AdLeap-MAS: An Open-source Multi-Agent Simulator for Ad-hoc Reasoning

Demonstration Track

Matheus Aparecido do Carmo Alves
Lancaster University
United Kingdom
m.a.docarmoalves@lancaster.ac.uk

Yehia Elkhatib
University of Glasgow
United Kingdom
yehia.elkhatib@glasgow.ac.uk

Amokh Varma
Indian Institute of Technology, Delhi
India
Amokh.Varma.mt618@maths.iitd.ac.in

Leandro Soriano Marcolino
Lancaster University
United Kingdom
L.marcolino@lancaster.ac.uk

ABSTRACT

Ad-hoc reasoning models are recurrently used to solve some of our daily tasks. Intending to avoid worthless investments or spend valuable resources, these smart systems require a proper evaluation before acting in the real-world. In this paper, we demonstrate AdLeap-MAS, a novel framework focused on enabling quick and easy testing of smart algorithms in ad-hoc reasoning domains.

KEYWORDS

Simulation Framework; Open-source; Ad-hoc Reasoning; Online Planning; Autonomous Systems.

ACM Reference Format:


1 INTRODUCTION

Autonomous systems play notable roles in contemporary society. Facing the increasing number of devices (agents) in the environment and the rising of problems with higher complexity, there is a need for new intelligence methods capable of solving tasks, learning about the context and handling uncertainties in an online-manner. A typical approach presented by the state-of-art is to assign multiple intelligent systems (agents) to solve a common objective and use an ad-hoc teamwork model to coordinate this Multi-Agent system [1, 10, 16, 18, 20, 23, 26]. However, they miss the opportunity to generalise their model by considering different roles in the environment (e.g., where agents can be potential teammates or opponents) – defining what we denominate as an *ad-hoc reasoning domain*.

On the other hand, building an intelligent system implies also testing and evaluating it. Simulators are important tools for the advancement of ad-hoc reasoning research [12]. However, current literature suggests that each researcher is implementing their own scenarios and using different custom-built simulators for similar purposes [2, 5, 7, 14, 15, 17, 18, 21, 25]. Therefore, a common platform that is capable of assembling different problems and algorithm implementations is still missing.

We propose the open-source *Adaptive Learning and Planning Multi-agent Simulator (AdLeap-MAS)*, a novel framework focused on simulating ad-hoc reasoning problems, where potential type(s)/policies for other agents are estimated, and sampled during an on-line decision-making process. We offer base classes for implementing new problems and algorithms, besides ready-to-use common benchmarks found in the literature. This proposal supports the execution of reactive algorithms, neural networks, estimation methods, reinforcement learning and on-line planning application over full and partial observability, only requiring the connection of algorithms to the ad-hoc reasoning model. In this way, our contributions can be summarised as: (i) first simulator that allows a quick switch of learning and planning algorithms across different ad-hoc reasoning scenarios; (ii) *AdLeap-MAS* enables the execution of multiple reasoning agents that run independently; (iii) our architecture guarantees information security while running scenarios under partial observability, i.e., agents do not have access to any forbidden information, and; (iv) a standard set of benchmark algorithms and problems to allow fair and quick experiments.

2 DESIGN FEATURES

The *AdLeap-MAS*’s architecture is based on unilateral and cyclical module communication, where the information within the framework must be delivered or received directly and exclusively by one module from another in the architecture. Such design enables the problem simulation as a step-by-step process, processing each fragment of the simulation independently. The 3 main modules are – *Environment*, *Decision-making*, and *Components* modules. The Environment module is responsible for all the simulations and also makes sure that certain information is hidden from the Decision-making module, to ensure that partial observability is not violated. The Components module controls the different dynamic parts of the environment – such as agents, tasks to be completed etc. This allows the user to use the Environment module as a black-box, which takes the action as input and returns the observation, without revealing any extra information. The high-level overview of *AdLeap-MAS* can be seen in Figure 1.
3 PROBLEM SCENARIOS
Currently, AdLeap-MAS offers 6 different environments: three of which are ad-hoc reasoning domains (1.abc) while the other three are traditional partial observability toy-problems (2.abc).

(1.a) The Level-based Foraging (LBF) represents an ad-hoc team-work domain where the ad-hoc agent tries to maximise the number of boxes collected by its team while learning the environment, teammate features and deciding its own actions [1].

(1.b) Capture the Prey (CTP) is a domain derived from the traditional “pursuit game” [3, 4], where a team of hunters tries to catch all the preys in the environment. They need to surround all the preys in order to capture them before the timer expires.

(1.c) The “Truco” Card Game (TCG) is based on the popular Brazilian card game, where the agents have a small window of observation to make decisions and maximise their chance of winning.

(2.a) The Maze (MZ) is an active localisation problem where the ad-hoc agent navigates a toroidal grid-world to gather the available observation and figure out its actual position [22].

(2.b) The Rock Sampling (RS) domain, in which a robot moves around a grid-world and tries to maximise the number of “good” rocks collected while exploring the unknown map [19].

(2.c) The Tiger Domain (TG) is a classic POMDP domain where the ad-hoc agent must open one of two doors: one has a tiger and the other a treasure. The agent can listen to the tiger or make a decision of which door to open without this observation [11].

Figure 2 illustrates the above domains. We provide an easy-to-use template for the implementation of a new environment, which is versatile as the figure shows. The template is available at the framework GitHub’s page with more information about the environments and simulator usage. Additionally, AdLeap-MAS offers some ready-to-use baselines for planning experiments, besides state-of-art estimation methods and several other reactive methods.

4 RELATED WORK
Intending to point out the major differences between the AdLeap-MAS and the frameworks which are currently available for similar purposes, we discuss some state-of-art proposals in detail.

OpenSpiel is a reinforcement learning framework widely used for the evaluation of planning algorithms [13]. However, even though it presents a collection of environments and algorithms, OpenSpiel is not focused on the simulation of ad-hoc reasoning domains. Furthermore, AdLeap-MAS enables an easier swap of algorithms between environments and agents using its component-based architecture.

Similar limitations emerge in the GAMA project [8], which focuses on the Multi-Agents context, but does not address ad-hoc reasoning domains. Despite the issue of acquiring a world model, the framework also requires an understanding of its dedicated programming language before utilisation.

In the literature, we also find simulators that are focused on tackling the representation and simulation of real robotics, such as Gazebo [12] and Stage [24]. Even having the ability to present high-fidelity simulations of multi-robot problems, these frameworks do not support learning/planning algorithms.

Finally, Open-AI Gym [6] is a Python package that provides a collection of benchmarks to run reinforcement learning tests by abstracting the environment. We extend the benefits of the Open-AI Gym platform and also improve its range of applications. By directly modelling and offering support of ad-hoc reasoning applications, we specialised the package for evaluation and simulation into the context. Our framework also handles modifying visibility restrictions without requiring further implementation.

5 ADLEAP-MAS READY-TO-GO
Our GitHub’s page furnishes the users with extensive documentation on how to use the framework. Moreover, we also release an introductory video to demonstrate the functionalities of our simulator and facilitate understanding of its operation.

This paper represents our willingness to spread the current AdLeap-MAS’s results. We want to allow the development of research in a collaborative manner, capable of improving the overall results found by the community in the short and long term. As mentioned, our purpose is not to surpass the capabilities and functionalities of other simulators. Instead, our aim is to build a reliable solution that alleviates the difficulty of running experiments, and the complexity of fairly comparing algorithms for different problems without losing trust in the collected results.

Finally, we are continuing to work on AdLeap-MAS and improve its environments. Our current project is focused on developing a problem within the continuous action and state spaces – denominated Smart Fire Brigade Environment (based on [9]) – where we want to go towards more realistic scenarios, evaluating real-systems constraints, costs and application.

1 AdLeap-MAS’s GitHub: https://github.com/lsmcolab/adleap-mas/
2 Introductory Video: https://youtu.be/aCXFHyvoiHo