Iran University of Science & Technology

School of Computer Engineering

Assignment #2

Multi-agent system

BY:

DR. Naser Mozayani, Fall 2024

Teaching Assistants:

Aysan Ghayouri

Due: 1404/02/06

Contents

Notes:

1. Submit the answers in a complete PDF file and the code for the questions in the .ipynb format (including the notebook cell outputs) in a compressed file named HW2_StudentID.zip by the specified deadline.
2. A total of 72 hours of delay in submitting the answers is allowed across all projects. After that, for each additional day of delay, 10% of the score will be deducted.
3. If a student submits the project earlier than the deadline and achieves 75% of the score, up to 24 hours will be added to their allowable delay time.
4. The maximum delay for submitting each assignment is 5 days, and after 5 days, submission will not be accepted.
5. The exercises must be performed individually, and group participation is not allowed.
6. It is important to note that the explanation of the code and the obtained results must be included in the PDF file. Code without a report will result in a score deduction.
7. The evaluation of the assignment will be based on the correctness of the solution and the completeness and accuracy of the report.
8. Assignments must be completed individually, and group work on assignments is not allowed.
9. Please allocate sufficient time for the assignment and avoid leaving it until the last days.
10. You can ask your questions in the relevant group.

good luck.

Exercise 2: Designing an Intelligent LLM-Driven Agent (City in Flames)

In this exercise, you will build upon the environment from Exercise 1 by replacing the random agent with an intelligent agent powered by a Large Language Model (LLM). The LLM-driven agent must use real-time observations, strategic planning, and reasoning to navigate the city, extinguish fires, rescue civilians, and make informed decisions in a partially observable, dynamic environment.

Scenario Description:

A 15×15 grid-based city is initialized with fires, buildings, emergency stations, and civilians. The environment is unknown to the agent at the beginning, except for the fixed positions of the emergency stations. The agent must explore the city, discover threats and targets, and perform actions based on a structured prompt processed by an LLM. The goal is to maximize score through firefighting and civilian rescue operations within a limited number of time steps.

Environment Details and Game Rules:

Environment Components:

- Buildings: 7 immovable obstacles that block the agent's movement. The agent cannot enter or pass through these cells.
- Emergency Stations: Located at the top-left corner ([0, 0]) and the bottom-right corner ([14, 14]). Rescued civilians must be delivered to these locations. Fire cannot spread into these cells.
- Civilians: 8 individuals randomly distributed across the city. The agent must rescue and transport them to an emergency station.
- Fires:
    - Type 1 Fires: 5 small fires randomly placed, each occupying one cell and independently extinguishable.
- Agent: A single agent initialized at a random location. It can move in four directions: up, down, left, and right.

Movement and Interaction Rules:

- The agent can move to any of the four adjacent cells (up, down, left, right) at each step.
- The agent may revisit previously visited cells.
- To extinguish a fire, the agent must enter a fire-containing cell and execute a special action. Simply entering the cell does not automatically extinguish the fire.
- To rescue a civilian, the agent must enter the civilian's cell, execute the rescue action, and then transport the civilian to an emergency station.
- The agent can carry only one civilian at a time and should prioritize transporting the civilian to the emergency station.

Fire Spread:

- Every 5 consecutive agent moves, the fires spreads randomly to an adjacent cell (right, left, up, down).
- Fire cannot spread to a cell containing an emergency station and buildings.
- If all surrounding cells of a fire are already burning, it will not spread further.
- The agent can pass through burning cells.
- If the agent is carrying a civilian, the civilian is not harmed while passing through fire.

Perception and Exploration:

- At the start, the agent has no knowledge of the fire locations, buildings, or civilians.
- The environment is completely unknown, and the agent must explore to discover it.
- The agent only has knowledge of the emergency station's location.
- The agent has sensors that allow it to perceive only a 5×5 area around itself.
- To uncover the rest of the map, the agent must move and explore gradually.
- If fire spreads beyond the agent’s sensor range(5×5 area), the agent will not be aware of the change.

Scoring System:

- +10 points for extinguishing each Type 1 Fire.
- +50 points for rescuing and delivering a civilian to an emergency station.
- -100 points if a civilian dies due to fire spread.

Game End Conditions:

- The game ends after 100 time steps.
- If all fires are extinguished and all civilians are rescued within this limit, the game is successfully completed.
- If fires remain or not all civilians are rescued, the agent fails.

Additional Notes:

- All components (fires, buildings, civilians, emergency stations) are randomly and unpredictably placed.
- To extinguish fires or rescue civilians, the agent must perform specific actions—simply entering the cell is not enough.
- The agent must explore and observe the environment to gather information for decision-making.
- Use the Python programming language.
- Display the city map and the map discovered by the agent from its surroundings, and print them on the .ipynb output.

Key Enhancements from Exercise 1:

- Replaces random decision-making with LLM-guided action planning.
- Adds structured prompts and LLM reasoning for real-time interaction.
- Maintains fire spread rules and game dynamics from Exercise 1.
- Introduces explanation-based actions for transparency and evaluation.

Agent Design Requirements:

I. Perception and Memory

- Track the explored portions of the map (15×15 grid).
- Log discovered locations of:
  - Fires
  - Civilians
  - Buildings
- Monitor:
  - Whether the agent is carrying a civilian
  - Nearest emergency station

II. LLM-Driven Decision-Making

- At each step, generate a structured prompt for the LLM containing:
  - Agent's current coordinates and 5×5 perception view
  - Known map entities (fires, civilians, buildings)
  - Current score, remaining time steps
  - Carrying status (whether a civilian is onboard)
- The LLM should return one valid action:
  - Move up/down/left/right
  - Extinguish fire
  - Rescue civilian
  - Deliver civilian
- Invalid actions result in either no action or a fallback random valid action.

LLM Integration Guidelines:

Example Prompt Format

```
[AGENT STATE]
Position: (x, y)
Sensor View: [[...], [...], [...], [...], [...]]
Known Fires: [(x1, y1), (x2, y2), ...]
Known Civilians: [(x3, y3), ...]
Carrying Civilian: Yes/No
Score: N
Time Left: T steps

[GOAL and PRIORITY]
Extinguish fires, rescue civilians, and avoid penalties.

[ACTION]
Move right
```

Response Expectations

- Validate the LLM output to ensure it maps to a valid action.
- Require the LLM to include an explanation of reasoning, such as:

  "Moving south to approach civilian at (7,9) and avoid fire at (6,8)."

Implementation Requirements:

- Use the Python programming language.
- Use either open-source LLMs (e.g., models from Hugging Face) or free-tier APIs**. You can use the hugchat package from pypi.**
- Display the following in a Jupyter Notebook (.ipynb):
  - Full city map (ground truth)
  - Agent's discovered map over time
- Provide a performance report comparing:
  - LLM-driven agent

- Random agent from Exercise 1

Objective of the Exercise:

To apply concepts from multi-agent systems, LLM integration, and intelligent decision-making in a partially observable, dynamic environment. This hands-on task prepares you for more advanced simulations and real-world AI reasoning challenges.