Different Levels of Communications in Multi-Agent System

Zifan Nan, Wenxuan Zhu, Xiangqing Ding
Department of Computer Science, North Carolina State University
{znan, wzhu10, xding3}@ncsu.edu

Abstract

A multi-agent system consists of multiple interacting intelligent agents within an environment. Among the agents, communication could occur as a way of sharing information. The amount and quality of information shared may influence the agents' behaviors. A foraging model is implemented as a platform to figure out the influence of communication on the behaviors in the system and whether it earns its complexity.

Introduction

In multi-agent system (MAS), there are several agents attempting to jointly solve tasks to maximize the system's performance through their interaction [1]. Communications may occur among the agents for sharing information. And the amount and quality of information may influence their behaviors. From one perspective, more information shared allows agents with more knowledge of the environment. As a consequence, the agent could make more considerate decision. From another perspective, more information doesn't necessarily mean better performance. Redundant information could also lead to a waste of storage and processing time. What should be focused is measuring the performance of different communication strategy and whether it earns its complexity.

In this project, a multi-agent system is implemented as an evaluation platform. With goals and different levels of communication assigned to agents, their behaviors and performance would be analyzed to evaluate the influence and complexity of different communication strategies. More specifically, the platform is designed based on the foraging problem, which is a representative multi-agent problem. In the foraging problem, the basic tasks for agents include searching the unknown world and transporting food from food source to home. And there are some goals such as less time cost or less energy efficiency [2].

In many current games, different difficulties are achieved by adjusting environments attributes, such as enemy's status and amount of resource. However, this paper is aimed at changing the game difficulty by changing agents' communication strategies. In multi-agent system, from no information sharing to total information sharing, each agent would behaves different based on the information they owned. This method can be used in many game with multiple agent and where team cooperation and strategy are needed.

Besides, by evaluating the performance of different level of communications, it could be found out which level of communication is the most appropriate under certain environments or situations, and what is the advantage and disadvantage of each level respectively. Furthermore, from the perspective of foraging, studying how agent behaves in different situations may provide some useful approaches for multi-agent foraging problem.

Model Design

To evaluate the performance of different communication strategies, a foraging model is designed as the platform. Some details of the model would be specified in this section.

Entities

Based on the foraging problem, there are three fundamental entities in this model: food sources, foragers (agents) and hive.

Food source is a place which foragers collect food from. It would have two basic attributes: location and remaining amount of food. And there would be many food sources on the map.

For the foragers, their tasks basically include searching food sources on the map and transmitting the food to the hive. In addition to the basic status, they have attributes like capacities, energies etc. Certainly, with a higher communication level, it would communicate with companions.

Hive is the place that all foragers start from and store food. Only one hive is in one map.

Tasks and Techniques

The whole implementation is split into two parts: environment implementation and agent implementation. Environment implementation include world generation, world representation and environment manager. Agent implementation include character behaviors, data structures, decision making and communication strategies.

Unity

The famous game engine, Unity, is the main tool used in the platform development. For the most important reason, Unity has many useful tools for game development, such as NavMesh (Navigation Mesh) agent and well-developed physics engine, which help through the development, debugging and testing of this project. Furthermore, many free models from the asset store are utilized to improve the appearance of the platform.

World Generation

First of all, the game world should be constructed. The basic tasks of world generation mainly include distribution of food source and obstacles. The hive is generated on a fixed location.

The world is generated by script, also with some manual adjustment. It looks like a corner of a room with different obstacles and food sources randomly distributed on it (*Figure 1*). The numbers of obstacles and food sources could be changed by changing the parameters in script. Unity currently doesn't support generating navigation meshes at runtime, so the world generation is separate from running of the platform.

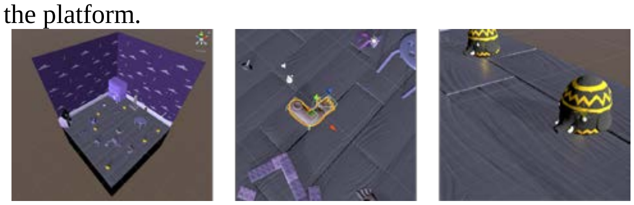

Figure 1: World, obstacles and food sources

World Representation

To ensure the movement of agents, one method is using collision avoidance methods combined with other moving method. However, there could be many potential problems, such as stuck when near the wall.

Another method is using division schema for world representation. Unity provides NavMesh agent and abundant APIs, which includes creating navigation meshes on the environment. After the world is generated with obstacles and hive set as NavMesh obstacles, the NavMesh is baked and generated (*Figure 2*).

Figure 2: NavMesh generated by Unity

Environment Managers

There are two mangers implemented in the platform, forager manager and game manager, which is two different scripts.

Forager manager is responsible for generating foragers at the runtime. The number of foragers in the platform is determined by it.

Game manager is a script that control the end of the game. In addition, it records important evaluation data such as time and amount of food collected, and displays some of them by UI component (*Figure 3*).

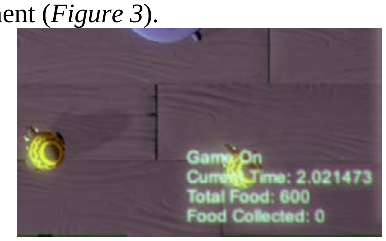

Figure 3: Data displayed

Basic Movement

Movement is the basic behavior of agents, not simply arriving behavior since there are obstacles in the game environment. In Unity, the navigation system allows agents move to one specific position by using *NavMeshAgent.SetDestination(position)* method. This method also ensures that agents move to target position in shortest path, which is very convenient and used in this platform.

Exploration

In the beginning of the game, agents are not aware of food source. So, in addition to the basic movement behaviors, they should move around to discover food source, which is a kind of wandering behavior without returning to the place where it has searched.

Without well-defined algorithm for the exploring behavior, an original exploration method is invented. First, the map is quantized into a 50*50 matrix divided into four equal sections (*Figure 4*).

At the start of exploration, the agent would randomly choose a unvisited point in one section as destination. While the agent is going to the destination, the points of the matrix that agent go through will be deleted by setting the value of that point in matrix to 0. As agent reaches the destination point, another unvisited point in other three sections will be chosen randomly. This is simply one pattern of exploration, which enforce agent explore the map widely. The parameter of this method is also adjusted based on the size of agent and detection field.

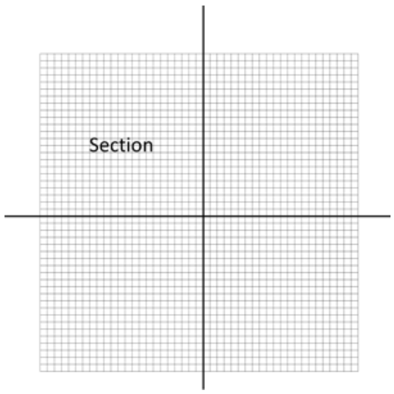

Figure 4: Exploration Quantization

Detection field

The detection field of an agent is the area that the agent would detect other entities such as food source or other agents. Usually, in a three-dimension environment, the detection field is a cone area. However, in this implementation, the field is a sphere centered with agent (*Figure 5*). Because the sphere field is easy to implement and increases the interaction between agents and other entities.

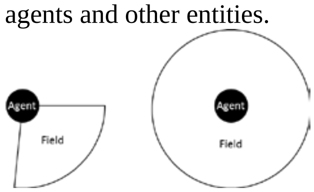

Figure 5: Cone area vs Sphere area (from vertical view)

One method to create a detection field is building a detection manager that continuously checks the distance between each entity. However, an easier way is discovered in the process of implementation. Unity provides many game physics tools, one of which is collider. User defined events would be triggered if one collider enters or exists another collider. Thus, one sphere collider (detection field) and one capsulate collider (body field) is assigned on each entity (*Figure 6*). For example, when the body field of entity A hits entity B's detection field, A would be put into B's list that store the detected entities. Also, it would be removed from the list if it leaves the detection field.

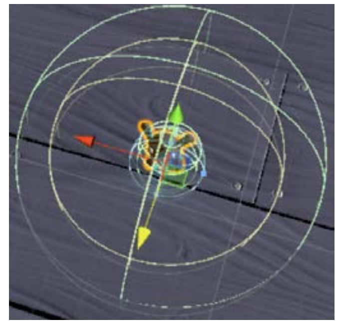

Figure 6: Detection field implemented by collider

Information Structure

Information is a set of data shared and stored in the agent. To decide its structure, which data are in what form should be considered. Basically, information in this model includes two types: food information and forager information. The food information contains the position and the remaining amount of food source, while the forager information contains the remaining capacity and current actions of each agent. They are implemented as two classes and each food source or agent is assigned with one instance of them. In another word, it is the information objects that stored and transferred among agents.

Apparently, there could be conflicts between information of two communicating agents, for example, different remaining amount of the same food sources. One important solution and feature in this model is that all information has a version, which is an integer starting from 0. Once the information is updated, the version will increase by 1. This feature ensures that newer information would replace older information in the process of communication.

Information Storage

For the information storage, there are two list of food sources information and a list of forager information in each agent. All the lists are empty at origin. Once a new information is obtained by the agent, the agent would process it and add it to corresponding list (*Figure 7*). One list of food information is called open list, which stores the information of food sources that haven't run out. And once a food source runs out, its information would be removed from the open list and added to the closed list. The advantage of using two lists of food information is that agent just needs to iterate the open list to find the which food source to collect food. And both lists will be shared when agent communicate with other agents, then agents could know which food sources are empty and avoid them. And when combining the lists with lists from other agents, agent checks the version of each information and keep the newest one, which is with larger version number.

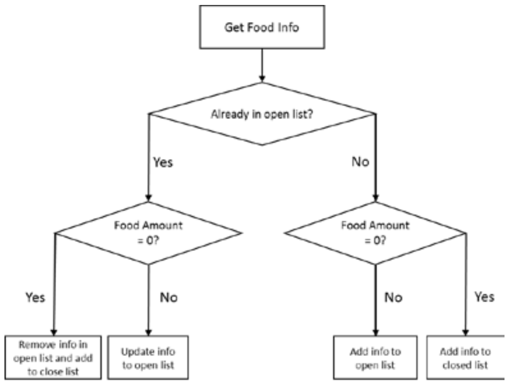

Figure 7: Processing new food information

Rules of Choosing Target

In this model, agents should make choice in many aspects. Since an agent could have information of different food sources, it should decide which to collect food from. In addition, if agents should positively communicate with other agents, it must decide whether to stop current work and which agent detected to communicate.

Choosing food source

When agents have multiple food sources to choose from, they need a strategy or heuristic to decide. Simple ones could be choosing the furthest or closest, or pick randomly. More complex ones could involves communicating with other agents and eliminate some options or decide on one together, or some combination of distance and the amount of food. All the strategies implemented and compared are listed below (*Table 1*).

Name	Description
Earliest	pick the earliest discovered food source
Nearest	pick the closest food source
Furthest	pick the furthest food source
Random	randomly pick one

Table 1: Strategies of choosing food sources

Choosing forager to communicate

In this model, the agent will communicate with the nearest agent in the detection filed and the agents should not be in the process of collecting food. Also, the communicating agents will be marked so it would not be continuously chosen. The mark would be removed after leaving the detection field.

Communication

Communication between agents is the core of this project. The communications are classified into three levels of communication in terms of amount and quality of information shared. In the first level, there is no information shared by the agents. In the second level, agents share the most useful information such as food locations with each other. In the last level, in addition to the information shared in the second level, agents could know what other agents are doing, which is a less important information used in decision making.

Furthermore, the type of communication would also influence agents' behaviors thus should be considered. There are two types of communication between agents. One is that agents can send or receive information everywhere, i.e. no distance limit. Another type is that communication only occurs between two agents that are within certain distance. In the second type, agents should take whether to communicate with companion into consideration.

Communication Process

First of all, in this model, one time of communication occurs between only two agents. In the beginning of communication, each agent would select the nearest agent in the detection field as communicating target. It would check whether the selected agent is communicating with other. If the selected agent is free, the agent would start communication without permission from the selected target. After the communication is established, the information in storage will be updated based on the information from another side. Then the agent would end the communication of both side and remove the target from the list that stores the agents detected. As a result, the agent would not continuously choose the same agent as the target to communicate.

Level 1

In level 1, there is no communication among agents. From test on this level, the results reflect the performance of original foraging model.

Level 2

In level 2, communications occur among agents and agents would share information about the food sources they discover with other agents. Both communicating agent will get two lists of food sources from each other and updates the lists with the ones they own. Since the location of food source is the most important information in the foraging model, the result would reveal how important information would influence the performance of communication.

Level 3

In level 3, amount and types of information shared increase, that is, agents will share their status and action with other agents. The status of other agents would be stored in the list of foragers if the agent is aware of what the target is doing. The agents' status would be used to decision making. Compared with location of food source, status of other agents is less important information. Then combined with the test results from level 2, it would be found out how the amount and quality of information shared would influence the performance of communication.

Communication Type

Usually the communication occurs between agents within certain distance. However, there could be no-limit communication, that is, agents could communicate everywhere.

This is like the level 1 with all agents sharing the same information storage. To achieve this, the open list and closed list in information storage are changed into class variables. Thus, all agents use the same open list and closed list.

State Machine

State machine is used in the implementation to control the switch among behaviors of agents. And each agent could have many states, such as searching state and collecting state.

Level 1

In the beginning of level 1, agents have no information of current environment and starts with searching. The agent would explore the environment until food source is found. If a food source is found and stored to the open list, it moves towards the food source and changes into collecting state. After its collect enough food, that is, capacity is full, it would deliver food home. But if the capacity is not full and food sources runs out, the agent would return to status of searching new food source (*Figure 8*).

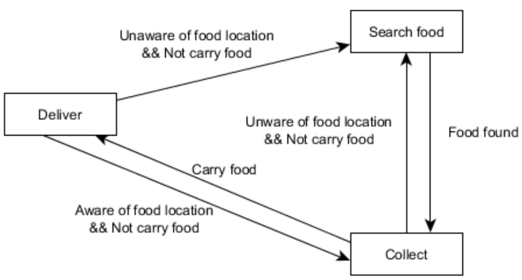

Figure 8: States Machine in Level 1

Level 2

From level 2, communication occurs among agents. In addition to the status in level 1, when an agent encounters other agents, it would turn to communicating with them (*Figure 9*).

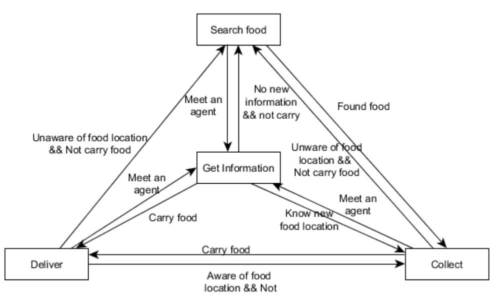

Figure 9: States in Level 2

Level 3

In level 3, now agents are aware of other agents' current actions. This is very similar to the state machine in level 2, except now agent would choose the food source also based on other agent's action. For instance, if A knows B is collecting one source, it would avoid collecting the same one, which is determined by the strategies.

Figure 10: States in Level 3

Evaluation

Questions Identified

Several questions about the system have been identified:

1. How does different information-sharing strategies affect foraging in terms of collected food (per minutes)?
2. Does environment affect the performance of each strategy?
3. Does sharing more information necessarily help find food faster?
4. Where is the balance between sharing too much information and too little information?

Evaluation Metrics

Time Usage

Instead of real time, logic time, unit of which is one running of update method, is used to represent time usage. The purpose of using logic time is reducing the influence of different machines on testing results. Furthermore, it would be more accurate than real time, for the graph processing time may vary.

The total time of collecting all the food and delivering to the hive is a criterion to evaluate the time efficiency of different communication strategies. And the total time usage of each state (e.g. searching, collecting) would also reflect the performance.

Energy Cost

First of all, each state is assigned with different energy cost (*Table 2*). Change in the cost assigned would influence the result. And since delivering is that agents go home with an amount of food, it would cost more energy. In the meanwhile, the cost of searching and collecting is the same.

State	Searching	Collecting	Delivering

Cost	10/unit time	10/unit time	12/unit time

Table 2: Behaviors and corresponding cost

The total energy cost is defined as follow and will be used to evaluate the energy efficiency.

$$Energy = \sum (state\_total\_time * cost)$$

Total cost

For each level, the time cost and energy cost will be recorded. Then all the time cost and energy cost will be scaled to [0,1]. The performance of each level is evaluated by the sum of energy cost and time cost.

$$Total\ Efficiency = Scaled\_time + Scaled\_energy$$

Expectation

Communication provides agents with alternative way to gain knowledge and cooperate. Thus, it is expected that the agent would behave better compared agents without communication. However, communication might be expensive in practice and reducing the amount of communication is often desirable. Considering those factors, the level with best performance might be level 2 or 3. The performance of level 1 is supposed to be worse than the other levels, since there is no communication and the agent must find food itself. Although, there has been cooperation strategy that doesn't include any communication and uses environmental information to cooperate like cooperative construction [3].

Result

Every combination with two different maps (with same amount of food, different distribution and of obstacles and food sources), four food selecting strategies, three communication levels and four different number of agents (3, 5, 10, 20) are tested in the experiment. In each test, the time usage, energy usage in each state are collected as raw data. Each test case would be executed for three times, using the mean value as data for evaluation and analysis.

Influence of Communication Levels

Influence of communication levels on the performance of agents is the main point of this project. To see how information sharing influence the performance, the map, number of agents (5) and food choosing strategy (Nearest) keep unchanged (*Figure 11*).

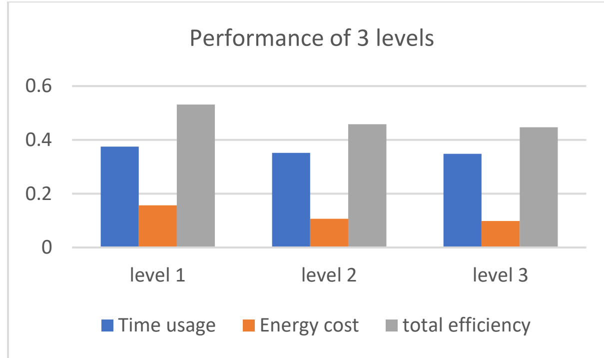

Figure 11: Performance of different levels

From the figure, it can be found out that, with higher level, i.e. with more information shared among agents, the time usage, energy cost and total efficiency decreased. This means with more information shared among agents, it takes less time and energy for them to collect all the food, as what is expected. And from the slopes of the figure, it can be seen that the difference between level 1 and 2 are larger than it between level 2 and 3. It can be explained in terms of the importance (or quality) of the information shared in the communication. In the foraging problem, the location of food sources is more significant than the status of other foragers. In conclusion, with the more important information shared, the influence of communication on agents' behavior becomes larger.

Time Usage of Each State

During the first round of test, there is an interesting phenomenon that agents spend a lot of time trying to get to the position with no food in level 1. Thus, it is necessary to record how much time an agent spend in each state. The state machine is almost the same as those in each level, with minor difference in state transition conditions.

The setting of the test case is as usual, five agents, Nearest food choosing strategy and the same map. The intuition of this test is finding out if, with more information shared, the time spent on searching food would decrease with higher level since the agent can obtain the location of food from other agents. However, opposite from the expected outcome, the search time slightly increases as more information is shared (*Figure 12*). One possible explanation is that, as agents know what other agents have, they will go to the closest food source to them first. Then they would miss the further food sources when collecting. Therefore, the time spent in search state increases while time spent on collecting known food sources is reduced.

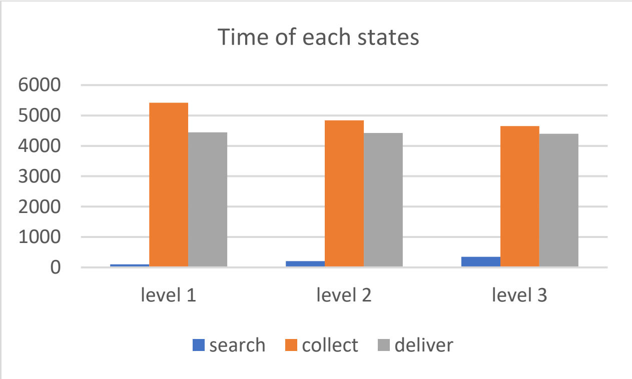

Figure 12: time of different states in each level

Influence of Maps

For the purpose of finding out whether environment, more specifically, the distribution of food sources and obstacles have influence on the performance of agents, the two maps with different distribution of food sources and obstacles and the same number of food and agent, is tested in 3 communication levels. One group of results is shown in table below, with 5 agents and Nearest food choosing strategy and level 2 in each map (*Table 3*).

Map	time usage	energy cost	total efficiency
1_original	1895.3	103562.6	0.4583
1	966.3	52758	0.9190
2	1070.7	58436.7	1.0258

Table 3: Performance with different food location

Assumption for this experiment is that the similar environment with same food amount but different distribution of obstacles and food sources would have slight influence on the performance. However, the original result data, i.e. the data collected in the first map seems quite different with the data collected in the second map. The reason could be that two different machines are used in experiment to run the tests with 2 maps separately, and the result is affected by the difference between machines.

Then again, the same machine is used to run the test cases for several times, and the result shows that the difference between the influence on the performance of agents of the two maps is small, which support the assumption.

Influence of Food Choosing Strategy

There are four different food choosing strategies in this model. The number of agents in this experiment is 5, while communication level is 2. It is aimed to find out the influence of different food choosing strategies (*Table 4*).

Strategies	Time Usage	Energy Cost	total efficiency
Earliest	2060.3	111845	0.3904
Nearest	1895.3	103562.7	0.3519
Furthest	2933	155475	0.5938
Randomly	1939	103239.3	0.3621

Table 4: Performance of different food choosing strategy

From the result, it can be found out that choosing the nearest food source has the best time efficiency and total efficiency. And random choosing costs less energy. In the meanwhile, choosing the furthest food would cause the worst performance. To find out why the nearest and furthest has such a huge difference, time use in each state is compared (*Table 5*).

State	Search	Collect	Deliver
Nearest	207	**4841.6**	4423
Furthest	113.3	**10109**	4437.7

Table 5: Time of each states

From the result, it can be found out that the main difference comes from collecting state. The furthest choosing strategy has lower time usage in searching, which is because when finding the furthest food, the agents have more chance of detecting other food. As for the larger time usage in collecting, the explanation could be that the food in long distance are more possible to be collected by other agents as the agent going the food sources. That will waste a lot of time compared with going to the nearest food source. As a result, choosing the furthest food will cost more time on collect behavior.

Influence of Agent Number

Since communication is a kind of cooperation behavior, it may be influenced by the number of agents, or more specifically, the density of agent's distribution. For example, if there are too few agents, they can hardly communicate with each other, then the communication is useless. To evaluate the performance of different numbers of, the test data on the first map, with nearest choosing strategy and level 2 communication is chosen to analyze (*Figure 13*).

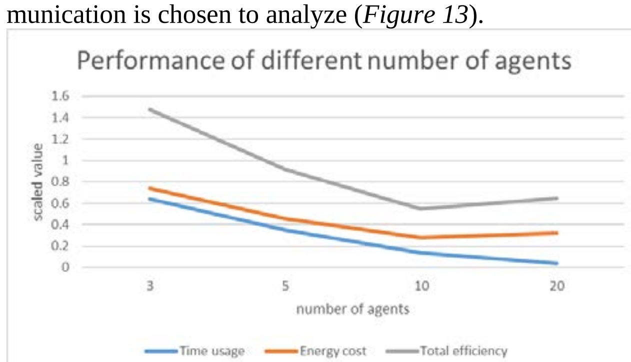

Figure 13: Performance of different agents numbers

As it can be seen from the result, more agents improve performance to some extent but doesn't necessarily always means better performance. That may be because with too much agents, even though the time cost is lower, the energy cost will be larger with the increase of agents' number. And there are a lot more agents running around, which gives higher possibility of wasted behavior, e.g. going to the food location where food is taken by another agent. If there are few agents, both cost will be bigger, because it takes longer to search and collect all the food.

Conclusion

Swarm intelligence, which is the collective behavior of decentralized and self-organized system, has been an attractive topic for many years. Based on foraging problem, which is a representative subject of swarm intelligence, a basic communication model is established specifying when and how agents communicate. Different communication strategies are camp up with and put into the model to find out the difference when amount and quality of information shared change. In the process of implementation, insights about multi-agent system is gained and other interesting approaches may be come up with and implemented.

Compared with the initial proposal, the focus of the project is switched into the communication process and comparison of different levels of communication in multi-agent system. One important reason is that it is hard to conclude that whether decentralized system or centralized has better performance. With different method and different strategies, both would behave different. Focusing on the communication allows deep study of the field.

From the analysis of the data collected, it can be concluded that, under current implementation and parameters, the performance of agents can benefit from sharing more information. In addition, different source selecting strategy would influence the performance of communication, and maybe there exists better strategies not used. And there is tradeoff between too many agents and too few agents. Emphasizing on current implementation and setting means that it is not guaranteed that the agents would behave better when parameters change. After all, it depends.

Future Work

Most of details of the project is designed by the authors. However, there are many related work that could be referenced. For further improvement on this project, deeper research into the field is needed [4, 5].

Manually testing and collecting data is quite time-consuming and the result is affected by that. Maybe with the aim of only collecting data, a program without graphic display would be considered. And there is more analysis that could be done, like memory usage of the communication.

Another interesting study direction is to look into theory of mind and see if an implementation without communication could be come up with.

References

[1] Panait, Liviu, and Sean Luke. "Cooperative multi-agent learning: The state of the art." Autonomous agents and multi-agent systems 11.3 (2005): 387-434.

[2] Zedadra, Ouarda, et al. "Multi-Agent Foraging: state-of-the-art and research challenges." Complex Adaptive Systems Modeling 5.1 (2017): 3.].

[3] Justin Werfel, Kirstin Petersen and Radhika Nagpal. "Designing Collective Behavior in a Termite-Inspired Robot Construction Team" Science. 14 FEB 2014: 754-758

[4] S. Garnier, F. Tache, M. Combe, A. Grimal and G. Theraulaz, “Alice in Pheromone Land: An Experimental Setup for the Study of Ant- like Robots”, Proc. of IEEE Swarm Intelligence Symposium, pp. 37–44, 2007.

[5] Ho NR III, Sago A, Wood RJ, Nagpal R (2010) Two foraging algorithms for robot swarms using only local communication. In: International conference on robotics and biomimetic (ROBIO). IEEE, pp 123–130