AI-2002 Artificial Intelligence

Assignment 1

You are tasked with designing an algorithm to find the optimal path for multiple robots in a dynamic, partially observable environment. Each robot must navigate the grid-based map while avoiding collisions with obstacles, dynamic agents, and other robots. The environment is now more complex due to the following additions:

1. **Grid Representation**:
    - The map is an N x M grid, where each cell can be either:
        - **Free**: The robot can move through this cell.
        - **Obstacle**: The robot cannot move through this cell.
        - **Dynamic Obstacle**: A moving agent that occupies a cell for a certain time interval.
        - **Goal**: The destination the robot must reach.
        - **Minimum Grid Size**: 4x4
2. **Robot Movement**:
    - Each robot can move in four directions: up, down, left, and right.
    - Each move takes 1 unit of time.
    - Robots cannot move into a cell occupied by an obstacle, a dynamic agent, or another robot at the same time.
3. **Dynamic Agents**:
    - There are K dynamic agents moving on the grid.
    - Each agent follows a predefined path and occupies a sequence of cells at specific time steps. When the agent reaches its last index it will start movement in reverse and then vice versa.
    - The paths of the agents are known in advance.
    - **Minimum number of agents:** 2
4. **Multiple Robots**:
    - There are R robots, each starting at a unique position on the grid.

- Each robot has its own goal position.
- Robots do not know the paths of other robots and must plan their paths independently.
- If two robots collide (occupy the same cell at the same time), both robots will randomly change direction and retry their movement.
- **Minimum number of Robots:** 2

5. **Objective**:
    - Find the shortest path for each robot from its starting position to its goal, avoiding collisions with obstacles, dynamic agents, and other robots.
    - The paths must be optimal in terms of both time and distance.
6. **Constraints**:
    - Each robot must start at time t = 0.
    - Robots cannot wait in a cell (they must move at every time step).
    - The algorithm must handle large grids (N and M up to 1000), a large number of dynamic agents (K up to 100), and multiple robots (R up to 10).
    - In case the goal state is an obstacle it will not have a valid path.

Algorithm Requirements:

You must design a hybrid algorithm to solve this problem efficiently. The algorithm should:

1. Explore the grid and identify potential paths while considering the dynamic agents' movements and the presence of other robots.
2. Prioritize paths that are more likely to lead to the goal, using a heuristic function that accounts for both ***distance*** and ***time***.
3. Handle the dynamic nature of the environment by updating the grid state at each time step.
4. Simulate the behavior of other robots by predicting their possible movements and avoiding collisions.
5. Implement a collision resolution mechanism where robots randomly change direction upon collision.

Input:

- A 2D grid representing the map (N x M).
- The starting positions of the R robots.

- The goal positions for each robot.
- A list of K dynamic agents, each with a predefined path (a sequence of cells and the time steps at which they occupy those cells).

Output:

- The optimal path for each robot as a sequence of cells from the start to the goal.
- The total time taken for each robot to reach its goal.

Example:

```
Grid (5x5):

S1 . . . .

. X . . .

. . . X .

. . . . .

. . . . G1


Dynamic Agents:

- Agent 1: [(1, 1), (1, 2), (1, 3)] at times [1, 2, 3]

- Agent 2: [(3, 3), (2, 3), (1, 3)] at times [1, 2, 3]


Robots:

- Robot 1: Start (0, 0), Goal (4, 4)

- Robot 2: Start (4, 0), Goal (0, 4)


Robot 1 Path: [(0, 0), (0, 1), (0, 2), (1, 2), (2, 2), (1, 2), (2, 2),
(3, 2),(4, 2), (4, 3), (4, 4)]

Robot 1 Total Time: 9
```

Robot 2 Path: [(4, 0), (4, 1), (4, 2), (3, 2), (2, 2), (3, 2), (3, 3), (3, 4), (2, 4), (1, 4), (0, 4)]

Robot 2 Total Time: 9