UNIVERSITY OF TRENTO - Italy
Department of Information Engineering
and Computer Science

AUTONOMOUS SOFTWARE AGENTS

Project Presentation

presented by:

Marina Segala
248447
marina.segala@studenti.unitn.it

Pietro Bologna
248715
pietro.bologna@studenti.unitn.it

1. INTRODUCTION

Objective

The goal is to develop a **BDI-based agents** to collect and deliver parcels efficiently.

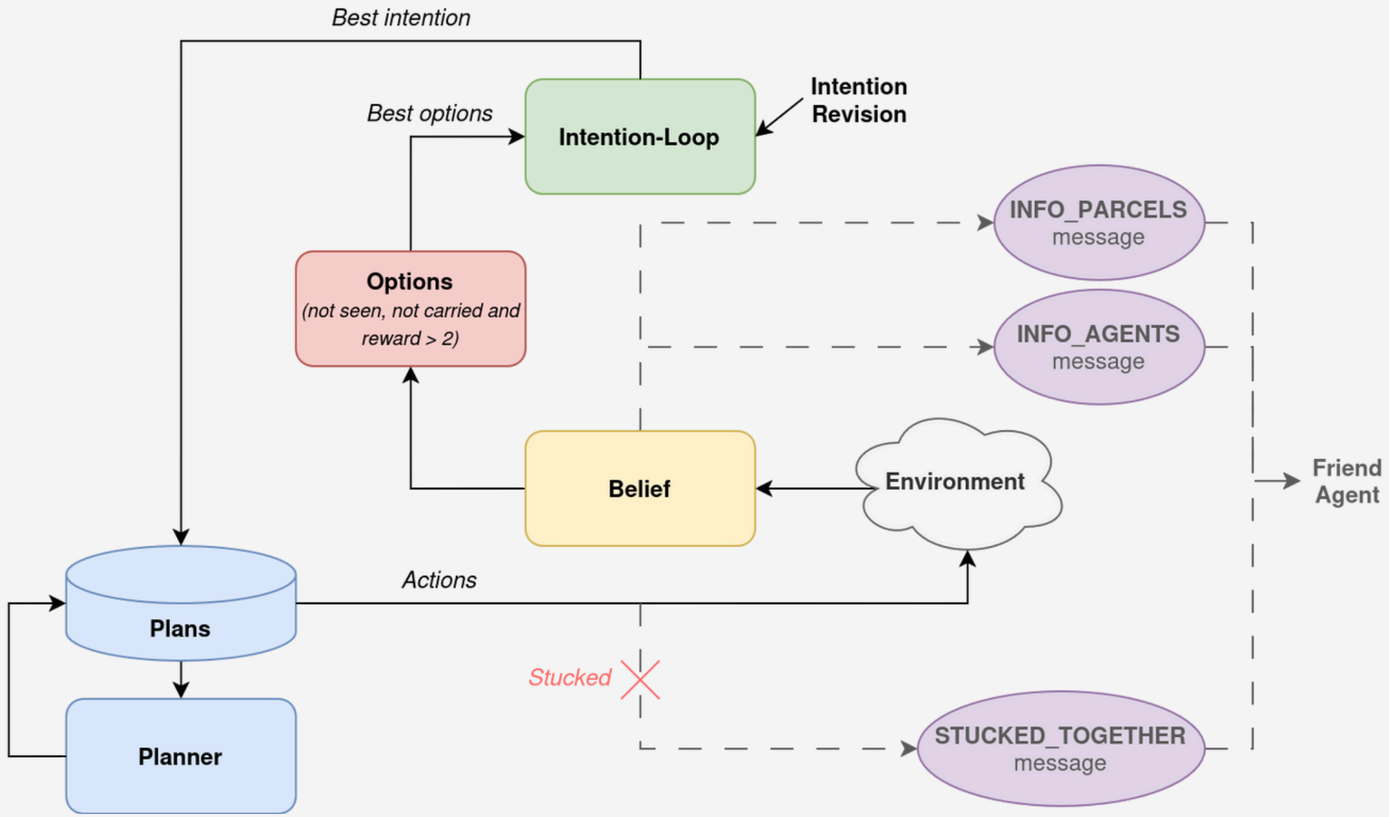
Our project is structured in two main parts:

- **Single Agent:** one agent that perceives, plans, and acts autonomously.
- **Multi-Agent:** two collaborative agents works together in the same environment.

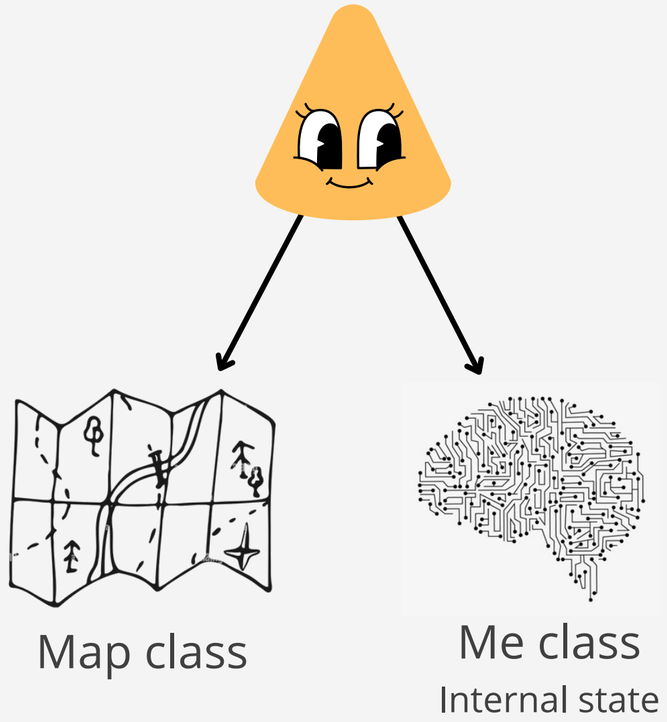
Performance is tested through simulations with varying challenges.

1. INTRODUCTION

Belief-Desire-Intention



Agent



Map class

Me class

Internal state

Belief-Desire-Intention

The agent builds and updates its **beliefs** asynchronously based on environmental stimuli, ensuring up-to-date knowledge.

The main observations that our agent can detect include

- **Map**: used to carry out operations needed to perform the selected intention.
- **Parcel Sensing**: detects and updates parcel positions dynamically.
- **Agent Sensing:** tracks other agents, removing them if unseen for 20 seconds.

Intention-Loop

It is responsible for continuously processing the intentions of the agent.

The **intention queue** is created, and the best option available is selected

- action pickup of nearest particle has to be found
- action putdown is created if some particles are carried and the closest delivery point has a lower distance than the nearest particle
- action goto is created with *RandomMove* if intention is not defined
- intention is added to the queue
- sorting of the queue according to the distance from the agent

Intention-Loop

It is responsible for continuously processing the intentions of the agent.

The intention queue is created, and the best option available is selected
If the agent is **unable** to complete the selected intention

- counter for failed actions will increase
- action putdown is executed immediately, if it is carrying some particles
- after **2** times in a row with no satisfied intentions, action goto the *prev_pros* coordinates

Planning

Primary goal: capture as many particles as possible without getting stuck.

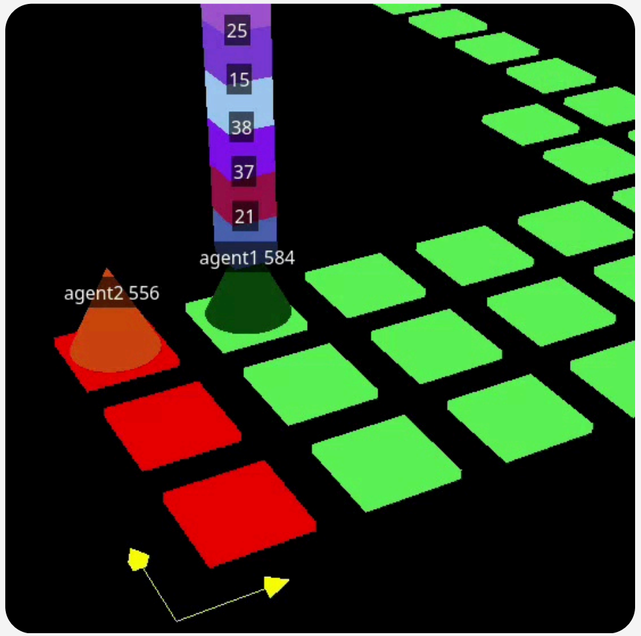
The path is calculated with two strategies:

- **BFS**: proved to be more faster
  - shortest path is computed
- **PDDL**: efficient path planning but slower;
  - fixed PDDL domain
  - map-based PDDL problem

Planning

The agent may not be able to complete the selected action:

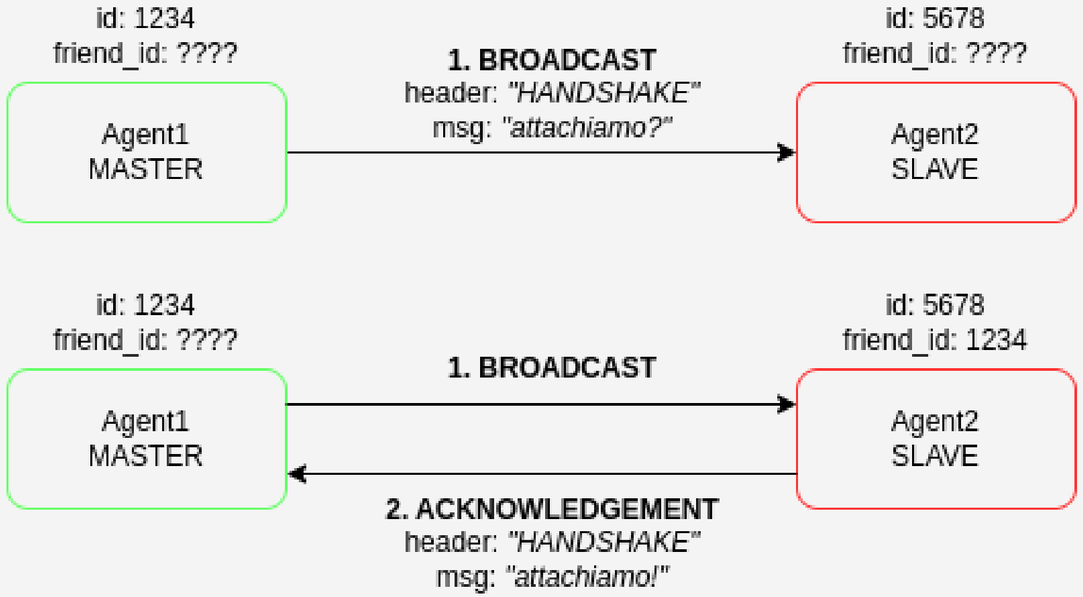
- **Universal** case:
  - a delivery point is blocked by an enemy;
  - the second-nearest delivery point has to be reached.
- **Multi-Agent** case:
  - handled with messages.

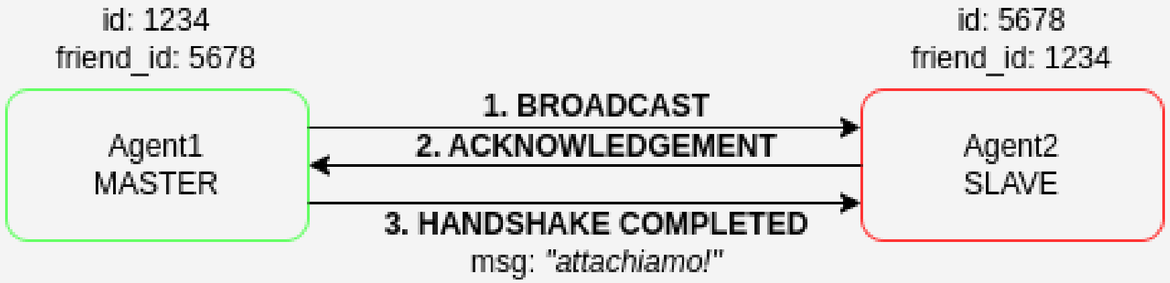


3. COMMUNICATION

Handshake

In the Multi-Agent scenario, it is necessary to establish a connection between Agent 1 (MASTER) and Agent 2 (SLAVE)





Agents set their *friend_id* to establish a mutual connection

Environmental information

INFO_PARCELS

Agent can decide to move towards the parcels to perform the pickup action.
Two conditions must be respected:

- particle has not yet been collected;
- particle reward > 4 or the distance from the agent < 20 steps

INFO_AGENTS

Agent updates its knowledge about other agents in the environment.
The map remains up-to-date

Messages of stucked

An agent block its friend during the execution of an intention

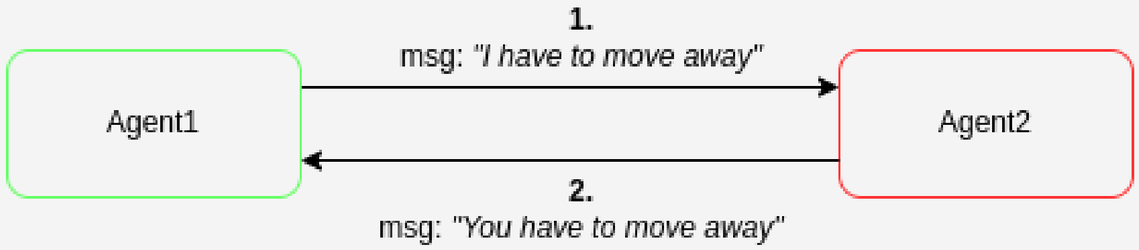


Figure 1: Case "*I have to move away*"

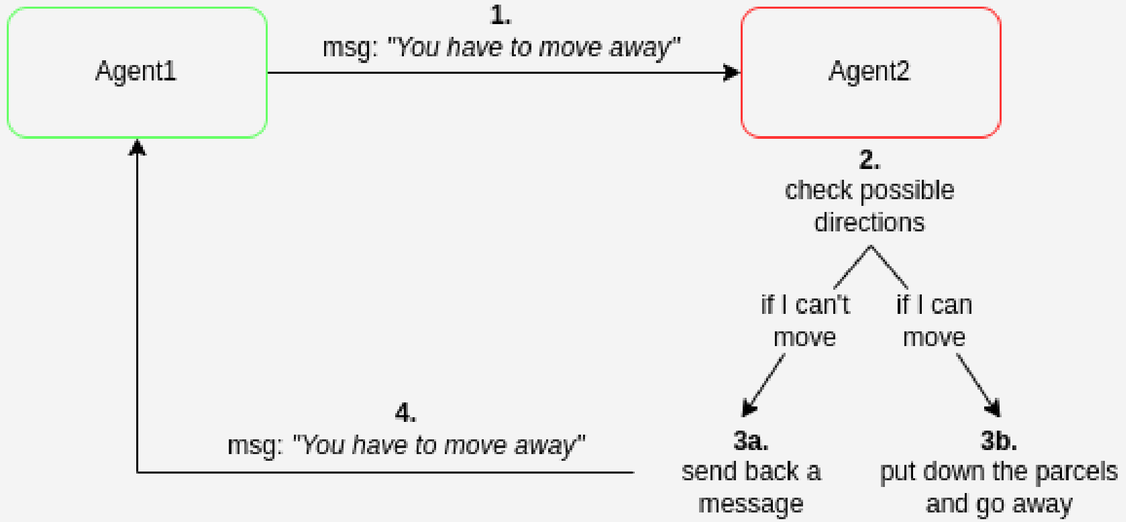


Figure 2: Case "*You have to move away*"

Results

Single-agent results

Level	Score
24c1_1	*1220*
24c1_2	*758*
24c1_3	*1603*
24c1_4	*148*
24c1_5	*6941*
24c1_6	*443*
24c1_7	*586*
24c1_8	*631*
24c1_9	*778*

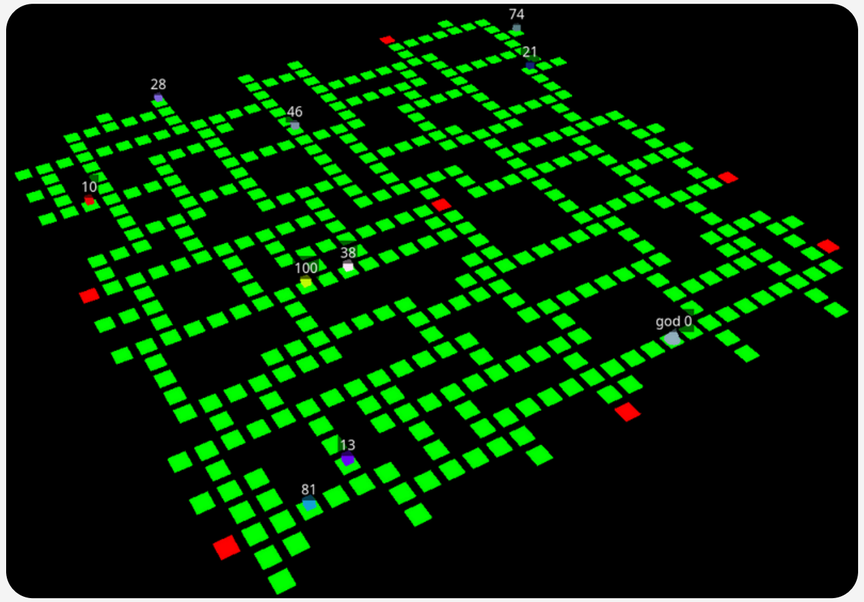
Multi-agent results

Level	Score	Agent 1 Score	Agent 2 Score
24c2_1	*1640*	*700*	*940*
24c2_2	*4343*	*2203*	*2140*
24c2_3	*2657*	*841*	*1816*
24c2_4	*1849*	*1346*	*503*
24c2_5	*1297*	*603*	*694*
24c2_6	*14908*	*8259*	*6656*
24c2_7	*6986*	*3454*	*3532*
24c2_8	*320*	*155*	*165*
24c2_9	*5204*	*2318*	*2886*

Single-agent evaluation

In single-agent scenarios, performance was stronger on **randomized spawning maps**.

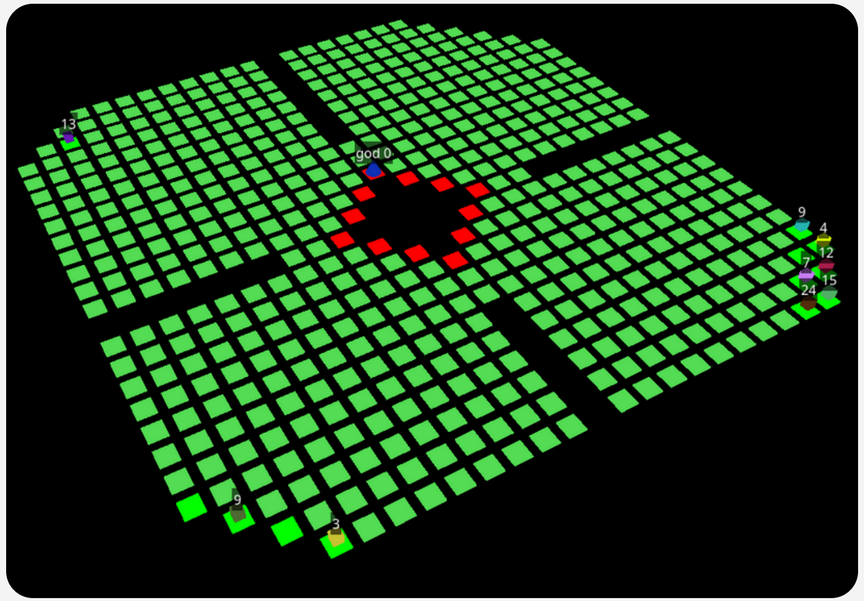
For instance, on maps like *24c1_5*, where parcels spawn randomly, our agent performs very well.



Single-agent evaluation

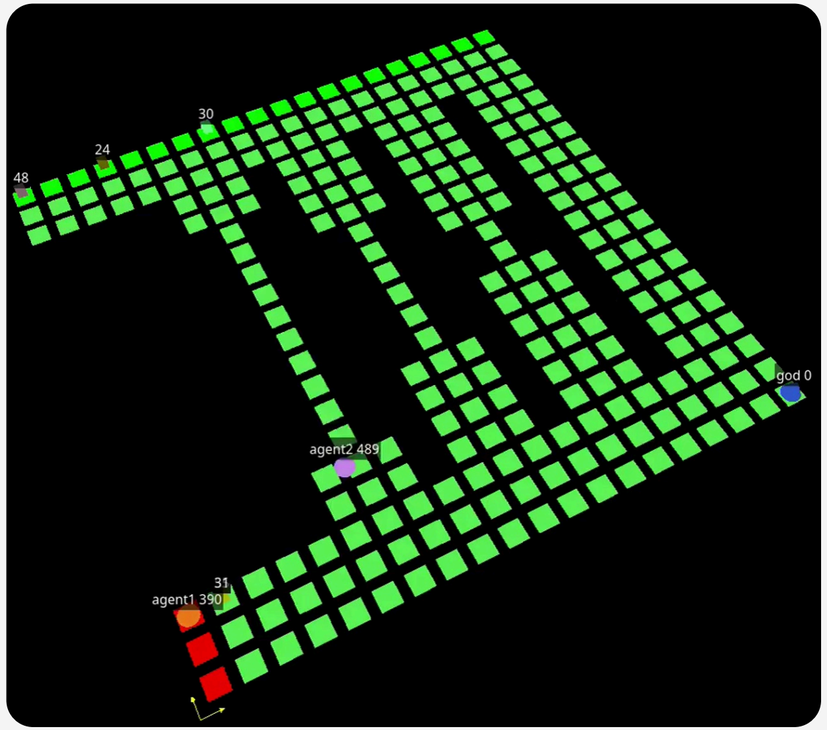
But the performance is **slightly weaker** on fixed spawning ones.

For instance, on maps like *24c1_4*, where parcels are placed along the edges, our agent encounters some difficulties.



Multi-agent Evaluation

Thanks to **effective communication** and **efficient handling of stuck situations**, our agents cooperate seamlessly, achieving strong performance and reliable results.



Future improvements

The results are promising, but there is still significant potential for improvement:

- **Enhancing multi-agent coordination** and optimizing path planning for smoother collaboration.
- **Exploring alternative planning methods** to achieve a better balance between speed and accuracy.
- **Integrating learning-based approaches** to enable agents to adapt dynamically over time.

UNIVERSITY OF TRENTO - Italy
Department of Information Engineering
and Computer Science

THANKS FOR THE ATTENTION!

Marina Segala & Pietro Bologna

https://github.com/bolognapietro/AutonomousSoftwareAgents_Deliveroo-Agents