Multi-agent System based on an Ant Colony Behavior

Carolina Brás
Instituto Superior Técnico
Lisbon, Portugal
carolina.bras@tecnico.ulisboa.pt

Guilherme Pereira
Instituto Superior Técnico
Lisbon, Portugal
guilhermecnpereira@tecnico.ulisboa.pt

Miguel Belbute
Instituto Superior Técnico
Lisbon, Portugal
miguel.belbute@tecnico.ulisboa.pt

ABSTRACT

Our goal is to implement a multi-agent environment based on the natural behavior of ants, which have previously inspired algorithms to study and solve pathfinding problems. More specifically, We plan to simulate an ant colony with multiple ants (agents) whose goals are to search for food (points of interest). By analyzing how these agents indirectly communicate through pheromones and cooperatively define the best paths to travel between the food and the colony (home base), we gain insights on how we can transfer this algorithm to real-life applications.

KEYWORDS

AAMAS; Multi-Agent; Path-finding; Ant; ACO

1 INTRODUCTION

In the scope of the Autonomous Agents and Multi-Agent Systems (AAMAS) course, we decided to pursue a multi-agent environment with a high number of agents which communicate with each other and pursue simple objectives - somewhat mimicking a swarm/colony intelligence with decentralized, self-controlled systems. In this sense, the implied interactions of an ant colony were chosen to be studied in depth, due to the social behaviors of these insects.

1.1 Related Works

Ants have previously been the basis for topics of research. For example, Ant Colony Optimization (ACO) is a probabilistic technique regularly applied to path-finding problems, as showcased in "Shortest Path Problem Solving Based on Ant Colony Optimization Metaheuristic" by Mariusz Glabowsky et al [3] and "Solving Vehicle Routing Problem using Ant Colony Optimisation (ACO) Algorithm" by Wan Othman et al. [4] Moreover, ACO has also been applied to reach near-optimal solutions for the famous traveling salesman problem as in "TSP Solving Utilizing Improved Ant Colony Algorithm" by Xuan-Shi Yao [5].

1.2 Problem Definition and Objectives

Our proposed implementation aims to study how a group of agents (represented by ants) can find efficient ways to explore the environment for points of interest (represented by food) and commute between the home base (represented by the colony) and the aforementioned locations. In a broad sense, our goal is to explore the effectiveness of a multi-agent solution to path-finding problems, manipulating the behavioral variables and environmental settings of our simulation to investigate how agents will respond in favorable and unfavorable conditions. We can then infer how these findings could be transferred to some of the real-life applications.

2 APPROACH

2.1 Environment

Our virtual environment aims to approximate the characteristics of a real-life ant colony while incorporating certain modifications to facilitate the implementation and study of the proposed system.

2.1.1 General Description. The environment is represented by a 2D, square, grid-like map with variable size. Each component of the simulation, including colonies, ants, food piles, and pheromones, occupies a single square (or tile) on this map. A more detailed list regarding the simulation's elements is indicated here:

- Colony - a colony is represented by a single brown tile. Its location is randomly chosen at the beginning of each simulation run. Each colony also displays its current food supply (initiated at 100 units, by default), which gradually decrements over time (by default, 1 unit/step). Moreover, colonies are tiles which the agents can interact with: by performing "DROP_FOOD" near them, they can increase the colony's food supply (by default, 20 units). In its current state, only one colony spawns each run.

- Food pile - a food pile is represented by a single green tile. Its location is randomly chosen at the beginning of each simulation run. Each food pile also displays its current "food value" in white text (which, by default, is either 2, 4, or 6). This value can be decremented if an agent decides to perform "COLLECT_FOOD" near the food pile. Normally, multiple food piles are spawned each run.

- Ant - an ant is represented by a single black tile. The agent's initial location is randomly chosen at the beginning of each run (it does not necessarily spawn near a colony - we assume it was already exploring the environment when the simulation starts). A white text also displays the agent's id above the black tile. The ant is always surrounded by 4 yellow tiles which aim to represent its "field of action" - the tiles with which the ant can directly interact with. An ant which is carrying food will also turn purple. Since we are mostly focusing on a multi-agent environment, multiple ants are spawned each run.

- Pheromone - a pheromone is represented by a cyan tile. Its placement results from an ant's movement when carrying food. Each pheromone also displays its current intensity value in white text, which has an initial default value and can be incremented as

AAMAS Project, IST, 2023, Lisbon, Portugal

https://doi.org/

a result of agents walking over it. Furthermore, pheromones have a global evaporation rate which decrements their intensity values with time (by default, -1 unit/step). Normally, multiple pheromone tiles are created each run.

2.1.2 Characteristics. Like most real-life environments, several important features can be observed in an ant colony scenario - inaccessibility, non-determinism, dynamicity, continuity, and non-episodicity. The virtual environment differs from the real-life environment regarding the following properties:

- Determinism: in the virtual environment, actions performed by ants are designed to have expected outcomes. Each action has a predictable effect based on the defined rules and dynamics.

- Staticness: the virtual environment will not change while the agents are deliberating - the simulation will occur in a turn-based manner.

- Discreteness: the virtual environment is discretized, both in the way it's displayed - discretized in a grid - and in the sense that ants have a well-defined, finite set of actions they can undertake and perceptions they can experience.

2.2 Agent Architecture

Each agent in the system has access and interacts with multiple sensors and actuators which allow it to perceive and act upon the environment, respectively. The way the information received is conveyed into actions depends on the agent's overall architecture. In our case, 4 architectures were considered, but before going into further detail regarding each of these, we should explain both the sensors and actuators utilized.

2.2.1 Sensors. Each ant possesses a 5 by 5 tiles field of view, centered on their current position, which allows them to observe the environment and detect important points of interest (such as colonies, food piles, pheromones, and other agents) and the corresponding relative positions. This field of view also obtains complementary information when certain elements are in range: the colony's current food supply, food piles' food quantity, pheromones' intensity levels and the amount of food other agents are carrying. This data is later conveyed from the environment to each ant in the form of observations.

2.2.2 Knowledge. Apart from the aforementioned observations, ants hold some sort of "universal" knowledge related to their virtual environment - at every instant, they know their own global position and the colony's global position on the grid. This approximation is not the most life-like way of implementing ant agents but it allowed for further analysis opportunities by decreasing randomness. Efforts were being made to create an "unknowledgeable" version of the agents which did not account for the colony's global position. However, this idea was later scrapped due to time constraints. It is important to not the agents do not know the global position of any other structure (like food piles)!

2.2.3 Actuators. Each ant can perform 11 different actions, 9 of them related to movement. The first 4 represent the possible movement directions for when the ant is simply exploring or moving towards a destination - "DOWN", "LEFT", "UP", "RIGHT". Action number 5 corresponds to "STAY", where the agent holds its current position. Action 6-9 represent the possible movement directions for when the ant is carrying food and thus leaving behind a trail of pheromones - "DOWN_PHERO", "LEFT_PHERO", "UP_PHERO", "RIGHT_PHERO". Initially, the agents, whether carrying food or not, would leave a trail of pheromones behind them as they traversed the environment, similar to what happens in real life. The pheromones associated with having food would have a higher intensity value than the "normal" ones. However, after careful consideration and perceiving this as an added unnecessary complexity, we decided to disregard the "normal" pheromones. Finally, actions 10 and 11 correspond to "COLLECT_FOOD" and "DROP_FOOD", which can only be performed when the agent does not have food and is near a food source, and when it has food and is near a colony, respectively. Lastly, the ant's field of action is displayed as 4 yellow tiles surrounding its current position.

2.2.4 Decision-Making. As previously mentioned, 4 different architectures were explored for the ants' decision-making behavior: random agents, deliberative agents, reactive (greedy) agents, and collaborative agents. We will now explain each of these architectures in further detail:

- Random Agent - as the name suggests, this type of agent performs actions randomly and, therefore, is not expected to perform well in the virtual environment. However, it is implemented to offer a baseline for comparison with the other architectures. One key aspect to note is the following: if the arbitrarily chosen action is not valid at a given instance (like dropping food when the agent is not carrying any) the environment will not let the ant proceed and it will instead consider the "STAY" action.

- Deliberative Agent - mostly based on the Belief-Desire-Intention model (BDI), this agent can, from its perceptions, construct its beliefs (what it sees and what it knows), formulate 1 of 3 possible desires ("GO_TO_COLONY", "EXPLORE" and "FIND_FOODPILE") and finally work towards those goals by defining intentions. The first desire is quite self-explanatory - by default or whenever they have food, agents will travel to the colony. "EXPLORE" represents a way to "randomly" explore the surrounding environment in hopes of finding undiscovered food piles (in reality, in order to prevent a chaotic, totally random exploration which may yield few results, this desire is associated with a mechanism that arbitrarily chooses an action and keeps said action for a given amount of steps, after which another non-opposite direction is chosen to move towards. This increases the overall effectiveness of exploring the map). Lastly, "FIND_FOODPILE" encompasses both traveling to observed food sources as well as following intense pheromone trails. To determine which desire to choose between these last two, the agents verify the food supply of their own colony and, considering a given threshold, will "exploit" pheromone trails if the food gets too low and explore the map if it is at a comfortable level.

- Reactive Agent - unlike the deliberative agent, the reactive agent mainly reacts to and acts upon immediate stimulus from the virtual environment in order to accomplish its goals. With regards to the implementation, we mostly followed a set of condition-action rules with varying priorities. For example, if the agent detects having food, it will move towards the colony, this being the highest priority endeavor it can assume. If the agent does not possess food, it will instead check for food piles in view and if that condition fails, then it will resume any pheromone trail it might have been following. The lowest priority rule is to, naturally, explore!

- Collaborative Agent - in order to foster collaboration among the agents, a reduced movement speed was implemented for ants carrying the maximum amount of food (by default, 2 units). A collaborative behavior was introduced, wherein if a certain ant observes another carrying 2 units of food, it will take half of that amount. This cooperative action allows both ants to move faster and expedite the delivery of food to the colonies. For this architecture, 2 roles were defined: "GO_HELP" and "GO_WORK." In the former, the potential was determined as the negative value of the Manhattan distance between the agent's position and the position of the observed ant carrying food. In the latter role was used as a default with a higher potential than "GO_HELP" (when the agent does not observe any ant carrying food). The potential changes when the agent sees a food pile, becoming zero in that case. A new desire, "GO_HELP," was also added to support the food collection action of the ant carrying the maximum amount of food. When the agent's role is set as "GO _HELP," it will update its desire accordingly. Conversely, if the agent's role is "GO _WORK," it will continue to follow the pre-established loop of desires and actions already defined in the deliberative agents.

2.3 Multiagent System

With the aforementioned architectures, a multi-agent system was developed, exhibiting the following traits: communication (agents indirectly communicate through pheromones in order to coordinate their actions) and coordination (agents collectively examine unexplored areas and, once an agent identifies a path to a food source, it marks the trail, allowing other agents in the colony to follow it without redundancy or conflicts. On the other hand, in the collaborative agent scenario, two agents are able to transport the same amount of food in a more timely manner by helping one another).

3 COMPARATIVE EVALUATION

Considering all the previously mentioned characteristics of the implemented system, some tests were run to evaluate the effectiveness of the developed autonomous agents. These tests were separated into two distinct groups - the first one is aimed at studying the decision-making processes of multi-agent teams, while the second one is focused on the effect of environmental and behavioral parameters on overall performance. All the tests were performed on a 16x16 map, with a single colony and 4 food piles.

3.1 Decision-Making Analysis

For the first group of testing, 4 different teams were considered: Random Team (4 random agents), Deliberative Team (4 deliberative agents), Reactive Team (4 reactive agents), and Hybrid Team (2 deliberative agents and 2 reactive agents). These teams will henceforth be referred to as RT, DT, ReT, and HT, respectively. Each multi-agent team was tested in the same 100 maps (a different one per episode), in order to closely approximate evaluations conditions.

3.1.1 Heat Map Analysis. In this first study, 3 heat maps were generated per team: after the first episode, at the middle of the test run and at the last episode, each respectively represented by (a), (b) and (c). Each episode terminated as soon as 100 steps were reached or when all the food was depleted.

- RT: since the agents perform random actions at each time step, they show unorganized movement patterns, with no significant connections being accomplished between the colony and the food piles. This is corroborated by the presence of small hot spots throughout the grid, independently of episode number, most likely each corresponding to one of the agents of the team.

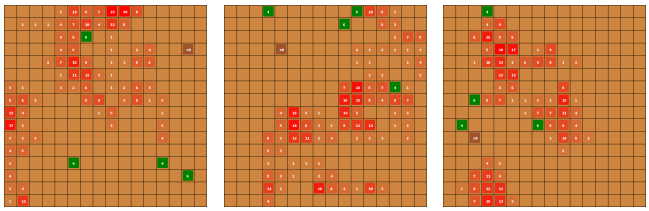

Figure 1: RT's heat maps for episodes 0, 50 and 100

- DT: in relation to the previous team, the deliberative agents explore a bigger portion of the environment, being noticeable in the multiple constructed paths connecting the colony to the food piles. One should also note how there is a significant concentration of "heat" near the colony. This is due to deliberative agents' implementation - recollect how, by default (whenever the agents do not have another desire), they ponder their next goal by traveling to the colony and verifying its food level, in hopes of balancing exploration and exploitation.

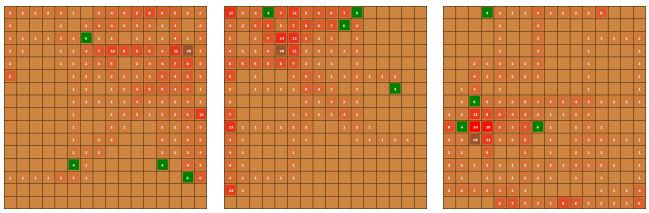

Figure 2: DT's heat maps for episodes 0, 50 and 100

- ReT: the reactive agents seem to share somewhat similar heat maps to the deliberative agents, although their exploration efforts are far lesser than the latter - their quick responsiveness favours immediate rewards (food piles close to the colony), thus not allowing a deeper exploration of the environment.

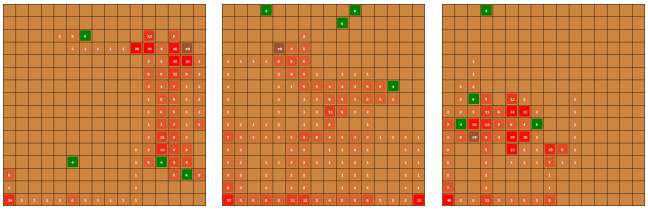

Figure 3: ReT's heat maps for episodes 0, 50 and 100

- HT: finally, the hybrid team also displays satisfactory results - the joint efforts of deliberative and reactive agents promote the discovery of the environment, whilst maintaining a certain urgency in exploiting food piles... Whereas the deliberative ants can focus more on exploration (discovering new food piles, possibly), the reactive agents act quickly to follow known pheromone paths (laid mostly by the deliberative agents) and raise the colony's food level in a timely manner.

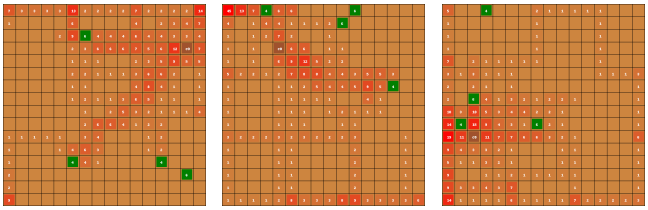

Figure 4: HT's heat maps for episodes 0, 50 and 100

3.1.2 Time Taken To Complete Environment. In this study, the simulation was run 3 times, each producing a graph that compares the time steps each team needed to complete the simulation, either by collecting all the food or by running out of time (reaching a maximum of 100 time steps). Observing Figure 5, the results for each team are quite similar - unfortunately, in most episodes the agents failed to collect all of the environment's food within the time limit. Despite this, an important remark can be made: on average, the reactive (greedy) agents managed to complete the environment faster than the deliberative agents (although only by a small margin). This does make sense if one recalls the more "planned-out"/"conservative" model of the deliberative agents, despite them seemingly exploring more, as we could tell from the results of the previous study. HT appears to be the middle-term between DT and ReT (as one would expect), but we cannot necessarily say it performs better than DT. By increasing the time limit to 200, we managed to achieve a graph (Figure 6) which confirms our initial conclusion - given a time constraint, RT performs better than DT.

3.1.3 Food Storage Evolution. This study aimed to analyze the colony's food storage evolution in order to once again get a grasp on the speed at which the different teams explored the environment and consequently gathered food. Supporting the last study's results, we can tell from Figure 7 that ReT managed to reach a higher food quantity (say, 130) sooner than the remaining teams. The highest recorder food level for ReT was also greater than the other teams - 160, in comparison to HT's 150 and DT's 130. Despite this, HT

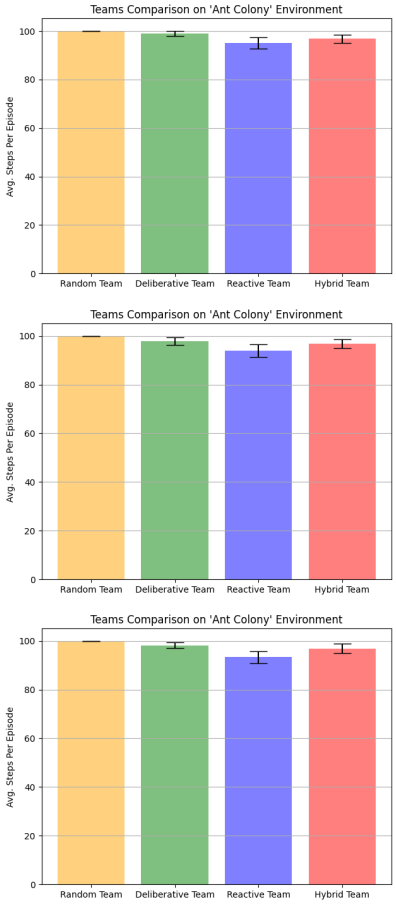

Figure 5: Team comparison regarding environment completion (100 steps, one of the runs)

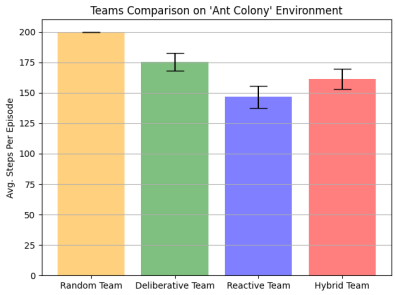

Figure 6: Team comparison regarding environment completion (200 steps)

managed to overtake RT after around 75 steps, while DT did so after around 85 steps, having both finished with a greater food storage level! Considering all this information, one can state that despite ReT's speed in gathering a considerable amount of food, the reactive agent's strategy is quite "volatile", being mostly limited by the immediate surroundings of the colony (which is explored in a

few time steps, hence the greater food collection speed and higher maximum food storage reached). After that initial area is explored, reactive agents have a harder time exploring the environment, similar to what is displayed with the heat maps. Although DT and HT took a longer period to gather food, they are clearly preferable options on the long term (and, extrapolating, considering a larger environment.) By incorporating the collaborative agents into the simulation, we also introduced the factor of reduced speed for the ants carrying the maximum load. This factor affects our final results, making it challenging to obtain reliable outcomes when comparing the different agent architectures against each other. To make a more accurate comparison, it would have been beneficial to include the same factor of reduced speed in the other agents as well. This would have allowed for a more comprehensive evaluation of the different agent architectures.

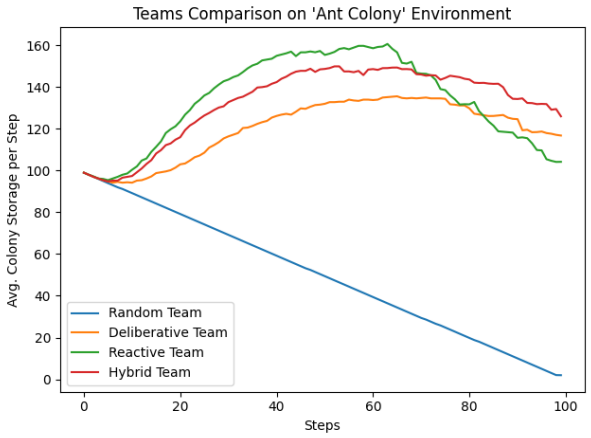

Figure 7: Average colony storage evolution during 100 time steps

3.2 Decision-Making Analysis

For the second group of testing, environmental and behavioral parameters were modified to analyze their implications on the overall performance of the agents. In the first study, the previously mentioned teams were used, whereas in the second one, only deliberative agents took part.

3.2.1 Effect of Pheromones' Evaporation Rate. For this study, the pheromone's evaporation rate was manipulated as a means to test its influence on the multi-agent system's balance between exploration and exploitation. The value of this rate was halved (-0.5 units/step) and doubled (-2 units/step), yielding the results below. One would expect that increasing the pheromone's evaporation rate would make it harder for the agents to indirectly communicate pathways to different food piles, thus enabling hot spots on the heat map (in other words, agents would have difficulties reaching the same food piles multiple times). However, the heat maps for the same agent type are approximately the same, which contradicts this hypothesis. Except for the ReT: reactive agents clearly have a higher tendency to explore the environment if the evaporation rate increases (notice how with the doubled rate, agents explore a much wider region of the map), displaying some sensitivity to this behavioral parameter. In the other agent architectures, this sensitivity is not significant.

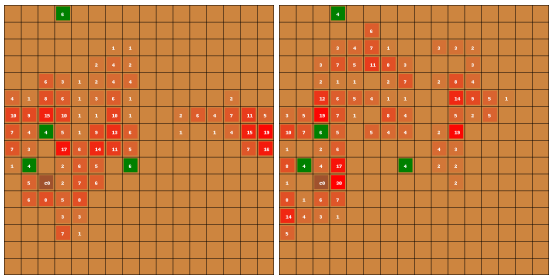

Figure 8: RT's heat maps, episode 100, evap. rate halved and doubled

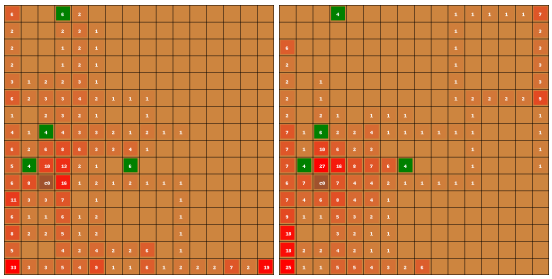

Figure 9: DT's heat maps, episode 100, evap. rate halved and doubled

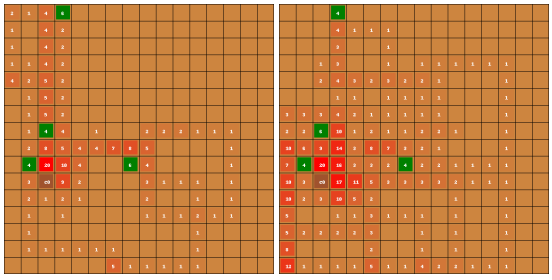

Figure 10: ReT's heat maps, episode 100, evap. rate halved and doubled

3.2.2 Exploration/Exploitation Threshold. Lastly, this final study was developed to examine the effect of the exploration/exploitation threshold considered in the deliberative architecture: as an ant approaches the colony, it verifies its food level and based on said value determines the best course of action - explore or exploit (any value below the threshold corresponds to the desire of actively looking for food piles and pheromones, whereas a value above this threshold induces exploration). This threshold's default quantity

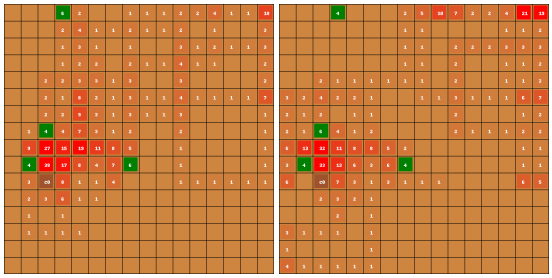

Figure 11: HT's heat maps, episode 100, evap. rate halved and doubled

(50 units in colony storage) was increased to 200 units and the results are explicit, besides expected: a lower threshold foments exploration and vastly expands the portion of the environment covered by the agents, as seen in Figure 12. We can also analyze the colony storage's evolution considering the different thresholds, as seen in Figure 13: allowing a higher threshold results in a more "conservative" behavior which, in actuality, hinders the colony storage's evolution - the maximum food level reached is lower, as well as the final food level. These results go to show the importance of balancing exploration and exploitation in the deliberative agent architecture.

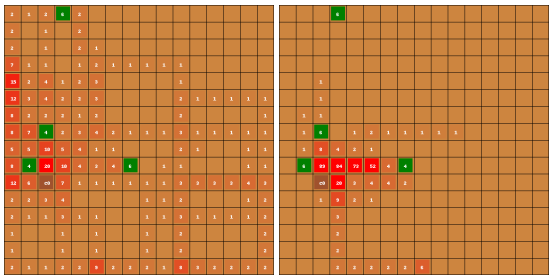

Figure 12: DT's heat maps, episode 100, normal and doubled exploration/exploitation threshold

4 FINAL THOUGHTS

To sum up all the different findings, we have verified that the efficiency of an ant colony multi-agent system in finding pathways to unknown points of interest is highly dependent on the architecture applied and contextual setting. Generally speaking, for faster results, one should opt for reactive agents, which prioritize immediate actions and quickly exploit a small area of the map. However, in applications that require longer periods of contact with the environment, deliberative agents or even a hybrid team are options to consider. If one would take computational time into consideration, then reactive agents would surely come up on top. That being said, another viable option could be an increased amount of reactive agents per team - this would improve the exploration area without losing the previously mentioned advantages of these agents. Lastly, although we were unable to compare the outcomes of the collaborative agents directly, we infer that the mutual assistance among

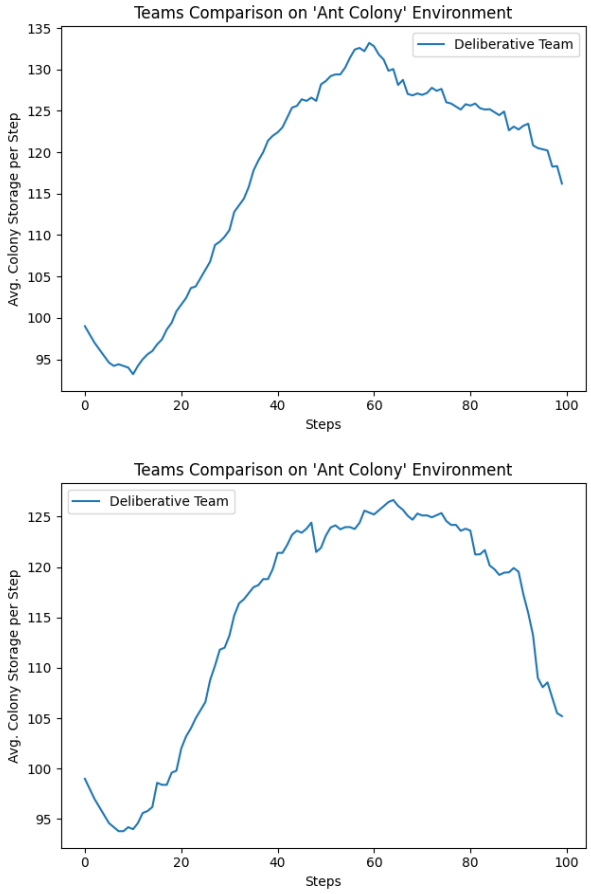

Figure 13: DT's average colony storage evolution during 100 time steps, with normal and doubled exploration/exploitation threshold

ants proves to be an efficient solution to the slower movement of the food-carrying ants.

REFERENCES

[1] [n. d.]. Ant Colony Optimization Algorithms. https://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms. ([n. d.]). Accessed: 2023-05-12.
[2] Jelmer Ast. 2023. Ant Colony Optimization for Control. (05 2023).
[3] Mariusz Glabowski, Bartosz Musznicki, Przemysław Nowak, and Piotr Zwierzykowski. 2012. Shortest Path Problem Solving Based on Ant Colony Optimization Metaheuristic. *Image Processing & Communications* 17 (11 2012), 7–18. https://doi.org/10.2478/v10248-012-0011-5
[4] Wan Othman, Aeizaal Wahab, Syed Alhady, and Haw Wong. 2018. Solving Vehicle Routing Problem using Ant Colony Optimisation (ACO) Algorithm. *International Journal of Research and Engineering* 5 (11 2018), 500–507. https://doi.org/10.21276/ijre.2018.5.9.2
[5] Xuan-Shi Yao, Yun Ou, and Kai-Qing Zhou. 2021. TSP Solving Utilizing Improved Ant Colony Algorithm. *Journal of Physics: Conference Series* 2129 (12 2021), 012026. https://doi.org/10.1088/1742-6596/2129/1/012026