Artificial Intelligence Lab
AL-2002

Lab 04

Instructor: Muhammad Saood Sarwar

Semester: Spring 2024

Artificial Intelligence Lab 04

Objective

The objective of this lab is to understand and apply informed search algorithms in problem-solving.

Learning Outcomes

1. Define informed search algorithms and distinguish them from uninformed search algorithms.
2. Explain the advantages and disadvantages of using informed search algorithms.
3. Implement the best-first search algorithm (greedy search) to solve a problem.

Table of Contents

Informed Searches

So far, we have talked about the uninformed search algorithms which looked throughsearch space for all possible solutions of the problem without having any additional knowledge about search space. But informed search algorithm **contains an array ofknowledge** such as how far we are from the goal, path cost, how to reach to goal node, etc. This knowledge helps agents to explore less to the search space and find more efficiently the goal node.

The informed search algorithm is more useful for large search space. Informed search algorithm uses the idea of heuristic, so it is also called Heuristic search.

Heuristics function: Heuristic is a function which is used in Informed Search, and itfinds the most promising path. It takes the current state of the agent as its input and produces the estimation of how close agent is from the goal. The heuristic method, however, might not always give the best solution, but it guaranteed to find a good solution in reasonable time. The heuristic function estimates how close a state is to the goal. It is represented by **h(n)**, and it calculates the **cost of an optimal path betweenthe pair of states**. The value of the heuristic function is always positive.

Admissibility of the heuristic function is given as:

h(n) <= h*(n)

Pure Heuristic Search

Pure heuristic search is the simplest form of heuristic search algorithms. It expands nodes based on their heuristic value h(n). It maintains two lists, OPEN and CLOSEDlist. In the CLOSED list, it places those nodes which have already expanded and in the OPEN list, it places nodes which have yet not been expanded.

On each iteration, each node n with the lowest heuristic value is expanded and generates all its successors and n is placed to the closed list. The algorithm continuesunit a goal state is found.

In the informed search we will discuss two main algorithms which are given below:

- **Best First Search Algorithm(Greedy search)**
- **A* Search Algorithm**

Best-first Search Algorithm (Greedy Search)

Greedy best-first search algorithm always selects the path which appears best at that moment. It is the combination of depth-first search and breadth-first search algorithms. It uses the heuristic function and search. Best-first search allows us to take the advantages of both algorithms. With the help of best-first search, at each step, we can choose the most promising node. In the best first search algorithm, we expand the node which is closest to the goal node and the closest cost is estimated by heuristic function, i.e.

f(n)= g(n)

Where, h(n) = estimated cost from node n to the goal.

The greedy best first algorithm is implemented by the priority queue.

Best first search algorithm:

- Step 1: Place the starting node into the OPEN list.
- Step 2: If the OPEN list is empty, Stop and return failure.
- Step 3: Remove the node n, from the OPEN list which has the lowest value of h(n), and places it in the CLOSED list.
- Step 4: Expand the node n, and generate the successors of node n.
- Step 5: Check each successor of node n, and find whether any node is a goal node or not. If any successor node is goal node, then return success and terminate the search, else proceed to Step 6.
- Step 6: For each successor node, algorithm checks for evaluation function f(n), and then check if the node has been in either OPEN or CLOSED list. If the node has not been in both list, then add it to the OPEN list.
- Step 7: Return to Step 2.

Advantages:

- Best first search can switch between BFS and DFS by gaining the advantages of both the algorithms.
- This algorithm is more efficient than BFS and DFS algorithms.

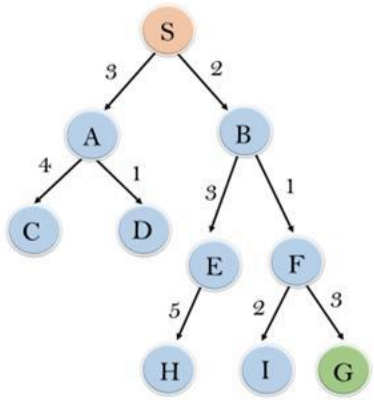
Disadvantages:

- It can behave as an unguided depth-first search in the worst case scenario.

- It can get stuck in a loop as DFS.
- This algorithm is not optimal.

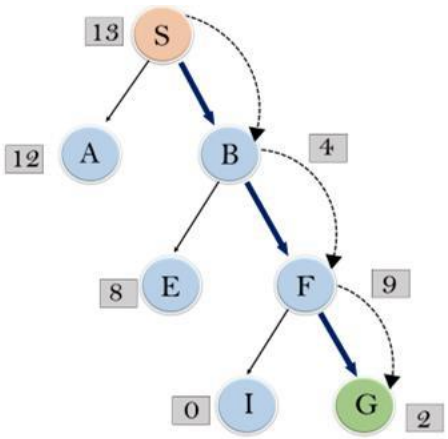
Example:

Consider the below search problem, and we will traverse it using greedy best-first search. At each iteration, each node is expanded using evaluation function f(n)=h(n) , which is given in the below table.



node	H (n)
A	12
B	4
C	7
D	3
E	8
F	2
H	4
I	9
S	13
G	0

In this search example, we are using two lists which are OPEN and CLOSED Lists. Following are the iteration for traversing the above example.



In this search example, we are using two lists which are OPEN and CLOSED Lists. Following are the iteration for traversing the above example.

Expand the nodes of S and put in the CLOSED list.

Initialization: Open [A, B], Closed [S]

Iteration 1: Open [A], Closed [S, B]

Iteration 2: Open [E, F, A], Closed [S, B]

: Open [E, A], Closed [S, B, F]

Iteration 3: Open [I, G, E, A], Closed [S, B, F]

: Open [I, E, A], Closed [S, B, F, G]

Hence the final solution path will be: S----> B----->F----> G

Time Complexity:

The worst-case time complexity of Greedy best first search is O(bm).

Space Complexity:

The worst-case space complexity of Greedy best first search is O(bm). Where, m is the maximum depth of the search space.

Complete:

Greedy best-first search is also incomplete, even if the given state space is finite.

Optimal:

Greedy best first search algorithm is not optimal.

Lab # 07: Informed Searches

In this search example, we are using two lists which are OPEN and CLOSED Lists. Following are the iteration for traversing the above example.

In this search example, we are using two lists which are OPEN and CLOSED Lists. Following are the iteration for traversing the above example.

Expand the nodes of S and put in the CLOSED list

Initialization: Open [A, B], Closed [S]

Iteration 1: Open [A], Closed [S, B]

Iteration 2: Open [E, F, A], Closed [S, B]

: Open [E, A], Closed [S, B, F]

Lab # 07: Informed Searches

Iteration 3: Open [I, G, E, A], Closed [S, B, F]

: Open [I, E, A], Closed [S, B, F, G]

Hence the final solution path will be: S----> B----->F----> G

Time Complexity: The worst-case time complexity of Greedy best first search is O(bm).

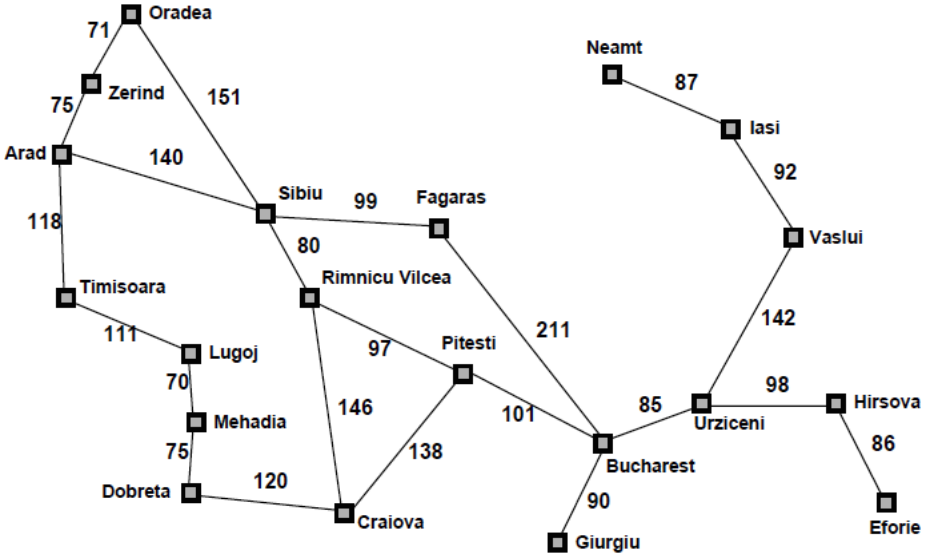
Space Complexity: The worst-case space complexity of Greedy best first search is O(bm). Where, m is the maximum depth of the search space.

Complete: Greedy best-first search is also incomplete, even if the given state space is finite.

Optimal: Greedy best first search algorithm is not optimal.

Informed search vs. Uninformed search

Suppose that you plan to spend vacations in Romania. Following is the map of Romania.



Suppose that you are to travel from Oradea to Neamt and want to determine the shortest path for it. As is seen from the problem, the only information you have is distances between cities and thus, the problem at this point can only be solved by using uninformed search techniques, e.g., by using depth first search and breadth first search. Here is the solution if you applied those search techniques (as already done by you in the previous lab).

```
Breadth First Search algorithm
Oradea Zerind
Oradea Sibiu
Oradea Zerind Arad
Oradea Sibiu Fagaras
Oradea Sibiu Rimnicu Vilcea
Oradea Zerind Arad Timisoara
Oradea Sibiu Fagaras Bucharest
Oradea Sibiu Rimnicu Vilcea Craiova
Oradea Sibiu Rimnicu Vilcea Pitesti
Oradea Zerind Arad Timisoara Lugoj
Oradea Sibiu Fagaras Bucharest Urziceni
Oradea Sibiu Fagaras Bucharest Giurgiu
Oradea Sibiu Rimnicu Vilcea Craiova Dobreta
Oradea Zerind Arad Timisoara Lugoj Mehadia
Oradea Sibiu Fagaras Bucharest Urziceni Hirsova
Oradea Sibiu Fagaras Bucharest Urziceni Vaslui
Oradea Sibiu Fagaras Bucharest Urziceni Hirsova Eforie
Oradea Sibiu Fagaras Bucharest Urziceni Vaslui Iasi
Oradea Sibiu Fagaras Bucharest Urziceni Vaslui Iasi Neamt

Depth First Search algorithm
Oradea Zerind
Oradea Sibiu
Oradea Sibiu Arad
Oradea Sibiu Fagaras
Oradea Sibiu Rimnicu Vilcea
Oradea Sibiu Rimnicu Vilcea Craiova
Oradea Sibiu Rimnicu Vilcea Pitesti
Oradea Sibiu Rimnicu Vilcea Pitesti Bucharest
Oradea Sibiu Rimnicu Vilcea Pitesti Bucharest Urziceni
Oradea Sibiu Rimnicu Vilcea Pitesti Bucharest Giurgiu
Oradea Sibiu Rimnicu Vilcea Pitesti Bucharest Urziceni Hirsova
Oradea Sibiu Rimnicu Vilcea Pitesti Bucharest Urziceni Vaslui
Oradea Sibiu Rimnicu Vilcea Pitesti Bucharest Urziceni Vaslui Iasi
Oradea Sibiu Rimnicu Vilcea Pitesti Bucharest Urziceni Vaslui Iasi Neamt

Shortest path
The shortest path from Oradea to Neamt is 835 miles and goes as follows:

Oradea to Sibiu
Sibiu to Rimnicu Vilcea
Rimnicu Vilcea to Pitesti
Pitesti to Bucharest
Bucharest to Urziceni
Urziceni to Vaslui
Vaslui to Iasi
Iasi to Neamt

_
```

Now let's suppose that you are to travel to Bucharest starting from Arad. But as compared to previous example, in addition to the map and their mutual distances between cities, you also are given the following table listing the Euclidean distance (heuristic) between a given city and Bucharest.

City	Heuristic value	City	Heuristic value
Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Eforie	161	Pitesti	100
Fagaras	176	Rimnicu Vilcea	193

Dobreta	242	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374

How will this change the problem? For one, now this becomes an informed search problem because now from each city, we have a heuristic value (Euclidean distance) which indicates if we are going closer to our goal with each move or moving away from it, unlike Example 1, where we were searching in complete darkness (uninformed) in comparison. Hence the problem has now shifted from being uninformed to informed.