Iran University of Science & Technology

School of Computer Engineering

Assignment #5

Multi-agent Systems

BY:

DR. Naser Mozayani, Fall 2025

Teaching Assistants:

Nafiseh Ahmadi

Due: 1404/03/24

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 **HW5_StudentID.zip** by the specified deadline.
2. A total of **72+48** 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. Please allocate sufficient time for the assignment and avoid leaving it until the last days.
9. You can ask your questions in the relevant group.

good luck :)

Designing a Competitive Multi-Agent LLM System (City in Flames)

In this exercise, you will extend the cooperative scenario from Exercise 3 by introducing a second team of intelligent agents. Each team consists of multiple agents powered by Large Language Models (LLMs) and must operate competitively in the same dynamic, partially observable city. The agents will need to extinguish fires and rescue civilians while strategically competing against the opposing team. This exercise emphasizes strategic competition, real-time decision-making, and adversarial reasoning within and across LLM-driven agent teams.

Scenario Description

A 15×15 grid-based city is initialized with fires, buildings, emergency stations, and civilians. Two teams of LLM-powered agents are deployed into the environment, each aiming to outperform the other in saving civilians and extinguishing fires. Initially, only the positions of the emergency stations are known to both teams. The agents must explore the city and make real-time decisions based on their perceptions and shared knowledge within their own teams. Inter-agent communication is handled through structured LLM prompts, enabling dynamic strategy updates. The objective is to maximize each team's individual score within a limited number of time steps, balancing rapid action and competitive tactics.

Environment Details and Game Rules

<u>Environment Components</u>

- Buildings: 7 immovable obstacles blocking movement.
- Emergency Stations: Located at [0,0] and [14,14], serve as safe zones for delivering rescued civilians.
- Civilians: 8 individuals randomly distributed.
- Fires:
    - Type 1 Fires: 5 small fires, independently extinguishable.

    - Type 2 Fires: 4 large fires, require simultaneous action by both agents on the same team to extinguish.

- Agents: Two teams (Team A and Team B), each with two LLM-powered agents initialized at random positions.

Movement and Interaction Rules

- Each agent can move to any of the four adjacent cells (up, down, left, right) at each step.
- Agents can simultaneously occupy a cell or pass through each other.
- 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.
- Each agent can carry only one civilian at a time and should prioritize

Fire Spread

- Fires spread every 5 agent moves (same as previous exercises).
- Same rules for spreading restrictions (cannot spread into buildings/ emergency stations).

Perception and Exploration

- Agents have a 5×5 perception range.
- Each team shares observations within the team, updating a shared team map.
- No direct communication between teams; partial observability of opposing agents limited to cells within perception range.

Scoring System

- +10 points for extinguishing each Type 1 Fire (requires 1 water tank).
- +25 points for extinguishing each Type 2 Fire (requires both team agents and 2 water tanks).
- +50 points for rescuing and delivering a civilian to an emergency station.

- -100 points of a civilian died due to fire for each team

Game End Conditions

- The game ends after 50 time steps.
- The team with the highest total score wins.
- If all fires are extinguished and all civilians rescued before time expires, the leading team wins.
- If tied, the team that extinguished more fires wins.
- If still tied, the game is a draw.

Additional Notes

- All components (fires, buildings, civilians) are randomly and unpredictably placed at the start.
- Agents must perform explicit extinguish and rescue actions; moving into cells is insufficient.
    - Use Python for implementation, focusing on modular, clean code with clear LLM integration.
- Display:
    - Full city map with all elements.
    - Each team's discovered map updated live.
    - Real-time scoring updates per team.

Key Enhancements

- Competitive Dynamics Introduced: Transition from cooperative to competitive multi-agent systems.
- Two-Agent Teams: Agents must now coordinate with teammates while competing with opponents.

- Shared Team Perception: Shared observation maps are used within each team.

Agent Design Requirements (Enhanced from Exercise 3)

Agents must now:

- Coordinate with a teammate in real-time to extinguish Type 2 fires and optimize team strategy.
- React to changing goals based on score tracking and time constraints.
- Communicate within the team (if allowed) using token-efficient prompts or shared blackboard memory.

LLM Integration Guidelines

LLMs must now handle:

- Strategic planning under adversarial conditions (e.g., prioritize saving civilians vs. extinguishing fires).
- Team-based collaboration prompts — enabling agents to negotiate plans, allocate tasks, and sync on actions like dual-agent extinguishing.
- Real-time decision adaptation using changing game state, partial observability, and risk estimation.

Evaluation Criteria and System Constraints

Evaluation Criteria

- Total Score per Team (primary metric).
- Efficiency: Fires extinguished and civilians rescued per time step.
- Coordination Quality: Quality of teamwork in achieving Type 2 fire extinguishing and optimized rescue paths.

System Constraints

- Time Limit: 50 steps.
- One Civilian per Agent at a Time.
- Shared vision only within teams.
- Perception limited to 5×5 grid.
- All placements randomized at game start.

Implementation Requirements

- Use the Python programming language.
- Use either open-source LLMs (e.g., models from Hugging Face) or free-tier APIs
- Display the following in a Jupyter Notebook (.ipynb):
    - Full city map (ground truth)
    - Agent's discovered map over time

Key Additions in This Exercise

- Team-based architecture: Each team manages 2 agents and a shared map.
- Scoring Tracker UI: Real-time score display for both teams.
- Logging Required: All agent decisions and score changes must be logged (resource interactions removed).
- LLM prompting must use updated context, including:
    - Team map
    - Current team score
    - Known positions of enemies (if visible)

Objective of the Exercise

To develop and evaluate a competitive multi-agent LLM-driven system where two opposing teams of agents operate in a dynamic, partially observable environment. Each team aims to maximize its score by extinguishing fires and rescuing civilians. The objective emphasizes intra-team cooperation and inter-team competition, introducing real-time strategy and competitive tactics to the agent design challenge.