Coursework #2 Report

Introduction

This paper describes an ant farm simulation implemented using MultiAgent System techniques. It shows how 2 ant colonies compete in resource gathering located in the same environment. The purpose of MAS is to illustrate decentralised decision-making and behaviours inspired by real-world ant biology and behaviour. The system integrates general principles of agent modelling, pheromone dynamics and resource-based competition.

Communication as the Overall Goal of the MAS

The main goal of the MAS is to simulate the behaviour and interaction within 2 ant colonies that compete for resources in a shared environment. The system shows how the agent, in the given scenario - ants, navigates, gathers resources and utilises a pheromone-based communication to retrieve the resources more efficiently.

The pheromone-based communication is considered indirect since there is no straight engagement between the ants. To clarify, pheromones are deposited on the grid, when the resources are located, leaving a trail that can be felt by other members of a colony, therefore giving the information that there is something in that area. Additionally, in simulation, the pheromones decay over time ensuring that the path to already collected resources is not congesting the communication.

Furthermore, the pheromone trails are strengthened by other ants that travel on this path, giving other ants extra information about the resource(s) that are located there. Since there are 2 colonies in the environment, ants prioritise their colony pheromone trail, to ensure competitive behaviour (in the application, pheromone trails are different colours for convenience).

Utilising those principles, the MAS appears to be a fairly realistic simulation of real-world ant behaviour, demonstrating basic principles of indirect communication using pheromone trails.

Features of the MAS

1. Environment

The environment is represented as a 2D grid where ants search for resources (leaves for 1 point and wood for 2 points), leave pheromone trails in order to communicate to other colony members and bring the resources back to the nest. The resources are randomly placed, but the colony locations are placed in the bottom left and top right corners (3 blocks away from the edge). Additionally, the environment itself is dynamic, therefore, if the ant gets to an edge, she appears on the opposite side of the map.

2. Agent-based modelling

Each colony consists of a number of ants that autonomously navigate through the environment, search for resources and bring them back to the nest. Ants make decisions based on pheromone strength, resource type and random movement. Ants are gaining score points upon delivering the resource back to the nest, the points quantity is determined by its quality (c. 1. Environment).

3. Pheromone utilisation

The pheromones are placed when the ant finds a resource and carries it back to the nest, creating a pheromone trail that leads from the nest towards a resource. Additionally, as previously mentioned, the pheromones decay over time, to ensure the correctness of a navigation path.

4. Resource respawn

Resources are limited therefore if an ant completely gathers it the resource disappears. Therefore, in order to prolong the simulation the resources respawn after a chosen time period, making the unusable environment and continuous exploration.

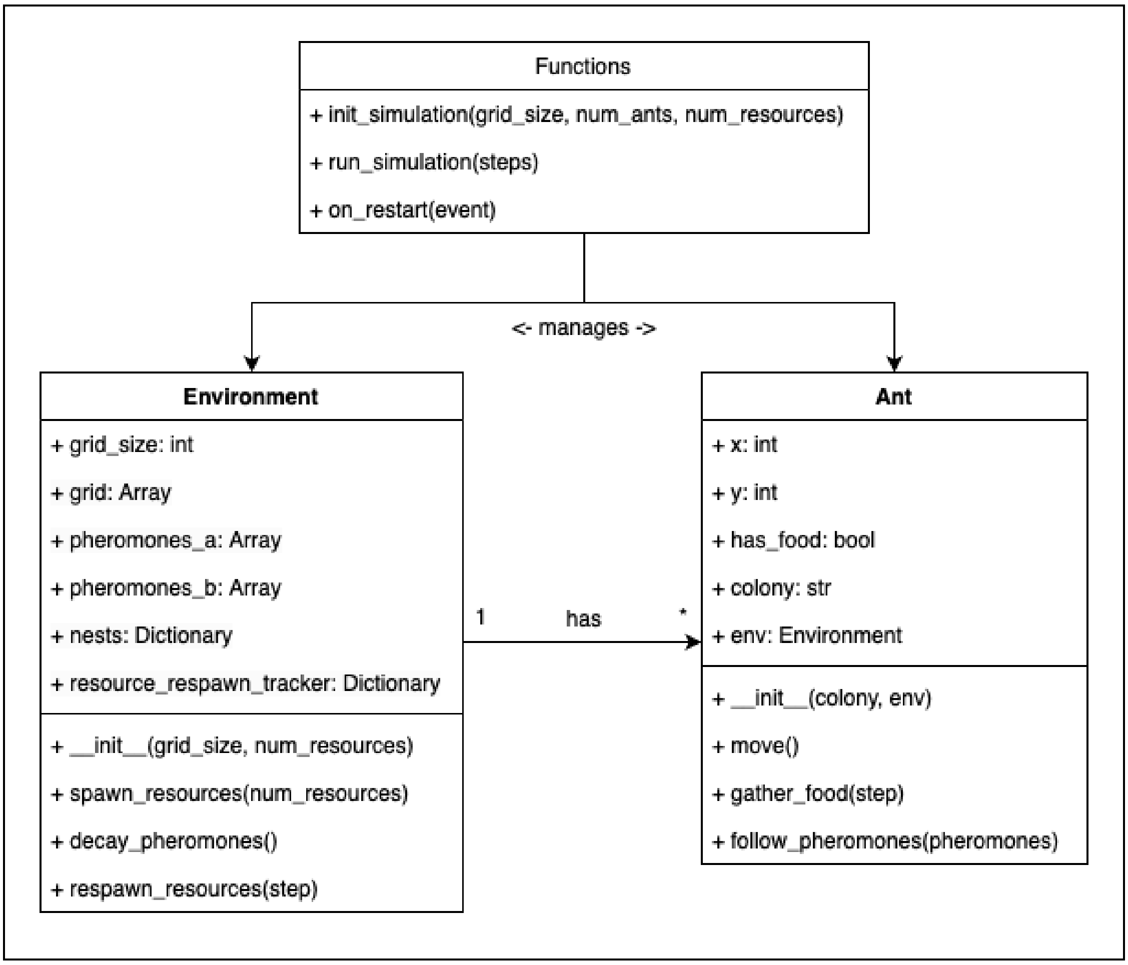
5. Visualisation and GUI

The simulation is visualised by Matplotlib, which displays a grid-based environment, ants from different colonies (blue and red), respectively coloured pheromone trails, their nests and resources (green leaves and brown wood).

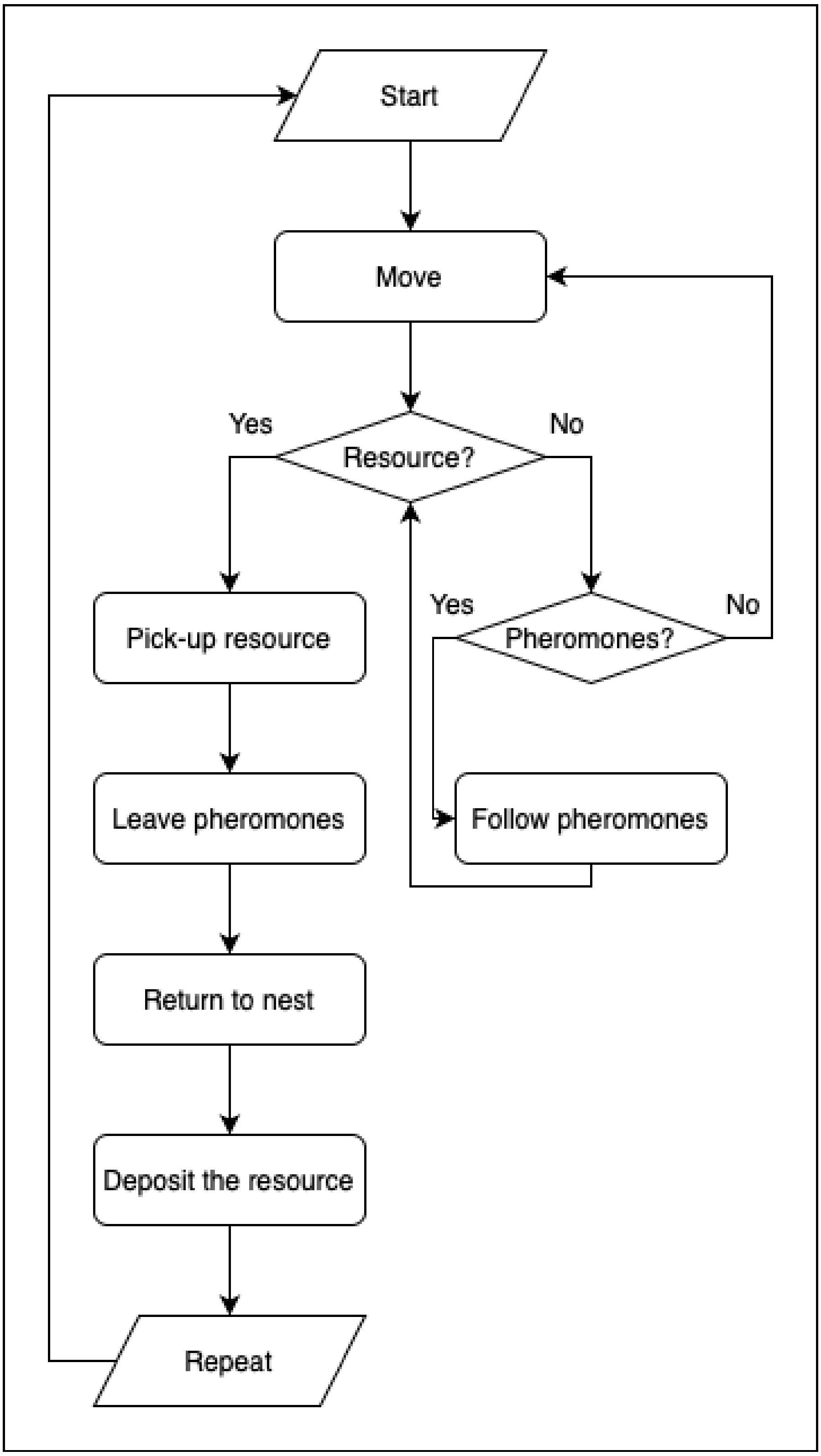
Additionally, the interactive GUI element is implemented that allows an uninterrupted continuance for the simulation, it allows to change grid size, ants and resource quantity, steps limitation, and restart the simulation in general.

Agent-oriented Design

Entity Relationship Diagram



Agent Decision-Making Tree



MAS improvements

1. Adding obstacles to simulate more realistic terrain conditions
2. Adding “enemies” to be avoided
3. Adding new types of resources
4. Adding more colonies + GUI option to do so
5. Make certain ants hostile to other ones
6. Enhance decision-making, based on smell and vision
7. Add direct communication upon stumbling
8. Add the lifespan or stamina simulation so ants will rest after a certain time

Challenges

The development of MAS was quite simple, but the fine-tuning of pheromone trails posed a significant challenge, because initially, the trails were not disappearing, and maximum steps were introduced, so they either succeeded or failed. Afterwards, I thought that in the real world organic things are not permanent, so I added the pheromone decaying system. In order to make the system show the results, I had to randomly pick out pheromone decay time, to avoid congesting the map.

Additionally, one of the hard development parts was visualisation. Initially, it was planned to use the MESA as the agent development library, and it had a built-in visualisation plugin, but for certain reasons, I could not get it to work. Afterwards, I tried to use a compatible visualisation plugin SolaraViz, which is Jupyter notebook-based. Consequently, the matplotlib as the most basic but useful library could handle the MAS visualisation.

Overall, the Ant Farm simulation using the Multi-Agent System is successful in demonstrating the basic flow of events in the scenario of a clean environment that has collectable resources. The application is reusable and has a GUI to configure most of the MAS parameters.

Conclusion

To recapitulate, this Multi-Agent System successfully simulates ants’ behaviour and competition using simple rule-based agents. The application integrates a 2-dimensional environment, pheromone-based communication, resource depletion and respawns. The agents, after setting some pheromone trails become more and more efficient in gathering the resources and navigating on the map. Despite multiple challenges along the way, the system features real-time visualisation and show-cases similar to real-world ant behaviour.