Assignment 1 - A* Search DA221

Author: Aryan Gupta
Roll Number: 230150003

Introduction

A* search finds the lowest-cost path in a weighted graph using

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

where:

- $g(n)$: Cost from the start to node n.
- $h(n)$: Estimated cost from n to the goal.

Pseudocode:

```
function A*(start, goal):
    open_set ← priority queue containing start
    g_score[start] ← 0
    f_score[start] ← h(start)

    while open_set is not empty:
        current ← node in open_set with lowest f_score
        if current == goal:
            return reconstruct_path(came_from, current)

        remove current from open_set

        for each neighbor of current:
            tentative_g_score = g_score[current] + cost(current, neighbor)
            if tentative_g_score < g_score[neighbor]:
                came_from[neighbor] ← current
                g_score[neighbor] ← tentative_g_score
                f_score[neighbor] ← tentative_g_score + h(neighbor)
                if neighbor is not in open_set:
                    add neighbor to open_set

    return failure
```

Heuristic Function

The heuristic $h(n)$ estimates the remaining cost to the goal. Here, it is the straight-line distance from each city to Bucharest.

Purpose & Importance:

- **Guidance:** Directs the search toward the goal.
- **Efficiency:** Reduces the number of nodes expanded.
- **Optimality:** Its admissible and consistent nature guarantees that A* finds the optimal path.

Implemented Graph

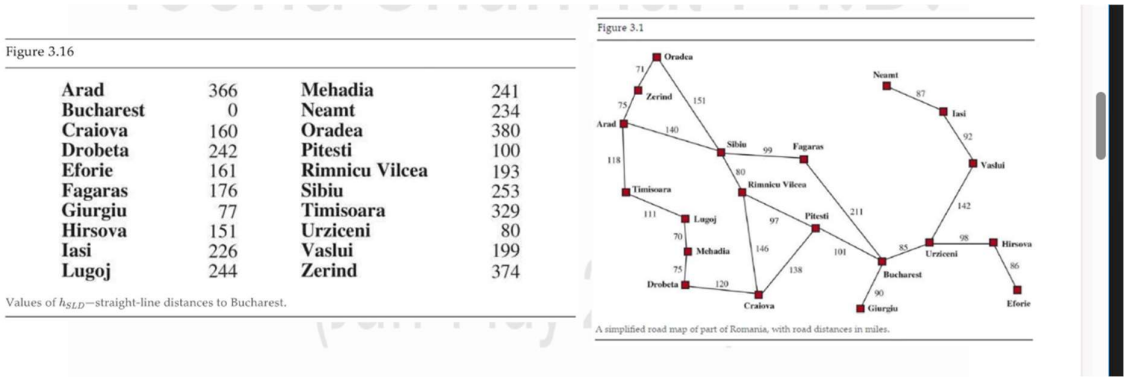

Figure 3.16

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

Values of h_{SLD}—straight-line distances to Bucharest.

Credit- Russell, S. & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach* (4th ed.)

Graph Description:

- **Example:** A simplified Romania road map with 20 nodes
- **Nodes:** Romanian cities (e.g., Arad, Sibiu, Bucharest).
- **Edges:** Roads with distances (miles).

Heuristic:

- Uses straight-line distances to Bucharest.
- **Optimality:** Since the heuristic never overestimates and respects the triangle inequality, A* efficiently finds the shortest route.

Path Obtained

Start: Arad to **Goal:** Bucharest

- **Optimal Path:** Arad → Sibiu → Rimnicu Vilcea → Pitesti → Bucharest (418 miles)

Conclusion:

The A* algorithm, leveraging an admissible and consistent heuristic, efficiently computes the optimal route in our Romania map. This balance of actual cost and heuristic estimate minimizes search efforts, confirming the method's optimality for navigation problems.

References:

- Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). *A Formal Basis for the Heuristic Determination of Minimum Cost Paths*. IEEE Trans. on Systems Science and Cybernetics.