COST-BASED GRID PATHFINDING PROBLEM

PROJECT PROPOSAL

1. Introduction and Motivation

In many intelligent systems, an agent is required to navigate an environment efficiently while dealing with different constraints. Unlike simple navigation problems where all movements have the same cost, real-world environments often involve varying movement costs, such as difficult terrain, obstacles, or energy consumption. This project focuses on modeling and solving a Cost-Based Grid Pathfinding problem, where an intelligent agent must determine the most cost-effective path to reach a target location. The problem is studied from an Artificial Intelligence perspective, emphasizing problem formulation, state-space representation, and the application of search algorithms

2. Problem Definition

The selected problem is a Cost-Based Grid Pathfinding Problem, defined as follows: An agent is placed in a two-dimensional grid environment consisting of traversable cells, cells with varying movement costs, and obstacles. The agent's task is to move from a specified starting position to a goal position while minimizing the total accumulated movement cost, not merely the number of steps taken. This problem differs from uniform-cost grid navigation because different grid cells represent different terrain types, each associated with a distinct traversal cost.

3. Environment Modeling

The environment is discretized into a 2D grid, where:

- Each cell represents a possible position of the agent.
- Some cells are blocked and cannot be entered (obstacles).
- Other cells have different traversal costs representing terrain difficulty.

This grid-based representation allows the environment to be treated as a state space, enabling the application of classical AI search techniques.

4. AI Problem Formulation

The problem is formulated using the **Problem-Solving Agent framework**:

Agent

The agent represents an intelligent entity responsible for navigating the grid environment toward the goal.

State Representation

A state is defined by the agent's current position in the grid:

```
State = (row, column)
```

Initial State

The initial state corresponds to the agent's starting cell in the grid.

Goal State

The goal state corresponds to the target cell that the agent aims to reach.

Actions

From any given state, the agent can perform movement actions to neighboring cells:

- Move Up
- Move Down
- Move Left
- Move Right
- Actions are only valid if the resulting cell is within the grid boundaries and is not an obstacle.

Successor Function

The successor function generates all valid neighboring states reachable from the current state, along with the cost associated with moving into each neighboring cell.

Path Cost Function

The total cost of a solution path is calculated as the sum of the movement costs of all visited cells along the path.

5. Search Algorithms Applied

To solve the problem, several search algorithms covered in the course will be implemented and evaluated:

Uninformed Search Algorithms

- Breadth-First Search (BFS)
- Depth-First Search (DFS)
- Uniform-Cost Search (UCS)
- Iterative Deepening Search (IDS)

These algorithms will demonstrate the limitations of uninformed search when dealing with non-uniform movement costs.

Informed Search Algorithms

- **A*** Search
  An admissible heuristic (Manhattan distance) will be used to guide the search efficiently toward the goal.
- Hill Climbing
  Used as a local search method to explore approximate solutions and highlight its limitations.

6. Evaluation Methodology

All algorithms will be tested on the same grid configurations to ensure consistency. The evaluation will focus on:

- Optimality of the solution
- Total path cost
- Number of expanded nodes
- Memory usage

- Completeness

The results will be analyzed to compare the effectiveness of each algorithm in solving the cost-based pathfinding problem.

7. Significance and Applications

Although the problem is implemented using a grid representation, it models a **general AI navigation problem** applicable to multiple domains, including:

- Robot motion planning
- Autonomous navigation systems
- Game AI pathfinding
- Intelligent agents operating in structured environments

This demonstrates the generality of the problem and its relevance beyond a single application domain.

8. Expected Outcomes

T

By completing this project, we expect to:

- Gain a deeper understanding of AI problem formulation
- Observe the practical differences between uninformed and informed search
- Demonstrate why cost-aware search strategies such as A* are essential for realistic navigation problems

9. Deliverables

The project deliverables include:

- Source code for all implemented search algorithms
- A public GitHub repository
- A detailed project report documenting implementation and analysis