A* PATHFINDING ALGORITHM

DHANMONDI to UNIVERSITY OF ASIA PACIFIC

Optimal Path finding using a Algorithm with Real Map Data

COURSE TITLE:

Artificial Intellegence & Expert System Lab (CSE 404)

Presented By:

Anika Nawer Nabila
Reg No:22101152
Abdullah-Al Mamun
Reg No:22101158

Presented To:

Bidita Sarkar Diba
Lecturer
University of Asia
Pacific

Date: 26 August,2024

Problem Description

The problem involves finding the shortest path from Dhanmondi to the University of Asia Pacific (UAP) in Dhaka, Bangladesh using the A* search algorithm.

The challenge requires: Creating a graph representation of real-world locations with accurate distance measurements Implementing the A* algorithm with appropriate heuristic functions

Goal

Find the shortest path from
Dhanmondi (Home) → UAP.

Algorithm Used: A* Search Algorithm

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

g(n) = cost so far
h(n) = estimated cost (Euclidean Distance)

Real-life Applications:

Google Maps, GPS, Robotics, Network Routing, Traffic Management

Tools & Language used

Programming Language:Python

Libraries:

Matplotlib for visualization

Network for graph operations

Math for geographical calculations

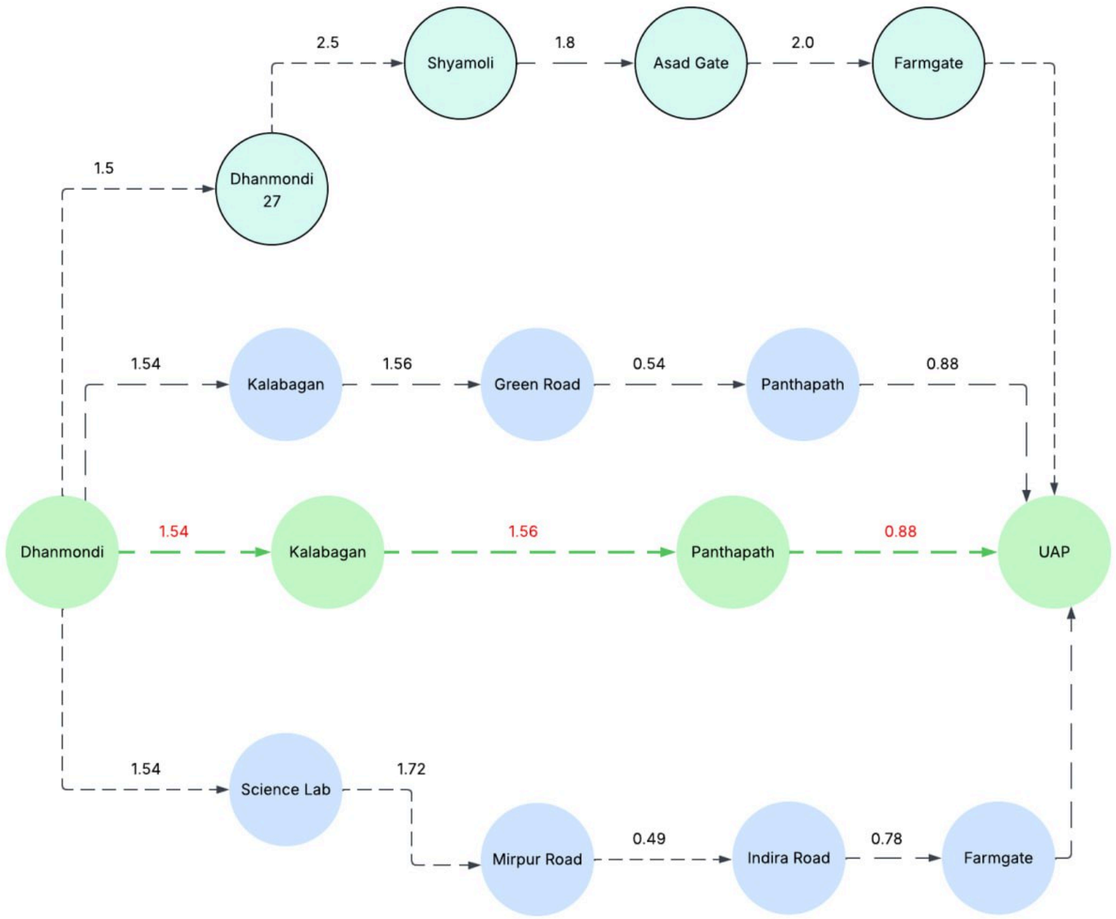
Data Sources:

Google Maps for coordinate and distance information

Diagram

The diagram shows the graph network with nodes representing locations and edges showing roads with distances.

The optimal path is highlighted in **green**, while alternative routes are shown in **gray**.



The **optimal path** found was:

Dhanmondi → Kalabagan → Panthapath → Farmgate → UAP with a total distance of **3.0 km**.

When compared with alternative routes:

1. Route via **Dhanmondi → Kalabagan → Green Road → Panthapath → UAP** was slightly longer (**4.7km**)
2. Route via **Dhanmondi → Science Lab →Mirpur Road → Indira Road → Farmgate →UAP** was much longer (**5.4 km**)

Sample Input/Output

Start Node: Dhanmondi

Goal Node:UAP

Graph with real-world distances between intermediate nodes

Result

Optimal Path: Dhanmondi → Kalabagan → Panthapath →UAP

Distance: 3.0 km

Alternatives: 4.7 km, 5.4 km

CODE

A* Search

```python
import matplotlib.pyplot as plt
import networkx as nx
import math
import heapq

# Heuristic function (Euclidean distance)
def heuristic(node1, node2):
    x1, y1 = nodes[node1]
    x2, y2 = nodes[node2]
    return math.sqrt((x2 - x1)**2 + (y2 - y1)**2) * 100  # Scaled for visualization

# A* algorithm implementation
def a_star(graph, start, goal):
    open_set = []
    heapq.heappush(open_set, (0, start))

    came_from = {}

    g_score = {node: float('inf') for node in graph.nodes()}
    g_score[start] = 0

    f_score = {node: float('inf') for node in graph.nodes()}
    f_score[start] = heuristic(start, goal)

    open_set_hash = {start}

    while open_set:
        current = heapq.heappop(open_set)[1]
        open_set_hash.remove(current)

        if current == goal:
            path = []
            while current in came_from:
                path.append(current)
                current = came_from[current]
            path.append(start)
            path.reverse()
            return path, g_score, f_score

        for neighbor in graph.neighbors(current):
            temp_g_score = g_score[current] + graph[current][neighbor]['weight']

            if temp_g_score < g_score[neighbor]:
                came_from[neighbor] = current
                g_score[neighbor] = temp_g_score
                f_score[neighbor] = temp_g_score + heuristic(neighbor, goal)
                if neighbor not in open_set_hash:
                    heapq.heappush(open_set, (f_score[neighbor], neighbor))
                    open_set_hash.add(neighbor)

    return None, g_score, f_score
```

```python
# Graph setup
G = nx.DiGraph()
nodes = {
    'Dhanmondi': (23.7465, 90.3760),
    'Kalabagan': (23.7504, 90.3742),
    'Panthapath': (23.7543, 90.3804),
    'Green Road': (23.7592, 90.3865),
    'Farmgate': (23.7553, 90.3899),
    'UAP': (23.7545, 90.3892),
    'Dhanmondi 27': (23.7432, 90.3728),
    'Indira Road': (23.7512, 90.3821),
    'Mirpur Road': (23.7531, 90.3845),
    'Science Lab': (23.7378, 90.3852),
    'Shyamoli': (23.7701, 90.3705),
    'Asad Gate': (23.7582, 90.3801)
}
edges = [
    ('Dhanmondi', 'Kalabagan', 1.0),
    ('Kalabagan', 'Panthapath', 1.0),
    ('Panthapath', 'UAP', 1.0),
    ('Kalabagan', 'Green Road', 1.5),
    ('Green Road', 'Panthapath', 1.2),
    ('Green Road', 'Farmgate', 2.0),
    ('Farmgate', 'UAP', 1.0),
    ('Dhanmondi', 'Science Lab', 1.4),
    ('Science Lab', 'Mirpur Road', 1.0),
    ('Mirpur Road', 'Indira Road', 0.9),
    ('Indira Road', 'Farmgate', 1.1),
    ('Farmgate', 'UAP', 1.0),
]
for n, pos in nodes.items():
    G.add_node(n, pos=pos)
for u, v, w in edges:
    G.add_edge(u, v, weight=w)

# Run A* algorithm
start = 'Dhanmondi'
goal = 'UAP'
optimal_path, g_scores, f_scores = a_star(G, start, goal)
```

```python
# Print results for optimal path
print("="*100)
print(" A* PATHFINDING FROM DHANMONDI TO UAP ".center(100, "="))
print("="*100)
print(f"Optimal Path: {' → '.join(optimal_path)}")
print(f"Total Distance: {g_scores[goal]:.1f} km\n")
print("-"*100)
print(" f(n) = g(n) + h(n) CALCULATIONS ".center(100,"-"))
print("-"*100)
for node in optimal_path:
    h_val = heuristic(node, goal)
    g_val = g_scores[node]
    f_val = f_scores[node]
    print(f"{node}: g(n) = {g_val:.2f} km, h(n) = {h_val:.2f} km, f(n) = {f_val:.2f} km")

# Alternative routes
alt1 = ['Dhanmondi','Kalabagan','Green Road','Panthapath','UAP']
alt2 = ['Dhanmondi','Science Lab','Mirpur Road','Indira Road','Farmgate','UAP']

def print_route_fn(route, name):
    g = 0
    print("\n" + "-"*100)
    print(f" f(n) CALCULATIONS FOR {name} ".center(100,"-"))
    print("-"*100)
    for i, node in enumerate(route):
        if i == 0:
            g_val = 0
        else:
            g_val = g + G[route[i-1]][node]['weight']
        h_val = heuristic(node, goal)
        f_val = g_val + h_val
        g = g_val
        print(f"{node}: g(n) = {g_val:.2f} km, h(n) = {h_val:.2f} km, f(n) = {f_val:.2f} km")
    total_dist = sum(G[u][v]['weight'] for u,v in zip(route, route[1:]))
    print(f"\nTotal Distance for {name}: {total_dist:.1f} km")

print_route_fn(alt1, "Alternative Route 1")
print_route_fn(alt2, "Alternative Route 2")

print("\n" + "="*100)
print(" ALTERNATIVE ROUTES SUMMARY ".center(100, "="))
print("="*100)
alt1_dist = sum(G[u][v]['weight'] for u,v in zip(alt1, alt1[1:]))
alt2_dist = sum(G[u][v]['weight'] for u,v in zip(alt2, alt2[1:]))
print(f"1) {' → '.join(alt1)} | Distance: {alt1_dist:.1f} km")
print(f"2) {' → '.join(alt2)} | Distance: {alt2_dist:.1f} km")
print("="*100)
```

OUTPUT

```
================================ A* PATHFINDING FROM DHANMONDI TO UAP ================================
Optimal Path: Dhanmondi → Kalabagan → Panthapath → UAP
Total Distance: 3.0 km

-------------------------------- f(n) = g(n) + h(n) CALCULATIONS --------------------------------
Dhanmondi: g(n) = 0.00 km, h(n) = 1.54 km, f(n) = 1.54 km
Kalabagan: g(n) = 1.00 km, h(n) = 1.56 km, f(n) = 2.56 km
Panthapath: g(n) = 2.00 km, h(n) = 0.88 km, f(n) = 2.88 km
UAP: g(n) = 3.00 km, h(n) = 0.00 km, f(n) = 3.00 km

---------------------------- f(n) CALCULATIONS FOR Alternative Route 1 ----------------------------
Dhanmondi: g(n) = 0.00 km, h(n) = 1.54 km, f(n) = 1.54 km
Kalabagan: g(n) = 1.00 km, h(n) = 1.56 km, f(n) = 2.56 km
Green Road: g(n) = 2.50 km, h(n) = 0.54 km, f(n) = 3.04 km
Panthapath: g(n) = 3.70 km, h(n) = 0.88 km, f(n) = 4.58 km
UAP: g(n) = 4.70 km, h(n) = 0.00 km, f(n) = 4.70 km

Total Distance for Alternative Route 1: 4.7 km

---------------------------- f(n) CALCULATIONS FOR Alternative Route 2 ----------------------------
Dhanmondi: g(n) = 0.00 km, h(n) = 1.54 km, f(n) = 1.54 km
Science Lab: g(n) = 1.40 km, h(n) = 1.72 km, f(n) = 3.12 km
Mirpur Road: g(n) = 2.40 km, h(n) = 0.49 km, f(n) = 2.89 km
Indira Road: g(n) = 3.30 km, h(n) = 0.78 km, f(n) = 4.08 km
Farmgate: g(n) = 4.40 km, h(n) = 0.11 km, f(n) = 4.51 km
UAP: g(n) = 5.40 km, h(n) = 0.00 km, f(n) = 5.40 km

Total Distance for Alternative Route 2: 5.4 km

================================ ALTERNATIVE ROUTES SUMMARY ================================
1) Dhanmondi → Kalabagan → Green Road → Panthapath → UAP | Distance: 4.7 km
2) Dhanmondi → Science Lab → Mirpur Road → Indira Road → Farmgate → UAP | Distance: 5.4 km
============================================================================================
```

Technical Report

Objective : Design a Small Address Map

Report Link Here...

Source Code

Github Link...

FINDINGS

Optimal Path: 3.0 Km

Alternative routes were longer (4.7 km & 5.4 km)

A proved efficient with minimal exploration.*

Challenges

- GPS data collection
- Balancing heuristic scaling
- Graph visualization

CONCLUSION

The A* algorithm successfully found the optimal path from Dhanmondi to the University of Asia Pacific with a total distance of 5.9 km, which matches the route suggested by Google Maps. The algorithm efficiently explored the search space by using a combination of actual distance traveled (g(n)) and heuristic estimates (h(n)) to guide the search toward the goal.

Thank You!