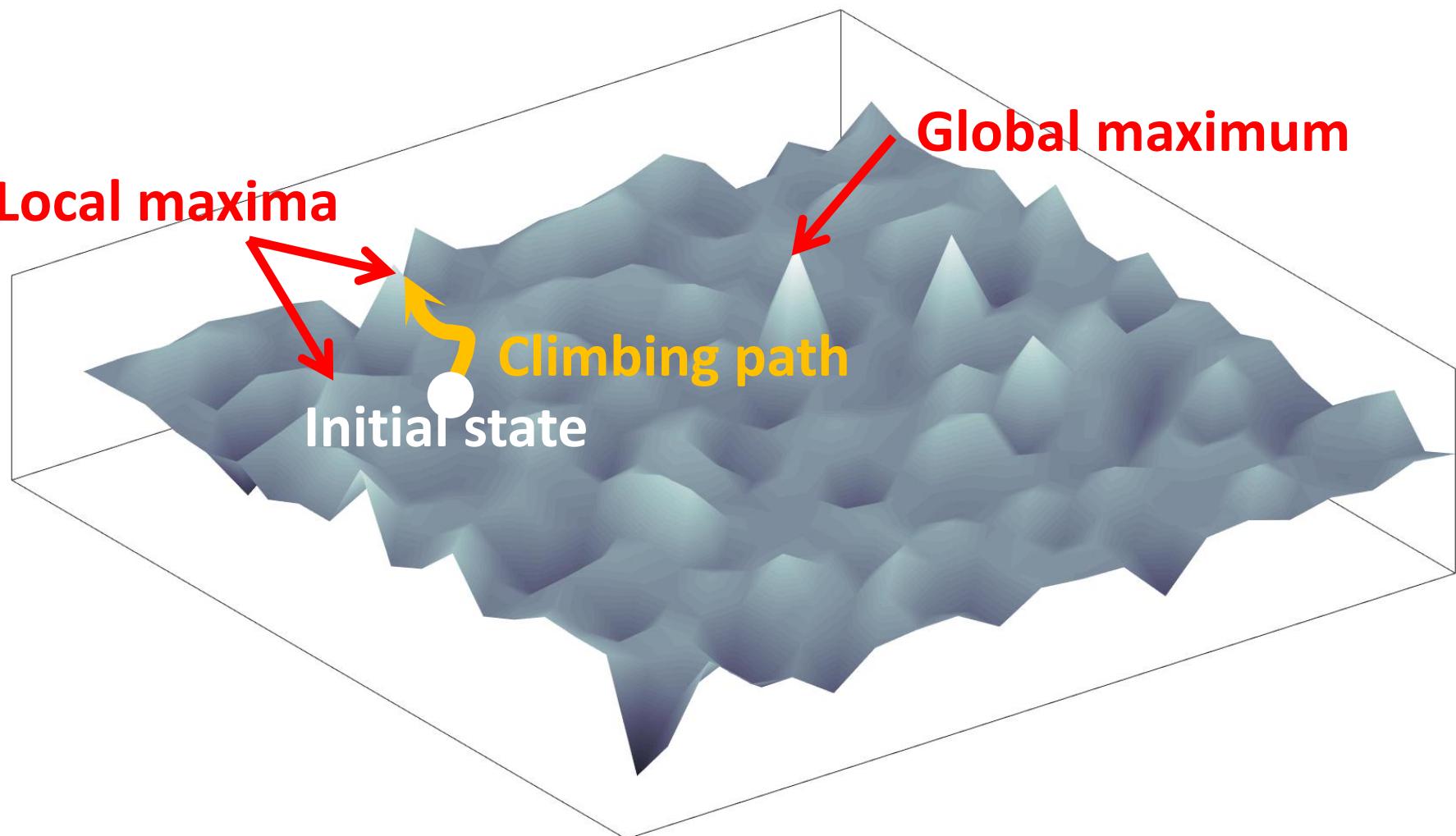
Hill Climbing Problems: Local Maxima



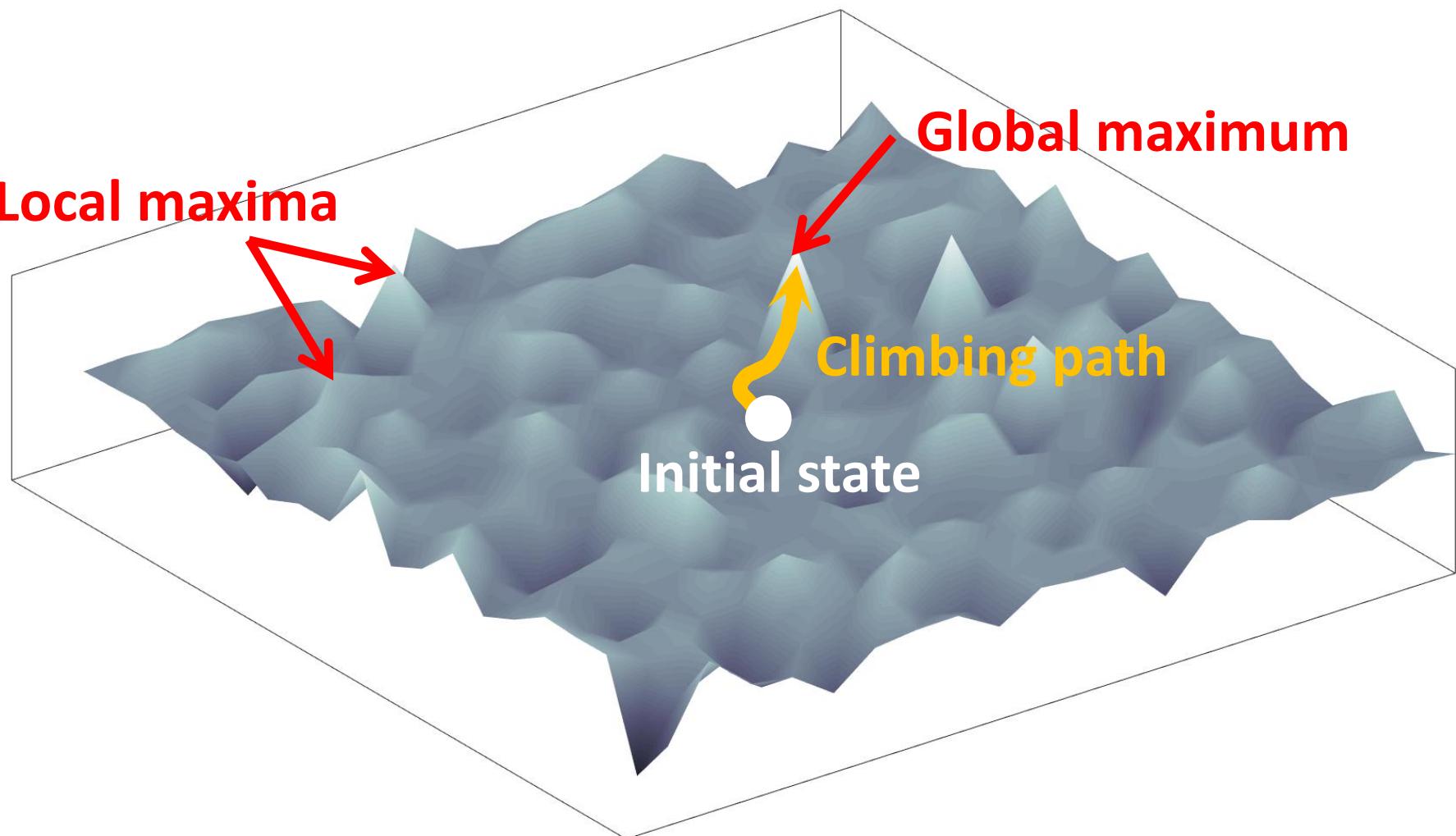
Hill Climbing Problems: Local Maxima



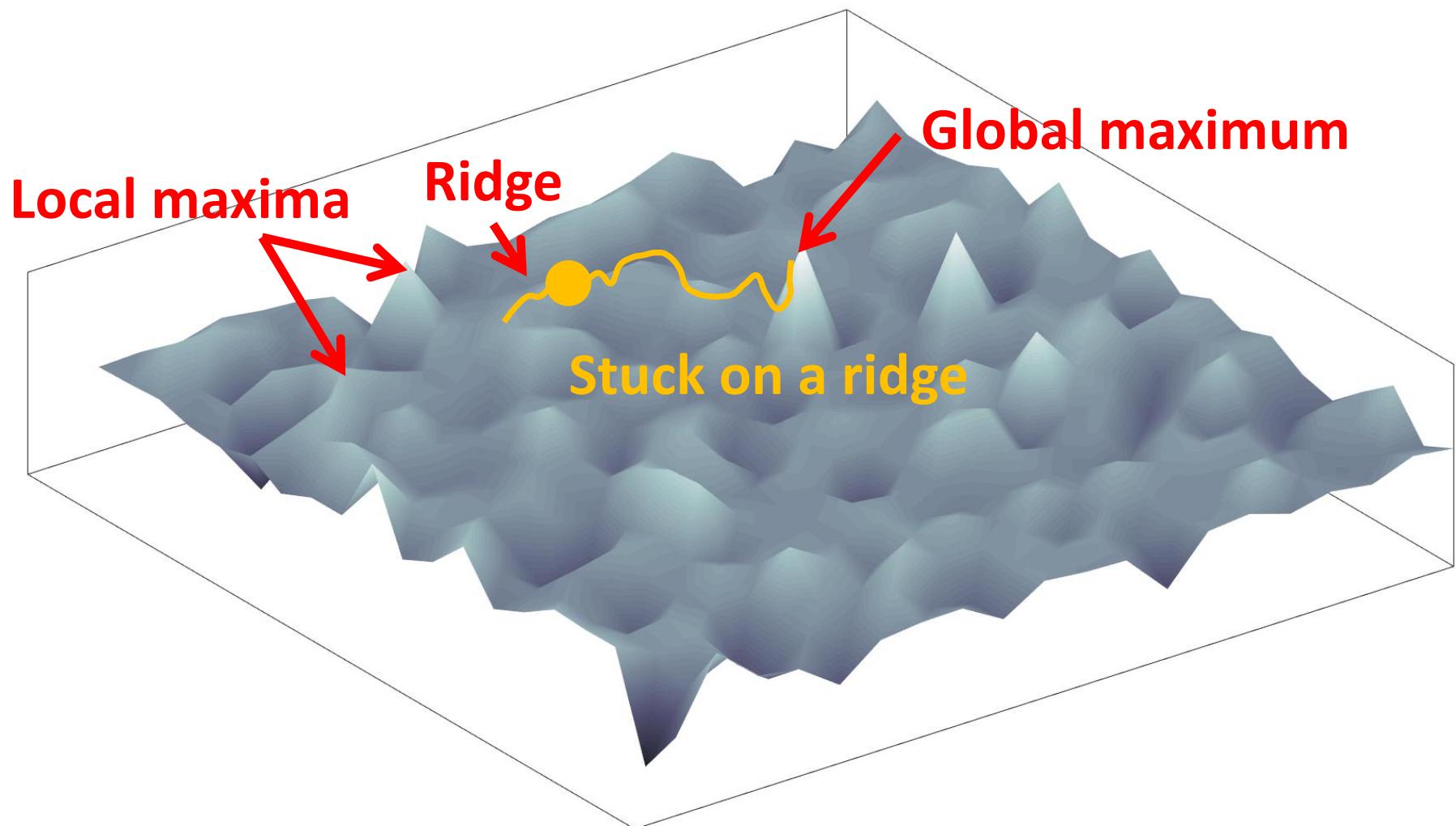
Hill Climbing Problems: Local Maxima



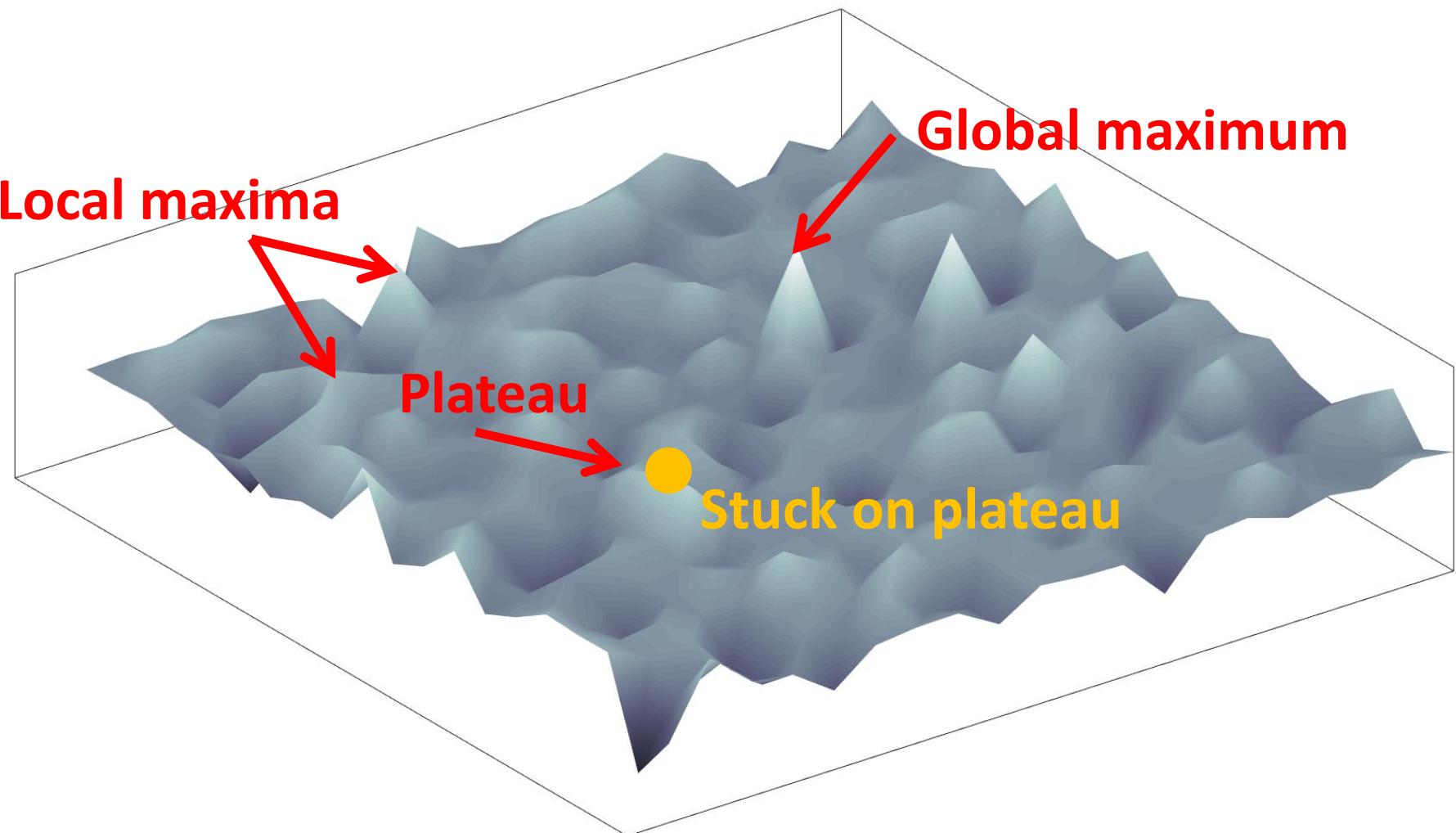
Hill Climbing Problems: Local Maxima



Hill Climbing Problems: Ridges



Hill Climbing Problems: Plateaus



Greedy Best First Search

- Also a rather primitive informed search approach
 - a naive greedy algorithm
 - evaluation function: **heuristics $h(n)$**
 - tries to not “move farther away” from the goal
 - does not care about the total path cost
- Practicalities:
 - it keep track of search history:
 - tracks visited states / nodes
 - tracks frontier states / nodes

Greedy Best First: Evaluation Function

Calculate / obtain:

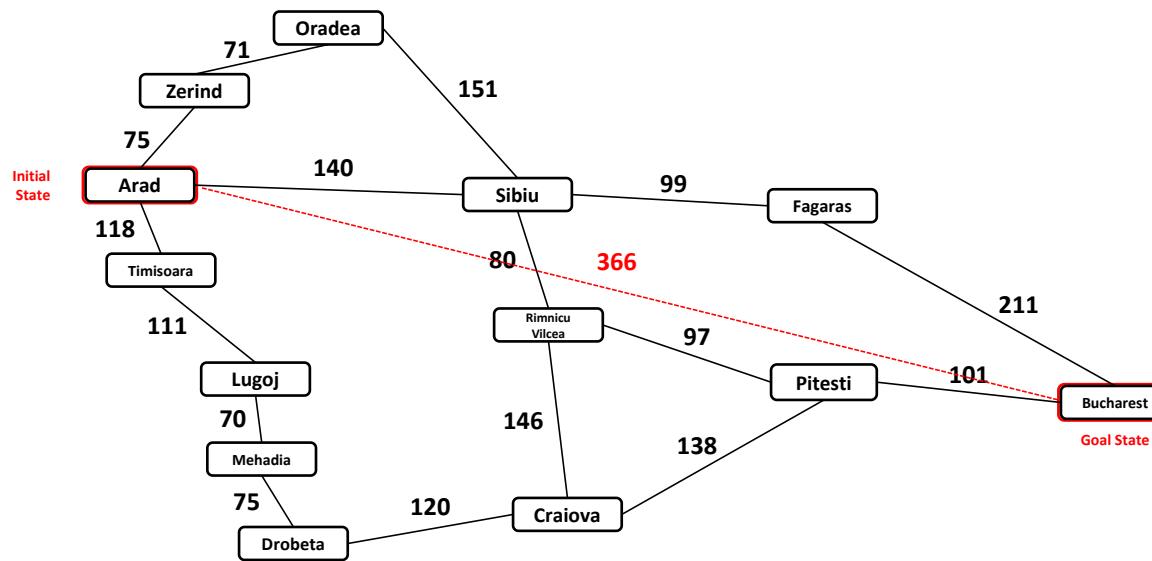
$$f(n) = h(\text{State}_n)$$

A state n with minimum (or maximum) $f(n)$
should be chosen for expansion

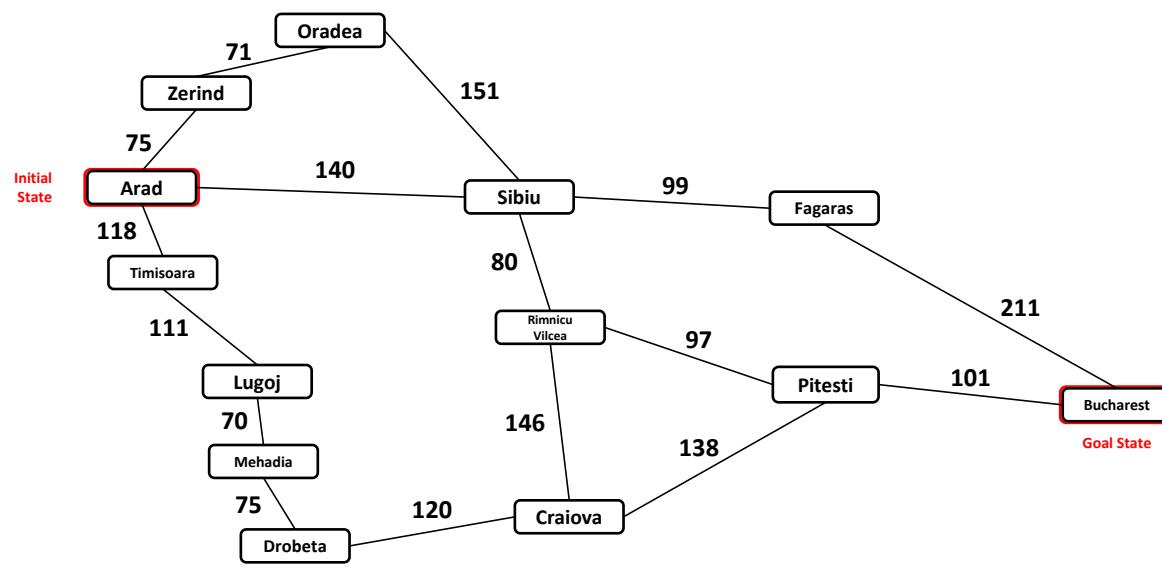
Dracula's Roadtrip: Heuristics $h(n)$

For this particular problem the heuristic function $h(n)$ is defined by a **straight-line (Euclidean) distance** between two states (cities).

“As the crows flies” in other words.



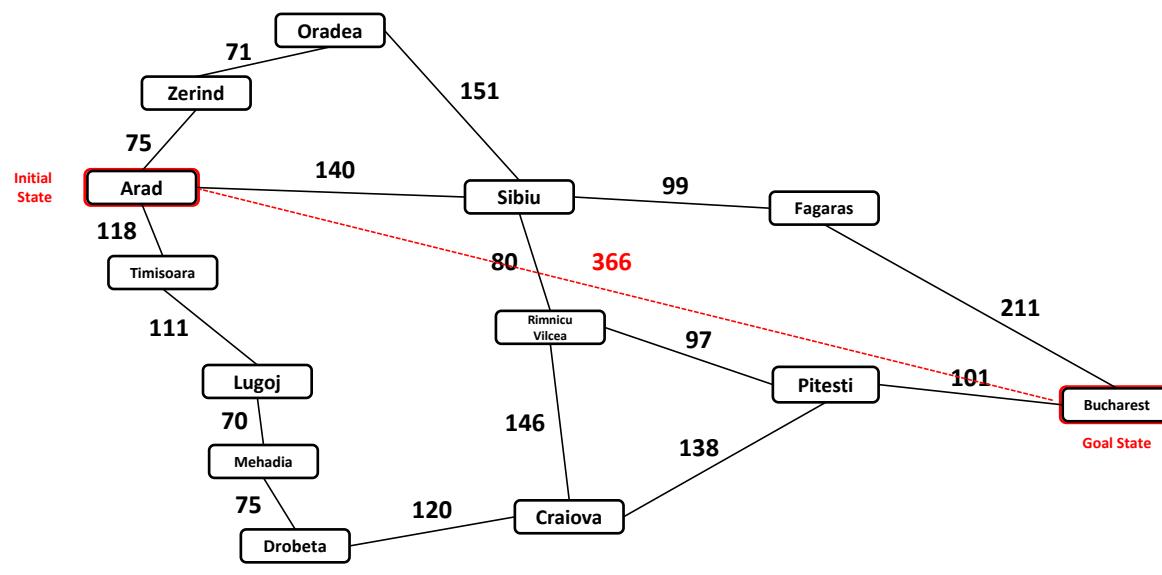
Dracula's Roadtrip: Greedy Best First



Straight-line distance to Bucharest ($\text{h}(\text{State})$):

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

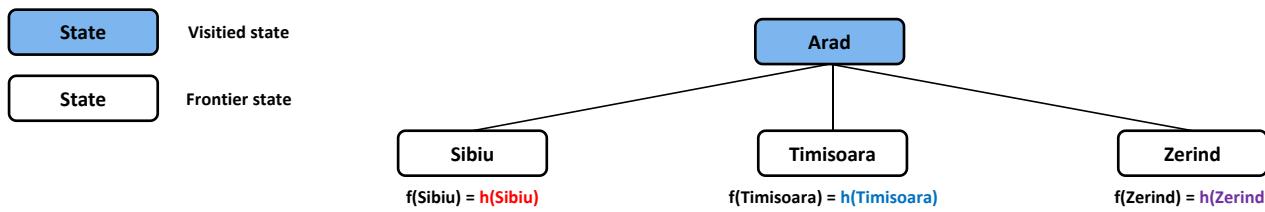
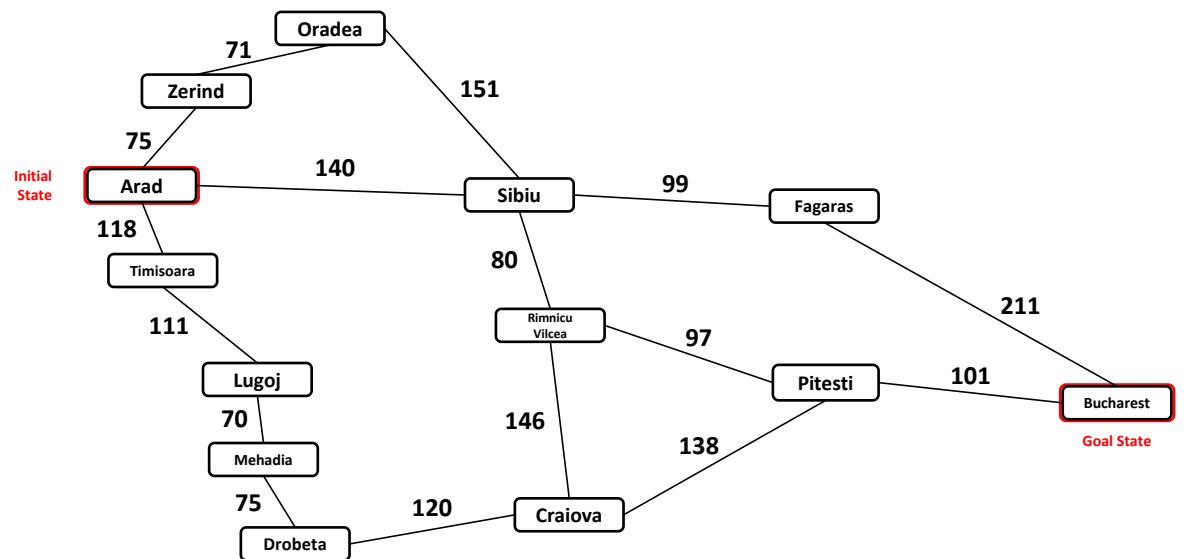
Dracula's Roadtrip: Greedy Best First



Straight-line distance to Bucharest ($h(\text{State})$):

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

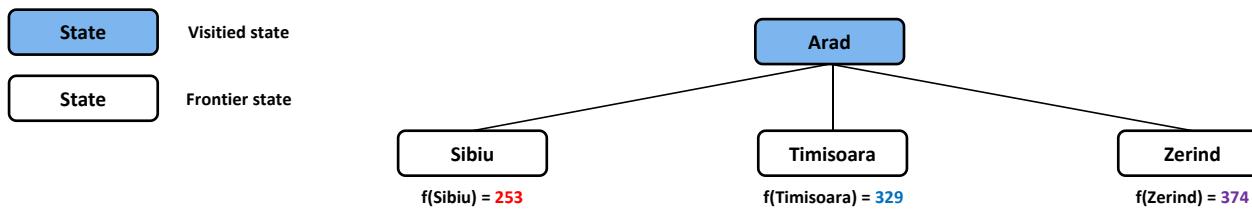
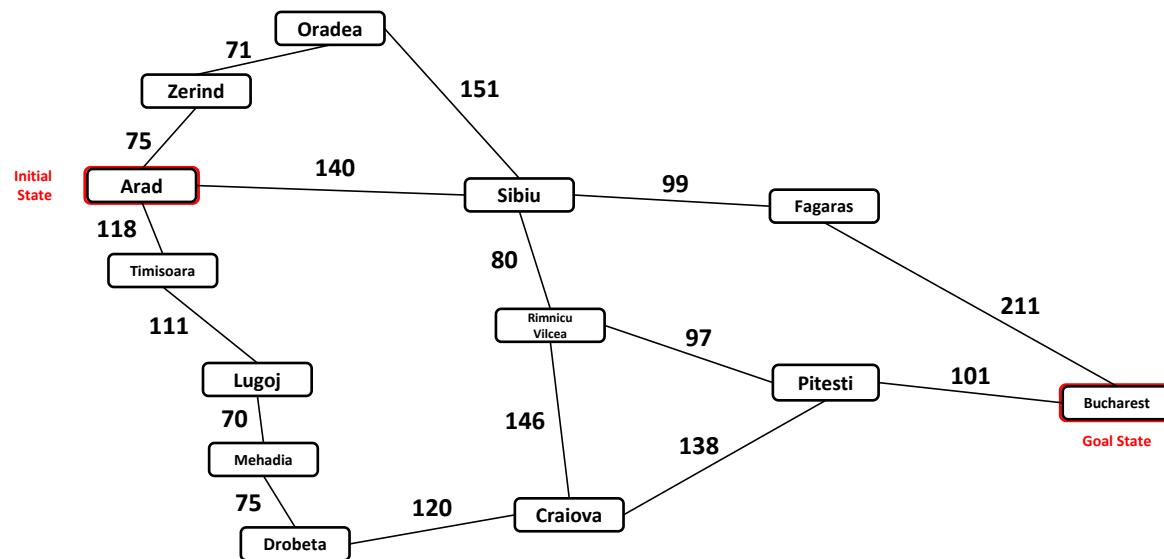
Dracula's Roadtrip: Greedy Best First



Straight-line distance to Bucharest ($h(\text{State})$):

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

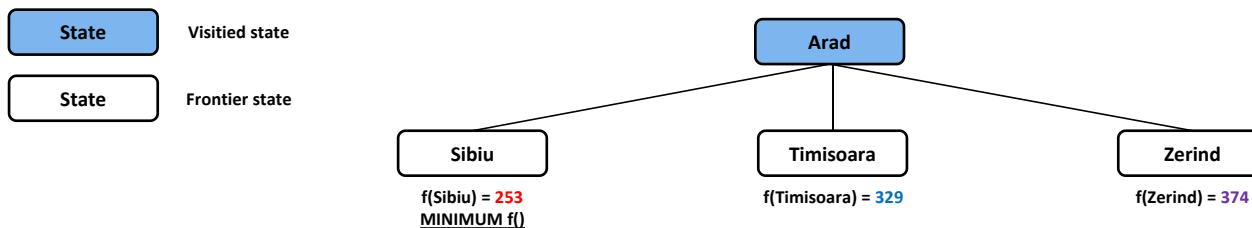
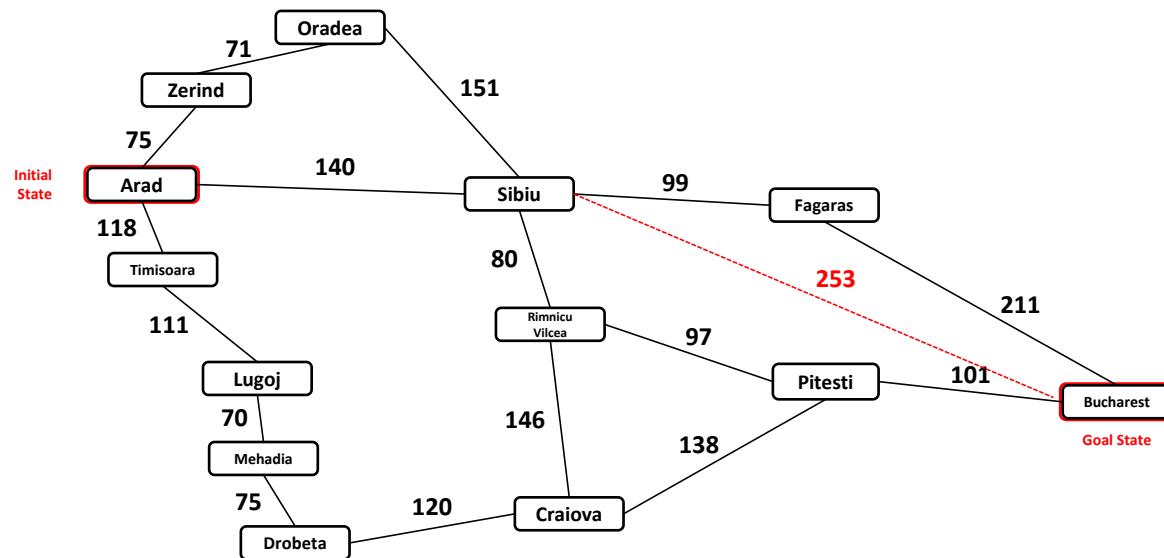
Dracula's Roadtrip: Greedy Best First



Straight-line distance to Bucharest ($h(\text{State})$):

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

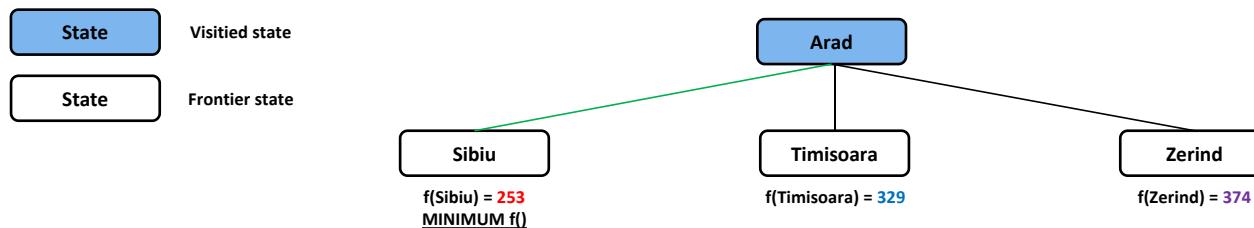
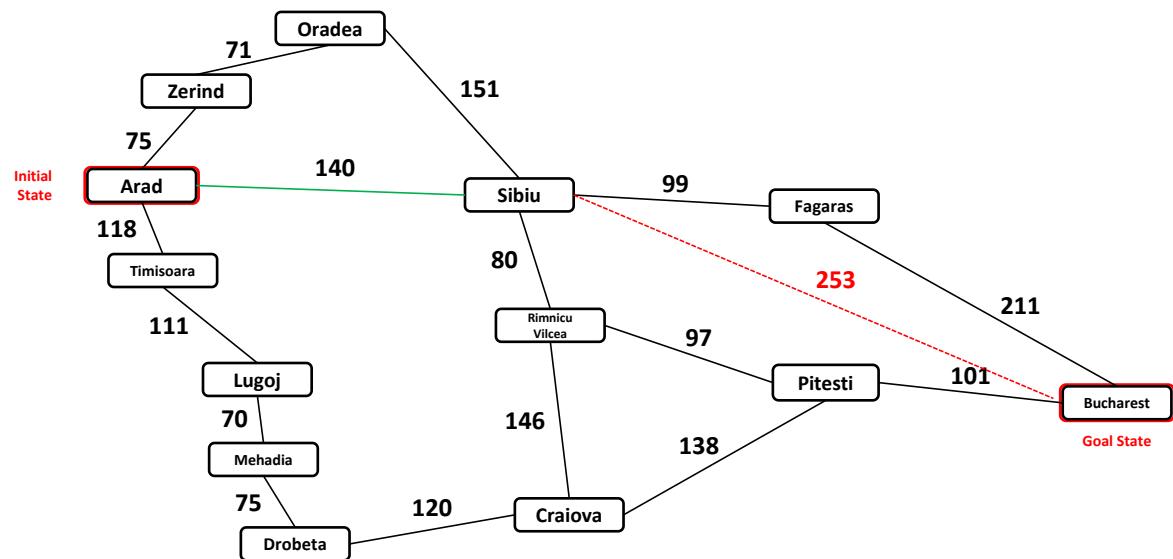
Dracula's Roadtrip: Greedy Best First



Straight-line distance to Bucharest ($h(\text{State})$):

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

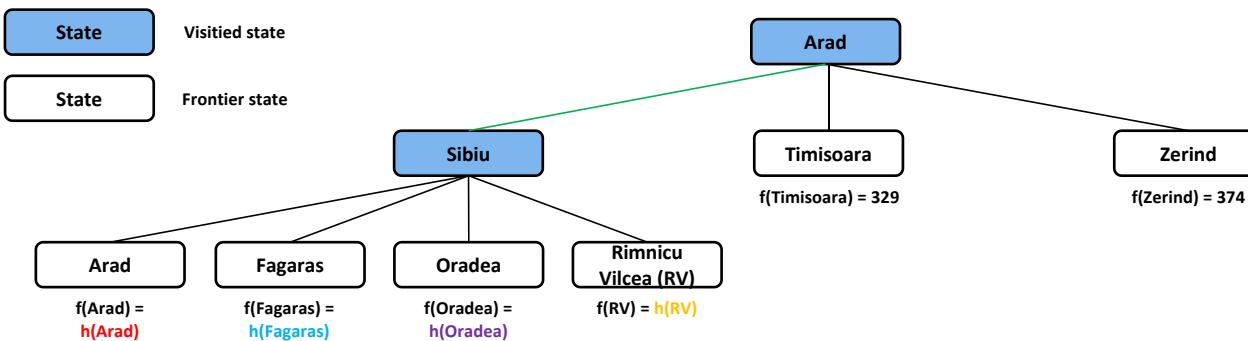
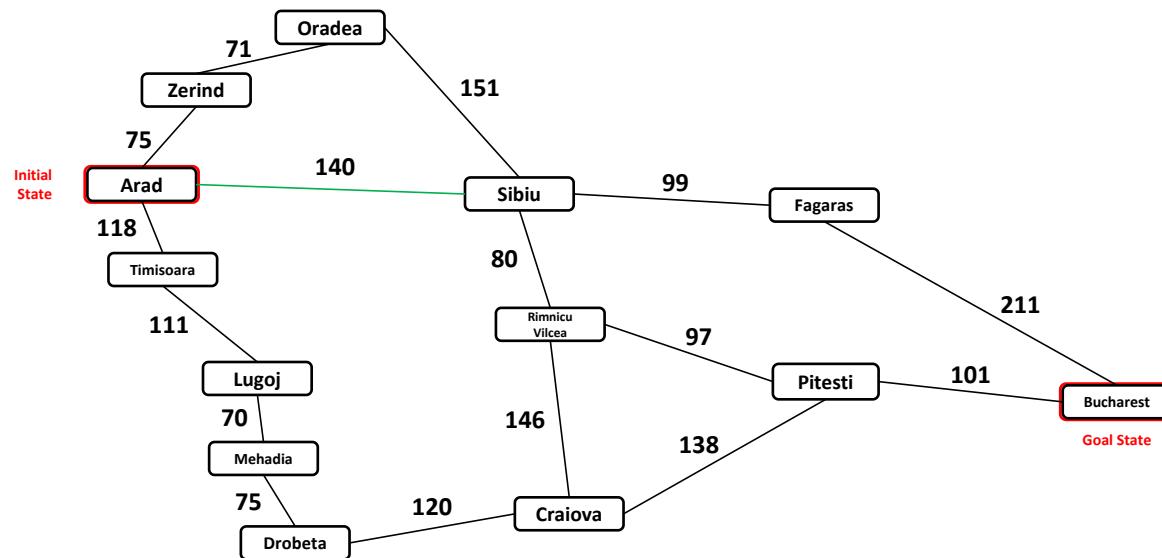
Dracula's Roadtrip: Greedy Best First



Straight-line distance to Bucharest ($h(\text{State})$):

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

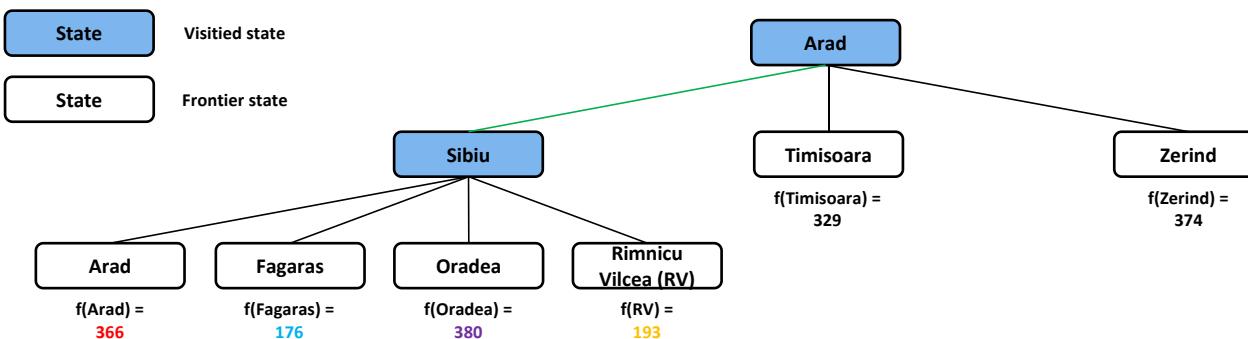
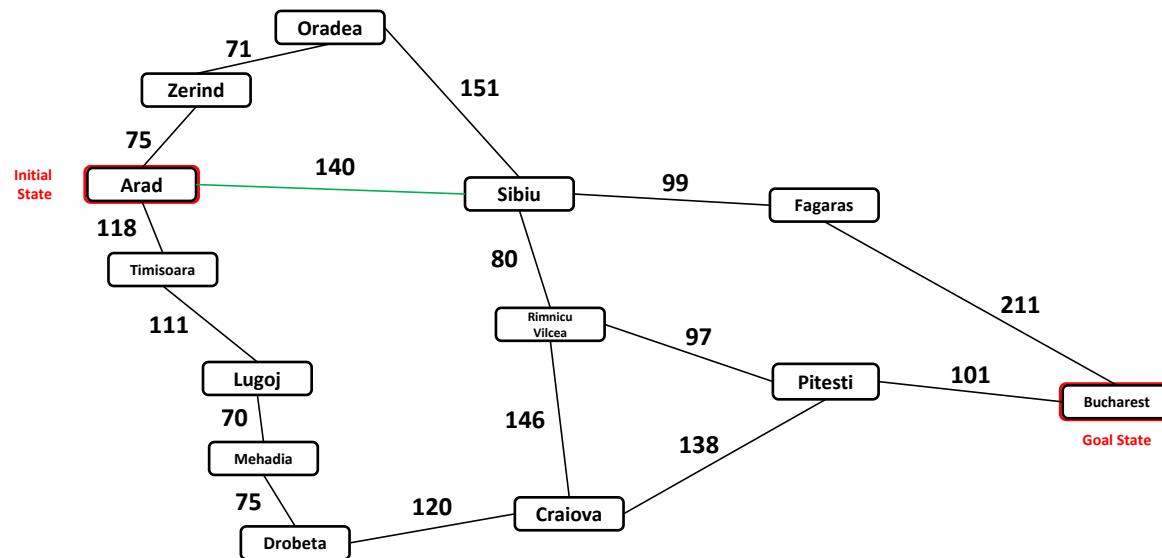
Dracula's Roadtrip: Greedy Best First



Straight-line distance to Bucharest ($h(\text{State})$):

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

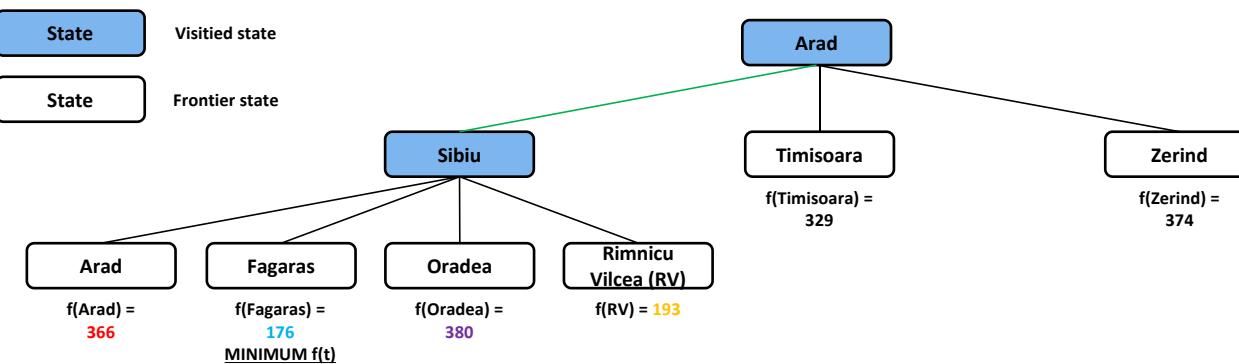
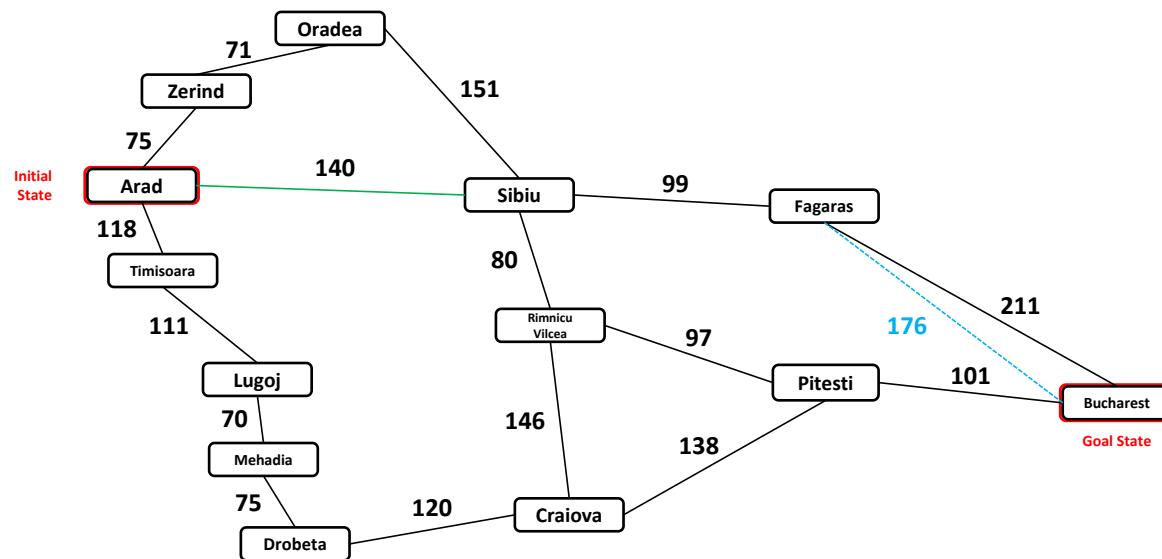
Dracula's Roadtrip: Greedy Best First



Straight-line distance to Bucharest ($h(\text{State})$):

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

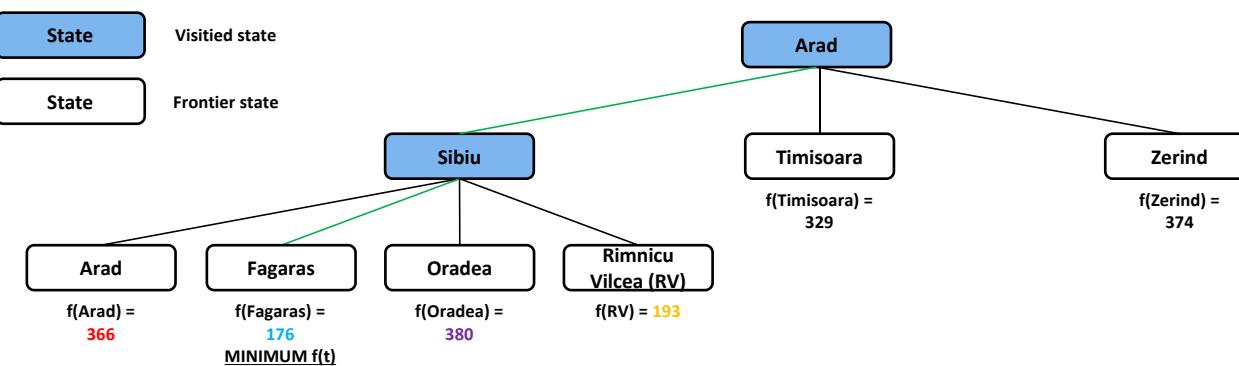
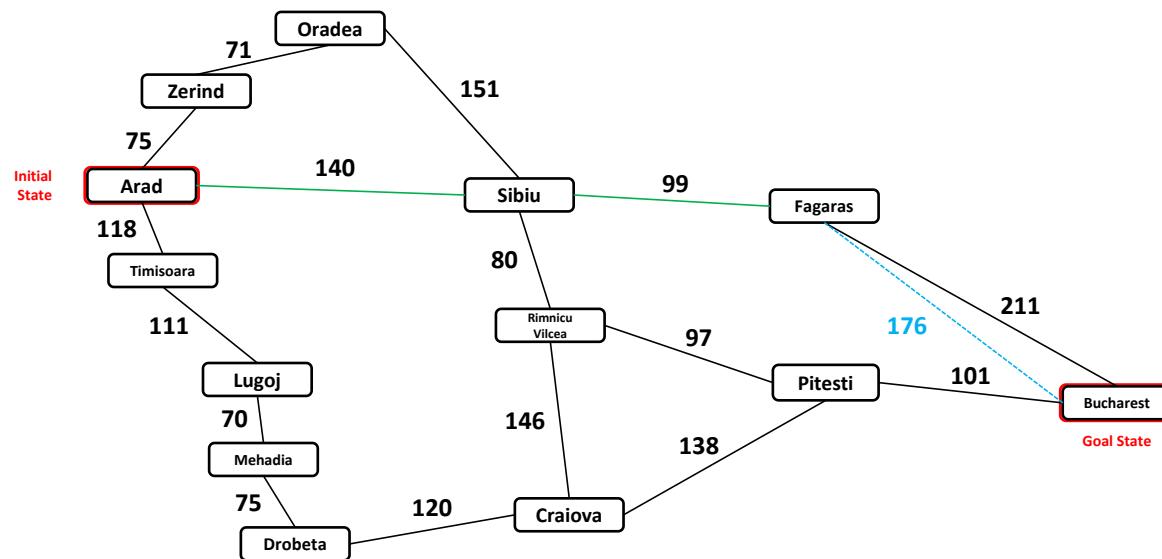
Dracula's Roadtrip: Greedy Best First



Straight-line distance to Bucharest ($h(\text{State})$):

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

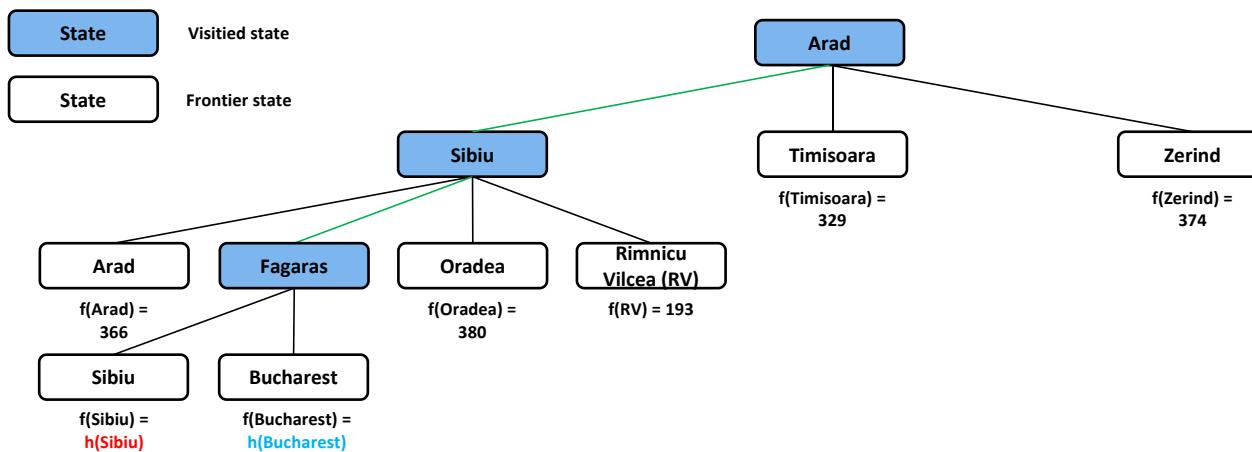
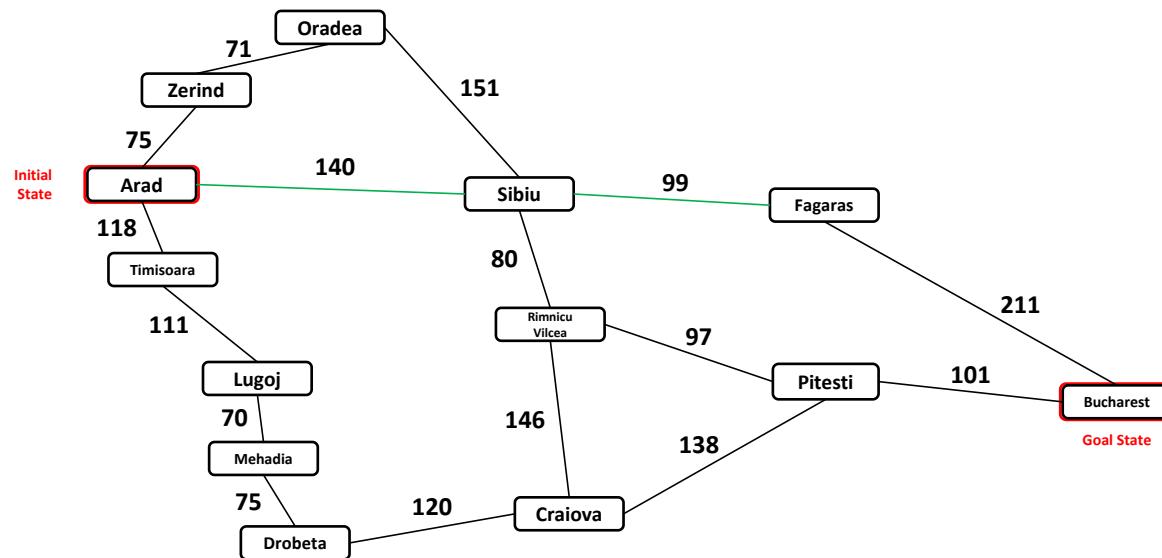
Dracula's Roadtrip: Greedy Best First



Straight-line distance to Bucharest ($h(\text{State})$):

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

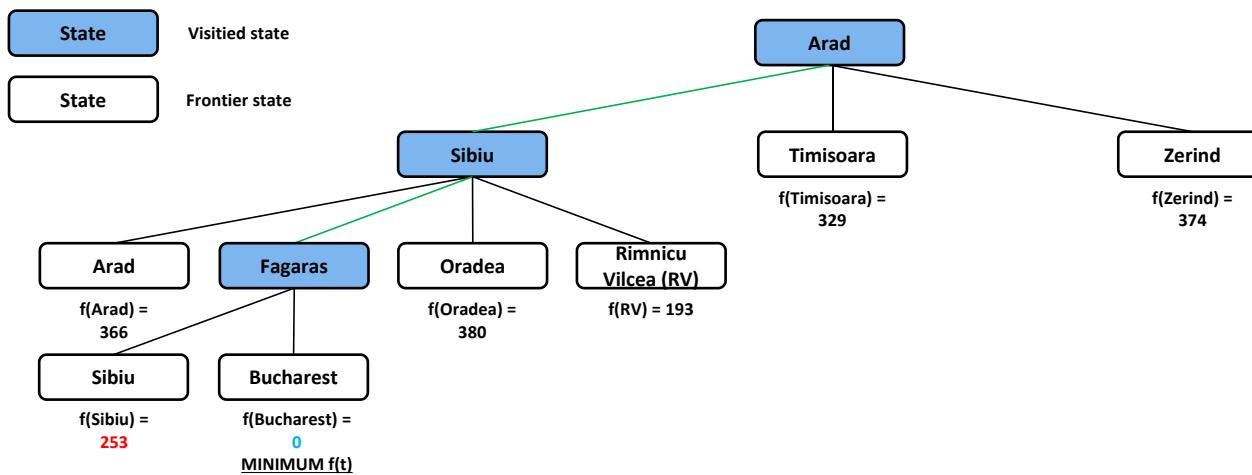
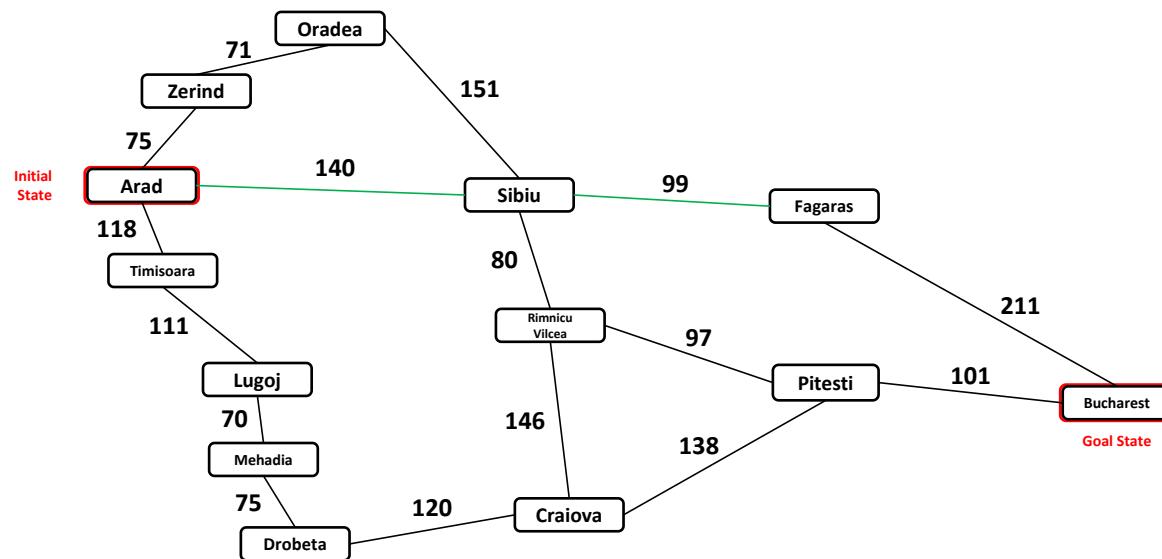
Dracula's Roadtrip: Greedy Best First



Straight-line distance to Bucharest ($h(\text{State})$):

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

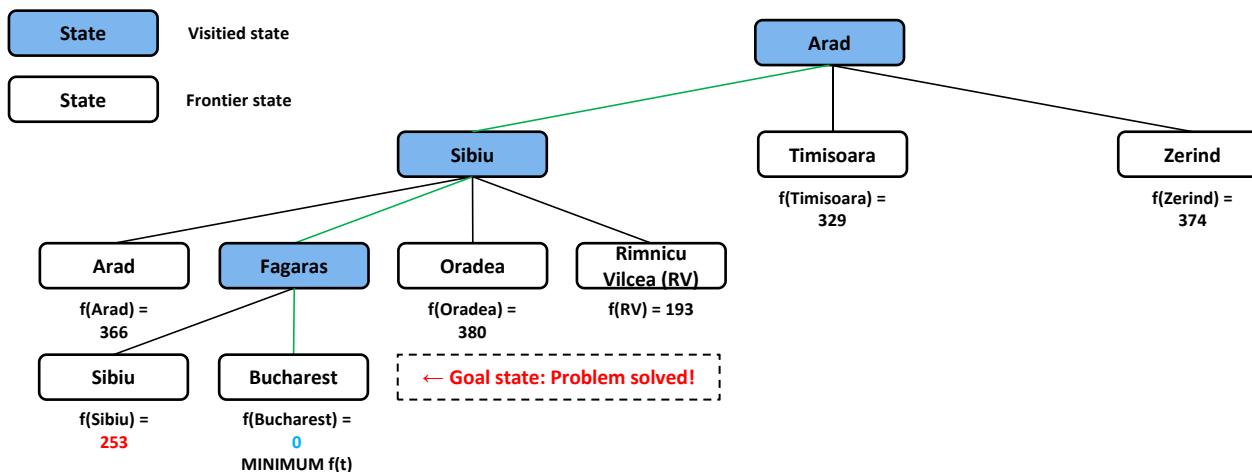
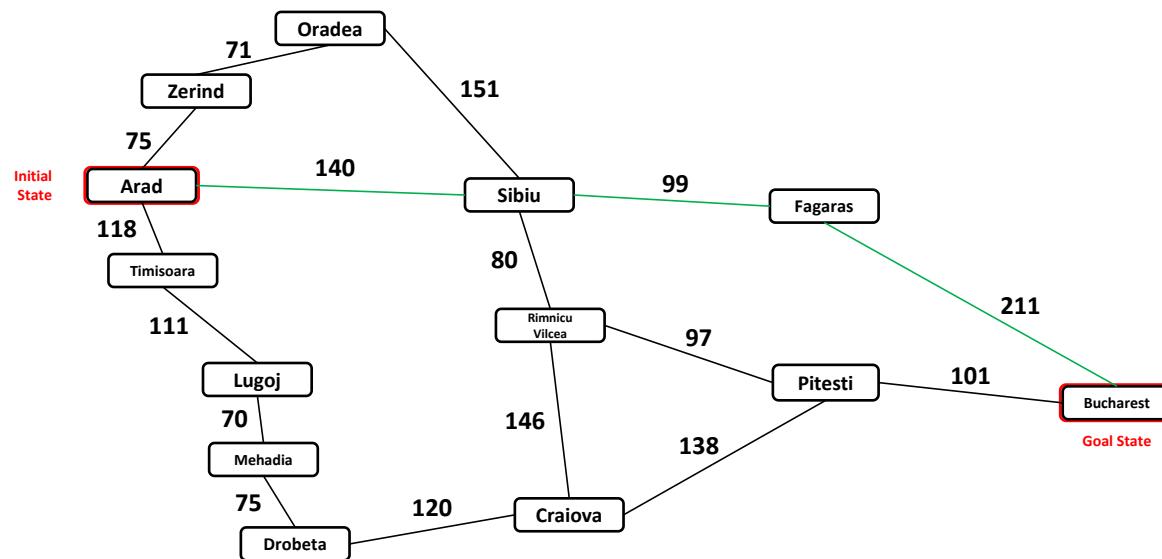
Dracula's Roadtrip: Greedy Best First



Straight-line distance to Bucharest ($h(\text{State})$):

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

Dracula's Roadtrip: Greedy Best First



Straight-line distance to Bucharest ($h(\text{State})$):

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

A* Algorithm: Evaluation Function

Calculate / obtain:

$$f(n) = g(\text{State}_n) + h(\text{State}_n)$$

where:

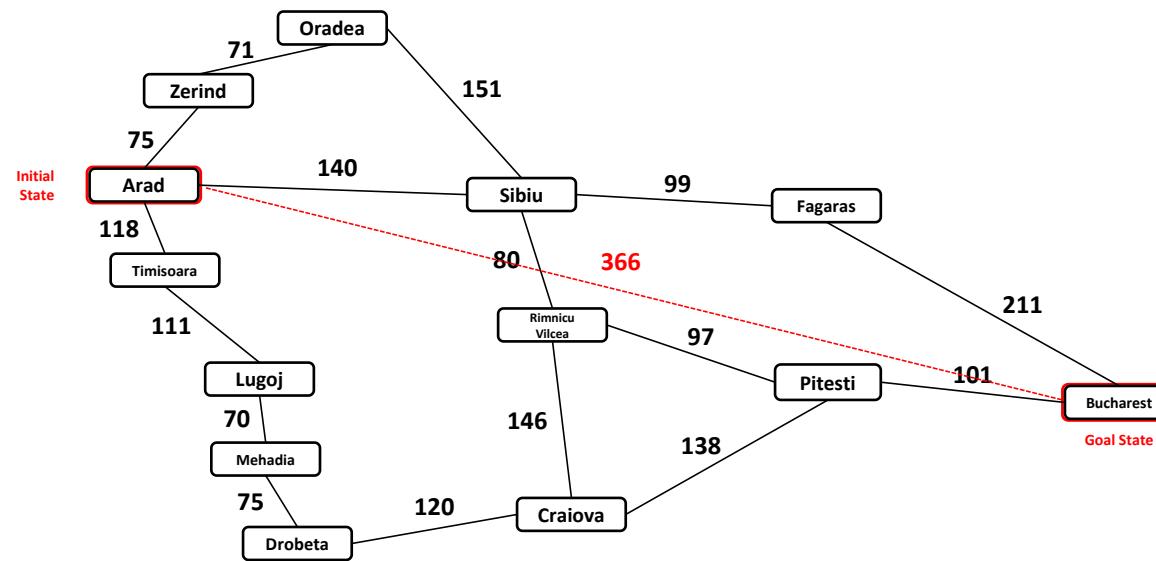
- $g(n)$ - initial node to node n path cost
- $h(n)$ - **estimated cost** of the best path that continues from node n to a goal node

A state n with minimum (maximum) $f(n)$ should be chosen for expansion

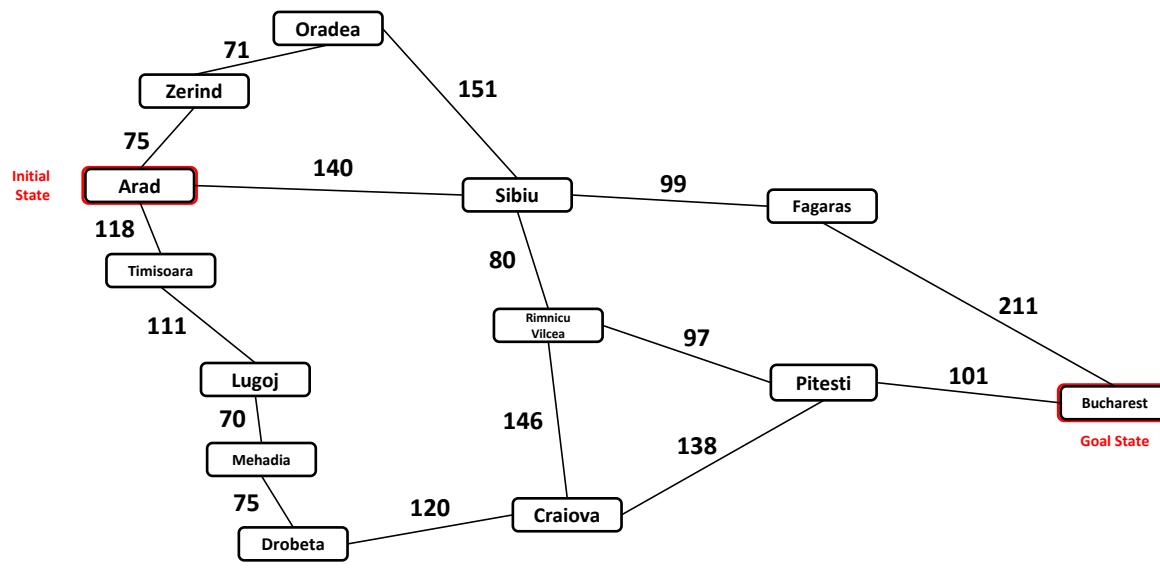
Dracula's Roadtrip: Heuristics $h(n)$

For this particular problem the heuristic function $h(n)$ is defined by a **straight-line (Euclidean) distance** between two states (cities).

“As the crows flies” in other words.



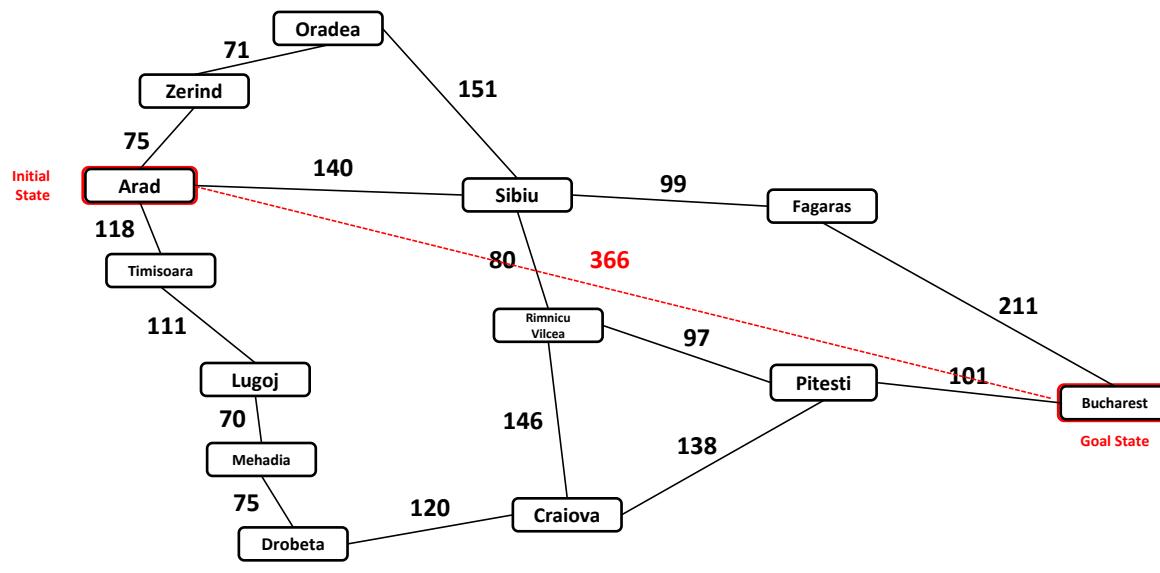
Dracula's Roadtrip: A* Search



Straight-line distance to Bucharest ($h(\text{State})$):

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

Dracula's Roadtrip: A* Search



State

Visited state

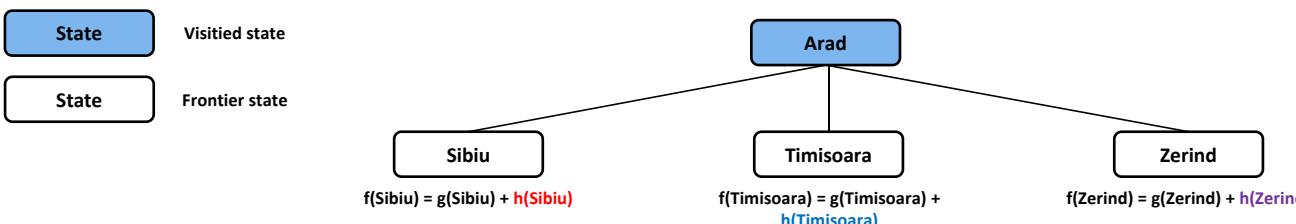
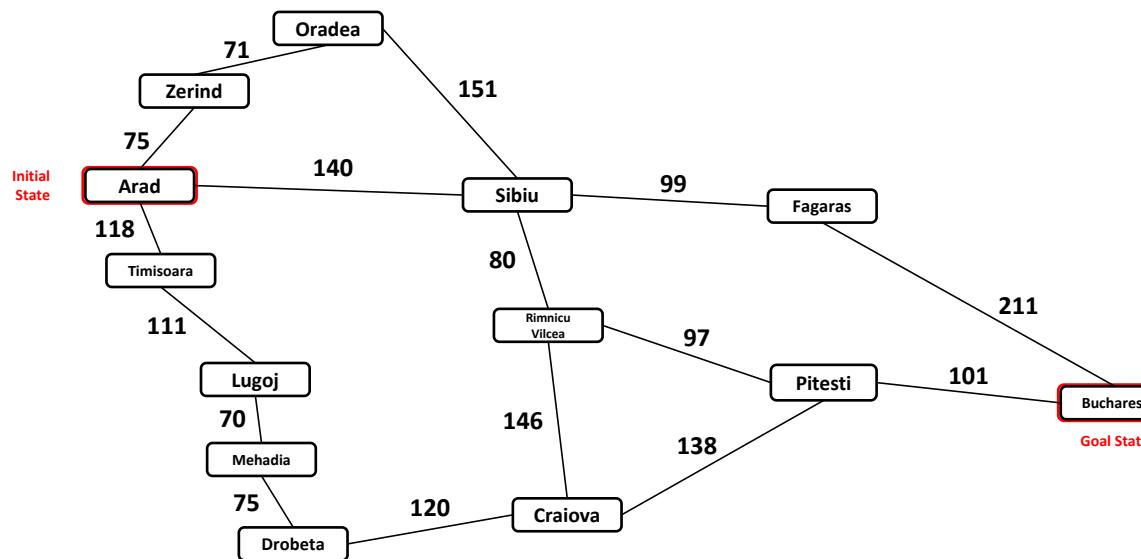
State

Frontier state

Arad

$$f(\text{Arad}) = 0 + 366 = 366$$

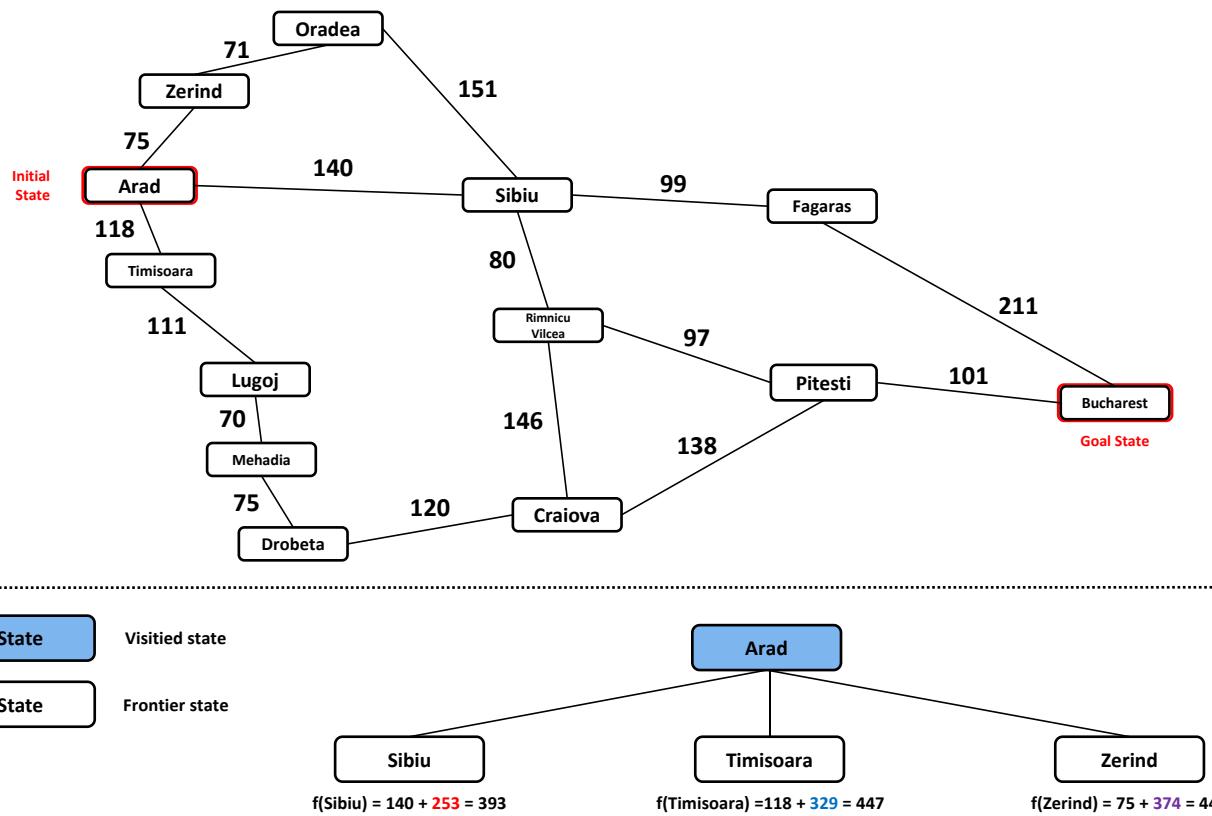
Dracula's Roadtrip: A* Search



Straight-line distance to Bucharest ($h(\text{State})$):

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

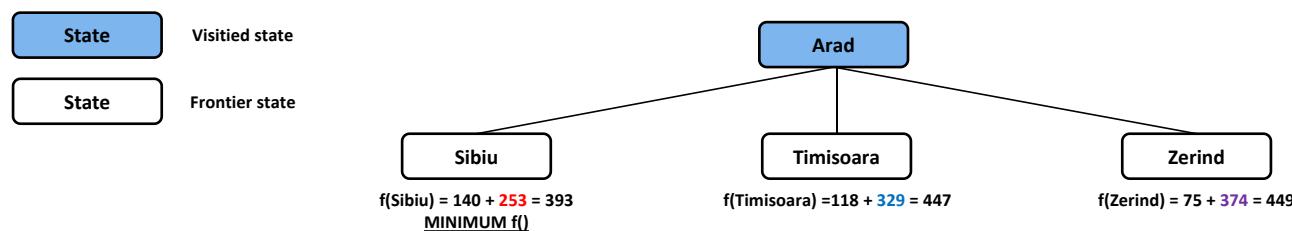
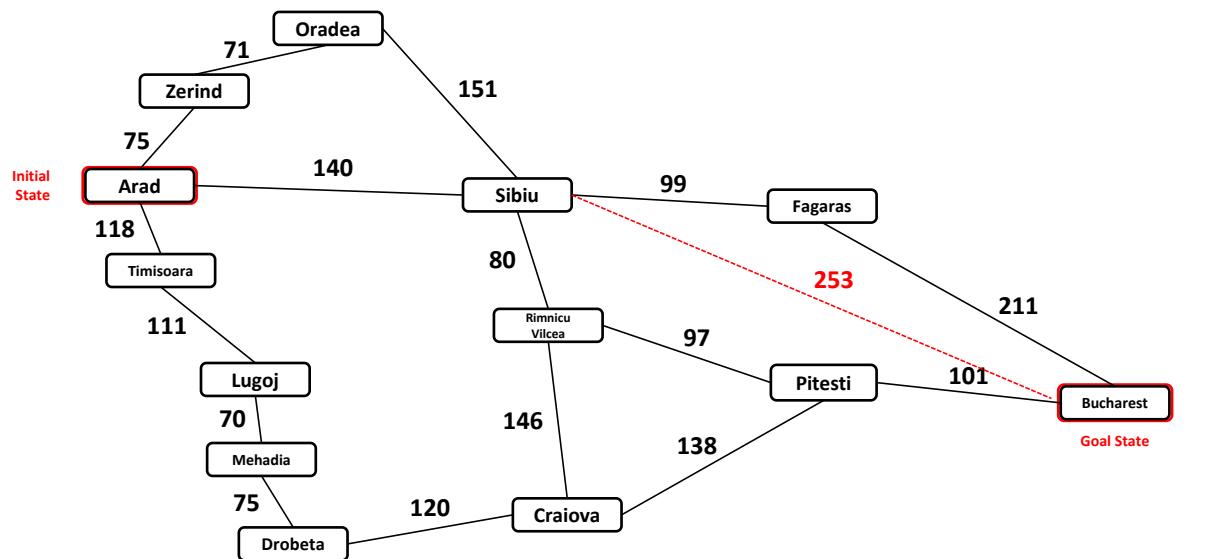
Dracula's Roadtrip: A* Search



Straight-line distance to Bucharest ($h(\text{State})$):

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

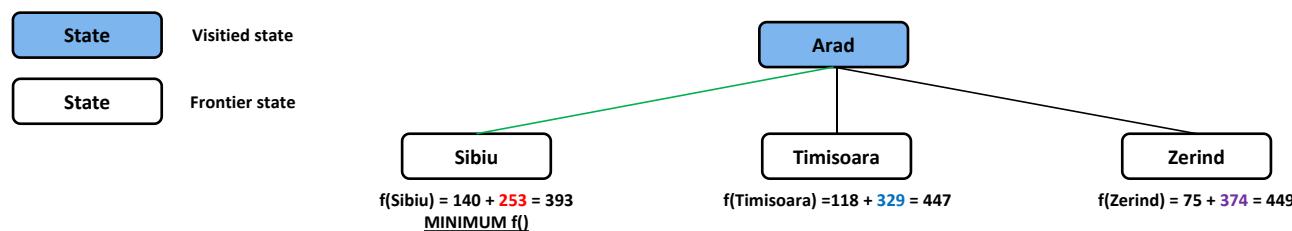
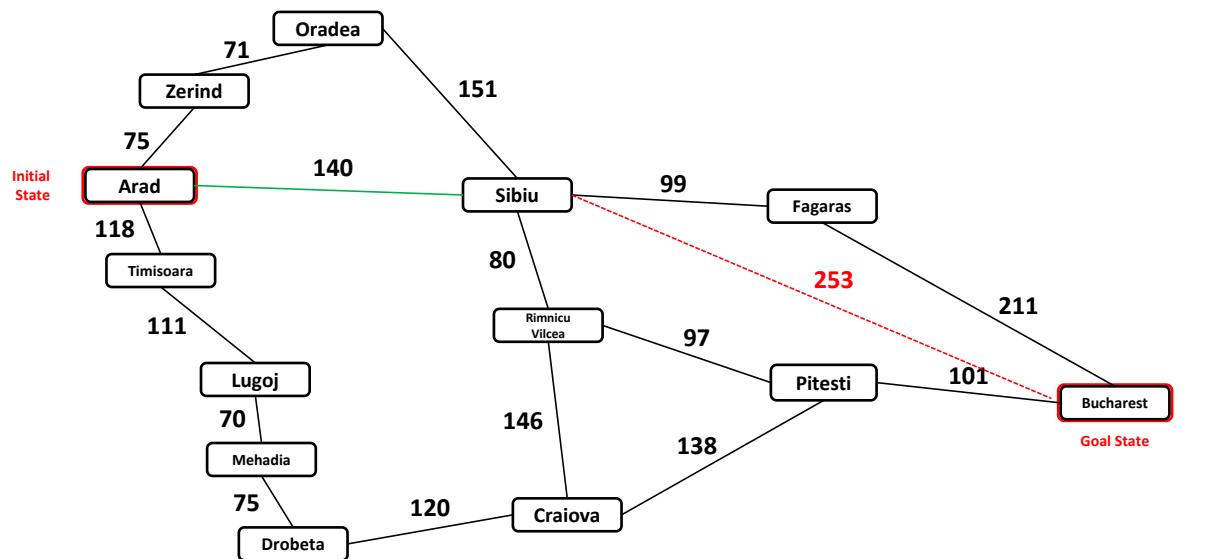
Dracula's Roadtrip: A* Search



Straight-line distance to Bucharest ($h(\text{State})$):

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

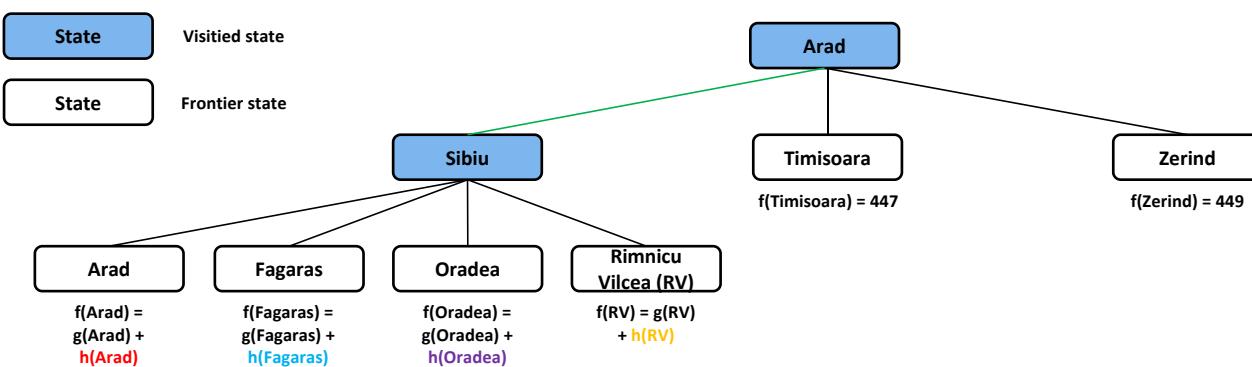
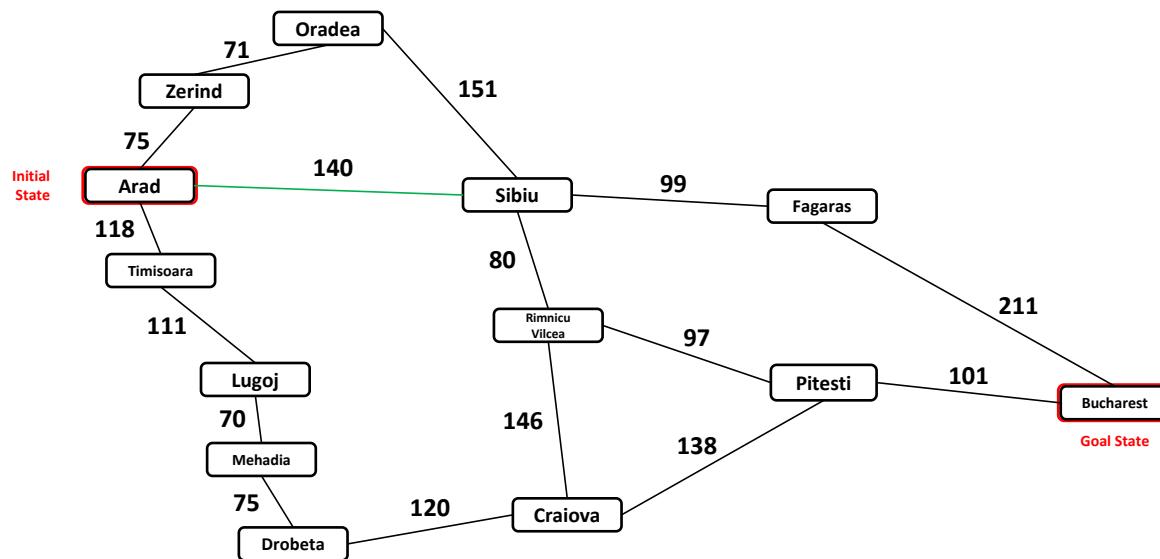
Dracula's Roadtrip: A* Search



Straight-line distance to Bucharest ($h(\text{State})$):

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

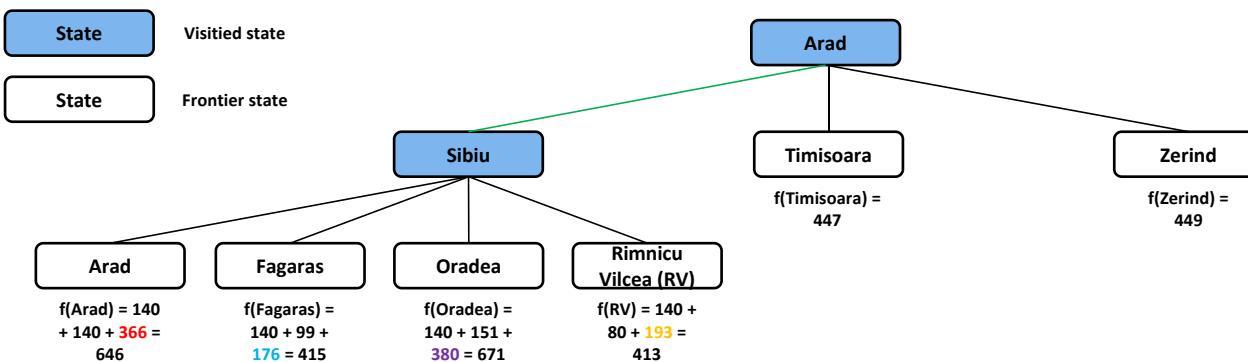
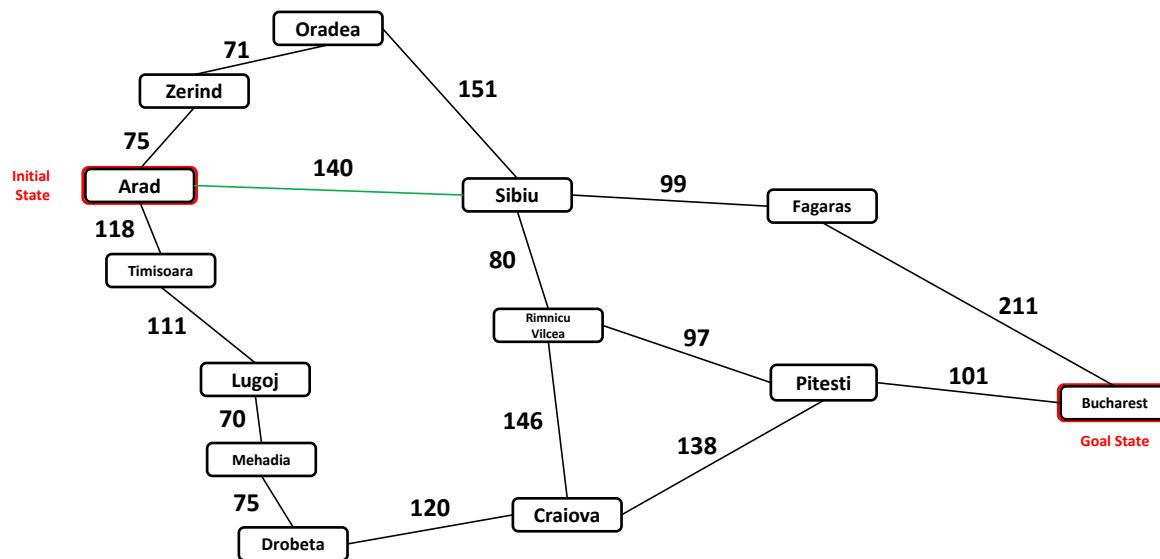
Dracula's Roadtrip: A* Search



Straight-line distance to Bucharest ($h(\text{State})$):

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

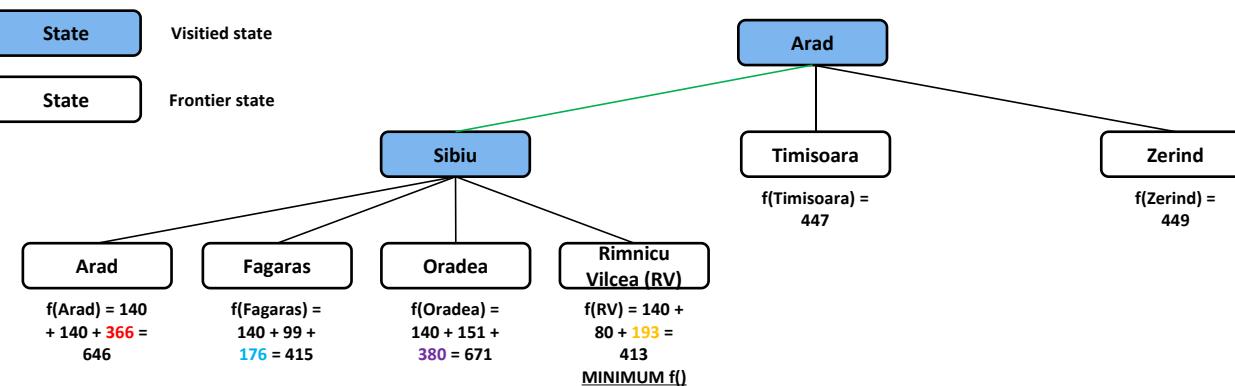
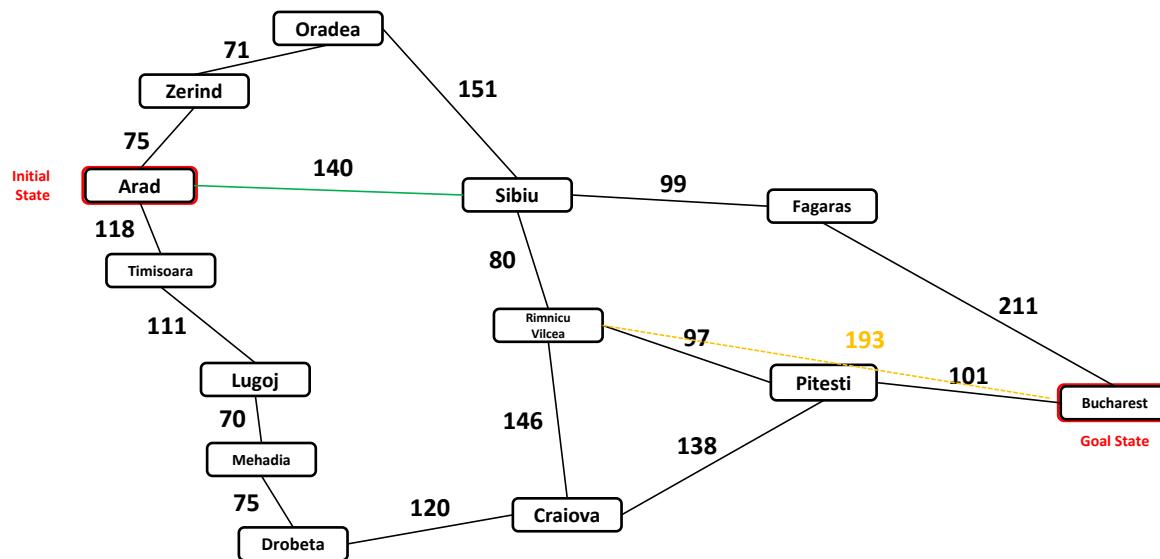
Dracula's Roadtrip: A* Search



Straight-line distance to Bucharest ($h(\text{State})$):

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

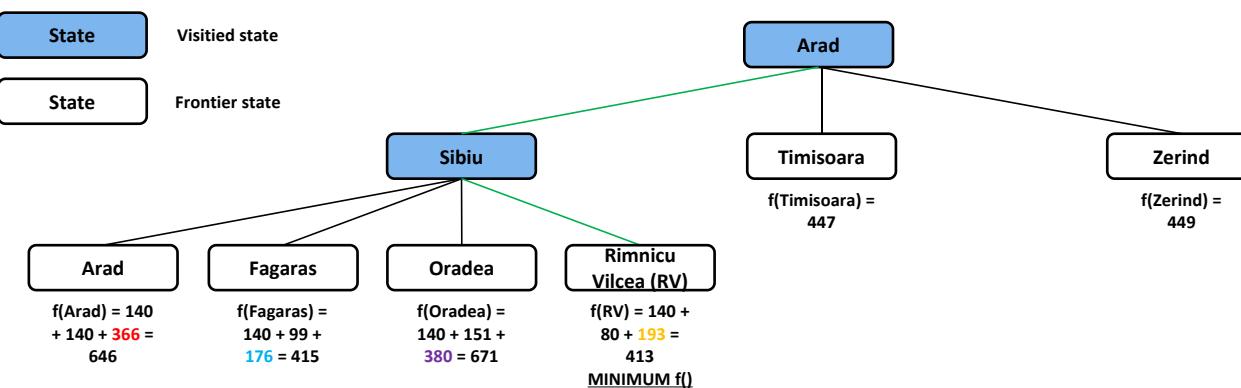
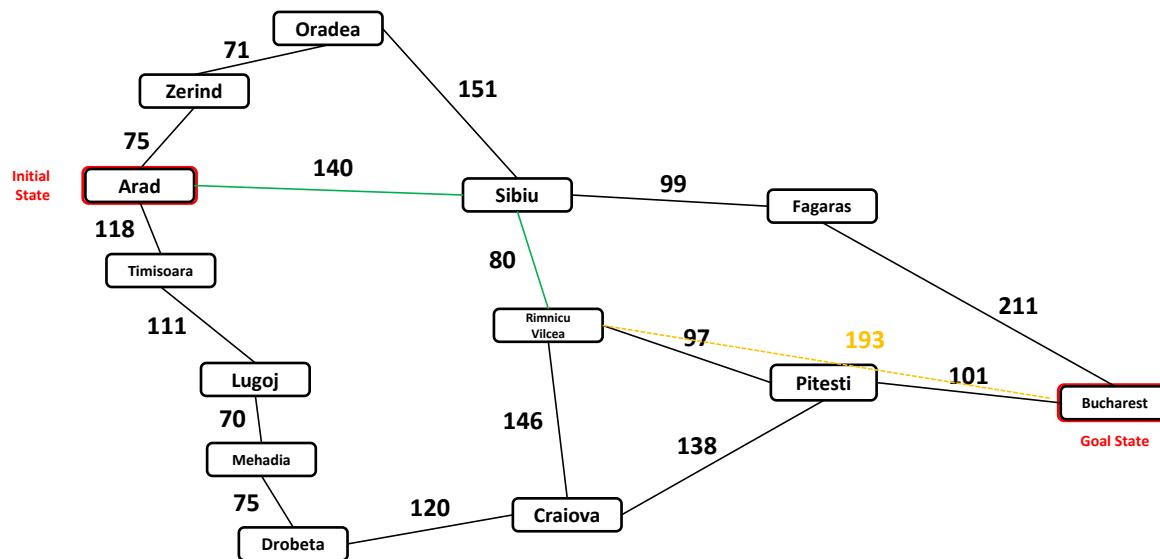
Dracula's Roadtrip: A* Search



Straight-line distance to Bucharest ($h(\text{State})$):

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

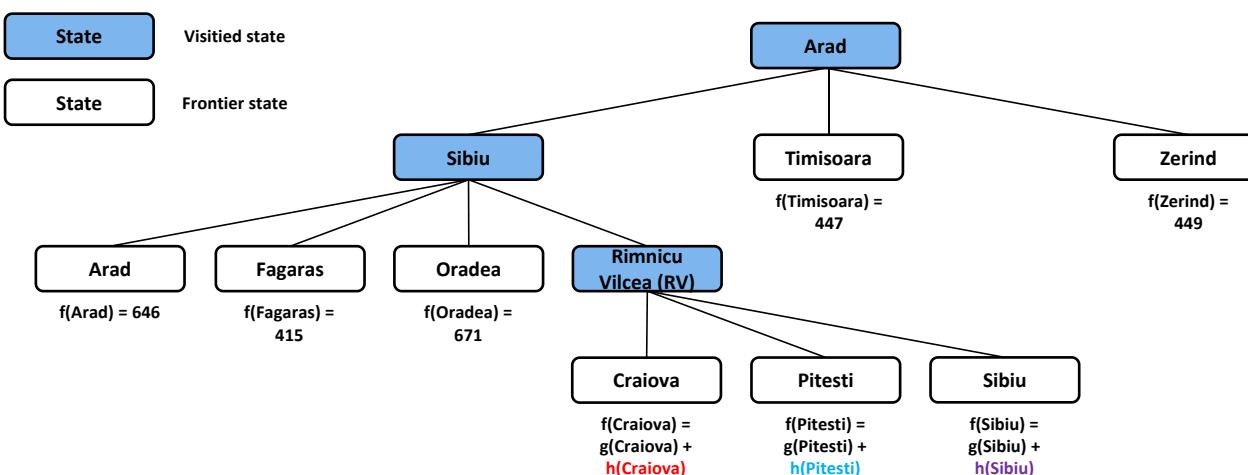
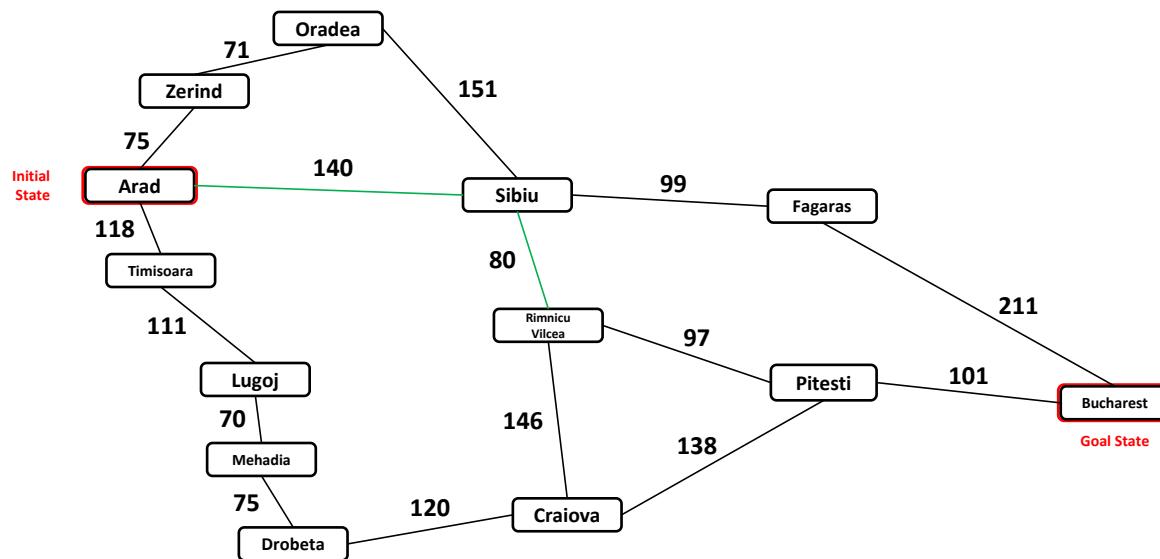
Dracula's Roadtrip: A* Search



Straight-line distance to Bucharest ($h(\text{State})$):

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

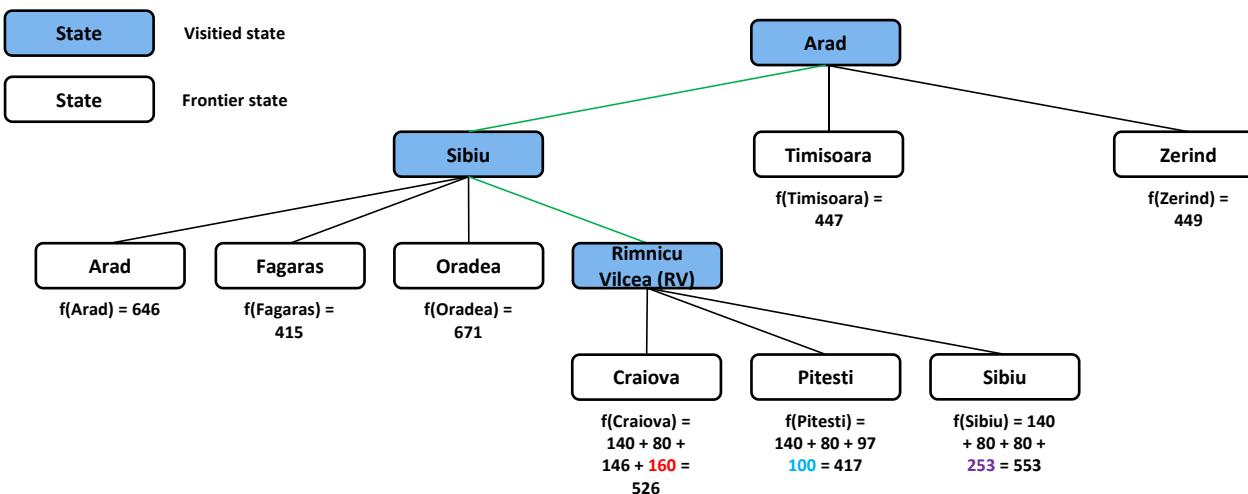
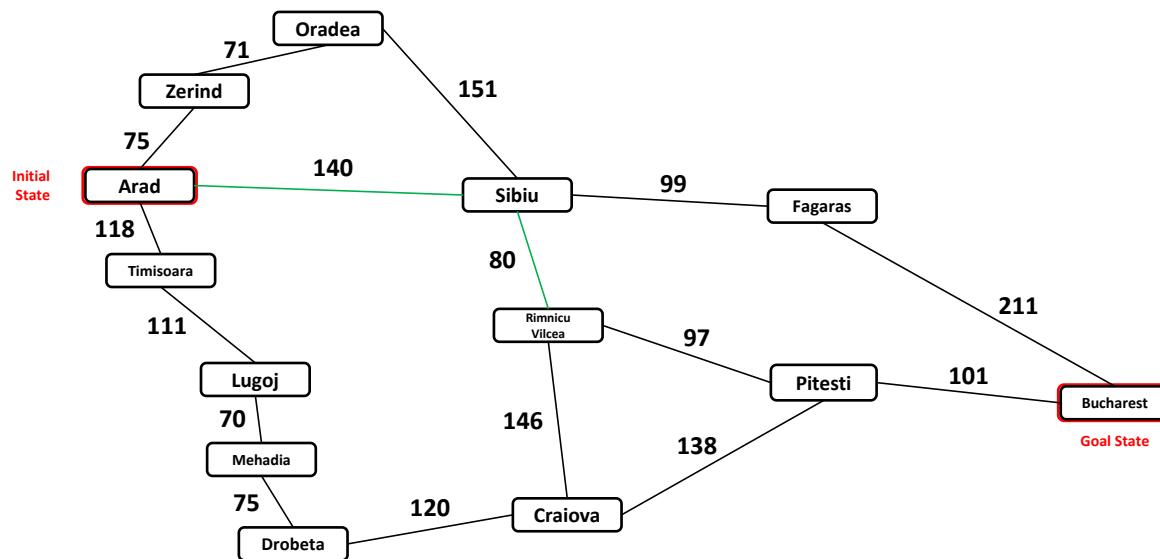
Dracula's Roadtrip: A* Search



Straight-line distance to Bucharest ($h(\text{State})$):

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

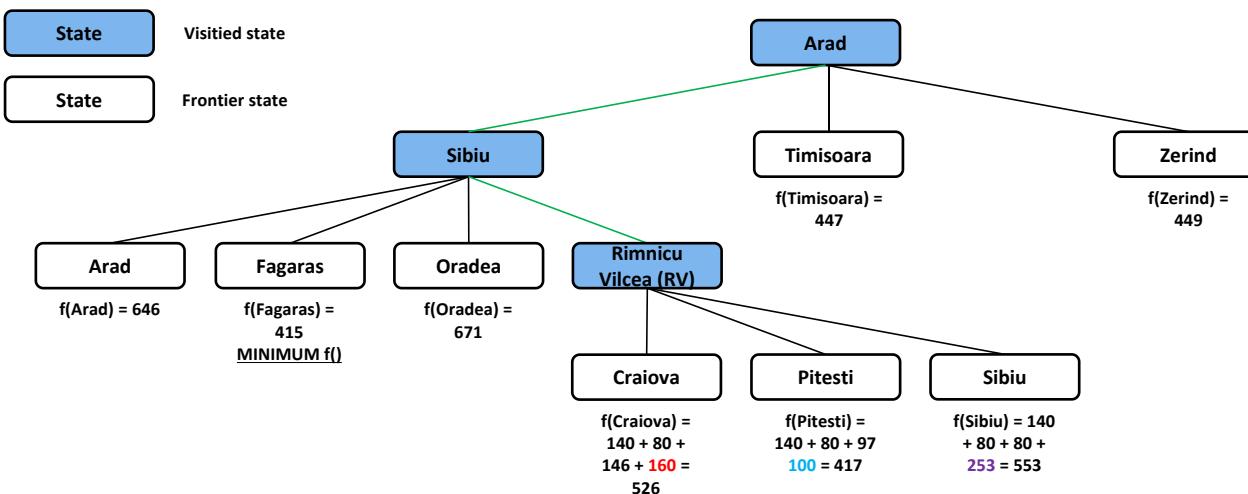
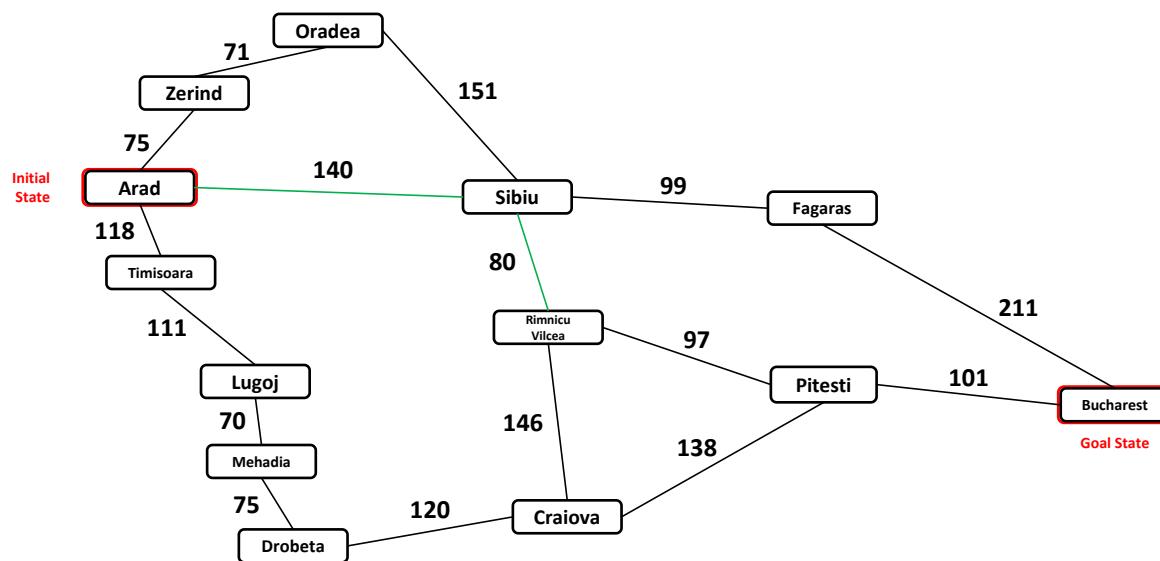
Dracula's Roadtrip: A* Search



Straight-line distance to Bucharest ($h(\text{State})$):

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

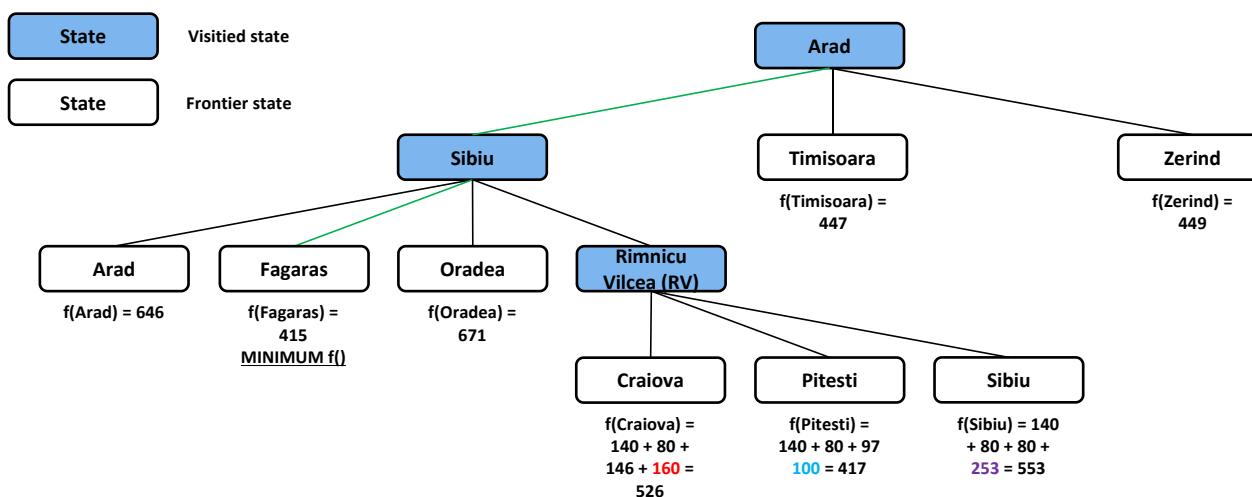
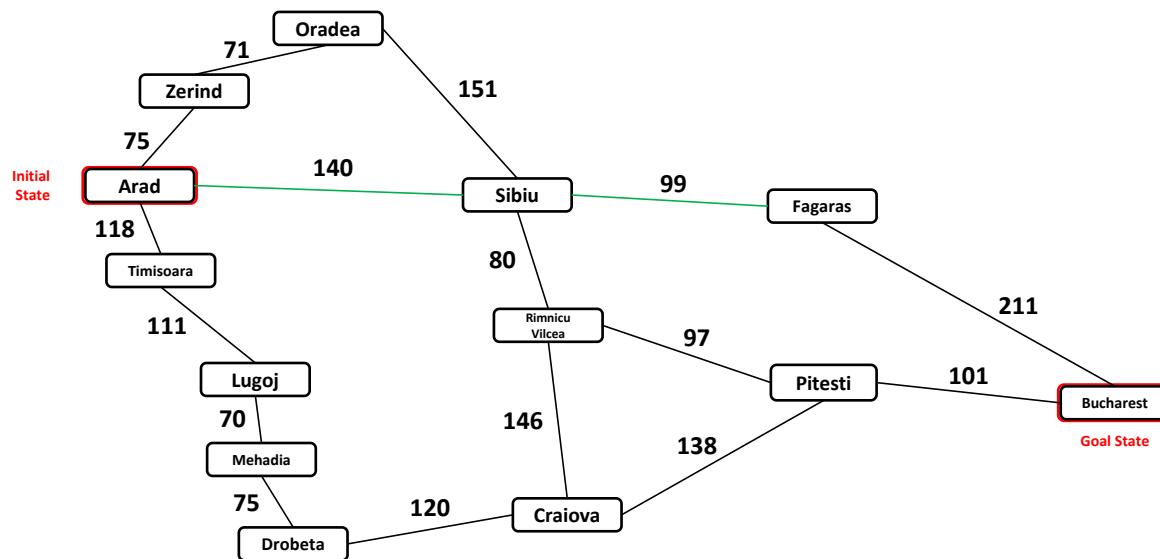
Dracula's Roadtrip: A* Search



Straight-line distance to Bucharest ($h(\text{State})$):

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

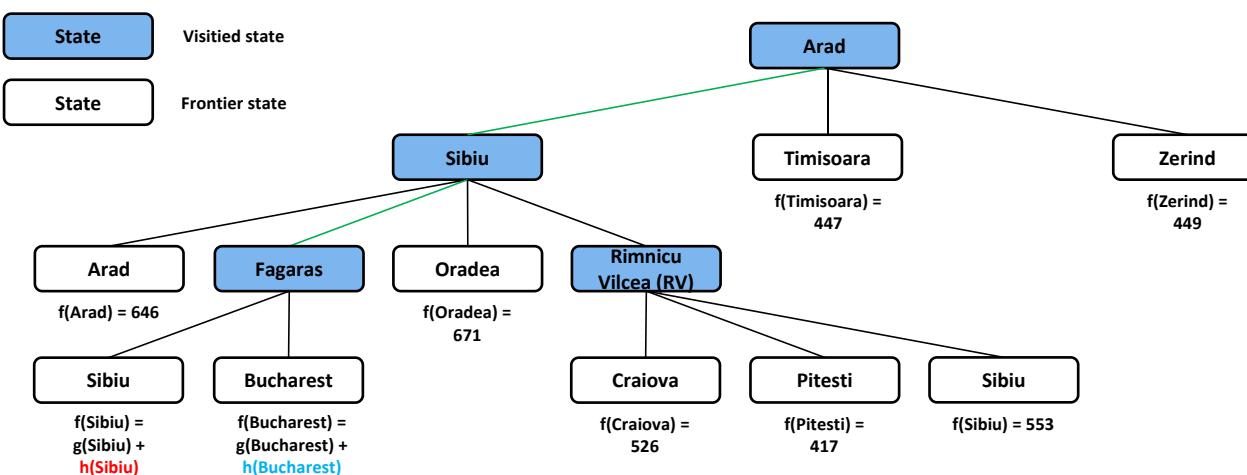
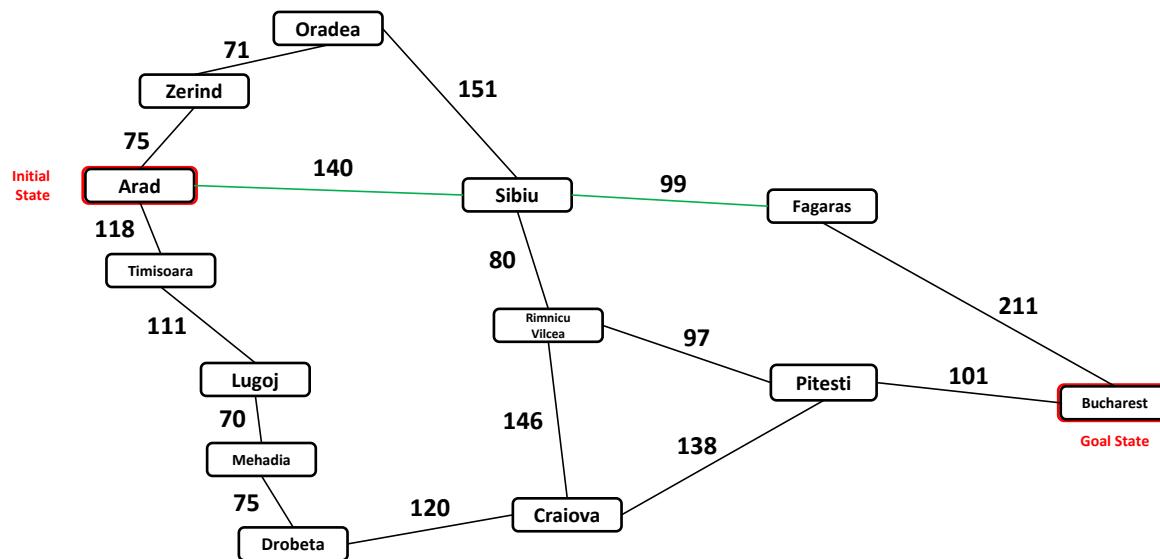
Dracula's Roadtrip: A* Search



Straight-line distance to Bucharest ($h(\text{State})$):

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

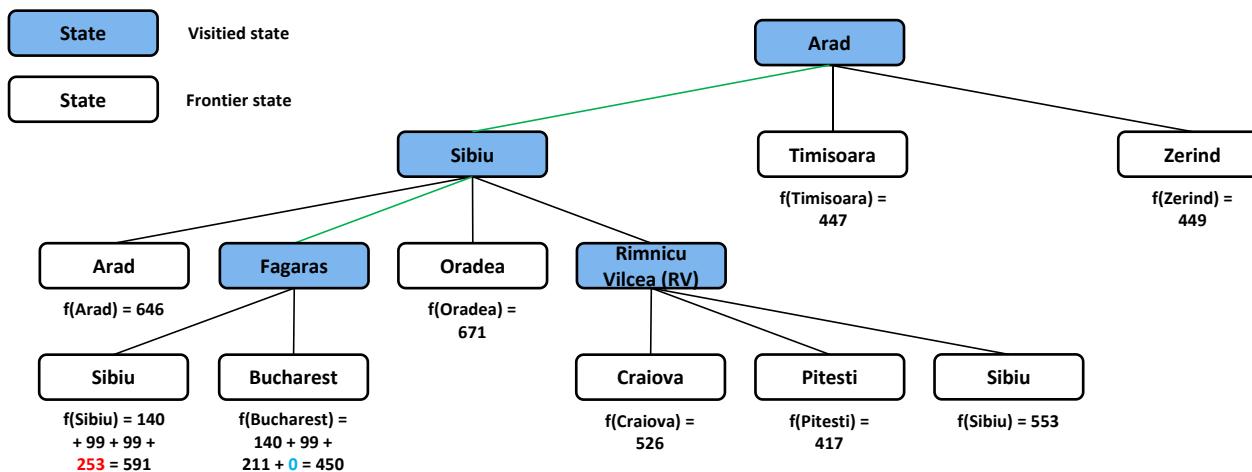
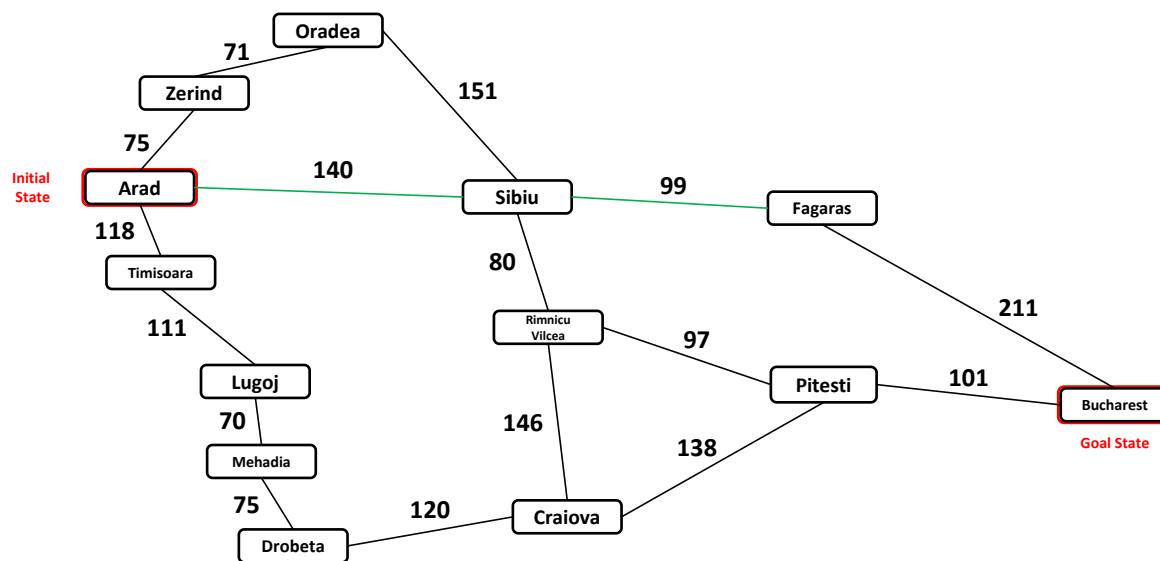
Dracula's Roadtrip: A* Search



Straight-line distance to Bucharest ($h(\text{State})$):

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

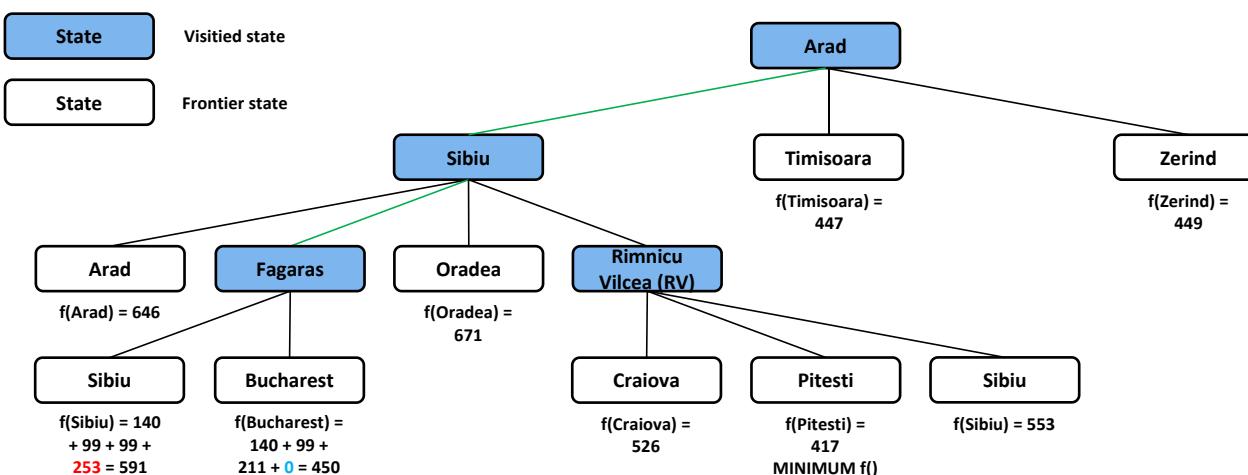
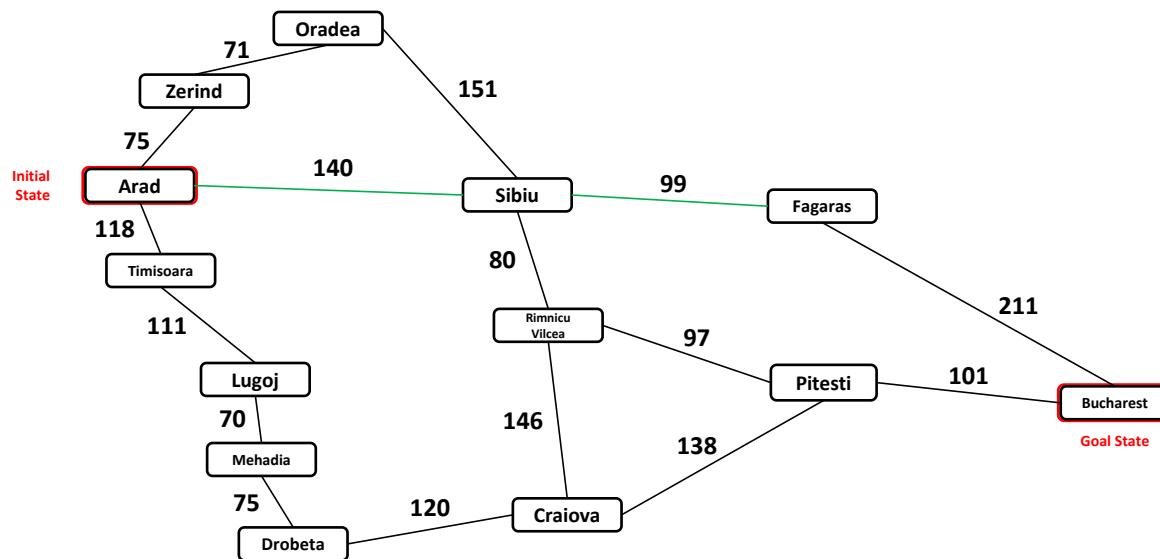
Dracula's Roadtrip: A* Search



Straight-line distance to Bucharest ($h(\text{State})$):

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

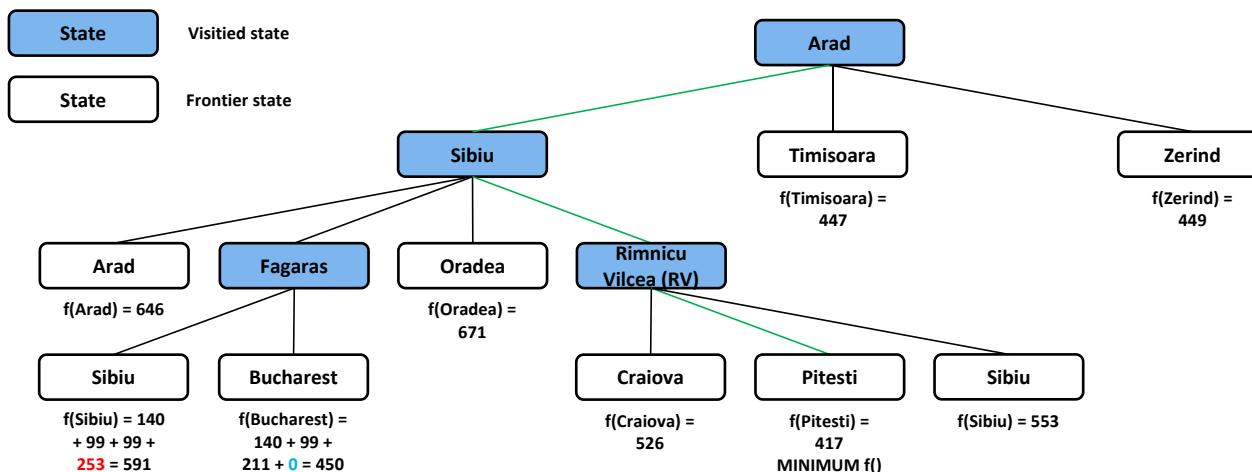
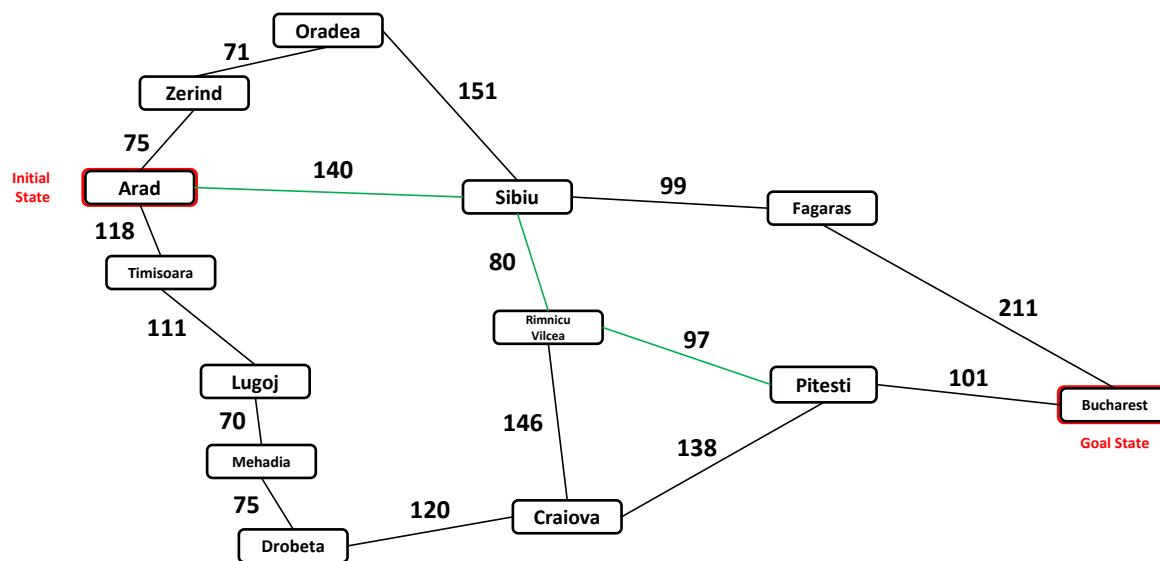
Dracula's Roadtrip: A* Search



Straight-line distance to Bucharest ($h(\text{State})$):

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

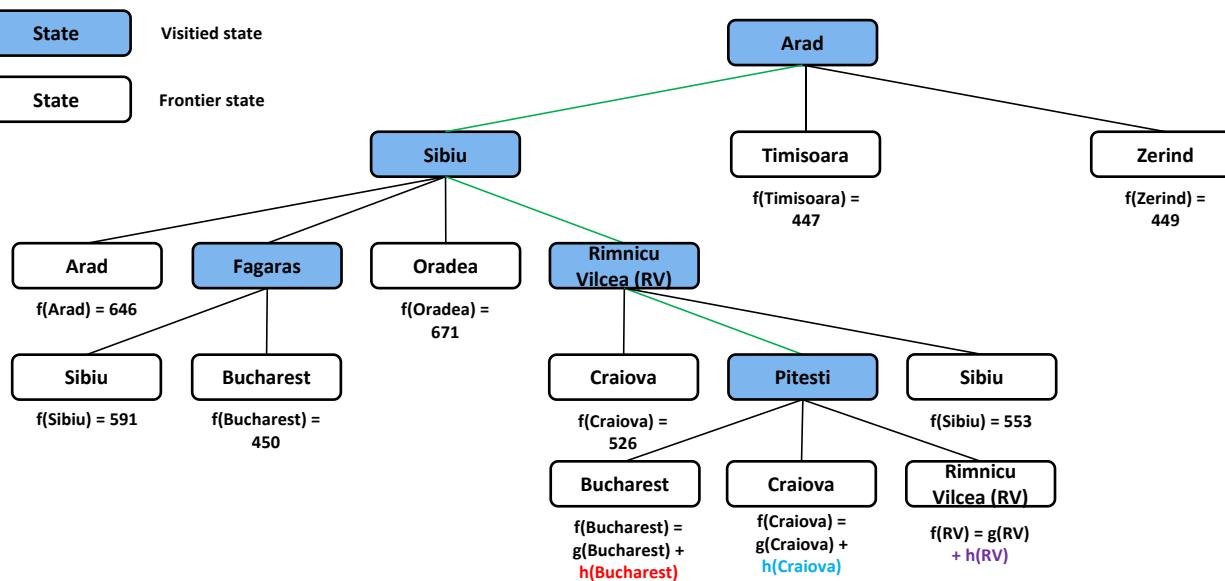
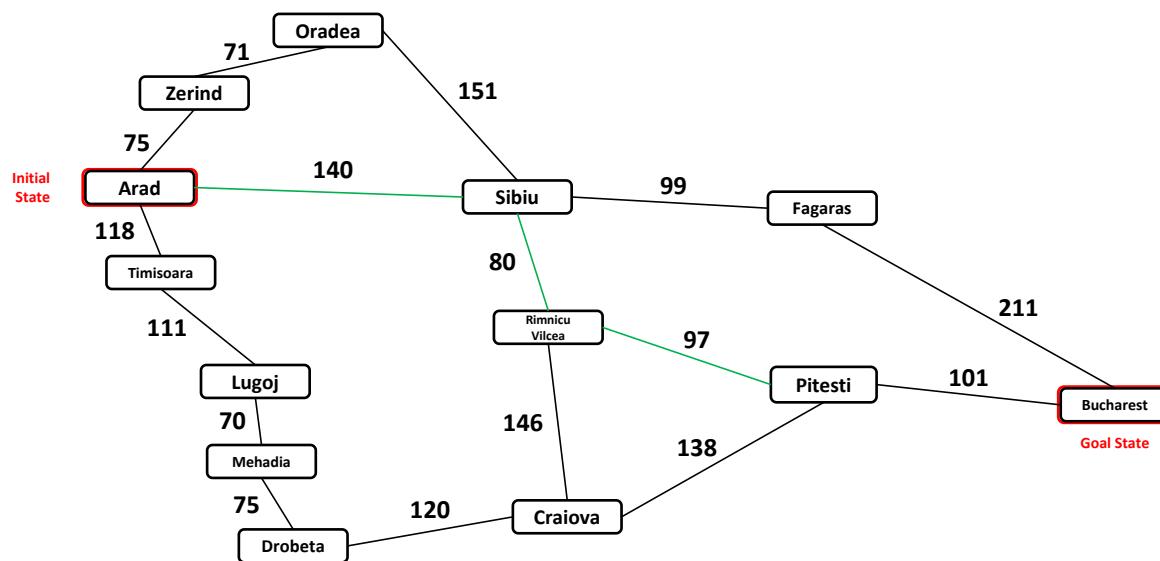
Dracula's Roadtrip: A* Search



Straight-line distance to Bucharest ($h(\text{State})$):

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

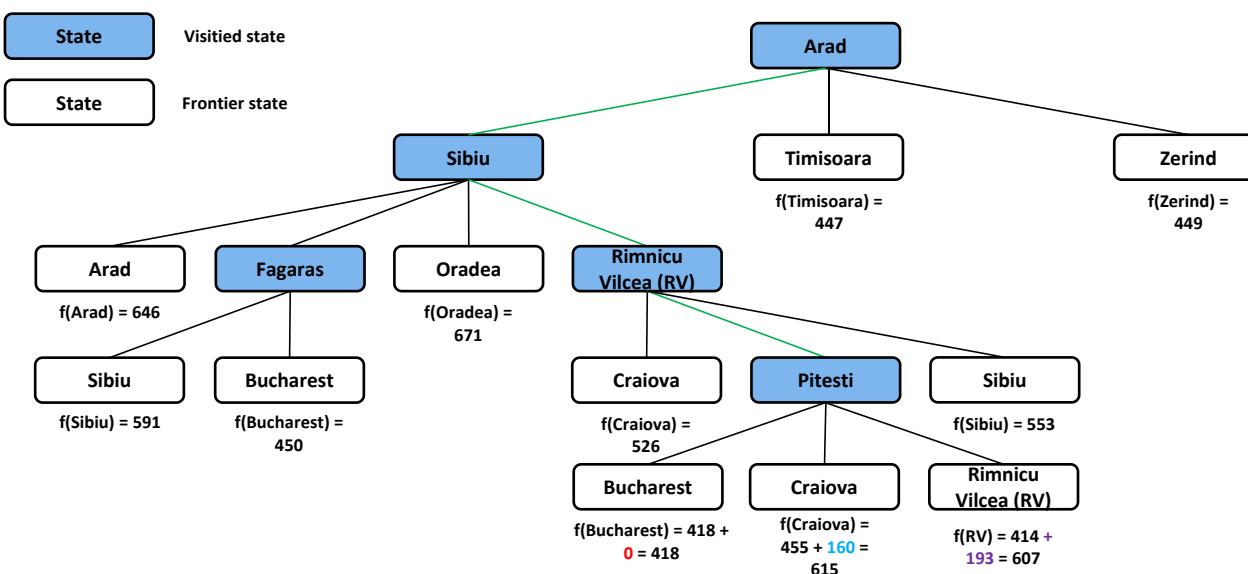
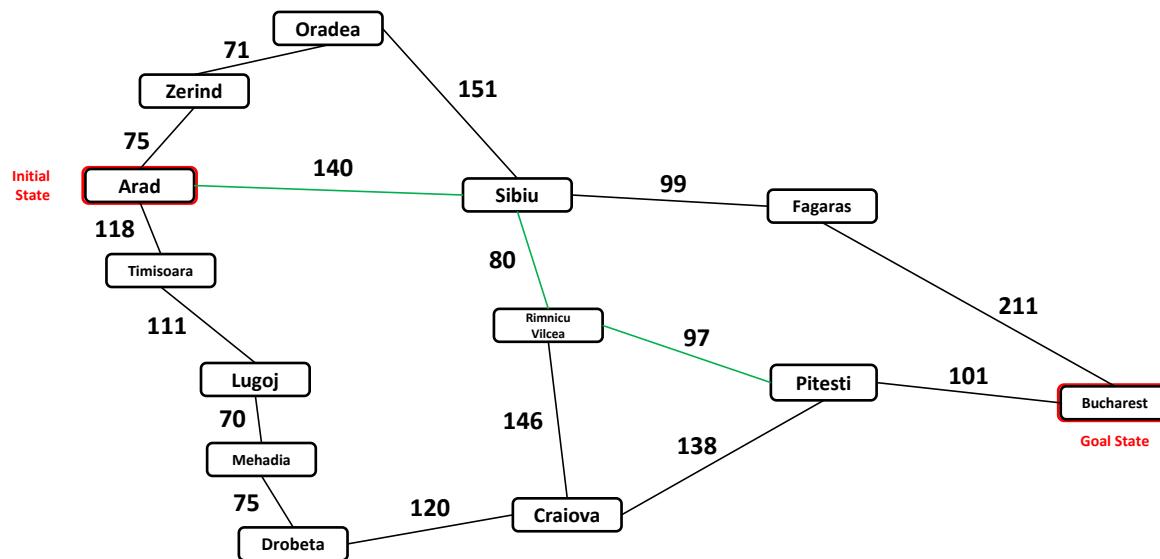
Dracula's Roadtrip: A* Search



Straight-line distance to Bucharest ($h(\text{State})$):

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

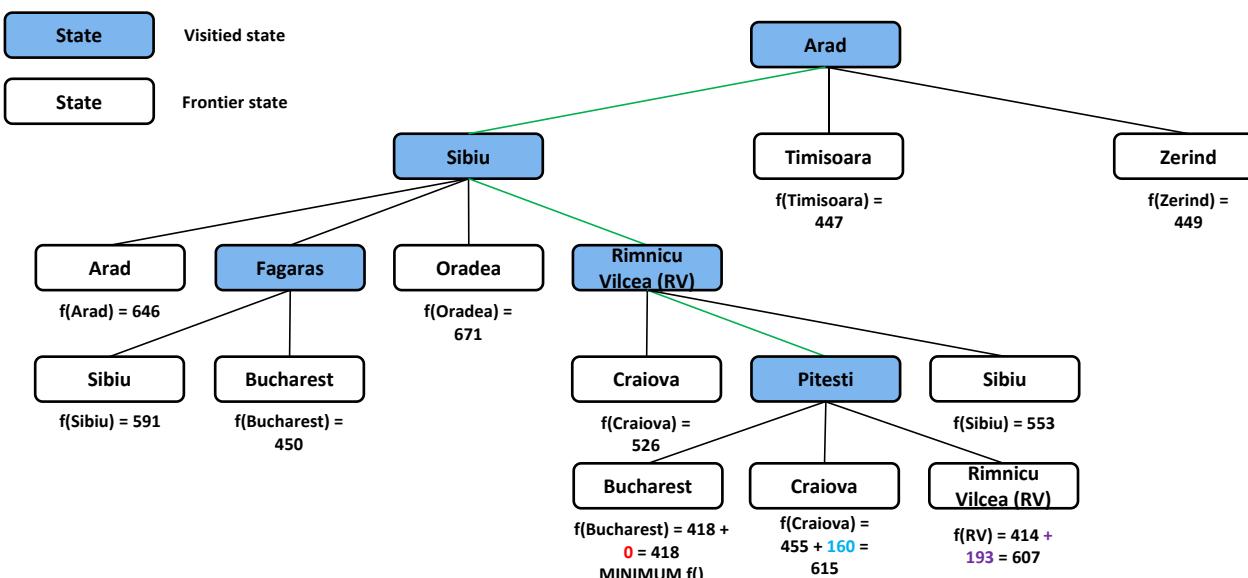
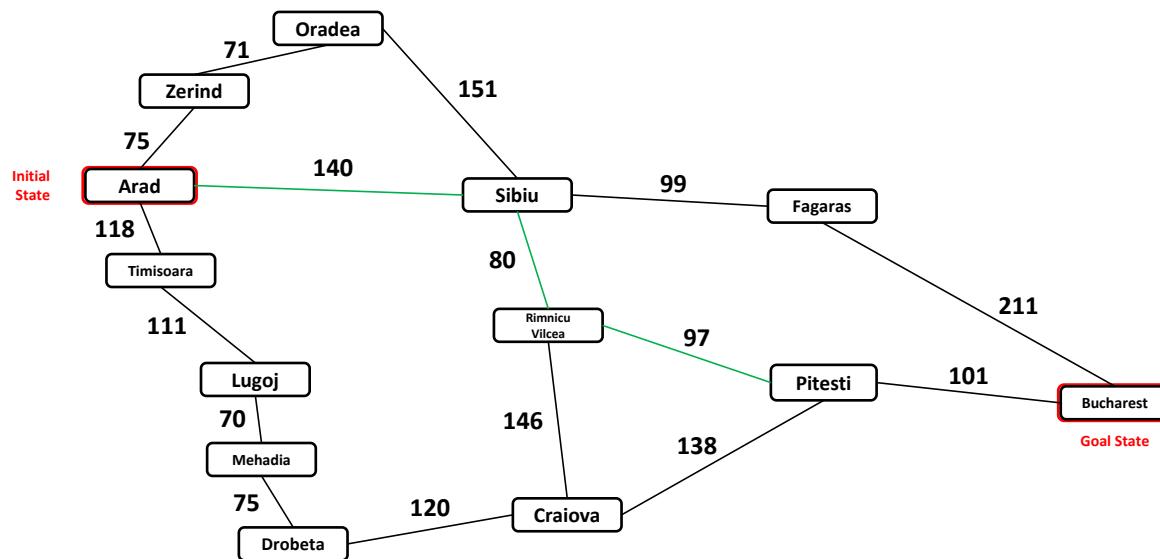
Dracula's Roadtrip: A* Search



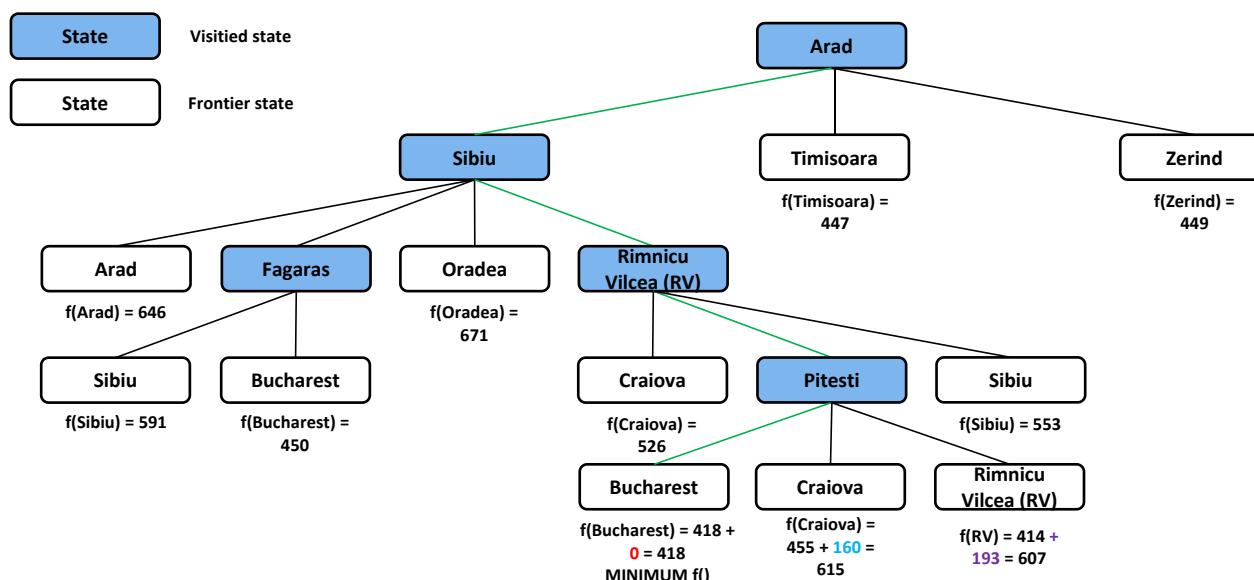
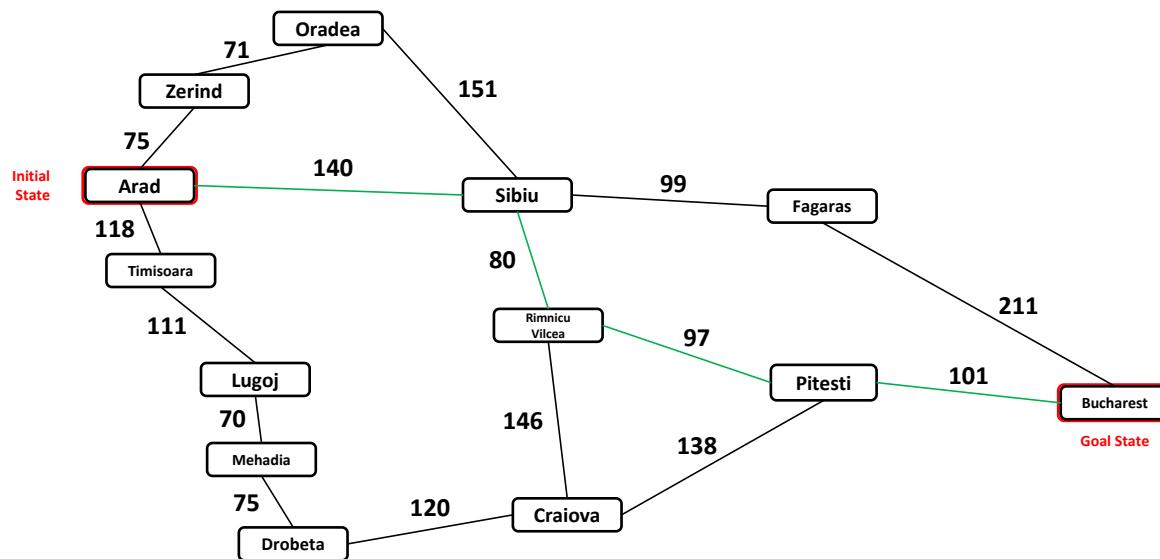
Straight-line distance to Bucharest ($h(\text{State})$):

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

Dracula's Roadtrip: A* Search



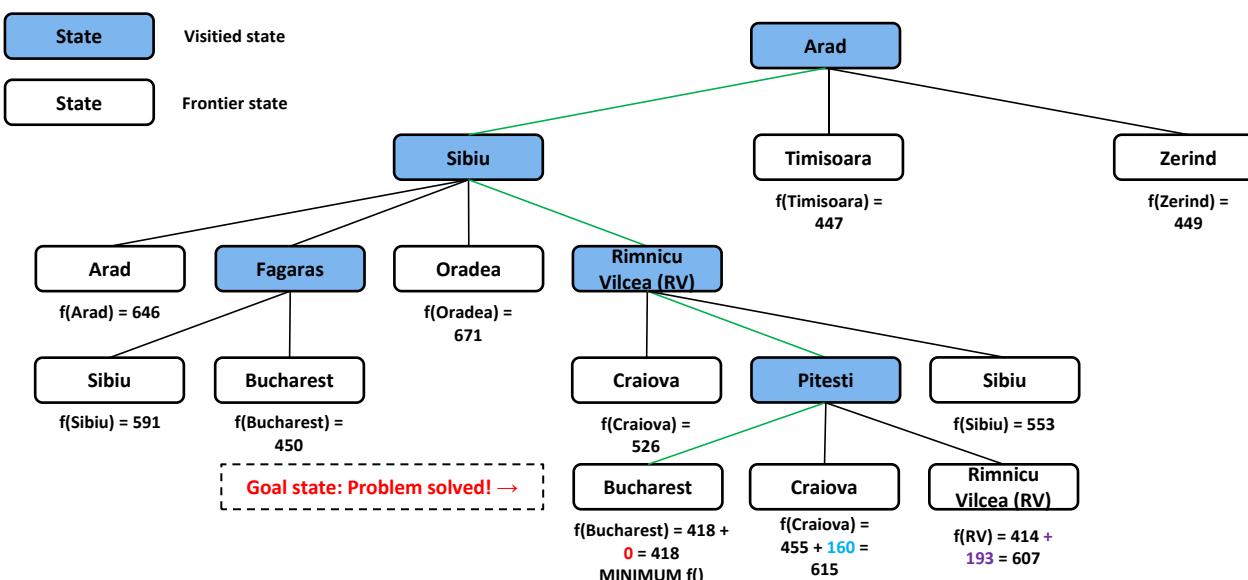
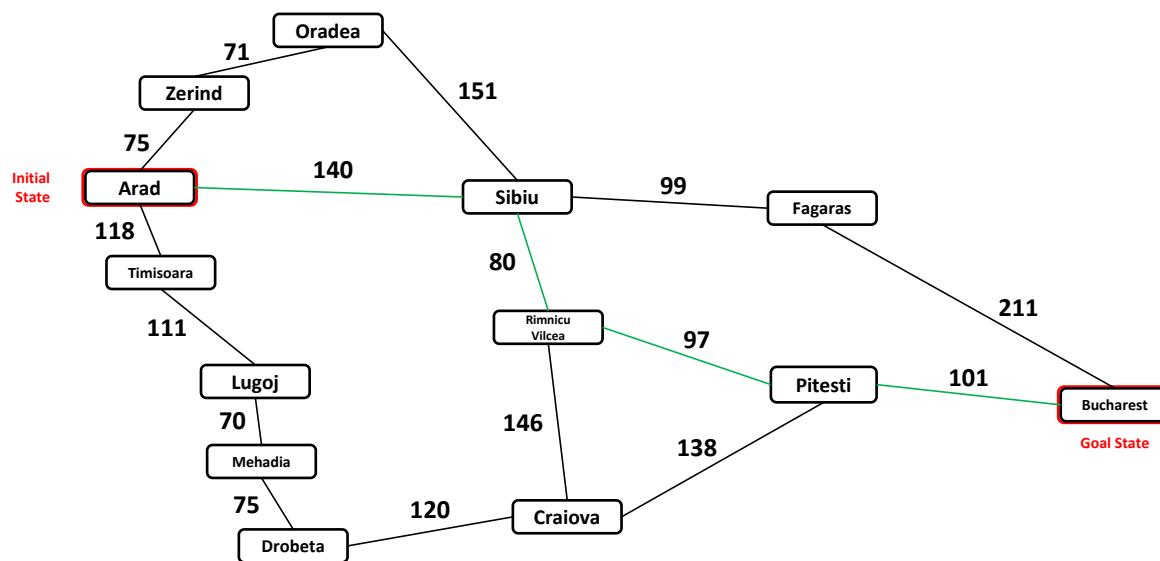
Dracula's Roadtrip: A* Search



Straight-line distance to Bucharest (h(State)):

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

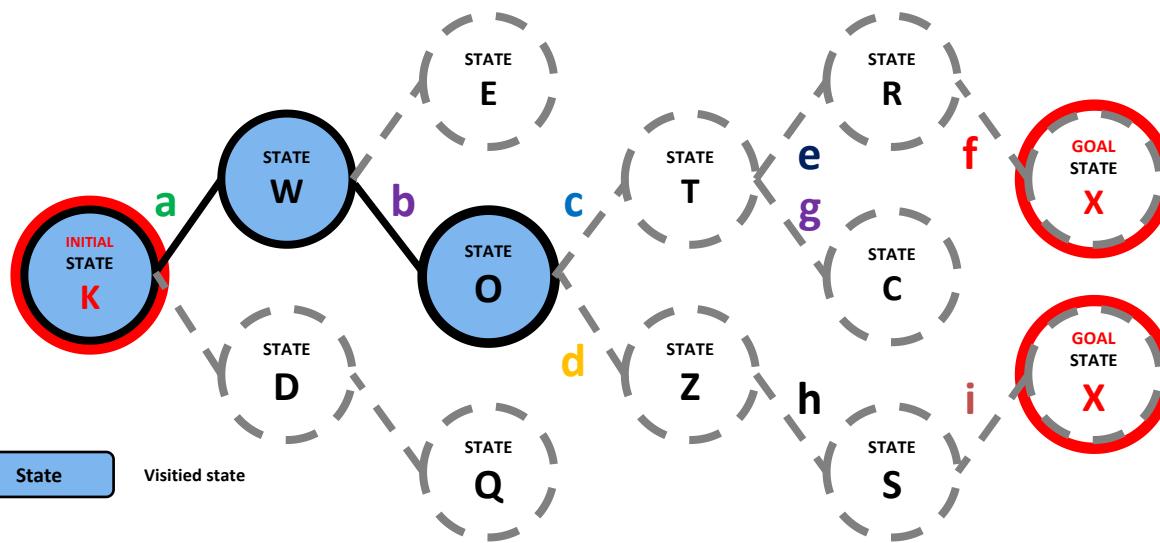
Dracula's Roadtrip: A* Search



Straight-line distance to Bucharest ($h(\text{State})$):

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

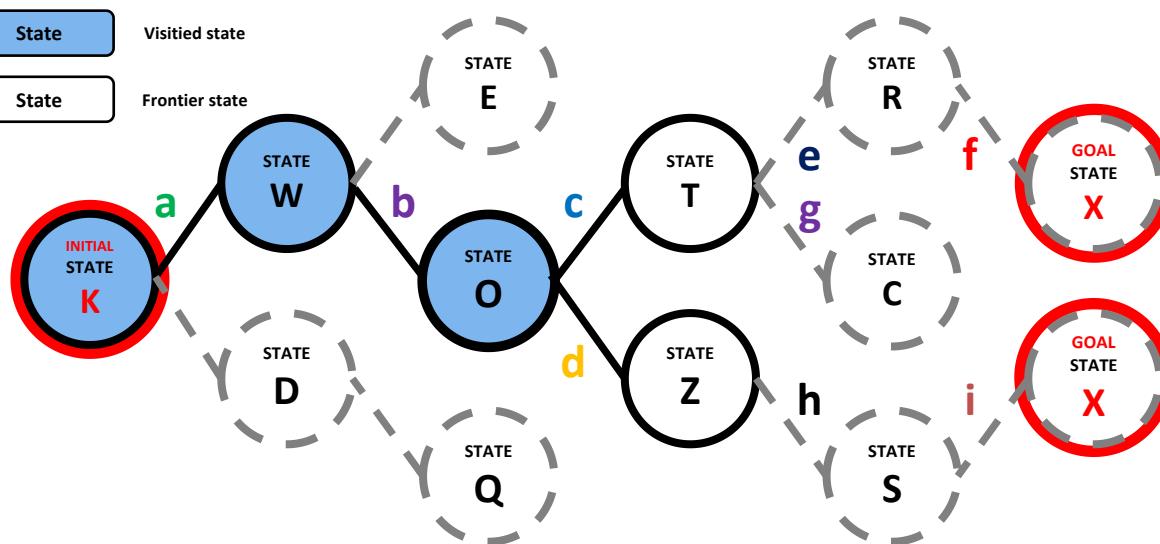
Hill Climbing Search vs. A* Search



Best First Search:
Go to T or Z?
 $f(T) = c$

$$f(Z) = d$$

Pick state with min $f()$



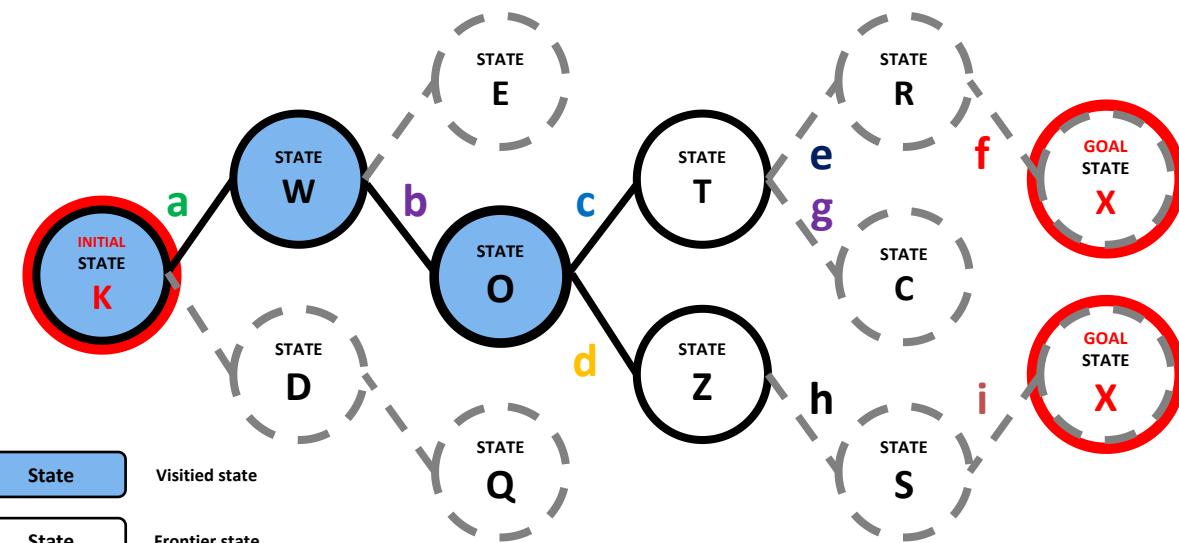
A* Search:
Expand T or Z?
 $f(T) = g(T) + h(T)$
 $f(T) = a + b + c + h(T)$

$$f(Z) = g(Z) + h(Z)$$

$$f(T) = a + b + d + h(Z)$$

Pick state with min $f()$

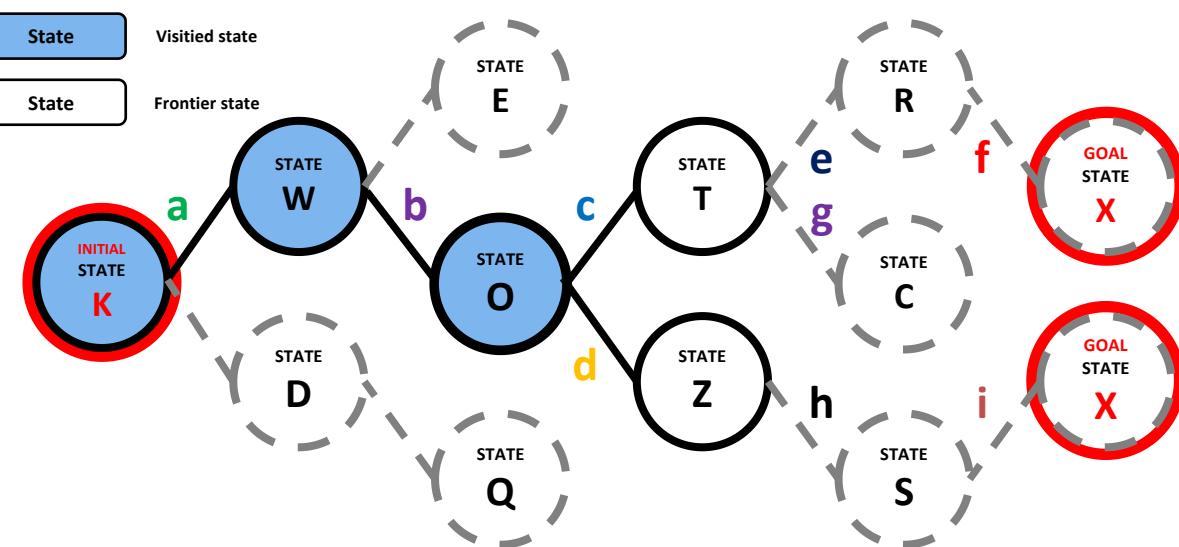
Greedy Best First Search vs. A* Search



Greedy Best First:
Expand T or Z?
 $f(T) = h(T)$

$$f(Z) = h(Z)$$

Pick state with min $f()$



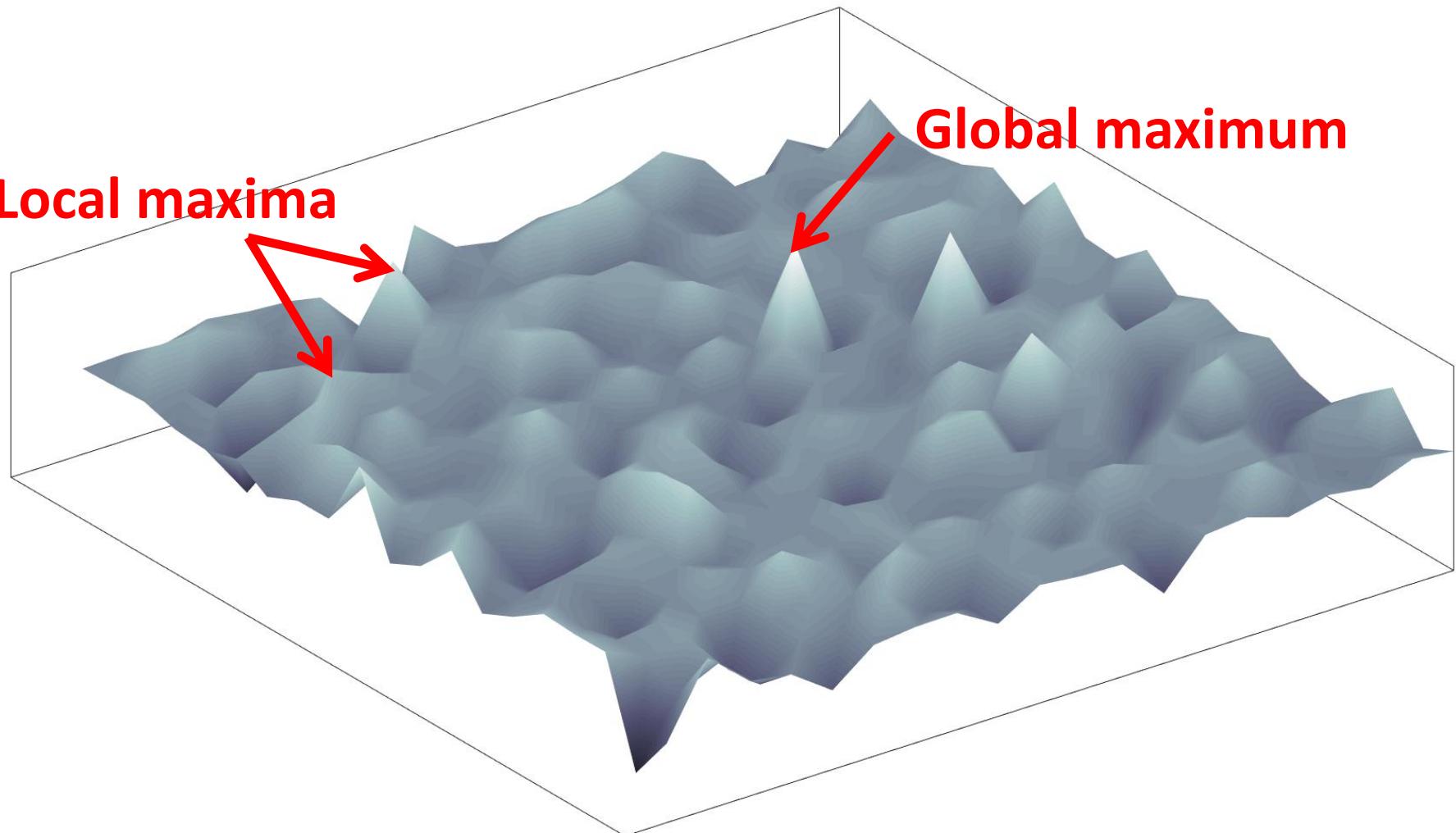
A* Search:
Expand T or Z?
 $f(T) = g(T) + h(T)$
 $f(T) = a + b + c + h(T)$

$$f(Z) = g(Z) + h(Z)$$

$$f(T) = a + b + d + h(Z)$$

Pick state with min $f()$

Complex Environments



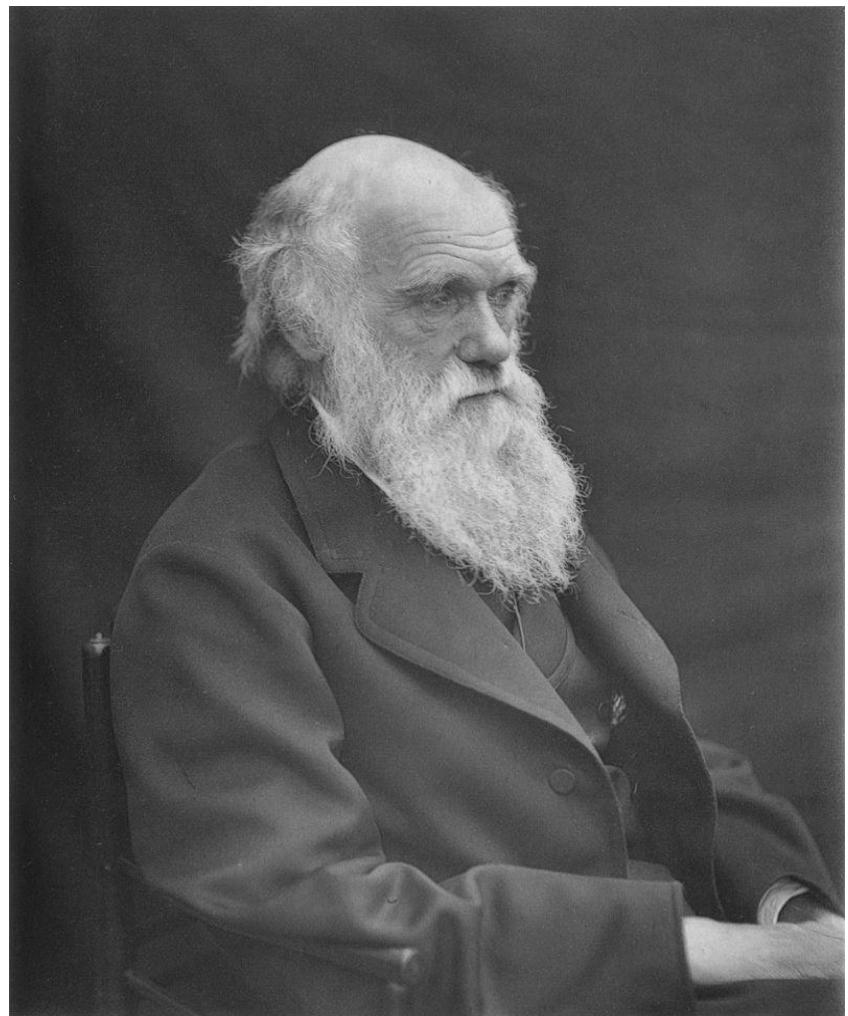
Bonus Material

Chapter 4 - related

(NOT ON EXAMS!)

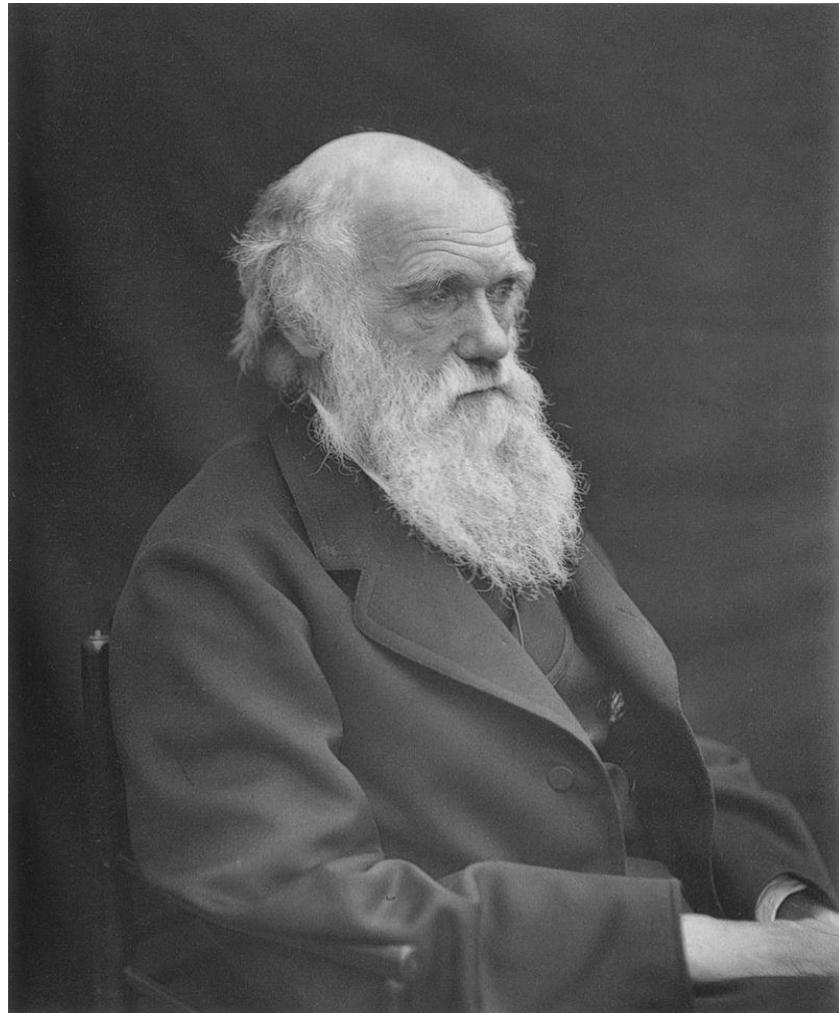
Search in Complex Environments

What's the Connection Here?



Source: <https://wikipedia.org/>

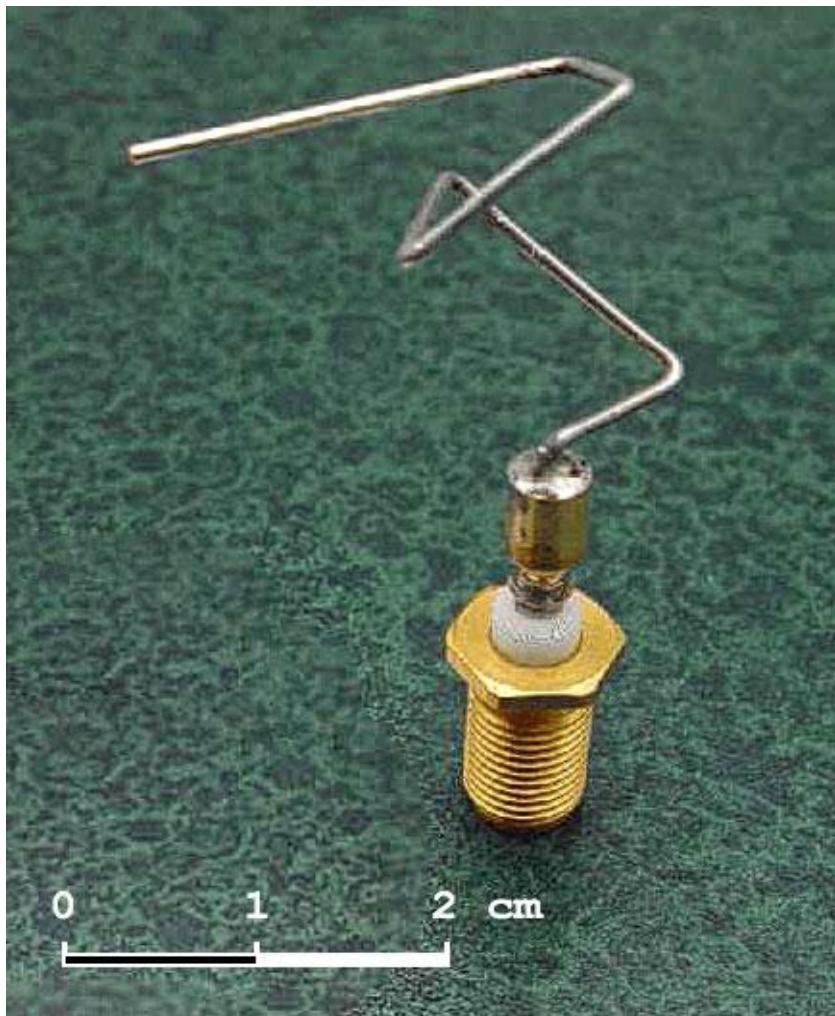
Charles Darwin



Source: <https://wikipedia.org/>

Charles Robert Darwin was an English naturalist, geologist and biologist, best known for his contributions to the science of evolution. His proposition that all species of life have descended over time from common ancestors is now widely accepted, and considered a foundational concept in science.

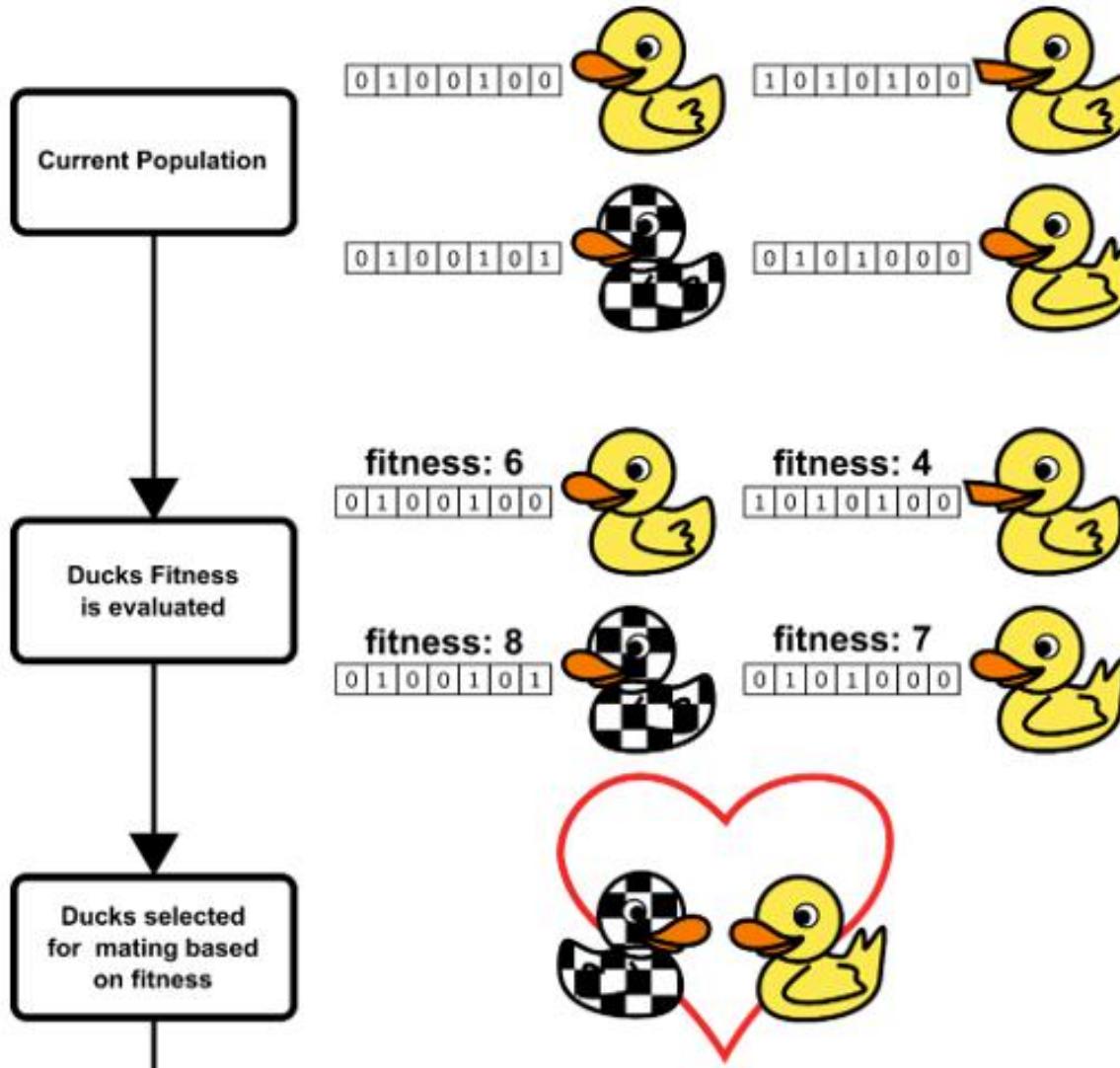
Evolved Antenna



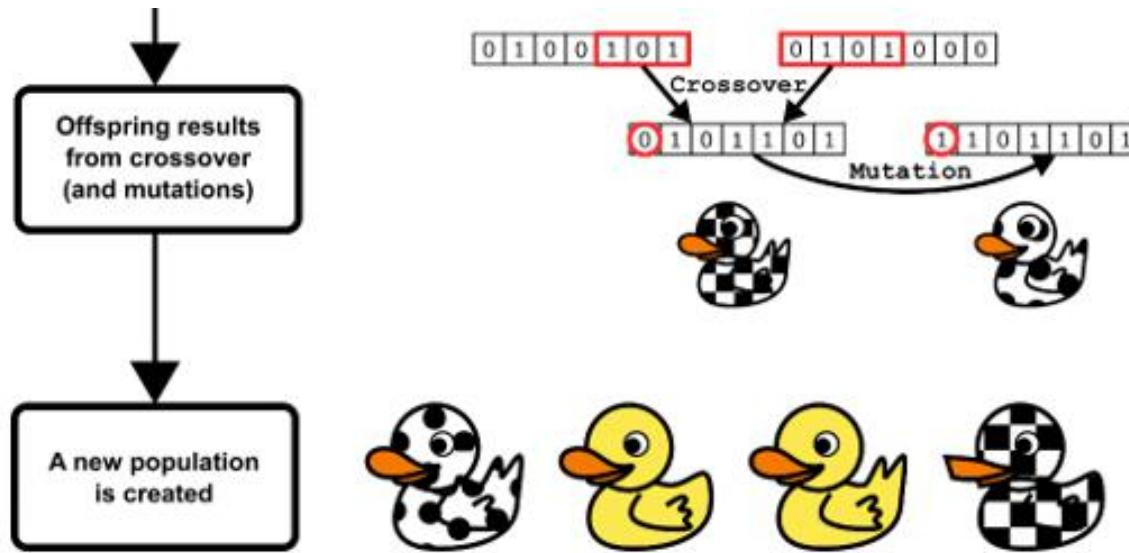
An evolved antenna is an antenna designed fully or substantially by an automatic computer design program that uses an evolutionary algorithm that mimics Darwinian evolution.

Source: <https://wikipedia.org/>

Genetic Algorithm: The Idea



Genetic Algorithm: The Idea



Source: <https://livebook.manning.com/book/algorithms-and-data-structures-in-action/chapter-18/v-14/102>

Genetic Algorithm: Example

Population of points
(solutions)

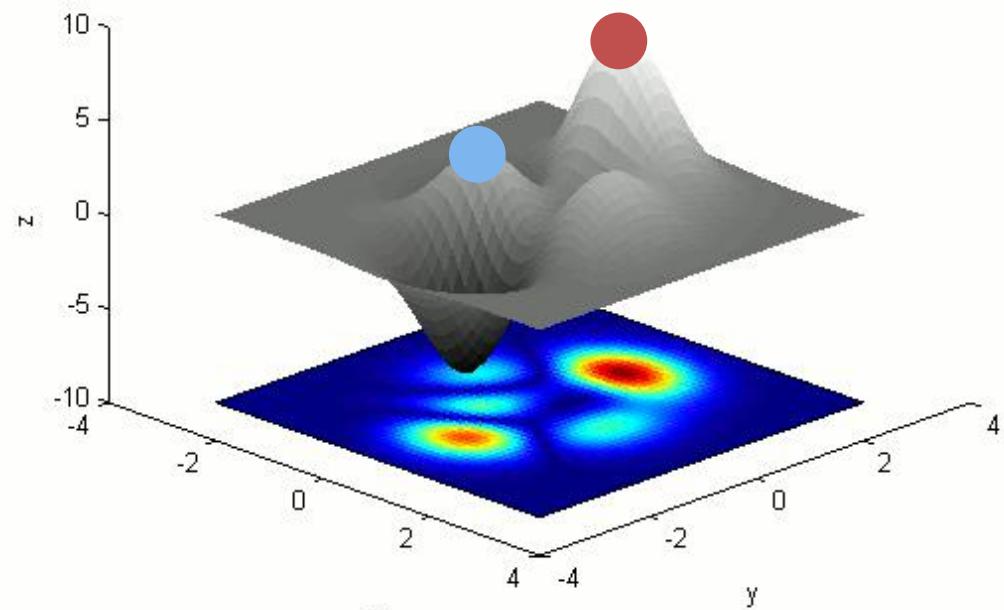
	x		y	
0	1	0	1	0

	x		y	
0	1	1	1	0

	x		y	
0	0	0	1	0

	x		y	
0	1	0	1	0

	x		y	
0	0	1	0	0



$y = f(x,y)$ - fitness function



“Good enough” / local maximum



Best / global maximum

Traveling Salesman Problem



A traveler needs to visit all the cities from a list, where distances between all the cities are known and each city should be visited just once. What is the shortest possible route that he visits each city exactly once and returns to the origin city?

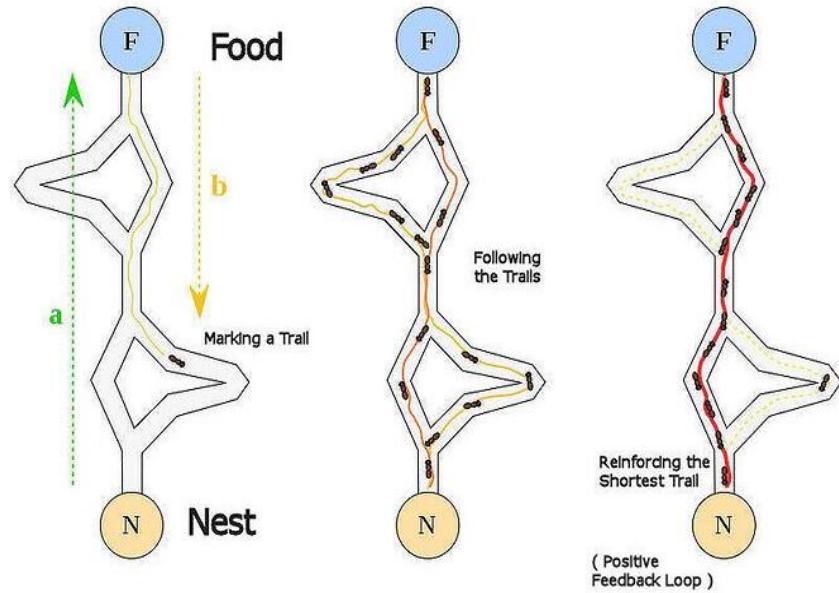
N cities $\rightarrow (N-1)!/2$ paths | 15 cities $\rightarrow 43589145600$ paths

Source: <https://medium.com/ivymobility-developers/traveling-salesman-problem-9ab623c88fab>

Example: Genetic Algorithm

<http://ostap0207.github.io/web-ga-tsp/>

Ant Colony Optimization: The Idea

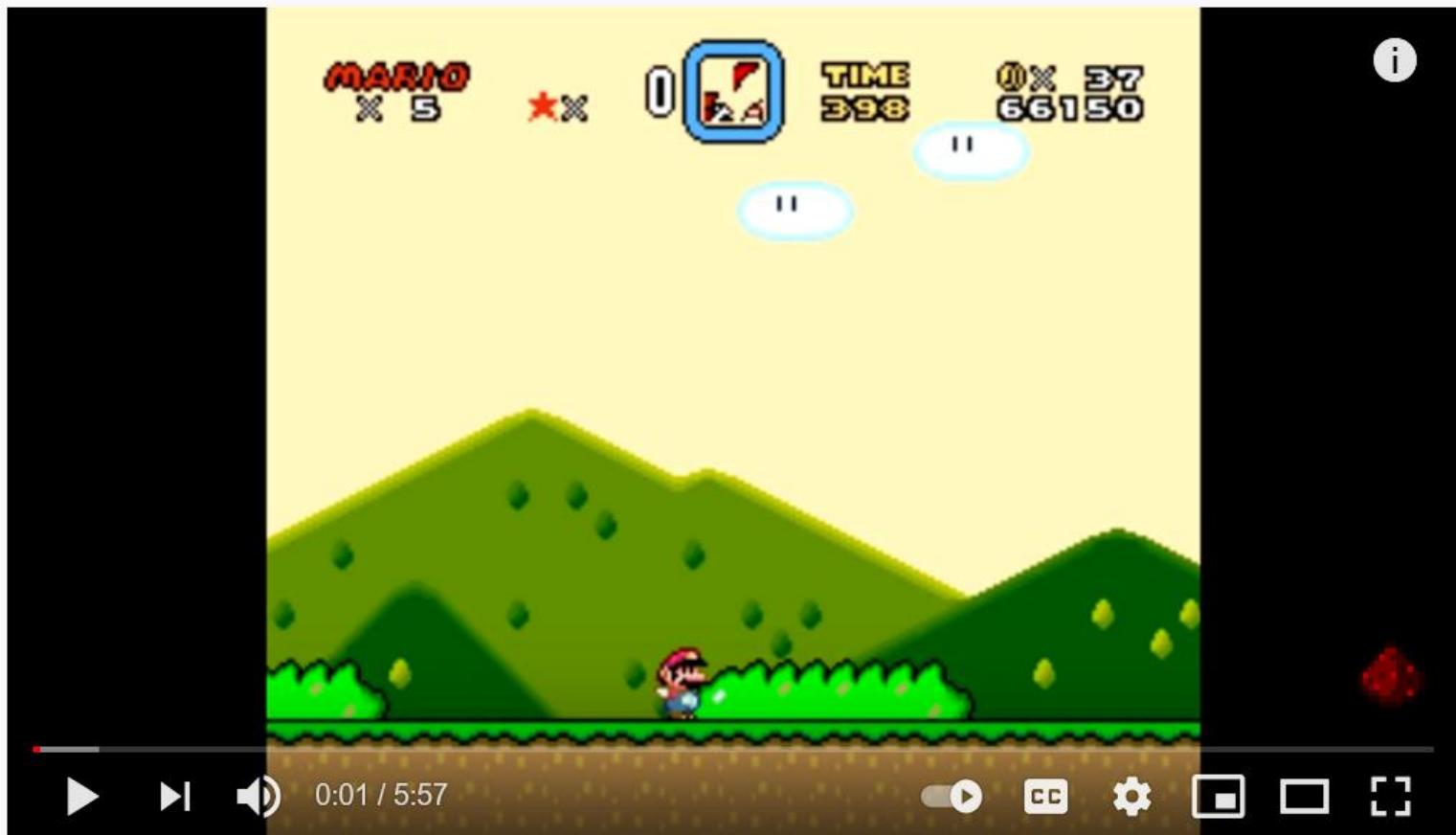


Source: <https://wikipedia.org/>

Example: Ant Colony Optimization

<https://courses.cs.ut.ee/demos/visual-aco/>

Genetic Algorithm in Action



Marl/O - Machine Learning for Video Games

10,471,977 views • Jun 13, 2015

170K

3.4K

SHARE

SAVE

...

Source: <https://www.youtube.com/watch?v=qv6UVOQ0F44>

Bonus DEFINITELY OPTIONAL Material

