TÉCNICO LISBOA

AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS

2023

PROJECT

Projects must aim to address a real-world problem by developing an intelligent system that includes: a) an autonomous agent or b) a multiagent system. Students are free to choose the programming language to develop their system.

1. AGENTS

1.1 AUTONOMOUS AGENT SYSTEM

Projects may focus on designing and implementing a single agent. For instance, groups may consider forms of interaction between a user and an agent. However, the project must need to go beyond a project for an undergraduate AI course, including several of the following features:

- An agent within a complex environment;
- Agent with deliberative, adaptive, curious or/and emotional behavior;
- Agent with several goals, sensors, and actuators;
- Environment with uncertainty and noise;
- Agent system with rich interactions and feedback.

1.2 MULTIAGENT SYSTEM (MAS)

Projects may focus on designing and implementing a multiagent system while considering cooperation, coordination, or negotiation strategies. A system with centralized control over a set of agents I not acceptable (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*;
- Agents should have a variety of sensors and actuators (and they *should not be too limited)*;
- Communication and coordination mechanisms (cooperation, teamwork);
- Complex negotiation or cooperation between agents.

2. GOALS & DELIVERABLES

All projects have to develop/implement the following components:

1. **A conceptual model of the agent or multiagent system:**
    - definition of the **problem**: requirements and objectives;

- specification of **agent and environment properties**: sensors, actuators, environment details, decision-making behavior, etc.;
- **design choices regarding the architecture** for the implementation of the agent system;

2. **Advanced decision making**: agents with advanced decision-making capabilities, such as coordination mechanisms or learning techniques (e.g., reinforcement learning);
3. **Engineering the agent system:** includes agents with different behaviors (e.g., baseline versus advanced behaviors);
4. **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 embedded mechanisms of emotions, teamwork, learning, etc.;
- 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).

4. Project Proposal, Final Delivery & Due Dates

4.1 Project Proposal and Presentation

The project proposal is due on Friday, **May 12th, 2023.** The project proposal is a two-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 succinct 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 **May 15th** - **May 19th**).

4.2 Final Delivery

The final delivery of the project is due on Friday, **June 9th, 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 presentation of the video) followed by a 5-minute session of questions and answers (from **June 12th - June 16th**).

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

5. Grading

Grading is individual. Each student within a group should clearly present his/her contribution to the project. The evaluation of the project will consider the following criteria:

10% Project proposal;

10% Adequacy and correctness of the proposed intelligent system to address the target problem;

10% Sound and complete implementation using suitable architectural principles;

50% Experimental analyses, discussion and findings;

10% Project description, positioning and communication (report);

10% Video presentation.

6. Project Examples

6.1 Autonomous Agents

6.1.1 Collaborative game with a person

Check Sueca *http://gaips.inesc-id.pt/parceiro/*, and Fireboy and Watergirl *http://www.fireboynwatergirl.com/*. In these settings, agents need to be proactive and adapt to the user. Visuals and game engines will not be evaluated.

6.1.2 Logic-geometric Programming

Optimize joint motions of multiple agents to solve cooperative sequential manipulation tasks, which require planning at symbolic and motion level.

https://ipvs.informatik.uni-stuttgart.de/mlr/papers/15-toussaint-IJCAI.pdf

https://ipvs.informatik.uni-stuttgart.de/mlr/marc/source-code/16-LGP.tgz

6.2 Multiagent Systems

6.2.1 Geometry Friends

Implement agents for a two-player cooperative puzzle game developed at GAIPS:

http://gaips.inesc-id.pt/geometryfriends/

6.2.2 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.

6.2.3 Emergency Responses

Use a multiagent system to effectively distribute ambulances and resources to respond to emergencies according to the type, gravity, and spatiotemporal distribution of emergency requests.

6.2.4 Traffic with Autonomous Vehicles

Study consequences of vehicles becoming autonomous and how to improve traffic when cars are autonomous.
https://www.youtube.com/watch?v=iHzzSao6ypE&t=148s

https://flow-project.github.io/videos.html

Model a circulation network with a multiagent system; research and implement human factors affecting driving. Study the impact of different control architectures (local vs. global), automatic coordination between vehicles, percentage of autonomous cars in the circulation, traffic lights, size and interconnection of the network, vehicle priority, and cost/payment systems.

6.2.5 Logistics

Consider *N* companies receiving requests to transport different products between various locations using trucks with a limited capacity. We want to understand how different mechanisms can lead to coordination, the formation of coalitions, monopolies, and side-payment schemes. Use a multiagent system to model a complex network (with different suppliers, clients, orders, and delivery mechanisms). Study the different forms of control architectures for each supplier and their impacts (e.g., reactive vs. non-reactive). Analyze the impact of different service modalities, pricing (cost/payment systems), number of companies, and clients.

6.2.6 Collaborative AI in Minecraft

https://www.microsoft.com/en-us/research/academic-program/collaborative-ai-challenge/

6.2.7 Simulated Robot Soccer

Develop a team of agents for the simulated RoboCup league.

https://www.robocup.org/leagues/23

https://www.cs.utexas.edu/~AustinVilla/sim/halffieldoffense/

6.2.1 Agent-based Competitions

Develop agents for competitions that have been organized within conferences.

https://ieee-cog.org/2022/#COMPETITIONS

https://ijcai-22.org/competitions/

6.2.1 PettingZoo

Develop agents for classical games (e.g., Atari, Card games, Board games).

https://github.com/Farama-Foundation/PettingZoo