Department of Computer Science & Engineering

Course Title : Artificial Intelligence and Expert Systems Lab

Course Code : CSE 404

Lab Report : 02

Submission Date : 13.04.25

Submitted To:

Noor Mairukh Khan Arnob

Lecturer,

Department of CSE, UAP

Submitted By:

Khan Jubayer

Reg: 21201090

Sec: B2

Problem Title:

Finding the Optimal Path from Mugda Medical College to UAP Using A* Search Algorithm

Problem Description:

The objective of this problem is to determine the optimal path from Mugda Medical College (home) to UAP (University of Asia Pacific) using the A* search algorithm.

A Search Algorithm*

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

Where:
f(n) = Evaluation function
g(n) = Actual cost from the start node to the current node
h(n) = Heuristic estimated cost from the current node to the goal node

Tools and Languages Used:

- Programming Language: Python
- Tools: Colab Notebook

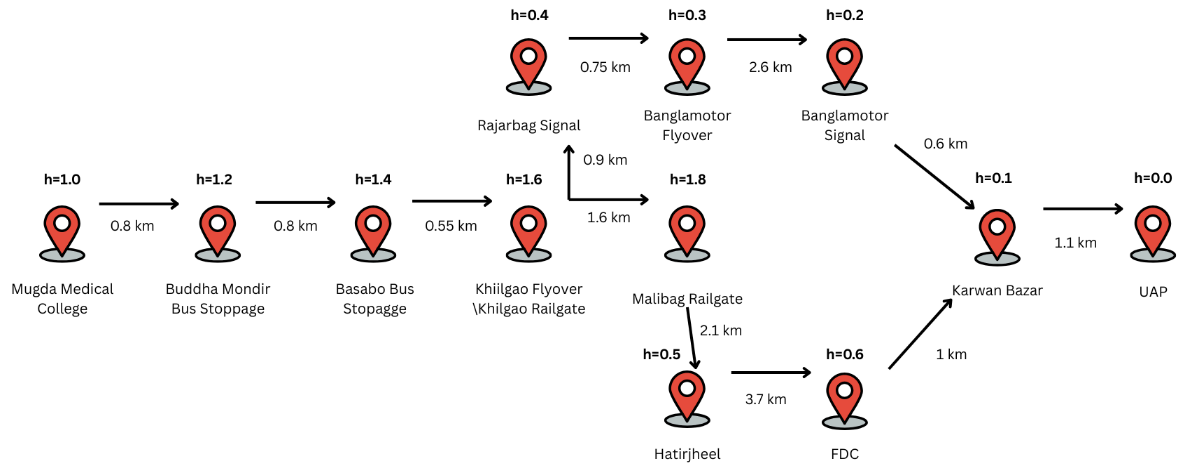
Diagram:

Designed Graph:

- Edge labels represent g(n) values (actual distances in km from Google Maps)
- Node labels include h(n) values (heuristic cost using Manhattan Distance)

Diagram:

Designed Graph



Here,

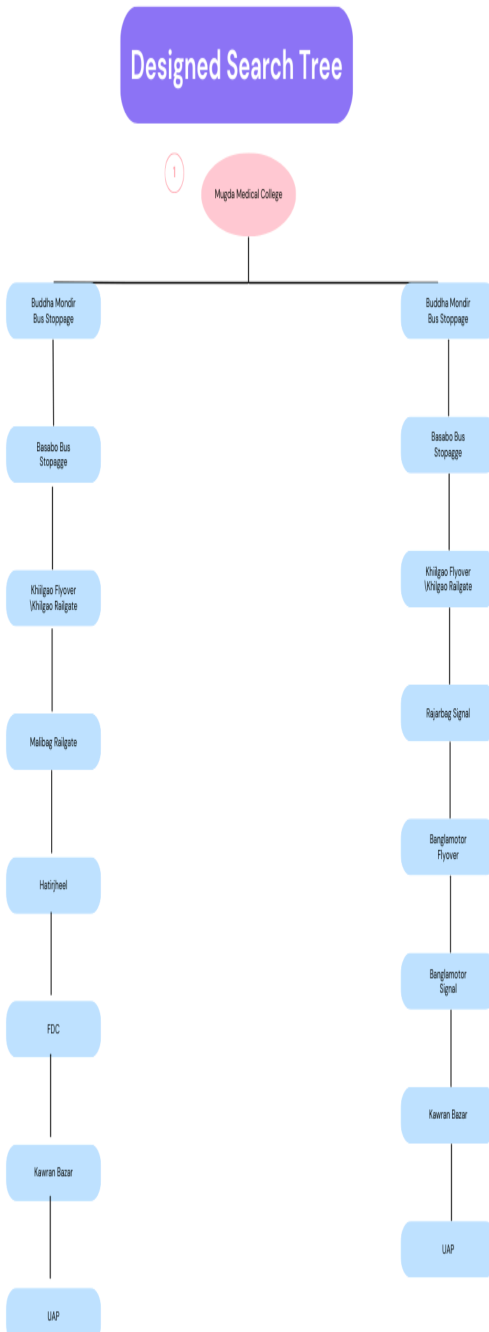
Start Node : Mugda Medical College (Home)

Goal Node: UAP (University of Asia Pacific)

g(n): Calculated in kilometers (km) from Google Maps

h(n): Calculated from Google Maps (longitude, latitude) using Manhattan Distance (longitude - latitude).

Designed Search Tree



Sample Input/Output:

```
# Install required libraries
!pip install networkx matplotlib

import networkx as nx
import matplotlib.pyplot as plt
from queue import PriorityQueue

# Create directed graph
G = nx.DiGraph()

# Heuristics (h values)
heuristics = {
    'Mugda Medical College': 1.0,
    'Buddha Mondir Bus Stoppage': 1.2,
    'Basabo Bus Stoppage': 1.4,
    'Khilgao Flyover': 1.6,
    'Rajarbag Signal': 0.4,
    'Banglamotor Flyover': 0.3,
    'Banglamotor Signal': 0.2,
    'Karwan Bazar': 0.1,
    'UAP': 0.0
}

# Edges (u, v, distance)
edges = [
    ('Mugda Medical College', 'Buddha Mondir Bus Stoppage', 0.8),
    ('Buddha Mondir Bus Stoppage', 'Basabo Bus Stoppage', 0.8),
    ('Basabo Bus Stoppage', 'Khilgao Flyover', 0.55),
    ('Khilgao Flyover', 'Rajarbag Signal', 0.9),
    ('Rajarbag Signal', 'Banglamotor Flyover', 0.75),
    ('Banglamotor Flyover', 'Banglamotor Signal', 2.6),
    ('Banglamotor Signal', 'Karwan Bazar', 0.6),
    ('Karwan Bazar', 'UAP', 1.1)
]

# Add nodes and edges to the graph
for node in heuristics:
    G.add_node(node, h=heuristics[node])

for u, v, w in edges:
    G.add_edge(u, v, weight=w)
```

```python
# A* Search Algorithm
def a_star_search(graph, start, goal):
    pq = PriorityQueue()
    pq.put((0 + heuristics[start], 0, start, [start]))  # (f(n), g(n), node, path)

    visited = set()

    while not pq.empty():
        f, g, current, path = pq.get()

        if current in visited:
            continue
        visited.add(current)

        if current == goal:
            return path, g

        for neighbor in graph.neighbors(current):
            if neighbor not in visited:
                cost = graph[current][neighbor]['weight']
                new_g = g + cost
                h = heuristics[neighbor]
                new_f = new_g + h
                pq.put((new_f, new_g, neighbor, path + [neighbor]))

    return None, float('inf')

# Run A* to find optimal path
start_node = 'Mugda Medical College'
end_node = 'UAP'
path, total_cost = a_star_search(G, start_node, end_node)

# Display results
print("Optimal Path:", " → ".join(path))
print("Total Cost:", round(total_cost, 2), "km")
```

Output:

```
Optimal Path: Mugda Medical College → Buddha Mondir Bus Stoppage → Basabo Bus
Stoppage → Khilgao Flyover → Rajarbag Signal → Banglamotor Flyover →
Banglamotor Signal → Karwan Bazar → UAP
Total Cost: 8.1 km
```

Conclusion:

By implementing the A* search algorithm, we successfully determined the most optimal path from Mugda Medical College to UAP while minimizing travel distance. The algorithm efficiently balances actual travel cost (g(n)) and estimated distance (h(n)), ensuring the shortest possible route.