Introduction to Artificial Intelligence

Bachelor in Informatics Engineering and Bachelor in European Computer Science Engineering

Year 2 – 1st semester 2023/2024

Practical Work No. 1 - Rational Agents

1. Introduction

The goal of this work, worth 2 values, is to conceptualize, implement and analyse rational behaviours for reactive agents. The project must be completed using NetLogo.

Within a virtual world, two types of agents coexist. The detailed description of the environment is given in section 2. Each type of agent has well-defined characteristics, and the agents interact with each other competitively. The characteristics of the agents and their rules of interaction are detailed in section 3. The main objective of the simulation is to ensure the survival of the agents as long as possible.

2. The Environment

The environment is characterized by a two-dimensional toroidal grid, creating an open-world setting. In this world, mainly composed of black cells, there should be poor food elements (brown cells) and rich food elements (red cells). The percentage of cells in the environment that contain food should be user-configurable (between 0% to 20% for brown cells and between 0% and 10% for red cells). Once a rich food is consumed, it undergoes a transformation into poor food. When the poor food is consumed, the cell reverts to its empty state (black). The energy gain by eating food is also configurable (ranging from 1 to 50). Poor foods must randomly reappear in the environment in such a way that the configured levels remain similar throughout the simulation. Additionally, the environment should have randomly positioned blue cells. Its quantity must be configurable by the user (ranging from 0 to 5).

3. The Agents

In the environment there are *lion* agents and *hyena* agents. The initial quantity of each of them should be configurable by the user. The agents walk through the environment and react to the cells of the environment or to the other agents that they perceive, according to their nature. Given that the primary goal of these agents is to maximize their survival, each agent is equipped with an energy level. At the time of their creation, the agents receive the same initial amount of energy (configurable by the user). Furthermore, *hyena* agents are assigned a clustering level of 1 upon their creation. At each iteration (*tick*) the agents lose the amount of energy corresponding to the action they perform. If the energy of the agents reaches zero, they die.

3.1. *Lion* Agent

The characteristics of the *lion* agents are as follows.

3.1.1. Perceptions

Each *lion* should be able to perceive the contents of the cells shown in Figure 1, where the arrow represents the *lion* and its travel direction.

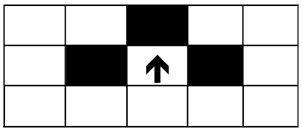

Figure 1: Perceptions of the *Lion* agent.

3.1.2. Actions

Each *lion* can feed, move to the cell immediately in front, rotate 90 degrees to the left, rotate 90 degrees to the right, fight hyenas, or perform a special movement action when it detects two or more hyenas in the cells it perceives. On each tick, it can only perform one of these actions. In every action other than combat and the special movement, the lions lose one (1) unit of energy. The energy loss in combat is defined in the following subsection. The special movement allows the *lion* to move away as far as possible from the perceived agents. The possible retreat actions are as follows:

- If the *lion* has two or more *hyenas* to its left, it jumps to the cell to its right, losing two (2) units of energy.
- If the *lion* has two or more *hyenas* to its right, it jumps to the cell to its left, losing two (2) units of energy.
- If the *lion* has two or more *hyenas* in front, **or** if the *lion* has *hyenas* on its left side and *hyenas* on its right side, it jumps to the cell behind, losing three (3) units of energy.
- If the *lion* has one or more *hyenas* on its left side and one or more *hyenas* in front, it jumps to the cell behind on the right side, losing five (5) units of energy.
- If the *lion* has one or more *hyenas* on its right side and one or more hyenas in front, it jumps to the cell behind on the left side, losing five (5) units of energy.
- If the *lion* has one or more *hyenas* on its left side, one or more *hyenas* on its right side, and one or more hyenas in front, it jumps to two cells behind, losing four (4) units of energy.

3.1.3. Behaviour

- Operate as a reactive agent.
- Give the highest priority to eating, provided that the energy level falls below a user-defined threshold.
- Give the highest priority to the special movement when the energy level is equal to or greater than the user-defined threshold mentioned above.
- Consume food when located in a cell containing it.
- Engage in combat with *hyena* agents when they are the only agents in the perceived vicinity. Following combat, transforms the defeated *hyena* into a poor food source and loses a percentage of the value of the energy it had (configured by the user).
- Enter a resting state for a user-defined number of *ticks* upon perceiving a blue cell. During rest, the agent does not take any action, does not lose energy, and cannot die in combat (although he may be perceived by *hyenas* and chosen for combat).

3.2. *Hyena* Agent

The characteristics of the *hyena* agents are as follows.

3.2.1. Perceptions

Each *hyena* should be able to perceive the contents of the cells shown in Figure 2, where the arrow represents the *hyena* and its travel direction.

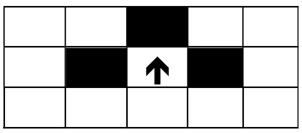

Figure 2: Perceptions of the *hyena* agent.

3.2.2. Actions

Each *hyena* can feed, move to the cell immediately in front, rotate 90 degrees to the left, rotate 90 degrees to the right, and fight *lions*. On each tick, *hyenas* can only perform one of these actions. In each action, the *hyenas* lose one (1) unit of energy, except in combat with *lions* (this energy loss is defined in the following subsection).

3.2.3. Behaviour

- Operate as a reactive agent.
- Continuously update the clustering level to match the number of *hyenas* in the perceived vicinity. When this level exceeds zero, the agent changes colour; it returns to the initial colour when the clustering level becomes one.
- Prioritize eating.

- Consume food when located in a cell containing it.
- Kill the *lion* when its clustering level is higher than 1 and only if there is a *lion* agent in the perceived neighbourhood, transforming it into a rich food source and, due to the combat, losing a percentage of the *lion's* energy (configured by the user) divided by the clustering level.
- Do not perceive blue cells.
- Impose its action, in terms of movement, on all *hyenas* in the perceived neighbourhood (all these agents should adopt the same direction, or *heading*, as the *hyena* imposing the action).

4. Project Structure

The work is divided into two main components: implementation and experimentation/results analysis.

4.1 Implementation

The project entails two key implementations:

a) Base Model - Incorporating all aspects outlined in sections 2 and 3. In cases of omissions or potential ambiguities, students are encouraged to select reasonable solutions and provide justifications in the report.
b) Improved Model - Enhancing the base model to optimize agent performance. Originality in these enhancements, backed by explanations in the report, is highly encouraged. Potential avenues for improvement include modifying agent memory and perceptions.

4.2 Experimental testing and Evaluation

This task involves conducting experiments to test both the base and improved models. For each experiment, a maximum number of iterations should be set, and the survival count for each agent type should be recorded. If an agent becomes extinct, the survival time in *ticks* should be recorded. To gain a comprehensive understanding of agent behaviour, each experiment should be repeated a minimum of 100 times, allowing for the calculation of average values.

For each experiment, parameters that may influence agent performance should be identified. Hypotheses should be formulated (e.g., 'food quantity affects agent survival') and tests to validate them should be performed. This entails adjusting parameters, such as 'food quantity,' and observing whether agent survival is impacted.

Each enhancement proposed in the improved model should be evaluated by comparing with the equivalent base model. This assessment aims to determine if agent performance has improved and provide justifications for the results.

5. Evaluation criteria

- Base Model Implementation (30%)
- Improved Model Implementation (30%)
- Experimental Testing and Evaluation (30%)
    - analysis of a minimum of three (3) hypotheses per model
- Documentation and Presentation (10%)

6. Report

- In the final report, which should **not** exceed 10 pages, provide a detailed description and justification for all implemented enhancements.
- Include information on the formulated hypotheses, tested configurations (parameter values), metrics, and explanations regarding the impact of each enhancement on agent performance.

7. Guidelines and Submission Instructions

- The project should be completed in groups of two students. In exceptional cases, with the permission of the teacher of the practical class, the work may be carried out individually.
- The project submission deadline is **23/10/2023**, by 00:00, via *Moodle*. The submission should include all implementations, the report, test results, and presentation slides (if applicable). All files must be compressed into a .ZIP file with a name that identifies the group members (name and student number), such as: "**StudentName1_StudentNumber1_StudentName2_StudentNumber2.ZIP**."
- Presentations of the projects will be scheduled during the week of **23/10/2023 to 27/10/2023** in the practical class where at least one of the students is registered. Students must confirm the presentation date with their practical class teacher. Each group will have a **10-minute** slot to present and justify their key decisions.