TÉCNICO LISBOA

Autonomous Agents and Multiagent Systems

Projects must aim to address a real-world problem by developing an intelligent system that includes a multiagent system with its mathematical formulation and decision-making process. Students are free to choose the programming language to develop their system.

1. PROJECTS

Projects should focus on providing a mathematical model, designing, and implementing a multiagent system while considering cooperation, coordination, or negotiation strategies. A system with centralized control over a set of agents is unacceptable (e.g., an AI agent that plays a Real-time strategy (RTS) game commanding all units and operations). The project needs to include several of the following features:

- Agents should have either *conflicting goals* or face complex *coordination problems*;
- Communication and coordination mechanisms (cooperation, teamwork);
- Complex negotiation or cooperation between agents.

The creation of the environment itself does not contribute to the grade, but students are free to use any environment that they are developing in personal projects or in other courses.

2. GOALS & DELIVERABLES

All projects have to develop/implement the following components:

1. **A conceptual model of the multiagent system:**
    - o definition of the **problem**: requirements and objectives;
    - o specification of **agent and environment properties**: sensors, actuators, environment details, decision-making behavior, etc.;
    - o **design choices regarding the architecture** for the implementation of the agent system;
2. A **game-theoretical formalism** describing the problem.
3. **Advanced decision-making**: agents with advanced decision-making capabilities, such as coordination mechanisms or learning techniques (e.g., reinforcement learning);
4. **Engineering the agent system:** includes agents with different behaviors (e.g., baseline versus advanced behaviors);
5. **Comparative analysis** (*very important*): a report should include a description of the experiments and a thorough analysis of the agents' behaviors for each implemented approach (with tables, graphs, etc.). The report should also include a discussion of the different approaches' suitability to address the target problem based on the experimental results.

3. Evaluation Criteria

The project will be evaluated according to the following criteria:

- A relevant, clear, and complete description of the addressed problem;
- Selected approach and architectural principles to address the problem;
- A correct conceptual model of the (simulated) environment and agent behavior;
- Suitability and correctness of any intelligent and emergent agent behavior;
- Adequacy and completeness of the undertaken experimental analyses;
- Relevant, clear, and complete conclusions regarding the empirical results;
- Clear and adequate communication of the findings (report, presentation, and accompanying video).

Grading is individual. Each student within a group should present his/her contribution to the project.

The evaluation of the project will consider the following criteria:

10% Project proposal;

20% Adequacy and correctness of the formal description of the problem and of the proposed intelligent system to address the target problem;

30% Sound and complete implementation using suitable architectural principles;

40% Experimental analyses, discussion and findings as verified in the report and presentation;

4. Project Proposal, Final Delivery & Due Dates

4.1 Project Proposal and Presentation

The project proposal is due on Friday, **April 26th, 2024.** The project proposal is a one-page document that describes the agent or multiagent system to be implemented. Please use the *ACM Primary Article Template* (2-column article).

The project proposal must include four brief sections:

- An **abstract** that summarizes the addressed problem, approach, and expected contributions;
- An **introduction** that includes motivation, related work, problem definition and relevance, and objectives of the project;
- A description of the **approach** with the specification of the environment, (multi)agent system, and system architecture. This section should also include text to explain why the design choices are adequate to address the problem;
- A description of the **empirical evaluation**. Define a set of metrics that can validate the project's objectives.

Be prepared to make a 2-minute presentation to the lab instructor in the class (from **April 29th** - **May 3rd**).

4.2 Final Delivery

The final delivery of the project is due on Friday, **May 24th, 2023**, with the following elements:

1. **Full source code** and **executable** with a README that explains how to run the system
2. **Final report** (5-to-8 pages) according to the *ACM Primary Article Template* (2-column article).
3. A 1-to-2 minutes **video** demonstrating the agents or algorithms *in-action* (N.B.: a video of text outputs within a terminal (system console) is not accepted). Some examples of acceptable videos:

a. The emergence of some social phenomenon in a population;
b. The effect of learning a new behavior (compare the performance before and after learning);
c. The contagion of emotions in a population;
d. The "team-behavior" exhibited by some group of agents;
e. The agents' behavior according to the different approaches.

Be prepared to make a 5-minute presentation of the project (including a presentation of the video) followed by a 5-minute session of questions and answers (from **May 27th** - **May 31st**).

***Important*: there will be no changes to the due dates (deadlines).**

5. PROJECT EXAMPLES

5.1 Petting Zoo

Develop agents for classical games (e.g., Atari, Card games, Board games).
https://pettingzoo.farama.org/
https://github.com/Farama-Foundation/PettingZoo

5.2 Multi-Agent Learning

Apply multi-agent learning algorithms in collaborative or competitive games. You can base your project on already implemented/existing frameworks, e.g. openspiel.

https://github.com/google-deepmind/open_spiel

https://colab.research.google.com/github/deepmind/open_spiel/blob/master/open_spiel/colabs/OpenSpielTutorial.ipynb

https://magent2.farama.org/

5.3 Sustainable Mobility

Model a transportation network to improve urban mobility. Use a multiagent system to understand the impact of bus routes, bicycles, and scooters on urban mobility issues. There are several interesting situations where the creation of a new road can increase the average commuting time, e.g. https://en.wikipedia.org/wiki/Induced_demand, https://en.wikipedia.org/wiki/Downs%E2%80%93Thomson_paradox, https://en.wikipedia.org/wiki/Braess%27s_paradox, https://en.wikipedia.org/wiki/Jevons_paradox, https://en.wikipedia.org/wiki/Lewis%E2%80%93Mogridge_position

5.4 Wrong Incentives and Public Procurement

Governments, or teachers, influence the behaviors of society by providing different incentives. A badly design incentive can make the problems worse, e.g. https://en.wikipedia.org/wiki/Perverse_incentive, https://en.wikipedia.org/wiki/Goodhart%27s_law, https://en.wikipedia.org/wiki/Campbell%27s_law. Find examples and create a model of the situation. Change the incentives to correct the problem. Most government contracts are given based on a public procurement process. Several rules exist to ensure that the selected offer is of good quality, e.g. ignoring offers that are too cheap, and of a good price, e.g. considering average prices of the proposals. Nevertheless, there are many cases where companies collude between them to inflate the price, or to ensure that they get the contracts.

5.5 Competitions

Develop agents for competitions that have been organized within conferences.
https://2024.ieee-cog.org/competitions/
https://ijcai-23.org/competitions/
https://www.aamas2024-conference.auckland.ac.nz/accepted/accepted-competitions/