TÉCNICO LISBOA

AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS

2ND SEMESTER - 2015/2016

LEARNING AGENTS IN *WOLF PACK*

Project Description

1. INTRODUCTION

The objective of this project is to implement a simulation with reactive, deliberative and learning social agents in the "Pursuit" environment, using the NetLogo tool. In this project the students should implement not only the agents but also the simulation environment itself.

This document presents the requirements for the implementation of the simulation. It overviews the characteristics and dynamics of the "Pursuit" domain and defines the role that communication and multiagent learning plays for improving the overall behavior of the group of agents. Finally, it sets out the objectives of the project and the evaluation criteria.

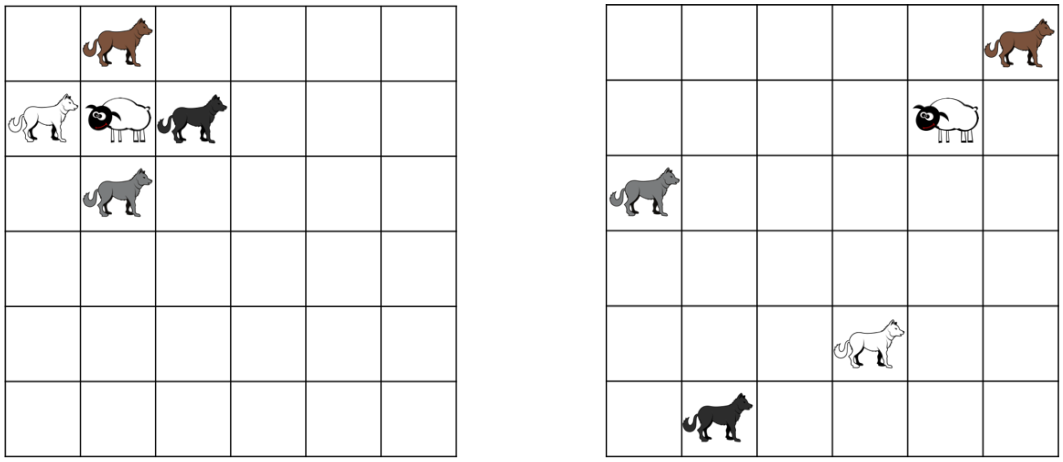
Note: It is assumed that the students are already familiarized with the NetLogo tool used in the laboratory classes.

2. DESCRIPTION OF THE "PURSUIT" ENVIRONMENT

This project will explore the creation of agents of different kinds in the Predator/Prey, or "Pursuit" domain. The "Pursuit" domain was introduced by Benda et al. [1] and is an interesting example for multiagent systems (MAS) as it has been studied using a wide variety of approaches in many different instantiations of the same problem.

The "Pursuit" domain involves the interaction between four predators, each with its own *color identifier*, trying to capture one prey, *e.g.*, a pack of wolves and a sheep. In this project we will consider

that the interaction occurs in a discrete *toroidal grid-world* environment of size n x n, *i.e.*, where movement actions performed over the border of the grid-world transport the entities into the cell in the opposite end of the environment. In this problem the agents are modelled as the *predators*, whose *goal* is to "capture" the prey by *surrounding* it so that it cannot move to an unoccupied position, as depicted in Figure 1(a). In this project, consider that the prey *cannot occupy the same position* as one of the predators at a given time. The *initial position* of the elements in the environment is randomly assigned, *e.g.*, as depicted in Figure 1(b).



(a) CAPTURE (b) RANDOM

FIGURE 1: POSSIBLE STATE CONFIGURATIONS FOR THE PURSUIT PROBLEM.

In a typical scenario, the predators have a *limited visual field* of depth d, and therefore cannot perceive the position of all entities in the environment in a given time. In this project we will follow the rule proposed in [2], that considers the following restriction: $2d + 1 < n$. At each time-step of the simulation, all entities *simultaneously* choose and perform some action according to its decision-making mechanism. As such, the state of the environment in the next time-step will be a result of all combined actions given the described restrictions, *e.g.*, if two predators decide to move to the same cell, both actions will fail and they will remain in the same position.

In this project we consider the agents to be *homogeneous* in the sense that they share the same goal (capture the prey), have the same capacities (same actions, same perceptual capabilities) and share the overall benefit (the "value" of capturing a prey is divided between them). In addition, consider that each agent is an *individual entity* behaving and learning according to its own perceptions at each time and *ignoring the actions* taken by the other agents. This also means that there cannot be a central controller combining the perceptions of all agents and assigning the next actions for each of them.

Notes:

- In the "Pursuit" domain the prey may also be modelled as an agent with a reactive or more intelligent behavior to escape the predators. However, for the purposes of the project you may consider that the prey has a random behavior, *i.e.*, at each time moves to an adjacent position or remains still with some probability;
- When preforming learning and evaluation of the agents' behaviors, consider that an interaction *episode ends* when the *prey is captured* or some predefined *time limit* has been reached.

3. Agent Architectures

The purpose of this project is to perform a comparative analysis of the prey-capturing ability of the group of agents using different approaches. This section overviews the several approaches and architectures that may be implemented and compared.

Note: We encourage the students to consult the survey in [3], where a description of several instantiations of the "Pursuit" domain and several approaches to solve this MAS problem are presented, along with the corresponding work references.

3.1 Reactive Agents

Several approaches of reactive agents have been proposed in the literature for the "Pursuit" domain. The students may implement any existing strategy (*e.g.*, choose one of prey's adjacent position and go towards it) or propose one of their own. See [3] for some examples and respective references.

Notes:

- Realistic restrictions regarding reactivity and observability must be ensured, *e.g.*, no communication or memory is allowed;
- Any problems that may arise regarding the joint behaviors of the agents must also be addressed in a reactive manner.

3.2 Deliberative Agents

The deliberative agents can extend the behaviors of the reactive approach and allow for more complex processing of the agents' perceptions, *e.g.*, search for a best path, target different positions, use formations through forces, etc.

Notes:

- The implementation of a BDI architecture and respective definition of desires is optional;
- If communication is used, realistic costs must be assumed.

3.3 Learning Agents

The implementation of agents that learn throughout time as a result of continuous interactions with the environment will be the main aspect of development for this project. In particular, the students will have to implement an algorithm based on the *reinforcement learning* (RL) framework. RL has been extensively studied as a framework for learning sequential decisions in an uncertain and, in the perspective of each agent, dynamic environment, as is the case of the "Pursuit" domain. Many extensions of RL to the MAS case have also been proposed to deal with the interdependence between agents' behaviors in order to achieve the task. Moreover, many are based on ideas from the *evolutionary game theory* literature to ensure that the resulting joint behavior of the agents reaches an *equilibrium*.

However, there are many complications when addressing these kinds of MAS learning problems. First, typical *joint-action learning* techniques, *i.e.*, where each agent has into account the actions taken by others during learning, assume that the agents are able to observe each other's actions, which in the case of this project would be *unrealistic* as the perceptual field of predators is limited. On the other hand, coordination between *individual learners* (IL) is hard to achieve if they cannot communicate or

share information about goals and observed states. Above all this there is the problem of combinatorial explosion in the state space – since multiple learning agents have to jointly perform a coordinated task, the state space for each learning agent grows *exponentially* in the number of partners.

Given the above problems, we encourage the implementation of a multiagent RL modular approach for the "Pursuit" domain proposed in [2]. The idea is to *decompose* the high-dimensional problem into smaller, more tractable problems, by having several Q-learning modules in *parallel* (for each agent), each dealing with part of the agent's perceptions. We advise the students to carefully review the paper and implement the proposed algorithm.

Notes:

- Approaches based on single-agent learning (*i.e.*, central learning and control of all agents) are not allowed. Each agent must be an IL;
- Again, the students may use (and justify their decision) any existing multiagent learning algorithm from the literature;
- More information of several (single-agent) RL techniques and algorithms may be found in [4].

3.4 Hybrid Approaches

You may also want to combine the multiple approaches by having mixed teams composed, *e.g.*, by two learning agents, a reactive predator and a deliberative one, and compare the overall performance of the group. Or, you may consider the heterogeneous learning agents case, where each agent has different perceptual capabilities (*i.e.*, a different d parameter for each agent). In addition, several parameterizations may be used to produce more interesting experiments (see below some suggestions).

4. Objectives

The overall objective is to develop intelligent learning agents corresponding to the predators as well as the entire "Pursuit" environment described above. In order to implement the described dynamics, the following aspects have to be taken into consideration:

4.1 Scenario Modelling

- The **"Pursuit" environment** (should be modelled as a rectangular area with varying size);
- The two **types of entities** inhabiting the environment;
- The necessary the **sensors** and **actuators** of the predator agents;
- The **dynamics** of the system, *i.e.*, the next state of the environment resulting from all agents' actions (including the prey).

4.2 Parameters

The simulation should be parameterizable and the effect of changing each parameter should be evaluated in the comparative analysis. Some parameters that may be considered include:

- **Size** and **shape** of the world (*e.g.*, toroidal vs. closed, with obstacles, etc.) ;
- Predators' **legal moves** (including diagonals vs. orthogonal only);

- Predators' **movement** (deterministic vs. stochastic);
- **Visible range** of elements (varying parameter d);
- **Prey movement** (random vs. reactive, *e.g.* flee from predators' center).

4.3 Performance Metrics

Many metrics evaluating the performance of the agents may be considered, including:

- Amount of **prey** captured after a given number of trials;
- **Time** spent until capturing the prey;
- Amount of **communication** messages exchanged, if communication was used;
- Number of **collisions** or colliding decisions between the predators.

4.4 Architectures

- A **reactive** architecture, *i.e.*, without internal state or communication;
- A **deliberative** architecture using some form of reasoning;
- A **learning architecture** in which the agents are individual learners.

Note: Please consult the project requirements document (`project-requirements.pdf`) and the report template (`report-template.docx`) for detailed information on the generic requirements and evaluation criteria for all projects.

5. References

[1] M. Benda, V. Jagannathan, and R. Dodhiawala. *"On optimal cooperation of knowledge sources - an empirical investigation"*. Technical Report BCS–G2010–28, Boeing Advanced Technology Center, Boeing Computing Services, Seattle, Washington, July 1986.

[2] N. Ono and K. Fukumoto. "Mulit-agent reinforcement learning: A modular approach". In *Second International Conference on Multiagent Systems*, pp. 252-258, Kyoto, 1996.

[3] P. Stone and M. Veloso. "Multiagent systems: A survey from a machine learning perspective". *Autonomous Robots*, 8(3), 2000.

[4] Sutton, R. S., and Barto, A. G. "*Reinforcement Learning*", MIT Press, Cambridge, MA, 1998. Online at: https://webdocs.cs.ualberta.ca/~sutton/book/the-book.html