# Analysis and Application of Trophallactic Network System in Swarm Intelligence

*Minchul Kim*

The Graduate School
School of Electrical and Electronic Engineering
Yonsei University

February 2020

# Analysis and Application of Trophallactic Network System in Swarm Intelligence

A Dissertation
Submitted to the School of Electrical and Electronic Engineering
and the Graduate School of Yonsei University
in partial fulfillment of the
requirements for the degree of
Master of Philosophy

Minchul Kim

February 2020

This certifies that the dissertation of
Minchul Kim is approved.

Thesis Supervisor: DaeEun Kim

Seong Lyun Kim

Euntai Kim

The Graduate School
School of Electrical and Electronic Engineering
Yonsei University

February 2020

# Abstract

Many researchers have observed numerous instances of clustering in nature. Many creatures that have no central leadership still cluster for a variety of reasons. Individuals require information to determine their behavior. However, if the size of the cluster grows large, global information is not accessible. Thus, clustered entities obtain information locally to determine their actions. This is called self-organization in cluster intelligence. Self-organization is facilitated by direct and indirect information. In indirect communication, the individual creature uses environmental changes to understand its current state and determines its actions accordingly. On the other hand, direct communication is a method of obtaining information using various sensory organs. In this study, we examine trophallaxis, one of the direct communication methods.

Trophyallaxis is typically the method typically used by ants and bees. In the trophallactic network system, individuals repeatedly bring food, which is passed from one mouth to the other until every individual is filled, from the outside. This system does not end with simply sharing food. As the creatures share their food, they obtain other information they need. Thus, although ants do not have access to global information, their activities appear to suggest they possess it. In such system, the entity bringing the food from outside is called a forager, while the one in the nest receiving the food is called a non-forager or recipient. Among real ants, the foragers deliver a random variable based on the exponential distribution. Based on this fact, we conduct robot simulations in which information is shared locally; however, we identify relationships with global information and analyze the various dactors determining the global actions selected by real ants.

We observe the robot simulations through various distributions, find the advantages of the exponential distribution, and identify the best behavior for the foragers. Beyond simply sharing food, trophallaxis involves one-to-one exchange between individuals. We implement this in a multi-robot system by creating a situation in which robots clean up pollutants, and do not end up merely sharing pollutants, but find the best behavior of the recipients of the information when they deliver it. We provided the location of the pollutant so that recipients of this information could move to that location and receive the pollutant smoothly. Although this information is conveyed one-to-one, the recipients are eventually observed moving in groups. This global action helps them to efficiently eliminate the contaminants.

FInally, we verify the effectiveness of the algorithm deployed in the robot simulation by creating a cluster robot using a vibrating motor to implement a system that removes contaminants and replicating the experimental set-up featuring in the robot simulation using a real multi-robot system. In this system, we divided the trophallactic network system into two categories, foragers and non-foragers, to demonstrate that every individual in this system, regardless of the category they belong to, can achieve optimal behavior through one-to-one information exchange. Our findings confirm the exchange of location information, the clustering, and the resultant increase in the number of interactions has a good effect on the swarm system.

# Table of Contents

# List of Figures

# Chapter 1

# Introduction

In recent years, there has been increased interest in multi-robot systems. Many researchers have proposed a variety of methods of controlling multi-robot systems, and accomplish desired tasks. To perform various tasks with multi-robot systems, it is necessary to exchange infromation. The efficiency of robots performing the same task is enhanced through the exchange of necessary information or objects, depending on the task. This study examines the way robots in a multi-robot system exchange information. Basically, all the robots give information to the central computer. The central computer then organizes the information, and sends it out as global information, which determines the next action. However, extensive research has been conducted on the possibility of exchanging information in a multi-robot system without using a central computer.

In a local exchange, it is important to know the information being exchanged, and the robots sharing it. To identify the robots exchanging information, topology methods (Ren and Atkins, 2007; Ren and Sorensen, 2008; Ma et al., 2016) are useful; usually, neighbor-to-neighbor interaction is observed (Schmickl and Crailsheim, 2006). Once the robots to exchange information are determined, the question of what information to share and how to share it arises. The task to be performed determines the information required. The success of the robots' task is hinged on the selection of the information or object to be exchanged and the robots to selected to share them. We consider how the multi-robot system can benefit from the swarm it forms. We attempt to solve these problems through using a bio-swarm.

## 1.1 Trophallaxis as communication

The inspiration for swarm intelligence was drawn from social insects. Despite having no central control, many social insects perform a variety of actions in a coordinated manner. Through this behavior, many social insects derive various benefits for different purposes. In swarm intelligence without centralized control, local communication is an important factor. Self-organization is used to determine behavior through information obtained through local communication. Of the various communication methods, we explored trophallaxis, a form of indirect communication. Trotrophallaxis is a way of sharing nutrients with each other. The nutrients are passed from mouth to mouth. However, It does not end at simply supplying nutrients.

By exchanging nutrients through trophallaxis, information on the total amount of nutrients needed for the nest can be indirectly obtained through the nutritional status and the duration of the exchange. This allows foragers to fetch as much nutrients as they desire. In a one-to-one exchange, executing the desired task brings great benefits to the swarm. This is because going outside is very risky. It also means protecting the nest. Further, social insects use trophallaxis, because performing the desired task ensures that nutrition is provided to those who cannot go out. To perform this task efficiently, a network is formed by delivering various information as well as nutrition, thereby reducing minimizing unnecessary movements.

## 1.2 Motivation and Objective

In this dissertation, within the concept framework of swarm intelligence, we consider the significance of trophallaxis as a means of communication and its benefits to the swarm, which we then harness, and by applyt to multi-robot system applications. Many social insects and animals share nutrients among themselves (Wilkinson, 1984; Isaac, 1978; Hölldobler, 1985). The ultimate aim is to ensure the survival of the entire community, not just the individual's. Social insects reside in colonies. Within this construct, the aim is to defend the colony against threats and store food. However, food must come from outside the colony. Because food is sourced from dangerous environments, not all members find their own food; the task falls to some select members. Thus, there are a variety of rules and methods to minimize the risks and ensure the task is performed more efficiently (Bird et al., 2002).

Among these social insects, bees and ants use the trophallaxis to exchange liquids food. Trophyallaxis is a way of storing liquid foods in the stomach, and delivering them from mouth to mouth to other agents in the colony. This process is described in detail in Chapter 2. Observing trophallaxis among these social insects raises many questions. In the case of bees, the waggle dance reveals the amount, direction, and distance of the food. The waggle dance that is associated with trophallaxis is a method through which the foragers that bring food into the colony communicate (Riley et al., 2005; Thom et al., 2007; Wenner, 1962). The number of waggle dances depends on the amount during trophallaxis (De Marco and Farina, 2001). Furthermore, the ants control the time and number of exchanges during the trophallaxis, and bring the exact amount of food needed in the colony. The implication of this is that besides passing on food, the creatures share information necessary to performing their tasks more efficiently.

Trophallaxis is not limited to delivering food. Several experiments have confirmed that trophallaxis often involves sharing information as well. We are interested in what information these social insects disseminate through trophallaxis and how they modify their behavior accordingly. Trophallaxis can be seen as an act of sharing information as well as sharing nutrition. This study is designed to verify trophallaxis can be deployed for performing global tasks in multi-robot systems. The trophallaxis is basically aimed at feeding. In other words, in a multi-robot system, the basic purpose is to be able to share some material with each other. Each agent also shares the necessary information for efficient sharing. We observe and analyze what information is shared and the modification in the behavior of each agent based on this information.

## 1.3 Organization of dissertation

This paper dissertation consists of six chapters. Chapter 1 is the introduction; we explain the scope of the study, and the motivations and objectives. Chapter 2 is the background of the study. Our research is based on swarm intelligence. Swarm intelligence refers to simple agents that perform global tasks by self-organization through local communication without central control. Thus, efficiency in global tasks depends on how well self-organization is performed.

Local communication between agents is important for self-organization. When information is obtained through local communication, the agent moves based on the information. Thus, from each agent's point of view, they seem to be free to act, but they are moving for one purpose. This article examines the emergence and current applications of swarm intelligence, and explains the research requirements and direction. Further, we provide a background on the trophallaxis. We will examine the definition of trophallaxis and present the state of knowledge on it, its current application, and the direction of future research.

Chapter 3 is the analyzes analysis of various aspects of robots based on actual trophallactic network system. We define ants that bring food from the outside as foragers and those that stay inside the nest as non-foragers. In ants' trophallactic network system, the exact amount of food needed by the non-foragers is brought inside. In other words, when the nest is full of food, the foragers no longer go out. Thus, the system goes beyond merely delivering food. In the process of delivering food, information and feedback necessary for self-organization are obtained. There are studies that have progressed and analyzed the behavior of ants within this context (Robinson et al., 2009; Greenwald et al., 2015). We analyze the factors required by ants to achieve their goals most efficiently in a trophallactic network, and implement them in a multi-robot system. In this chapter, we focuse on the behavior of foragers. From the actions of the foragers, the reasons for the different methods the ants have chosen to achieve their goal can be deduced; this suggests that ants can execute global tasks efficiently by altering their behavior as required.

In Chapter 4, we examine how well a global task can be performed when the trophallactic network system is extended to convey information rather than simply supply energy. In this chapter, a new global task is created. The global task aims to remove all the contaminants in a contaminated area. In certain areas, these pollutants can be cleaned. Therefore, the pollutants are brought from outside to the clean areas and cleaned up. The robots going out are referred to as foragers, and those receiving the contaminants in the clean area are the recipients. The task of each robot is to deliver contaminants and exchange additional information. The additional information is the location information. In Chapter 3, we observe the trophallactic network system from the perspective of the forager. However, in Chapter 4, we interpret the view of the recipient. Recipients move in groups with the location information. We explain what happens when you move in groups. Through the method detailed in Chapter 3, the foragers observe the movement behavior of the recipients as they effectively perform global tasks. In our study, we analyze and show the elements required by the recipients to efficiently perform the global task.

Chapter 5 shows the possibility that the trophallactic network system detailed in Chapter 4 can be implemented with real robots and applied in real situations. We deploy the trophallactic network system in a practical multi-robot system to ensure that each robot can serve the intended purpose according to its capabilities when it is implemented. The amount of pollutants delivered is determined by the brightness of the LED. We implement the system using real robots, and observe the movements of the recipients, and analyze the effects of these movements on the efficiency of the multi-robot system. The mathematical modeling of real robotic information and movements, information on situation implementation, and data are presented in detail in this chapter. Although the system has been deployed in robot simulation, applying it in an actual multi-robot system, and observing the movement of real robots may reveal other phenomena. However, We demonstrated that the same movements and results are observed when the experiment was repeated using real robots.

We present the conclusions and future research directions in Chapter 6. The entities in a trophallactic network system perform self-organization using a communication method called trophallaxis to execute the desired global task. This is achieved in several steps. First, the foragers examine the outside environment, and pass the information on to the non-foragers alongside the desired substance. Following this, it is observed that the recipients modify their movements according to the received information. We demonstrate that the trophallactic network system is applicable to a real multi-robot system to eliminate some of its complexities, as trophallaxis does not require a large system. We suggest various situations that can be studied to further verify and improve the effectiveness of trophallaxis for multi-robot system.

# Chapter 2

# Background

Our problem is complicated and difficult to solve everything with one robot. So many researchers are trying to solve these problems using multi-robots instead of just one robot. But with multi-robot there are new issues. In a multi-robot system, these issues are addressed in such a way that the task can be performed most efficiently and the swarm can benefit overall. Representative issues include how tasks are allocated to each robot(task allocation) and how information or objects are shared between robots(spreading information).

We are interested in how to share information or objects between robots. Multi-robots need to share information with each other or exchange objects depending on their mission. There are two ways for robots to share information or objects they need. First of all, the central system exists and transmits information of all robots or objects to the central system that all robots know. And the central system provides information or objects to the robots they need. The next method is to share information or objects locally. This means that robots can share tasks locally without knowing all the information.

Researchers attempt to solve this problem by sharing between robots rather than controlling a multi-robot over a central system. Since it is more limited than sharing all information or objects, it must be delivered with certain rules.

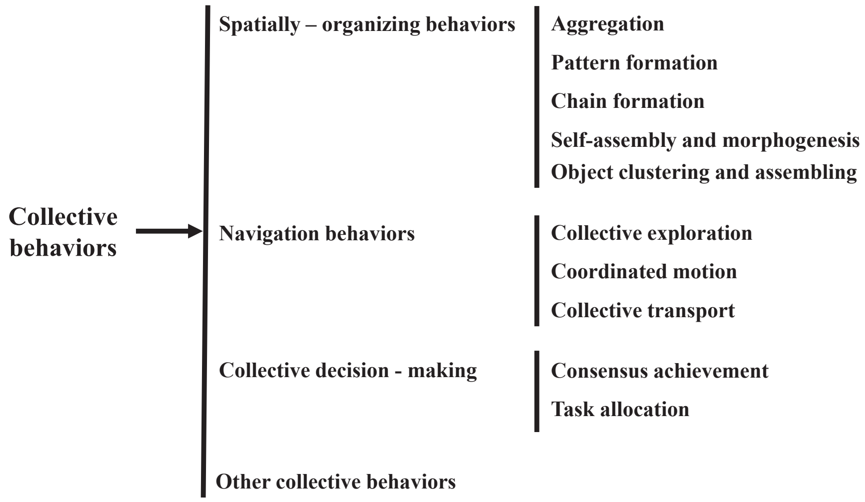



Figure 2.1: Type of collective behaviors (modified from (Brambilla et al., 2013)).

## 2.1 What is the swarm intelligence

In 1989 Gerardo Beni and Jing Wang introduced the cellular robotic system(Beni, 1988). The cellular robotic system has several rules. The cellular robotic system consists of a robot unit that moves autonomously. These robot units cooperate to achieve global tasks. The key concepts of this system are distributed computing, molecular computing, self-organization, and reliablility. Swarm intelligence exchanges information with limited communication between adjacent robots without a central system. Thus, swarm intelligence is less complex than controlling all robots with a central system. The system is also very stable because it receives distributed control. In addition, it has various advantages. In 1993, they first used the term swarm intelligence in analyzing cellular robotic systems (Beni and Wang, 1993). Robot intelligence and Robot system intelligence is defined by unpredictability of improbable behavior.

Unpredictability is analyzed by statistical unpredictability, inaccessibility, undecidability, intractability, and non-representability. This unpredictability results in different forms of intelligence behavior. Like the cellular robotic system introduced above, swarm intelligence(SI) is subset of artificial intelligence. SI has three properties (Blum and Merkle, 2008). First, set of simple agents perform collective actions. Each agent is not complicated. The SI then performs distributed control. In other words, there is no global control and each agent acts. Finally, agents in SI determine their behavior through self-organization. Self-organization is achieved by communicating between adjacent agents. The communication methods are direct communication and indirect communication. Direct communication can be achieved by contacting each other using vision or chemicals. On the other hand, indirect communication method is stigmergy. Stigmergy was introduced by Grasse in 1959 (Grassé, 1959). This method uses the environment to communicate. The same or different agents move in the same environment, leaving some traces. By repeatedly leaving these traces, the environment is strengthened. Agents get information from enhanced traces. As a result, *SI* can work together to accomplish complex tasks or actions that each agent cannot resolve in this way. And there is a big gain for the group. Such *SI* is basically common in nature. We show what *SI* is in nature. Naturally occurring *SI* provide a solution for running multi-robots.

### 2.1.1 Swarm intelligence in nature

We can easily see many insects and animals doing collective action. They do collective action for a variety of reasons. Insects can think in a variety of ways depending on the goal of group action. The first representative *SI* is division of labor or task allocation. If several agents have collective actions and there are several tasks, they try to work efficiently by distributing appropriate roles. This is an issue that needs to be addressed in collective action. Representative insects that share roles with each other are ants. Ants are insects that benefit the entire swarm through collective action. Ants have workers who work. It also shows the different tasks among workers, which perform their assigned tasks according to their age. Not only that, but it also flexibly changes its roles according to circumstances and circumstances to work on collective behavior (Sendova-Franks and Franks, 1993; Robinson et al., 2009).

Next is flocking, which is common in group behavior. Flocking is one of the common behaviors of fish and birds. In the case of fish, I do schooling for various reasons. They gain the advantage of predation by doing the schooling (Cushing and Jones, 1968). Another reason is due to hydrodynamical effects (Weihs, 1973). This effect gains speed and stamina. In the case of dolphins, too, the crowd acts to increase the rate of predation. They share the role of each other and increase the efficiency by allowing them to hunt a large amount of food (Shiqin et al., 2009). These actions do not exist and control the central system. Nor does a leader exist and lead everyone. Insects or animals in collective action do not know the situation of large groups. For this reason, you should judge yourself by using local information. A flock of fish and birds also acted by judging from the behavior of their peers. It's a local judgment and action, but when you look at it as a whole, it looks like you're leading a group of leaders who don't exist and disperse. This flocking movement is seen in sheep as well as fish and birds. We use only a few shepherd dogs to move large numbers of sheep using sheep moving in groups. The movement of shepherd dogs and the movement of herds of sheep is strongly related. As shepherd dogs get closer, the sheep try to move closer. In other words, the mean value of the distance of each sheep away from the center of the herd is smaller (King et al., 2012). Using this feature, shepherd dog's strategy was established to conduct efficient shepherding. Basically, it is to find a way to accurately move the herd of sheep to the desired place by dividing it into herding, covering, patrolling, and collecting (Lien et al., 2004, 2005; Lee and Kim, 2017).

This time, there is collective action that must be done for collective survival. Most collective actions of insects and animals are related to survival. FIre ants do special things to each other to survive natural disasters. Fire ants have the ability to float in water. These red ants keep together by rafting together in the event of a flood (Mlot et al., 2011). Fire ants build a raft to help keep the packs from scattering and create air bags to get oxygen in the water. Finally, it will help you easily climb up when you return to land after the flood. In order to prevent external threats like this, they may collectively act as survival strategies, but internally, they act collectively to supply energy. Energy supply is the most important for life. The most interesting part of this paper is the foraging of behavior related to energy supply. All the insects and animals in the group need different processes to get food and supply energy. First, workers start searching to find out where their prey is. There are many risks outside, and not all agents can get out. That is why some agents go out and search. When searching for food, it informs agents in the colony of its location.

As a way of telling where you are, you can tell a bee through a waggle dance. The optical flow method measures the distance and transmits the direction and distance to the waggle dance duration (Esch et al., 2001; Riley et al., 2005; Grüter and Farina, 2009). Ants use the Stigmergy method, which leaves a mark on the environment during indirection communication. The ant searches for a food source and returns to the colony, leaving behind a pheromone. Then the other ants see the pheromone and go to the food source to bring their food. When you bring food, if you still have food left over phermones. If you leave pheromone, the traces of pheromone on the way will be strengthened. These ant behaviors correspond to the representative ant colony optimazation(ACO) in the *SI* model. *ACO* is shown in detail in swarm intelligence models in 2.1.2. Next, they help each other get food from the colony.

Among other things, special actions are taken by sharing liquid food. Both bees and ants do the same thing, so that all agents in the colony are fed. That way the trophallaxis method is used. We use this trophallaxis to apply it to robots to solve complex and difficult problems with collective behavior. This trophallaxis method is also discussed in detail in Section 2.4. In addition to the behaviors described above, insects and animals get a variety of things through collective behavior. *SI* is an intelligence bahevior like the collective behavior of these insects and animals. Many people use these actions to solve many problems with multi-robots. We apply a variety of problems through the trophallaxis model. In Section 2.1.2 we get help from the typical *SI* model so far.

### 2.1.2 Swarm intelligence models

*SI* acts for a large number of robots to achieve the same goal. Many researchers have created a basic model for a particular situation among various group behaviors. Among them, known models include ant colony optimiazation (ACO) and particle swarm optimization (PSO). These two models are mathematically organized so that agents moving in groups can move optimally. Indeed, insects and animals use a variety of sensory organs, such as chemicals, vision, and smell, to conduct efficient collective behavior with their own rules. However, without certain rules, multi-robot can't have similar collective behavior. For this reason, many researchers have pre-mathematically summarized typical situations in order to establish rules. Many people will be able to bring and apply mathematical models for similar situations. We look at *ACO* and *PSO* to see what rules the mathematical model is based on. And finally, social insects play a role in choosing one another. This requires some behavior that must be shared with each other, such as trophallaxis. Therefore, we share the roles for some reason and look closely at what information is needed.

#### 2.1.2.1 Particle swarm optimization

Particle swarm optimization is a mathematical modeling of how birds and fish flock (Ahmed and Glasgow, 2012). It was first introduced in 1995 by James Kennedy (Kennedy, 2010; Eberhart and Kennedy, 1995). *PSO* is treated as the swam of the particles. Par-

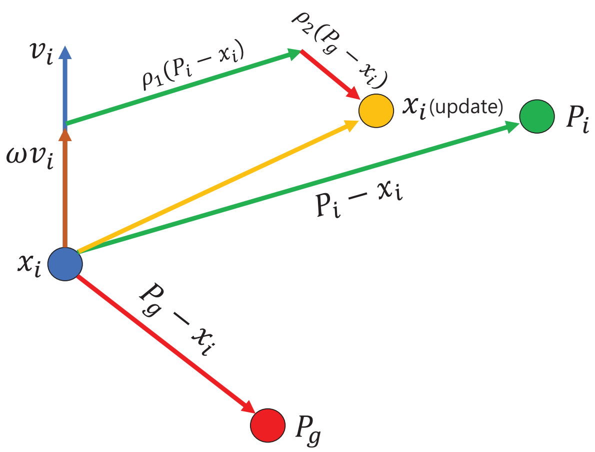



Figure 2.2: A visualization of the update on the location of the particles applied in particle swarm optimization. Particles move under three influences: inertia, aggregation, and cognition. The influence of the three effects can be adjusted appropriately with parameters to change the propensity of the particles.

ticle swarms are affected by neighboring particles. Thus, the swarm shape of the particles depends on how the neighboring positions are determined. Once the shape of the neighboring particles is determined, they are constantly updated with the speed, position and location of each neighboring particle.

$$\begin{aligned} v_i &:= \omega v_i + \rho_1 (P_i - x_i) + \rho_2 (P_g - x_i) \\ x_i &:= x_i + v_i \\ if\ f(x_i) &> f(P_i)\ then\ P_i := x_i \end{aligned} \tag{2.1}$$

Initially, the speed and position are randomly set. The velocity is determined by three terms. $v_i$ is a momentum term, $\rho_1\ (P_i - x_i)$ is a cognitive part, $\rho_2\ (P_g - x_i)$ is a social part. The momentum term has the effect of keeping the current direction of movement. The momentum term is also called inertia. A cognitive part means one's own best position. Finally, social part means the best position of the group. The effects of these three terms determine the direction and velocity, which determines where the next particle moves. The figure 2.2 visually shows the equation 2.1. In the case of *ACO*, they helped to make the collective behavior optimal by leaving traces in the environment. In other words, *ACO* corresponds to indirection communication. In the case of *PSO*, on the other hand, it looks directly at the agents around it and decides its behavior. As these behaviors come together, they create a collective action. *PSO* corresponds to direction communication. Indirection communication and direction communication are essential elements of self-organization. *SI* is a self-organization through limited communication, not central control, to achieve a global task.

#### 2.1.2.2 DIvision of labor and Task allocation problem

Social insects perform various tasks such as foraging and nest defense. Hundreds of thousands of insects share labor with each other. Some insects defend the nest and some have tasks to bring food from the outside. Using these aspects of insects, engineers similarly use multi-robots to perform a variety of complex tasks. In order for multi-robot to efficiently perform a variety of complex tasks, several things must be considered. What information do you need to perform tasks on a multi-robot system first? What functions should each robot have? Finally, how do you allocate task allocation?

These engineers consider task allocation issues in multi-robot systems centered on questions. Task allocation was basically done through a centralized control system. However, because a large number of robots must be controlled in noisy situations and tasks must be performed in various environments, a centralized control system has become a difficult algorithm. For this reason, many engineers are increasingly interested in decentralized control systems rather than centralized control systems. A decentralized control system is a system that performs various tasks through decentralized and self-organization. This is a basic properties of swarm intelligence (figure 2.4). Using this system, the problem of task allocation becomes a lot of motivation in biological situations. Ants perform different tasks by dividing various roles in the situation of bringing food. Ants are divided into forager and non-forager in colony. They quickly switch roles depending on the colony's nutritional status and body size, and perform the given task (Blanchard et al., 2000). Foragers perform tasks that bring food from the outside, and non-foragers perform tasks that receive food from the forager in the nest and deliver food to other non-foragers. Foragers and non-foragers change tasks dynamically depending on the situation. If an ant goes out to find food but finds it can't move alone if the food is large, the ant searching for an ant picks up an ant to nest with. The selected ants move to the area where the food is located and bring the food to them in cooperation (Robson and Traniello, 2002). In this way, ants distribute fluidly and efficiently search. In addition, it allows you to perform the tasks of the main workers

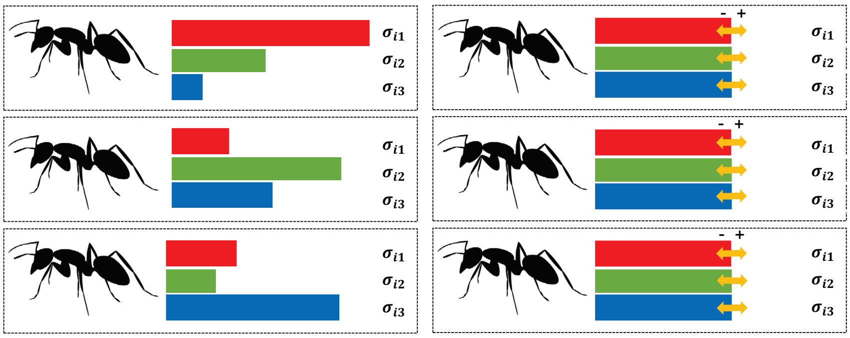



Figure 2.3: Threshold value for each agent's task. (a) Fixed threshold value. (b) Flexible threshold value.

in an emergency (Wilson, 1984). In this way, the ants have ants searching for food and only ants that exist only in the nest. In addition, there are many cases of division of labor by distributing tasks (Agrawal and Karsai, 2016; Ferrante et al., 2015; Ratnieks and Anderson, 1999).

In this way, the researchers consider how to distribute tasks in a multi-robot system. This is called a task allocation problem. Many researchers wonder how multi-robots can distribute tasks under the given conditions to get the most efficiency. There have been many ways to solve the task allocation problem. Typical methods include ratio method and threshold method. First of all, the ratio method looks at the ratio of the task to a certain bound of the robot and determines its role (Shehory and Kraus, 1998). Next is the threshold room, which is now more commonly known. The threshold method is that a given task sees it as a stimulus. Robots have a defined threshold for each task. Change the task if the stimulus crosses the threshold at a given task and is different from what is currently being done. Initially, when the threshold method was used, the threshold value of each robot was set. Proceeding to the specified threshold value is applied most efficiently only when the distribution of the whole task is known. But more often we don't know the percentage of tasks. In other words, it is good for the robot to change the threshold fluidly and distribute it accordingly.

The figure 2.3 shows how to do this. Each task has a threshol value that the robot has. When a task's stimulus crosses the threshold, change to that task. There is a high tendency to do tasks with low threshold values. This is because the threshold is low because you are trying to run the task on a small stimulus. In addition, research has been conducted to distribute work according to the situation by changing the threshold

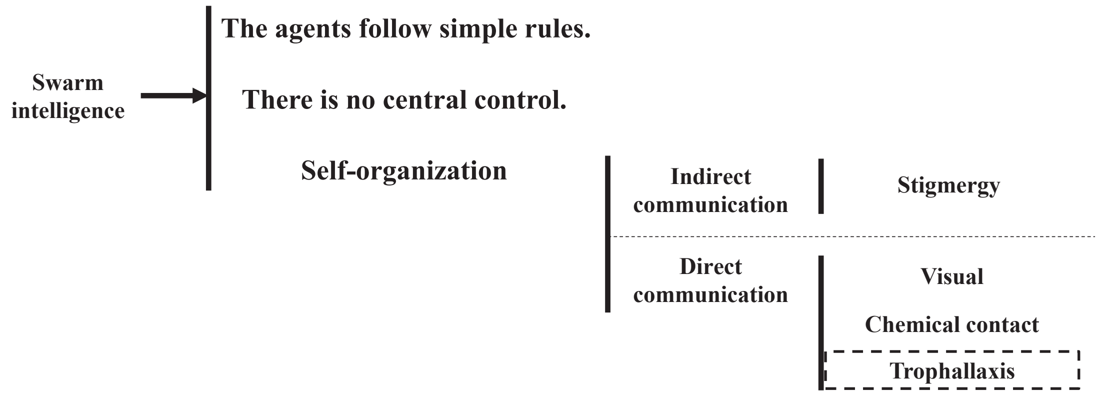



Figure 2.4: The properties of swarm intelligence. swarm intelligence basically requires that each agent unit be simple and there is no central control. And the key to swarm intelligence is self-organization. Self-organization sees a situation or takes information from another agent and uses it to feed back its status and decide what to do next. There are two ways of communication: direction communication and indirection communication. We are interested in the trophallaxis method of direction communication.

value flexibly. In reality, most multi-robots often suffer from changing tasks. Therefore, it is not good to cause frequent task changes. Considering this point, a method of calculating and changing the stimulus received for a certain period of time is proposed instead of immediately changing the task in response to the stimulus (Lee and Kim, 2014). The solution to the task allocation problem known so far is that it is applied only when the robot observes it directly in the area it can detect. If the robot communicates locally, the detection distance will be longer, which will allow for more efficient task changes. We looked into this possibility and carried out research of communication network system that can be applied to task allocation. We propose network system and show the change of robot's movement according to information.

### 2.1.3 Properties of swarm intelligence and methods

*SI* is the basic theory for controlling multi-robots. However, in order to apply the model of *SI*, it must have proper characteristics. The figure 2.4 shows the properties of *SI*. The ability of each individual of the multi-agents of *SI* is simply configured. Agents cannot solve all tasks by themselves. In fact, it is difficult to make one robot solve all the tasks from simple task to complex task. That's why a swarm of simple agents can solve both simple and complex tasks. The most important property is that there is no central control. In *SI*, the multi-agents run their tasks freely and individually, with no central control. How to control swarm through central control has many limitations. In the case of the central control system, there is an area where communication is possible and all the information is received from the central control system. The greater the amount of information, the more complex the system and the greater the chance of delay. In *SI*, on the other hand, without a central control system, each agent obtains information through local communication and uses the environment and the information it determines to determine the next action. Each agent judges and acts individually, but one swarm resolves the same global tasks. Therefore, the behavior of each agent depends on the method and information of communication, and the global tasks that can be performed accordingly.

Because *SI* is bio-inspired artificial intelligence, communication methods and giving information are also derived from the behavior of insects and animals. There are two ways to communicate locally: Direction communication and Indirection communication. Direction communication refers to the way in which adjacent agents meet in person and deliver information. This includes visual, chemical contact, and trophallaxis. In the visual case, the agent visually informs the other agent and decides the behavior. The chemical contact means the exchange of information by transferring chemicals to other agents. The trophallaxis discussed in this paper also corresponds to direction communication. The reason for this is discussed in section 2.4. On the other hand, indirection communication is a way of exchanging information through different environments, rather than in person.

Typical indirection communication is stigmergy. A stigmergy is a communication method that does not give information directly to an agent but leaves information in the environment so that other agents see and judge the changed environment and act. The representative insect that communicates by the method of stigmergy is an ant. Typically there is an action to find the shortest path between nest building and food source and nest (Dorigo et al., 2000). First of all, it is easy to see the pillars construction during nest building. When the soil pellet is initially released by ants, other ants see the changed environment. If another ant sees the soil pellets piled up, they pile up the soil pellets they bring. Repeatedly, pillars are built. Initially, an ant lays out a soil pellet, giving other ants information about where to build the pillars. This information does not directly inform each other, but rather the ants get information by looking at the environment where the soil pellets are stacked.

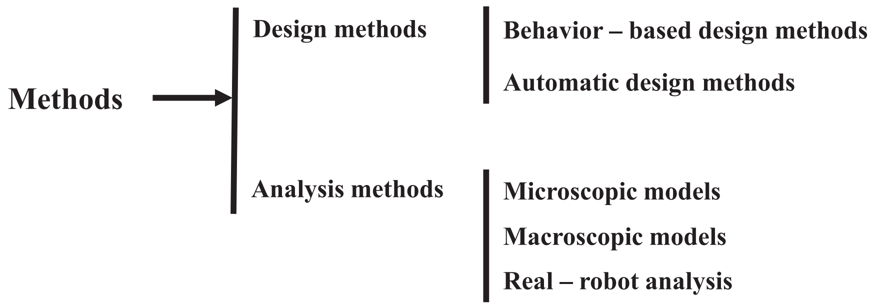



Figure 2.5: Design and analysis methods (modified from (Brambilla et al., 2013)).

The shortest path between food source and nest is found by ants leaving pheromone in the path. As the ants continue to move through the food source and nest, the ants move more and more along the shortest path. When ants move less on other paths, pheromones disappear. On the contrary, the shortest route is stronger pheromone. Therefore, the next ant to move will have a strong path for the phermones. As a result, if you iterate over it, you will only be able to find it by moving to the shortest path. We analyze the trophallaxis, which is the direction communication among the properties of *SI*, and show that the global task is performed by the trophallaxis communication method in the system satisfying all properties of *SI*.

The figure 2.5 shows how to design and analyze *SI* (Brambilla et al., 2013). *Behavior − based design methods* is a commonly used design method. The individual behaviors of the robots are developed manually to create collective behaviors. This design requires manual adjustments when errors occur. In the case of *Automatic design methods*, a robot is developed to automatically generate behavior. It is mainly designed to learn automatically through reinforcement learning and evolution. *Microscopic models* considers each robot individually, analyzing robot-to-robot and robot-to-environment interactions. The model analyzes and verifies the swarm robotics system through simulation. *Macroscopic models* considers the entire swarm robotics system. Individual robots are models that do not take into account.

Finally, *Real − robot analysis* creates a real robot to verify that group behavior is valid. Using real robots helps test how robust the swarm robotics system is against noise. Although it has been verified by simulation, if you are weak in noise when you actually run the algorithm, you will not see the desired collective behavior. In this way, the design and verification of the swarm robotics system can be confirmed. Our swarm robotics system is designed with *Behavior − based design methods*. Therefore, the operation of the robot is individually adjusted. And we model it with $Microscopic\ models$ and focus on the robot-to-robot interaction. Since the trophallaxis network system is a system in which group behavior is achieved by robot-to-robot interaction, $Microscopic\ models$ is appropriate. And we show that $Real-robot\ analysis$ actually produces the same collective behavior.

## 2.2 Swarm robotics

Swarm robotics is a new approach to the collective behavior of a large number of simple robots. It can explain the characteristics of the collective behavior of insects obtained through observations (Şahin, 2004). It is basically based on creating a collective action where each individual action performs a global task without a central system. It has three characteristics: robustness, flexibility and scalability. It aims to perform global tasks. In other words, even if individual behaviors go wrong for a moment, they can continue if they achieve the goal of swarm. Therefore, it is robust compared to other systems. The reason for the robustness is that even if one robot malfunctions, the other robot compensates for it. It also operates with distributed control, not central control. Central control causes all robots to malfunction if the central control malfunctions. However, swarm robotics communicate locally to determine behavior, so if a robot malfunctions, it does not affect other robots. Finally, because each robot is simple, it is less likely to have errors than a complex robot.

Another feature is flexibility, which allows you to easily change global tasks by changing the behavior of each robot. Finally, there is scalability. Increasing the number of robots makes it easy to scale up the group. Many researchers use robots to explain various $SI$. Harvard-made killobots were used to simulate various group actions. Representatively, the formation control creates the desired shape and shows that the algorithm induces the desired collective behavior (Rubenstein et al., 2012, 2014; Rubenstein and Nagpal, 2010). Another case is to use swarm robotics to extinguish an internal fire (Penders et al., 2011). As such, we can easily reproduce the collective behavior of insects with the characteristics of these swarm robotics, explain the reason for the behavior, and apply and verify for any goal.

### 2.2.1 Distributed network system

There are studies to solve many problems with multi-robots. Most of these multi-robot systems perform tasks at the discretion of individual robots, not central control. At this time, the robots exchange information with each other to perform tasks more efficiently. First, formation control is a task of a representative swarm robot. Formation control can be operated to the desired formation only if you know the location of each other. Information necessary here becomes positional information of each other. There is a lot of information to share the location information of all robots with each other. Thus, even if the robots share information locally, they can create the desired formation. At this time, a study of formation control using a topology method was introduced as a method of determining robots to be shared with each other (Ren and Sorensen, 2008).

Next comes a task to solve the patrolling algorithm problem. patrolling alogorithm is an algorithm for finding the hamiltonian cycle. The hamiltonian cycle is the path through all the vertices only once. It takes a long time to find a route. In order to solve this problem, it shows that not only one robot but several robots saves time (Hong et al., 2019). Creating a hamiltonian cycle locally can result in overlapping vertices. In order to prevent this, it is important that the robots do not overlap with each other. When each robot completes the hamiltonian cycle and informs it, it combines to form one large hamiltonian cycle. It shows that the task can be performed more efficiently and quickly when the task is executed by using more than one robot.

The task I'll introduce is Simultaneous localization and mapping (SLAM). SLAM is the method used to get the map. The robot collects and displays sensor data using a sensor to obtain a map. It takes a lot of time if the area we have to get is large. In order to improve this, a study was conducted to obtain a map using several robots instead of one (Tuna et al., 2014). The multi-robot SLAM presented in this paper shows a system that maps the role of a robot in real time. The map information obtained by the robot should be received by the server and synthesized for our viewing. However, because the distance to receive information is limited, the paper has set up a robot that receives information in the middle to overcome it. In other words, the robot has a robot explorer that collects map information and a relay that receives and transmits information in the middle.

Finally, the task for QoS Routing. Among them, the victim detection system refers to a system that finds and rescues victims with a robot. It shows that such a system becomes more efficient with several robots instead of one robot (Sugiyama et al., 2006). In this paper, the ad hoc network is used to drive the system. An ad hoc network is where each robot temporarily opens a network. Unlike the SLAM introduced, each robot receives information while performing tasks. It also introduces communication using topology rather than hoc network in QoS routing system (Wang et al., 2013)

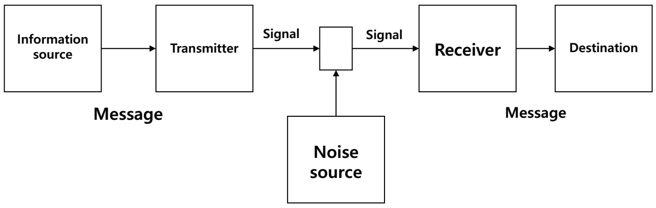
## 2.3 Spreading information problem

In Swarm robotics, information exchange is the most important factor in carrying out the desired task. There are some things to consider when exchanging information. First, we need to consider what information the robot needs to proceed with the task. Next, we need to think about how all the robots can quickly share the information they need. From these two perspectives, many researchers have been working to solve the problem.

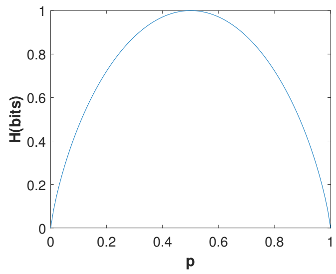
An important element of Swarm robotics is the exchange of information. It is important to select and exchange the necessary information to proceed with the task. Providing all the information is good for the task, but it is not easy to handle all the information. Therefore, in delivering information, it is effective to select and compress necessary information to deliver the information. Mathematical quantification is information theory. Information theory studies quantification, storage and communications. The theory of information was proposed by Claude Shannon in 1948 (Shannon, 1948). Information theory is based on probability and statistics. In information theory, the goal is to convey as much information as possible. In information theory, it is a branch of mathematics that determines how much data can be sent to a channel. The number of bits is determined by the information entropy, but the greater the information entropy, the higher the uncertainty. Figure 2.6 (a) shows the block diagram of the communication system and (b) shows the change in entropy H when there are two events. For example, if the coin has a probability p of 0.2 when face up and face down, the probability of face up is 0.8, which is 1-p. At this time, entropy H is as follows.

$$\begin{aligned} H &= -(plogp + qlogq) \\ q &= 1 - p \end{aligned} \tag{2.2}$$

In the equation 2.2 it is not defined when p is 0 or 1. Therefore, when p is 0 or 1, H




(a)




(b)

Figure 2.6: Communication system and entropy according to probability. (a) Block diagram of communication system(modified from Shannon (1948)). (b) If there are two events, the equation of entropy is $H = -(p \log p + q \log q)$. It represents the required entropy (bits) according to the probability. If p is 0 or 1, H is defined as 0.

is defined as 0. In this case, H has a maximum value when p is 0.5. Based on this, it is widely used to make a communication system that delivers information. Current multi-robot system exchanges a lot of information and proceeds tsak desired. You can do this by sending all the information to get the task you want, but sending a lot of information increases uncertainty and makes the data complex and time-consuming. Thus, we do not need to send data that is not needed or can be inferred. As an example, if you send a word, you can see that if the first received data is 'q', then 'u' will come out. Therefore, the word containing 'q' can be known without receiving data in 'u'. This inference within itself can reduce the number of bits needed to send information and reduce uncertainty. Our research identifies and identifies information that can help you perform tasks effectively when performing tasks in a multi-robot system.

Next, when information is selected, it must be quickly shared by all robots. The environment changes in real time, so you can't get the results you want if you don't share information quickly. Therefore, many researchers have been working to spread the data quickly on moving media to solve this problem. Initially, it was an issue to calculate the cover time with fixed nodes. Calculate and analyze the end-to-end delay in the network (Yu and Kim, 2010; Saifullah et al., 2014). But the nodes that receive the data often move. Since then, research has been done on moving nodes. When sending data from the base station, all users should be able to receive the data. At this point, I started thinking about spreading all the information quickly with the minimum delay. To experiment and verify this, we need to model the motion of the nodes. There was also research to model the movement of nodes as closely as possible to reality.

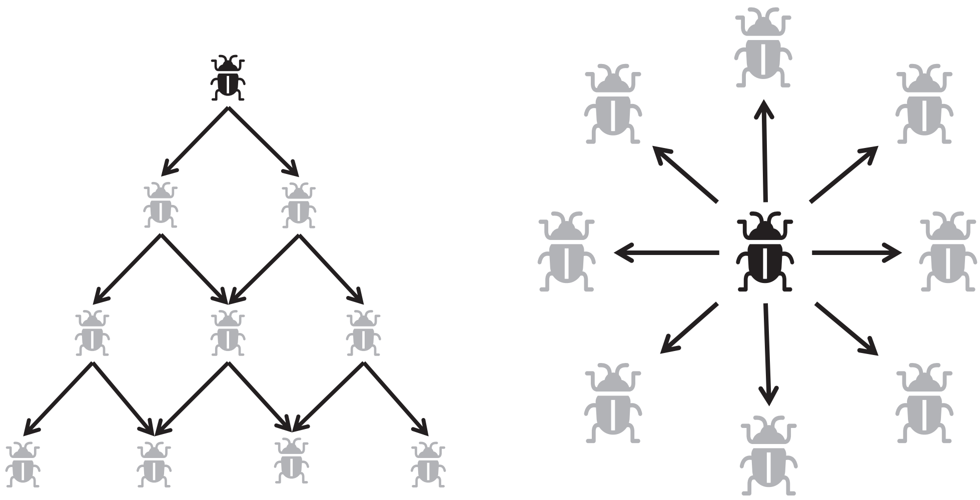
The current modeling model is the ”Smooth Random Mobility Model.” The model made more realistic modeling by considering the speed and direction of the node and by moving past values rather than being influenced only by the present. Based on this modeling, there are studies that have been conducted to evaluate and rapidly spread information when there are moving nodes. In this method, not only the information sent from the base station is directly received but also the information is shared between users so that the first user can transfer the information to other users. This is called device-to-device communication, and the addition of this method allows us to spread information faster (Choi et al., 2013, 2014).

To date, research has focused on the existence of a central control system and the sending of information from that system. However, in a multi-robot system, it is often impossible to install a central control system due to various problems such as environmental problems, a large amount of data, and algorithmic complexity. We use a trophallactic network system to show how robots in the role of external objects can take action and share information quickly and achieve their goals efficiently. Rather than simply sharing information quickly, we analyze how robots should behave in order to fulfill their desired purpose and apply it to applications.

## 2.4 Trophallaxis

Social insects are the most important factor in maintaining life. To get food, you have to go outside the colony and search. But going out and searching for food is dangerous. Also, all agents cannot go out to protect the colony. For these reasons, some agents go out and search for food, and others protect the colony. At this time, the searching agents should bring food from agents in the colony. It is possible to bring about common food and store it in colony. But for liquid food it is not possible to store it in colony. In this case, the trophallaxis method is used to allow all agents to eat liquid food. In some cases, trophallaxis can be used to deliver food, even if it is not liquid.

Trophallaxis is defined as the direct transfer of alimentary liquids, including suspended particulates and derivatives from one nestmate to another via regurgitation or anal feeding (modified from (Wilson et al., 1971)). Agents leaving the colony store food in their stomachs from the food source and bring it to the colony. When agents return to feed other insects, they take out the food stored in their stomach and deliver it to their



(a) (b)

Figure 2.7: The cascade of trophallaxis. (a) When a donor delivers to recipients, the recipients receive the donor role and deliver it to other recipients. (b) The donor delivers food directly to all recipients (modified from (Suárez and Thorne, 2000)).

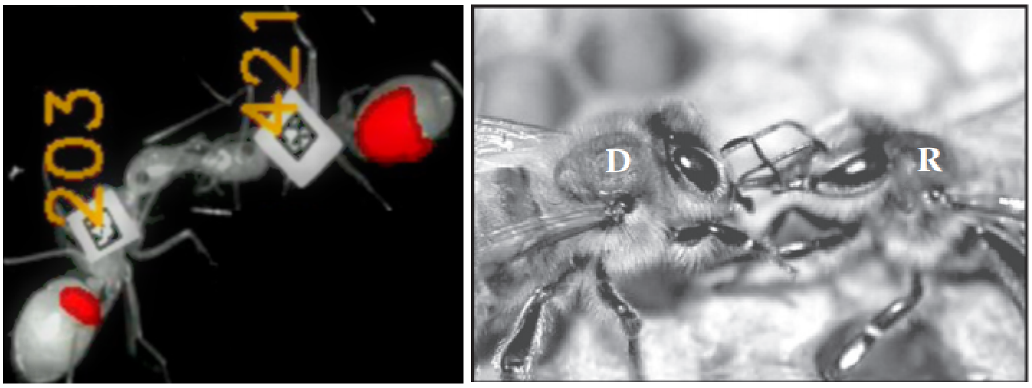
mouths. Agent that deliver food directly is called donor or forager, and agent that receive food indirectly is called recipients or non-forager. When these foragers deliver to non-foragers, there is a way to deliver them directly to all non-foragers, while there are ways to deliver only to some non-forgers and non-foragers fed to other non-foragers (Suárez and Thorne, 2000).

The figure 2.7 shows how foragers can deliver food to non-foragers by trophallaxis. social insects take advantage of the group by choosing the appropriate strategy based on their goals. Whether foragers deliver directly to all non-foreagers or to some non-foragers, the goal of using trophallaxis is the same. After all, social insects try to survive by all their members eating and receiving energy. However, there are questions about the social insects that are fed by the trophallaxis method. Basically, there is no central control system in the community of social insects. In other words, foragers do not know how many members of the colony are present and how much food they have. However, foragers bring in the amount of food needed for colony. This action is of interest to us. This is the question we are interested in. In addition, energy supply is linked to life. Therefore, it is important that there is no member who does not eat for a long time. We analyze how food should be delivered and in what quantity. We describe in detail in section 2.4.2 what the trophallaxis system means as the *SI* system.

### 2.4.1 Trophallaxis in nature

Insects or animals using the trophallaxis method are easily found. Typically bees and ants use trophallaxis (Korst and Velthuis, 1982). In the case of bees, there is a paper that observed trophallaxis through jelly. Bees each have a role. There is a worker who distributes food directly and there is a nurse who acts as a relay. Workers vary in their protein needs, absorption and utilization depending on their age and functional status. Workers use this to produce proteins and supply them to nurses. (Crailsheim, 1990). This balances the protein balance. Subsequently, experiments were conducted and the process of trophallaxis showed how food delivery varies with age (Crailsheim, 1992). In this way, the trophallaxis method among bees allows the entire community to obtain nutrients rather than to live alone. In the process of doing this, they act in such a way that the groups all benefit from their own rules, rather than creating and delivering them without rules. Not only bees, but also ants are representative social insects that share nutrients using trophallaxis.

Ants forager take food from above and bring it to the non-foragers when they return to nest (Cassill and Tschinkel, 1995). Feinerman Ofer observed the traophallaxis of ants using infrared cameras and barcodes (Greenwald et al., 2015). The experiment proceeded as follows. In the case of feeding, liquid food was used, and color dye was added so that it could be seen by an infrared camera. The experimental environment was partitioned and the proper entrance was made to distinguish the nest from the outside. A group of queen ants was placed in the nest for feeding outside. First, the foragers go out and look for food. The forager finds prey and feeds back to nest. The forager takes out the food and delivers it to other ants. After that, more foragers come out of the nest with their food. At this time, you can see that the food source is directly moved. This is possible because of the route left to phermones. Non-foragers fed by foragers move and deliver food when they encounter other ants. This is the same as figure 2.7 (a). Over time, all the ants in the nest will be fed and the number of outgoing forager will be reduced. These experiments allowed us to look closely at the trophallaxis of ants. In chapter 3, we analyze the trophallaxis system of ants under the same circumstances as swarm robotics via *Microscopic models*. The trophallaxis behavior of bees and ants is expected to help control multi-robot. To use it correctly, you need




(a) (b)

Figure 2.8: Example of social insect's trophallaxis. (a) Trophallaxis of ants ((Greenwald et al., 2018)). (b) Trophallaxis of honey bees ((Farina and Grüter, 2009)).

rules and mathematical models for the model, such as *ACO* and *PSO*.

### 2.4.2 Trophallaxis network system

We have been able to observe the actual trophallaxis of insects through the experiments of many researchers. In communities where trophallaxis is applied, exchange is carried out locally and all behavior is determined by the behavior of each individual. The only thing you do to feed the members of every nest is to deliver the food you bring. In other words, each member does nothing but simple things. The size of this group is very large. For example, ants have hundreds of ants in a nest. This means that no information is available about members of all communities. Thus, this community cannot have central control. Each member's behavior is determined by trophallaxis. The foragers decide whether they need to go for more food or not, simply by trophallaxis. This means that it is becoming self-organization. At this time, trophallaxis is the action that helps self-organization.

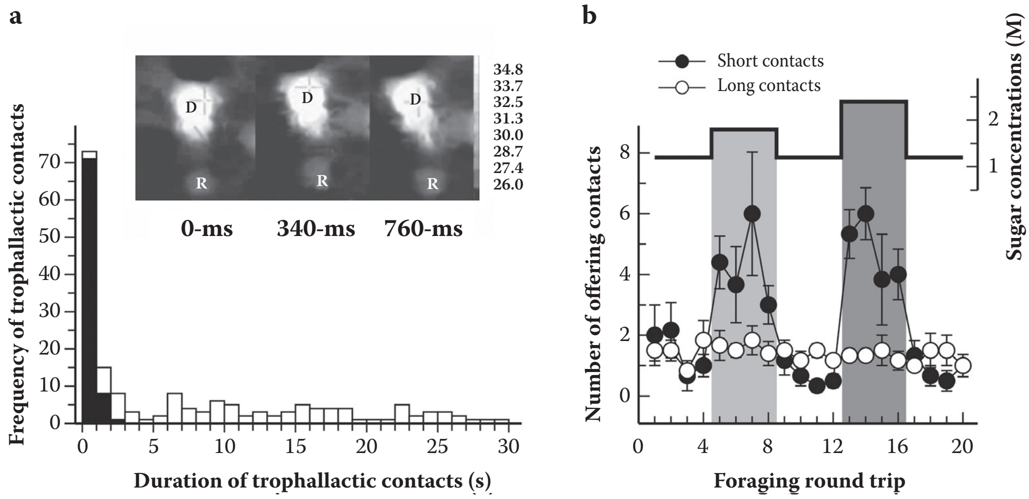
The trophallaxis acts as a communication that shares information with each other. As you can see in figure 2.1, all of the conditions of *SI* are satisfied. This means we can access this system from the perspective of *SI*. The most important element in *SI* is self-organizaion. If self-organization is not done well, each member generates an error, and if the error is large, the desired group behavior cannot be achieved. Examples of self-organization of social insects include moving large leaves and finding the shortest paths (Garnier et al., 2007). The trophallaxis not only meets and delivers food but also provides information. Therefore, it can be regarded as direction communication (Charbonneau et al., 2013). Thus, the group naturally forms a network.

The trophallaxis network system is formed by many social insects (Fewell, 2003). As mentioned in section 2.4, all members are at risk of finding food. And members must defend their colonies. The result is a naturally formed network system. The fact that trophallaxis is a means of communication also means that it also serves to convey information (Waters and Fewell, 2012). The information that can be obtained from trophallaxis is first the amount of crop load each other can get. We don't know exactly how much food you need. However, when your stomach is full, you will feel full, and you will know whether you have eaten enough food or not. This is also related to the time of exchange. The longer the exchange time, the more I still need a lot of food and try to get more food. They use this information to coordinate basic tasks related to the feeding process and to connect with different groups (Farina and Núñez, 1993). Not only feeding, but also obtaining other information to help them perform more efficient group actions. In chapter 4, we show how not only feeding but also other information can accomplish the task we want when used by the trophallaxis method.

### 2.4.3 Application

Many robots are produced to perform various tasks by simulating the behavior of social insects. The trophallaxis network system is also used for warm robotics. The trophallaxis is basically communication between the robot-to-robot or agent-to-agent. Therefore, when operating a multi-robot, it proceeds in a manner that delivers information or materials locally without central control. We introduce swarm robotics that perform missions by applying the trophallaxis network system. First, there is a robot that performs the task of dumping dirty materials in the trophallaxis network system method. (Schmickl and Crailsheim, 2008). This robot informs neighboring robots of its status. The information that can be given includes information about whether dirty material is in the vicinity, whether or not there is a material to be delivered, and dump information. The robot obtains the corrected parameters and proceeds with the corresponding actions. Repeating this locally will cause the dirty task to be moved to the global task dump.

Another robot transfers energy in line with the purpose of trophallaxis (Kubo and Melhuish, 2004; Schioler and Ngo, 2008). Robots always follow battery problems. When




(a)

Figure 2.9: The trophallactic contact time associated with the number of trophallaxis. The number of offering contacts varies depending on the short and long contacts. In general, the shorter contacts, the higher the number of offering conacts. With this information, foragers decide what to do when they bring food.(Farina and Grüter, 2009)

running a multi-robot, the battery problem is fatal. Because when the number of robots increases, it needs a lot of charging stations to fill them up. Charging the robot in a limited charging station takes a lot of time. Power distribution using the trophallaxis method has the potential to solve this problem. When exchanging such power, the power consumed by itself is also a problem to consider. If the consumption is large when receiving power, the power of the group continues to be seen. The trophallaxis network system is an essential system for the efficient energy supply of social insects. Using such a system has a great influence on the performance of multi-robots.

### 2.4.4 Inforamtion of trophallaxis

We do not end with the role of trophallaxis in providing nutrients to each other. The goal is that all community members have nutrition. Social insects collect information that can be obtained during trophallaxis in order to supply nutrients efficiently. As the trophallaxis begins to feed on the nest, the interaction time, interaction rate, and crop load change, and this information changes the behavior of the foragers. (Wainselboim and Farina, 2000; Farina and Núñez, 1993, 1995) The volume of the interaction is also reduced. This information is a past experience when the forager goes back to get food from the food source. Experience has shown that past harvesting times will vary and predict the amount of nested prey. This information controls the foragers' behavior out of the nest. Social insects change their information by looking at the amount of food they have and based on it, they decide their behavior. We analyze in Chapter 3 how to simulate this information by means of robot simulations of how this information relates to the amount of nested prey.

## 2.5 Summary of Chapter2

In this chapter, we introduce the basic background of swarm intelligence. And we show the trophallaxis behavior of the social insect we are interested in. Social insects are acting in groups and in various ways to benefit the community. The trophallaxis network system is one of them. The trophallaxis network system is a self-organization system. Each behavior is determined using the trophallaxis method, which is one of direction communication. There is more information about this trophallaxis network system and how it affects the community. And various multi-robot applications are also introduced.

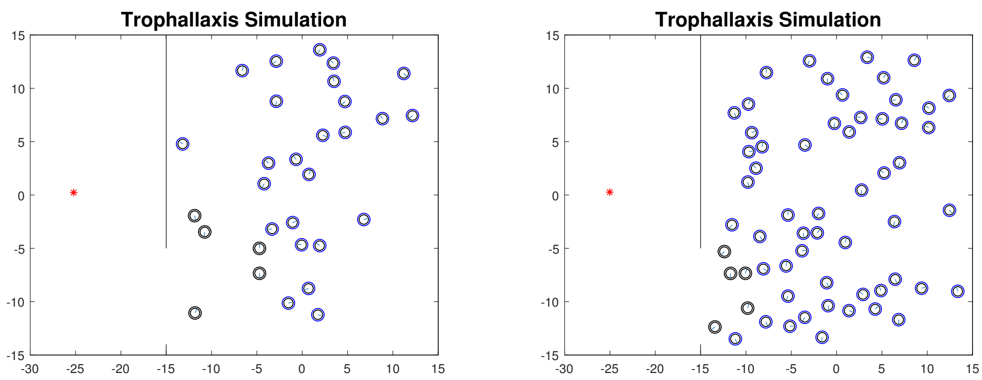
# Chapter 3

# Analysis of trophallactic network system using multi-robots

In this chapter, we implemented an ant trophallactic network system using robot simulation. The effectiveness of the robot simulation demonstrates that it behaves similarly to a real ant network. A robot that brings food from a food source is called a forager, and a robot that moves within a nest is called a non-forager. The amount of food exchanged is randomly determined by the an exponential distribution, and acts on the basis of local information but behaves as if observing global information as a whole. All the robots are fed as the colony state converges to its maximum value. We demonstrate how food inflow, interaction volume, interaction rate, maximum distance from entrance, and the number of interactions are related.

This result implies that the forager determines the food inflow as only the amount of the opponent ant's food, but indirectly includes the information of the colony state. In this chapter, we demonstrate the same behavior when applying mathematical modeling of real ants to robot simulation and reveal how the system is varies by changing variables that cannot be changed among real ants.

We also examine the system from various perspectives. The system demonstrates different effects depending on the number of foragers, the speed of non-foragers, and the distribution that determines the interaction volume. The system is evaluated according to the speed of filling, how well the robots are distributed, and the number of interactions. As a result, the robots can be operated by setting parameters according to the purpose of the system. We also confirms why ants determine the interaction volume by




(a) (b)

Figure 3.1: Robot simulation. The blue robots are represent non-foragers, and while the black robots is represent foragers. The small circle of robots represents the maximum amount of food, while. the star shape represents food source. (a) The total number of robots is 30, and the forager : non-forager ratio is 1 : 5., (b) The total number of robots is 60, and the forager : non-forager ratio is 1 : 11.

exponential distribution. The reason is that foragers often give a large amount of food to non-foragers because they bring the food from the food source. Then, non-foragers are not be able to feed evenly; however, non-foragers exchange their food with each other and alleviate this problem. Therefore, exponential distribution is the best way to quickly fill the colony and feed all the ants evenly as evidenced by. our results show this. The one of these reseaches is publisged (Kim and Kim, 2019).

## 3.1 Simulation

### 3.1.1 Trophallactic network system

A trophallactic network system refers to a system in which ants interact with each other through their mouths. Not all ants leave to find food sources, and each ant moves in a shared role. We divided the role of ants into two broad categories. First, we observed the food source outside and divided it into a forager that brought the food to the nest and a non-forager that traveled while feeding the food from the nest. Real ants find their way via pheromones when an initial forager comes in search of food. The foragers then move directly to the food source along the pheromone pathway to retrieve the food. The foragers then return to the nest and feed the non-foragers. Repeatedly, the total amount of food in the nest increases. Using this method, the amount of food required the nest can be accurately obtained.

In this study, we constructed a robot simulation using MATLAB. We assume that ants are robots and that pheromone paths are constructed. Thus, robots move directly to the location of the food and return to the nest immediately. Our point of observations are of the food inflow and the colony state of the nest, and. we analyze the type of system that it is according to this change.

### 3.1.2 Environment

The configuration of the simulation is similar to that of real ants presented in a previous paper, as illustrated in Figure 3.1. First, the size of the arena is 45 (cm) in width and 30 (cm) in length. The x-coordinates of the arena are -30 to 15, while the y-coordinates is are -15 to 15. At the point where the x-coordinate is -15, there is a wall that separates the nest from the outside. The entrance is from (-15, -14) to (-15, -5). In the initial stage, all robots are created on the nest side, and the foragers are selected in the order closest to the entrance.

For example, if there are two foragers, they in the order closest to the entrance. Because forager that if the fall so far produced in nest center and entrance takes coming with the food outside due to collision avoidance robot moves to the entrance is a long time to come back to the nest forager they have a food has no information when delivering the food. It is important to observe if this determines the food inflow. There are two types of robots: a forager and a non-forager. A forager is black, and while a non-forager is blue. The red star shape represents food outside the nest, which is provided infinitely without restrictions. The robot is observed by a camera from above.

### 3.1.3 Robot

The size of the robot is a circular shape with a radius of 0.4, and. the sensor detection range is 1. The robot has eight sensors from $-90°$ to $90°$. A circle with a radius of 0.3 indicates the maximum amount of food that can be stored by the robot. When the robot eats its food, it is displayed as a circle with a filled red circle. The amount of food that the robot stores is counted from 0 to 0.3. If it eats as much as 0.1, a radius of 0.1 and

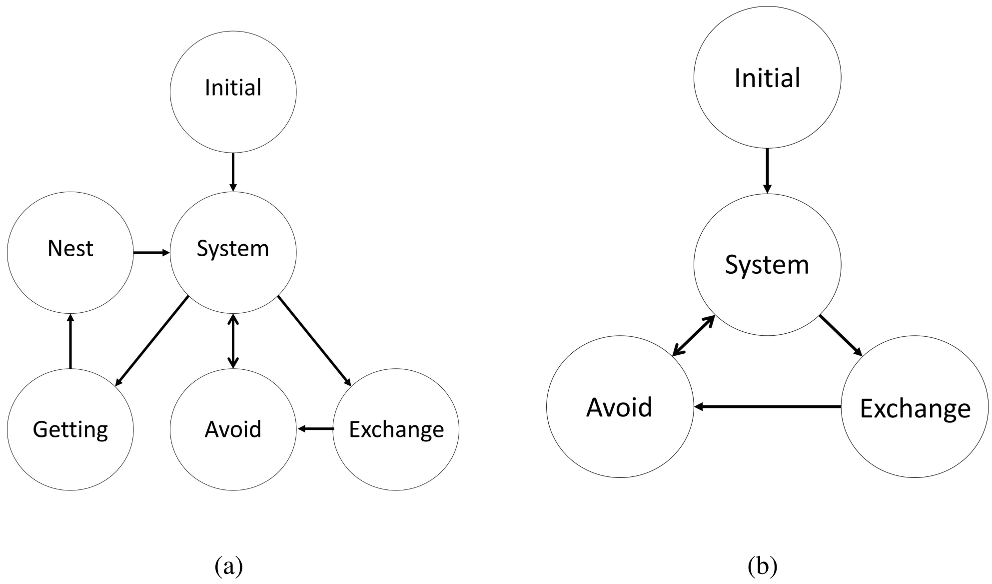



Figure 3.2: State diagram of forager and non-forager. The state is changed in the direction of the arrow. (a) Forager's state diagram. The forager's state consists of Initial, System, Avoid, Exchange, Nest, and Getting. The state changes according to the situation. (b) Non-forager's state diagram. The non-forager's state consists of Initial, System, Avoid, and Exchange. The non-forager's state is simpler than the forager's state. because non-foragers only move within the nest..

a red filled circle appear. We confirmed that the robot delivered the food well. Robots behave differently according to forager and non-forager. Figure 3.2 demonstrates that state changes as situation of forager and non-forager.

A forager behaves according to six states: initial, system, exchange, obstacle avoidance, getting, and nest. First, all the states of the robot are initialized. It then proceeds to the system state. In the system state, it is determined whether there is an obstacle or another robot, and the next action is selected. The forager detects the closest obstacle in the sensor range and determines whether it is ant. If it is not an ant, an action is selected with an obstacle avoidance state. If it is a robot, it selects an action according to the condition. If the forager is more prevalent than the opponent robot, and the opponent robot is not in exchange, then an action in the exchange state is selected. Otherwise, consider the action an obstacle avoidance state.

If there are no obstacles around the forager and the food is full, the forager moves in the nest randomly. However, if the food is not full, the forager moves to the entrance.

When it leaves the nest, it selects an action in the getting state. The getting state moves directly to where the food is located and collects food up to the maximum storage size. When the forager has filled up its food, it chooses to act in the nest state. It then returns to the nest and travels around in the nest to exchange. This is repeated continuously. If the forager goes outside and brings food, it is aware of any obstacles and avoids collision. Moving while considering collision avoidance is also part of the getting and nest states.

A non-forager selects actions according to four states: initial, system, exchange, and obstacle avoidance. The initial state initializes the state of the robot, as with the forager. The behavior is selected as a system state. In the system state, the non-forager verifies whether there is an obstacle. If there is an obstacle, it verifies whether the object is a robot. If it is not a robot, it avoids the obstacle in the avoidance state by selecting an action. However, if the obstacle is another robot, the non-forager has more food than the opponent robot, and the opponent robot is not in exchange, then the exchange state action is selected. If there is no obstacle, the non-forager moves randomly within the nest. This is performed repeatedly.

#### 3.1.3.1 Object avoidance

Obstacle avoidance is an important action for preventing conflicts; therefore, it is performed first. The robot uses an IR sensor to automatically detect obstacles. The maximum distance of the IR sensor is 1.5. An obstacle is avoided when the distance between robot is less than 1.5 as detected by the IR sensor. There are eight IR sensors from $-90^{\circ}$ to $+90^{\circ}$ of the robot. If only the sensors between $-90^{\circ}$ and $0^{\circ}$ detect an object, it is deemed that there is an obstacle on the left side.

Similarly, if the sensor is detects an object between $0^{\circ}$ and $+90^{\circ}$, it recognizes that there is an object on the right side. If there is no obstacle, the left wheel and right wheel speed of the robot change randomly from 0 to 10. However, when an obstacle is detected, the obstacle is controlled by changing the linear speed and rotation speed depending on the location of the obstacle.The rotation speed is defined as (+) counter-clockwise and (-) clockwise.

For example, if there is an obstacle on the left, then the straight forward speed is +10 and the rotation speed is -5. Assuming that there is an obstacle on the right side, the straight forward speed is -10 and the rotation speed is +5. If obstacles are detected on

**Algorithm 1** Forager Algorithm

```
forager state ← Initial
while running do
  if forager state = Initial then
    forager state ← System
  else if forager state = System then
    if obstacle then
      if obstacle = ant then
        if More than the opponent robot then
          forager state ← Exchange
        else
          forager state ← Avoid
        end if
      else
        forager state ← Avoid
      end if
    else if Lack of food then
      forager state ← Getting
    else
      Randomly move
    end if
  else if forager state = Exchange then
    if Exchange is possible then
      Give the opponent robot food
    end if
    forager state ← Avoid
  else if forager state = Avoid then
    Avoid obstacle
    forager state ← System
  else if forager state = Getting then
    Go to the food source
    if Arrive the food source then
      Get food
      forager state ← Nest
    end if
  else if forager state = Nest then
    Go to the nest
    if Arrive the nest then
      forager state ← System
    end if
  end if
end while
```

**Algorithm 2** Non-forager Algorithm

```
forager state ← Initial
while running do
  if forager state = Initial then
    forager state ← System
  else if forager state = System then
    if obstacle then
      if obstacle = ant then
        if More than the opponent robot then
          forager state ← Exchange
        else
          forager state ← Avoid
        end if
      else
        forager state ← Avoid
      end if
    else
      Randomly move
    end if
  else if forager state = Exchange then
    if Exchange is possible then
      Give the opponent robot food
    end if
    forager state ← Avoid
  else if forager state = Avoid then
    Avoid obstacle
    forager state ← System
  end if
end while
```

both the right and left sides, the straight forward speed is 0 and the rotation speed is -5, so that the robot can rotate in place. Each time the sensor is checked, the robot moves according to the detected sensor. Avoiding obstacles is important for preventing collisions between different robots. Therefore, avoidance must first be determined and other actions should be taken if the obstacle is determined to be safe.

#### 3.1.3.2 Exchange

Exchange is a behavior in which robots exchange food with each other. Specifically, a robot that stores more food delivers it to a robot that stores less food. If the robot does not have the amount that it must to deliver, or if the robot receiving the food can not accept the delivered amount, no trade is made. The size of the food inflow is determined based on an exponential distribution. When the robots perform an exchanged with each other, they are randomly determined according to the remaining amount to be received. Again, this signifies that the amount of food the robot receives is more important than the amount of food it receives. Food inflow is described in more detail later in the chapter.

## 3.2 Method

Our method is based on that in Greenwald et al. (2018). We define $f_i(t)$ as the total amount of food that forager $i$-robot delivers to the nest. $f_i(t)$ is determined by the exchange between a forager and non-forager, and the total amount of food delivered to the nest is called the colony state and is defined as $F(t)$.

$$F(t) = \sum_{i=1}^{N} f_i(t) \tag{3.1}$$

Equation (3.1) demonstrates the relationship between $f_i(t)$ and $F(t)$. N is the total number of foragers. If no food is delivered, $F(t)$ becomes 0. However, If the nest is full, $F(t)$ becomes 1. The value of $f_i(t)$ increases when a forager delivers food to a non-forager, and decreases when a non-forager delivers food to a forager. Foragers decide to leave the nest according to the amount of food they store; however, non-foragers do not leave the nest. This property has a large impact on determining the colony state. The amount of the forager's food is irregular because the foragers obtain food from a food source. Thus, it is difficult to determine whether the colony state is filled with the amount that the forager delivers. As a result, $f_i(t)$ is determined by the exchange of between a forager and non-forager, and the sum of $f_i(t)$ determines the colony state $F(t)$.

It is necessary to understand how $f_i(t)$ is determined to perform a robot simulation. $f_i(t)$ is determined when the $i$-robot exchanges food with another non-forager robot.

We therefore define $p_i(m)$ as the amount of food that the $i$-robot exchanges with other non-forager robots. $p_i(m)$ is determined based on the forager $i$-robot. If the forager delivers food to a non-forager, the nest's food has accumulatesd. Therefore, it has a a (+) value. In contrast, if a non-forager delivers food to a forager, it has a value of (-) because it is not delivered to the nest. Additionally, $\frac{df_i(t)}{dt}$ is defined as the time-average flow, and relates with $p_i(m)$.

$$\frac{df_i}{dt} = \frac{1}{\delta}\sum_{m=m_1}^{m_1+\delta} p_i(m) \quad (0 \le m_1 \le M - \delta) \tag{3.2}$$

Equation (3.2) illustrates how the value of $\frac{df_i}{dt}$ is defined. $M$ is the total length of $p_i(m)$, while. $\delta$ means represents the average intervals. That is, when performing an exchange, the $i$-robot records the amount of food exchanged and obtains the average as $\delta$ intervals. $p_i(m)$ is saved when a forager exchanges food with a non-forager. Therefore, $p_i(m)$ has an impulse function because it changes instantaneously. Figure 3.3 demonstrates that $p_i(m)$ has an impulse function.

The time-average flow, $\frac{df_i}{dt}$, obtained by using (Equation 3.2) is plotted against the colony state $F(t)$. It can be seen that $\frac{df_i}{dt}$ decreases as $F(t)$ approaches 1. This signifies that if the colony state $F(t)$ is reduced, the amount of food delivered on average is also reduced. In other words, because there are no more robots to deliver food, the interaction volume is reduced. These results suggest that the robot simulations are similar to those presented by Greenwald et al. (2018). As a result, it is suitable that the mathematical model obtained by observation of actual ants in the chapter.

The global flow must also be identified. Time-average global flow is defined as $\frac{dF}{dt}$, which is determined by $\frac{df_i}{dt}$. The relationship between the two values is as follows:.

$$\frac{dF}{dt} = \sum_{i=1}^{n(t)} \frac{df_i}{dt} \tag{3.3}$$

Equation (3.3) illustrates the relationship with $\frac{df_i}{dt}$, where. n(t) represents the number of foragers that start exchanging food at t seconds. In the paper by Greenwald et al. (2018), $\frac{df_i}{dt}$ exhibits a tendency to decrease according to $F(t)$ and $\frac{dF}{dt}$, and after all foragers participate. The same tendency is represented in the robot simulation results.

All situations are determined by $p_i(m)$. The interaction volume $\nu$ is the amount of food to be exchanged, and the relationship between $p_i(m)$ and $\nu$ is as follows:

$$p_i(m) = \begin{cases} -\nu & (If\ forager\ is\ fed) \\ \nu & (If\ forager\ feeds) \end{cases} \tag{3.4}$$

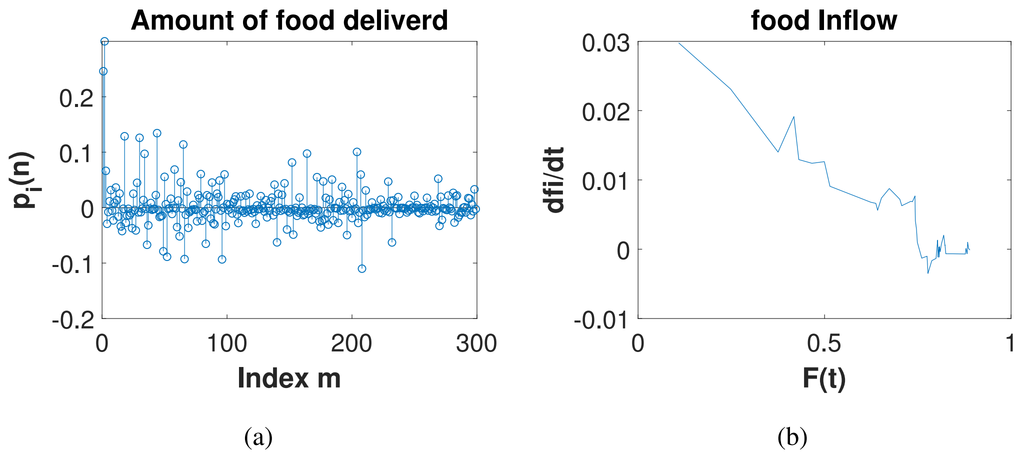



Figure 3.3: Interaction volume when exchanged and Relationship between colony state and time-average flow (a) Interaction volume when during an exchange between a forager and non-forager. $p(m)$ appears as an impulse function because the value exists only at the moment of exchange. (b) Time-average flow calculated by (equation 3.2). As colony state $F(t)$ increases, the time-average flows decreases.

This value of $\nu$ is determined by an exponential probability density function (PDF), and this is used to determine the amount of food to be exchanged. The form of the exponential PDF is as follows.

$$p(\nu|c) = \lambda_c e^{-\lambda_c \nu} \tag{3.5}$$

where $c$ represents the crop load, and $\lambda_c$ depends on the crop load. The exponential PDF is determined according to the value of $\lambda_c$ and randomly determines the value of $\nu$ according to this distribution. When the value of $\nu$ is determined, the value of $p_i(t)$ is also determined by (equation 3.4). For this reason, the form of the exponential PDF is important. The main factor that determines the graph is $\lambda_c$, which is determined by crop load $c$. The relationship between $\lambda_c$ and $c$ is as follows:

$$\lambda_c = \frac{\lambda_0}{C_0 - c} \tag{3.6}$$

where $c$ is $c_{forager}$ or $c_{recipient}$. $c_{forager}$ refers to the crop load of the forager, while $c_{recipient}$ refers to the crop load of the robot receiving the food. In actual real ants, $\lambda_c$ is more affected by $c_{recipient}$ than $c_{forager}$, and we must ensure that the same effect occurs when performing robot simulation.

Our results indicate that $\lambda_c$ is more affected by $c_{recipient}$ than $c_{forager}$ (see Figure 3.4). In the case of $c_{forager}$, the value of $\lambda_c$ does not change significantly. In contrast, the value

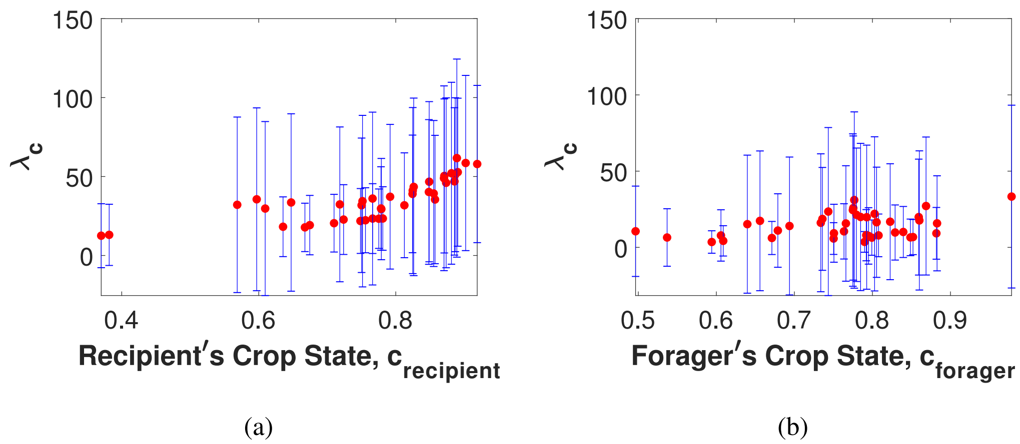



Figure 3.4: Changes in $\lambda_c$ according to crop load. (a) Relationship between $\lambda_c$ and $c_{recipient}$. As $c_{recipient}$ increases, $\lambda_c$ also increases. (b) Relationship between $\lambda_c$ and $c_{forager}$. Even if $c_{forager}$ changes, $\lambda_c$ remains constant. The crop load of forager, $c_{forager}$, does not affect $\lambda_c$.

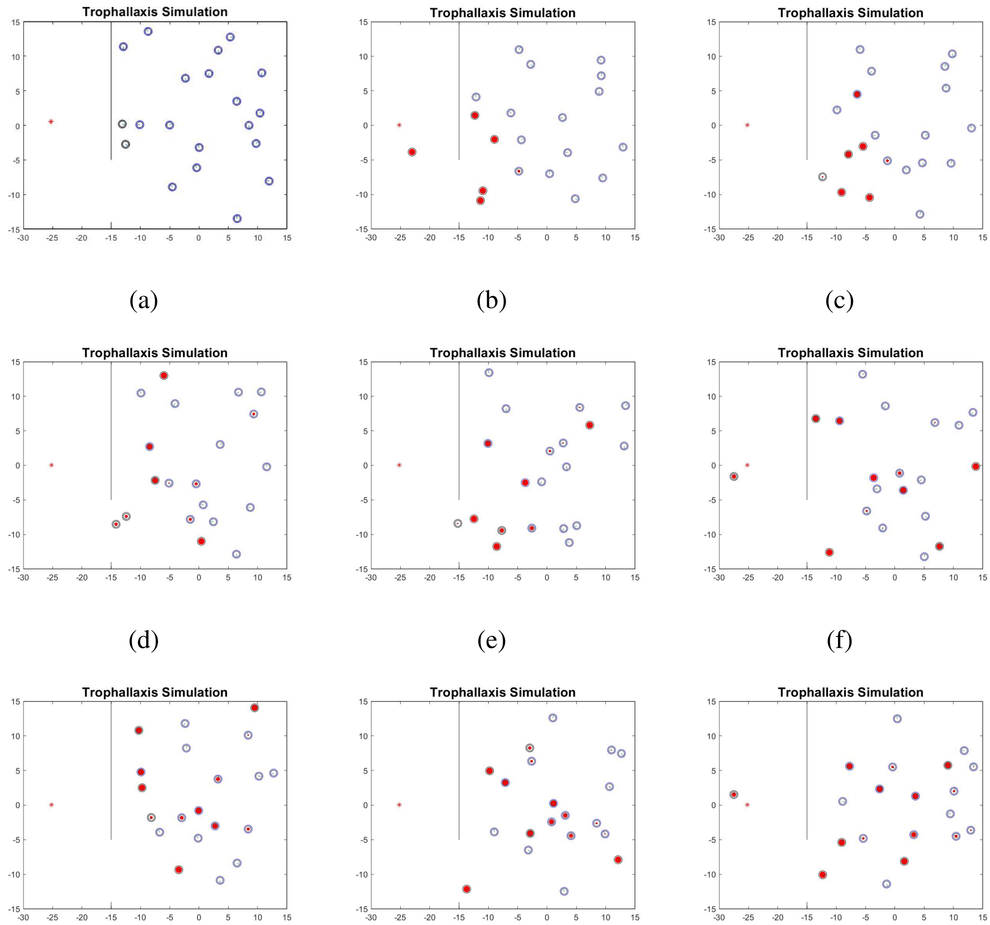
of $\lambda_c$ increases as $c_{recipient}$ increases. In other words, the crop load of the receiving robot is more important than the crop load of the robot when exchanging food. Based on this, we performed a robot simulation and verified the observed tendency. We not only observed a similar tendency to the appearance of a real ant, but also identified its characteristics by changing variables that normally can not be changed.

## 3.3 Simulation Results

### 3.3.1 Trophallactic Network System in Robot Simulation

First, we had to ensure that the robot system was similar to a real ant system. In the paper Greenwald et al. (2018), an actual ant was observed, and a graph was plotted. Based on this, we evaluated the similarity of our robot simulation to real ants. Each parameter had to be determined. Among real ants, non-foragers have a low mobility in the nest, where the ants move little or a short distance. Therefore, we set the speed of non-foragers to a low value. The maximum amount of food that each robot can hold is 0.3, and once it is exchanged, it does not change even if it meets another robot immediately. After a certain period of time, the food is exchanged again.

According to the paper Greenwald et al. (2018), the forager leaves when it has slightly




(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 3.5: Sequential progress of the simulation. Filled red circles represent the amount of food stored by each robo, and their radius is equal to the amount of food. Foragers are generated at regular intervals:. (a) t = 200 s, (b) t = 1,000 s, (c) t = 2,000 s, (d) t = 3,000 s, (e) t = 4,000 s, (f) t = 5,000 s, (g) t = 6,000 s, (h) t = 7,000 s, (i) t = 8,000s .

less than half the total amount of food that can be stored when leaving the forager. For example, if the total storage amount is 0.3, the forager will leave when it is between 0.1 and 0.15. We set the forager robot to leave when the amount of food was less than 40% of the total. The number of total robots was 20, and the number of foragers was 5. For the foragers, the $\lambda_0$ value was reduced to reflect the characteristics of delivering more food, while for the non-foragers, the $\lambda_0$ value was increased to reflect the characteristics of delivering less food. In this case, our results are similar to the those real ants

Our simulation moves the five foragers to a food source outside and returns them to the nest to deliver the food to non-foragers. Figure 3.5 illustrates the sequential progress. Initially, there is no food. However, foragers obtain food from the food source, return to the nest, and deliver the food to the non-foragers they meet. When the forager's feed is too low, the forager returns to the food source and obtains food. This is repeated. When enough time has passes, all non-foragers in the nest are fed. In other words, colony state $F(t)$ converges to 1. Real ants obtain information locally and determine $\nu$. However, foragers do not know the amount of food required by all non-foragers, but bring back as much food as they require. As a result, the forager appears to behave as if we has access to global information, colony state $F(t)$. To confirm this, we examined the relationship with colony state $F(t)$ sequentially.

### 3.3.2 Interaction volume and rate

In the current system, it is important how much food the forager and non-forager deliver during an exchange. In other words, the characteristics of the system depend on the value of interaction volume, $\nu$. The time the forager feeds and feeds other non-foragers is also affected. We used robot simulation to determine how these two parameters are related to the colony state, $F(t)$.

The results are illustrated in Figure 3.6 and reveal that. the iInteraction volume decreases as $F(t)$ increases. In addition, the figure illustrates the interaction volume average of five foragers. The convergence of the colony state to a value of 1 indicates that the forager can no longer deliver food. These results reflect the characteristics of real ants. The reciprocal of the time taken for a forager to feed two feeds in succession is defined as the interaction rate. That is, it refers to the time taken for the forager to make an exchange with the next non-forager based on the first exchange with a non-forager. A low interaction rate signifies that it takes a long time. The closer the colony state is to 1, the lower the amount of food that non-foragers can receive. The forager must deliver the food and feed the non-foragers; however, the surroundings are full and it therefore takes more time to deliver.

Our results indicate that as the colony state $F(t)$ increases, the interaction rate decreases. However, there is a difference between real ants and the robot simulation. When a forager leaves through the entrance, it sees the difference in the amount of food the foragers have. Real ants appear to have a similar amount of food when they

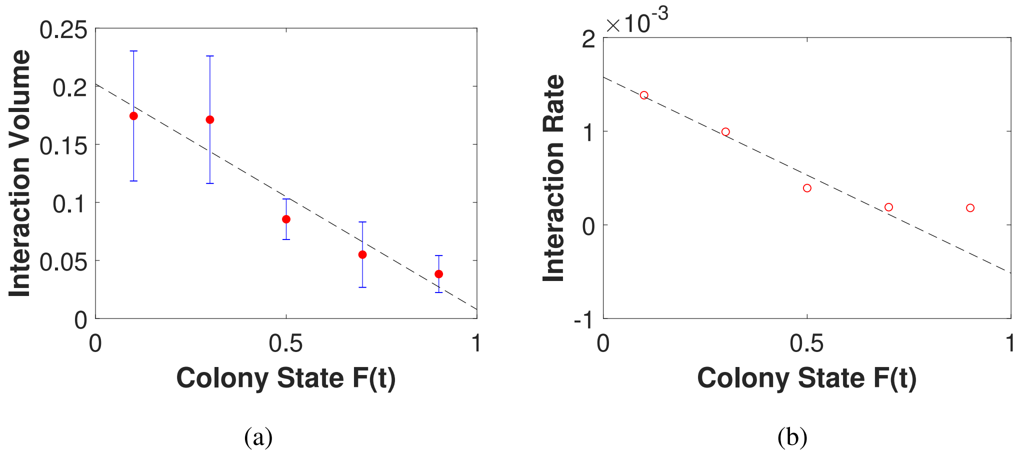



Figure 3.6: Linear relationship between interaction volume and interaction rate according to colony state. (a) Each point is an average of five foragers (mean $\pm$ std). As colony state $F(t)$ increases, the interaction volume decreases. Linear fit $y(F) = -0.1942 * F + 0.2020$ ($R^2$ = 0.9127). (b) Each point is the average of five foragers. As colony state $F(t)$ increases, the interaction rate decreases. Linear fit $y(F) = -0.0021 * F + 0.0016$ ($R^2$ = 0.9699).

leave. The difference between real ants and robots stems from obstacle avoidance. In the case of a real ant, it is not as important to recognize and avoid the same ant as an obstacle because real ants cross other ants and are not significantly affected when if they hit one another. In contrast, obstacle avoidance is very important for robots. Therefore, if there is another robot in the vicinity, the robot it is unable to leave. If the robot believes that the other robot is small, then it can leave; however, it can not leave by other robots and continues to exchange. The colony state $F(t)$ is between 0.2 and 0.3, which means that there are many non-foragers that do not have food. For this reason, it can be exchanged until it approaches zero. This can lead to various forms in a dynamic situation. If a forager is not disturbed by the obstacles, then you it can exit through the entrance with a similar crop load.

### 3.3.3 Crop State and Food Inflow

It is important for robots to have all the food in the a trophallactic network system. Therefore, it is important to examine the graph of crop load, crop state, and food inflow of the forager proceeds to confirm that the robot simulation is effective. Figure 3.7 presents a graph of crop load, crop state, and food inflow. First, the crop state is $f_i(t)$

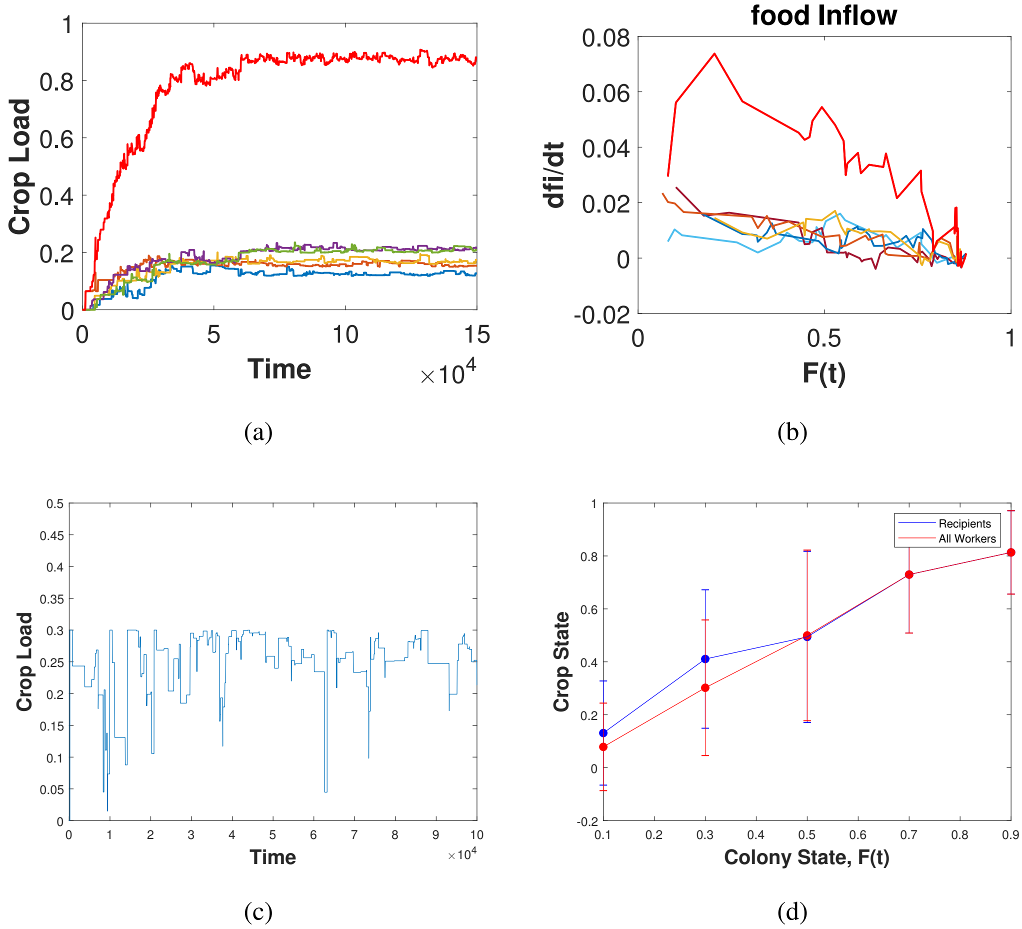



Figure 3.7: Various relationships between forager's crop load, colony state and food inflow. (a) The blue line represents colony state $F(t)$, while the remaining lines represent the amount of food that each forager delivers to the nest. (b) The blue line represents the time-average global flow $\frac{dF}{dt}$. The remaining lines represent the time-average flow $\frac{df_i}{dt}$ of each forager. The time-average global flow $\frac{dF}{dt}$, increases until all foragers start exchanging with non-foragers. After that, it continues to decrease. (c) Crop load of any forager. When exchanging, the crop load increases or decreases. If arriving at the food source, the forager is fed. Then, the crop load of the forager has a maximum value (0.3). (d) The recipients are non-foragers in interactions with foragers (blue line). All workers are all foragers (red line). As the colony state, $F(t)$, increases, foragers interact with all non-foragers.

and the colony state is $F(t)$ for each forager. The colony state $F(t)$ has a normalization from 0 to 1. Here, 0 signifies that the foragers did not deliver the food, while if 1 signifies that, the foragers delivered the food and the non-foragers in the nest are full. As illustrated in the figure, the colony state approaches 1, which indicates that the foragers fed the nest well. We set up a total of five foragers and simulated them, and the five graphs of $f_i(t)$ demonstrate that all foragers fed the nest effectively. The total amount of food delivered to the nest varies for various reasons, such as the situation of the robot and the amount of food it delivers. By (equation 3.1), the colony state is determined by the sum of each $f_i(t)$.

We then verified the relationship between the time-average flow $\frac{df_i}{dt}$ and time-average global flow $\frac{dF}{dt}$ to the colony state $F(t)$. According to Greenwald et al. (2018), $\frac{df_i}{dt}$ should exhibit a tendency to decrease with increasing colony state $F(t)$. $\frac{dF}{dt}$ tends to increase until all foragers participate in the exchange and then decrease (Figure 3.7 [b] - blue line). We plotted a graph based on the data obtained from the robot simulation. When we exchanged it by judging it locally, oscillation occurred; however, it demonstrated a tendency to decrease as a whole. The time-average global flow $\frac{dF}{dt}$ also had properties similar to those of real ants. In the robot simulation, all foragers do not initially go to the food source to make the environment similar to the real ants. This is why a forager exchanges food at different times. In other words, foragers do not exchange food at the same time, but rather, they exchange it at some interval. In the robot simulation, the interval increases up to the time of participation and then decreases continuousl.

Figure 3.7 (c) illustrates the change in the amount of food a single forager possesses. A forager's crop load is reduced or increased when exchanged for with non-foragers. When fed, it decreases, and when fed, it increases. When a forager travels to the food source and feeds on it, it rises to its maximum. Real ants may have more or less food. That is, when they bring food from a food source, it arrives in irregular quantities. However, in our robot simulation, the maximum amount of food that the robot can have is fixed, and can not be higher. In the crop load graph of the forager, the portion of the maximum feed that rises is the food source. In the current simulation, the maximum value is 0.3; thus, the part that rises to 0.3 in the graph is where the forager feeds in the food source. This value thus indicates that foragers bringing food from a food source and effectively exchange it with non-foragers.

Figure 3.7 (d) is a graph comparing a non-forager directly exchanging food with a forager and the crop load of all non-foragers. The recipient line represents the robot that exchanges directly with the forager, while the workers line represents all non-foragers. The colony state $F(t)$ is small in two cases, and as it increases, the difference decreases. Initially, it means that foragers are not equally distributed to all non-foragers. Foragers are more likely to exchange with non-foragers near the entrance when they enter the nest. For this reason, the probability of exchanging with only those non-foragers increases until the non-foragers near the entrance are completed. Because the mobility of non-foragers is not high in the current situation, there is not much exchange between non-foragers. Thus, after some time, the non-foragers near the entrance fill up, and the foragers move around. Then, as it happens to exchange with other non-foragers, the same applies for non-foragers and all non-foragers that exchange directly with foragers over time. This is demonstrated by our results.

### 3.3.4 Number of Interactions, Max Distance from Entrance and Balance

The number of interactions is the number of forager exchanges. Our results reveal a linear relationship with colony state $F(t)$ (Figure 3.8 [a]). This result implies that as the colony state $F(t)$ converges to 1, the number of interactions increases. In addition, as the colony state $F(t)$ increases, the foragers enter the nest deeply. Because the mobility of non-foragers is low, the exchange of non-foragers is low as well. Therefore, they first exchange with a non-forager close to the entrance, and when non-foragers close to the entrance begin to kick in a certain amount of time, they enter and begin to trade with non-foragers that lack food. Therefore, as the amount of food in the nest increases, it becomes deeper.

We have also confirmeded that the system is evenly distributed. Figure 3.8 (c) demonstrates how to maintain balance over time. We identified the variance of non-foragers to distinguish them. The large variance of non-foragers signifies that non-foragers do not feed equally, while a small variance signifies that non-foragers feed equally. Initially, there is an imbalance as the forager exchanges with a non-forager. Thus, non-foragers is increased. As time passes, the colony state $F(t)$ becomes 1, which signifies that non-foragers have all of their food. Thus, non-foragers have a smaller dispersion value as they obtain food. As the mobility of non-foragers is not high, the balance is dependent on the foragers. It is important to evaluate how quickly the colony state $F(t)$ increases and how uniformly non-foragers are distributed to determine the kind type of system that the robot simulation is. The average signifies how full the non-foragers are and is used to confirm that the nest is full so that it has been evenly fed, or that it is evenly distributed by exchanging with each other. We can see that the figure 3.3.5 (d) is evenly

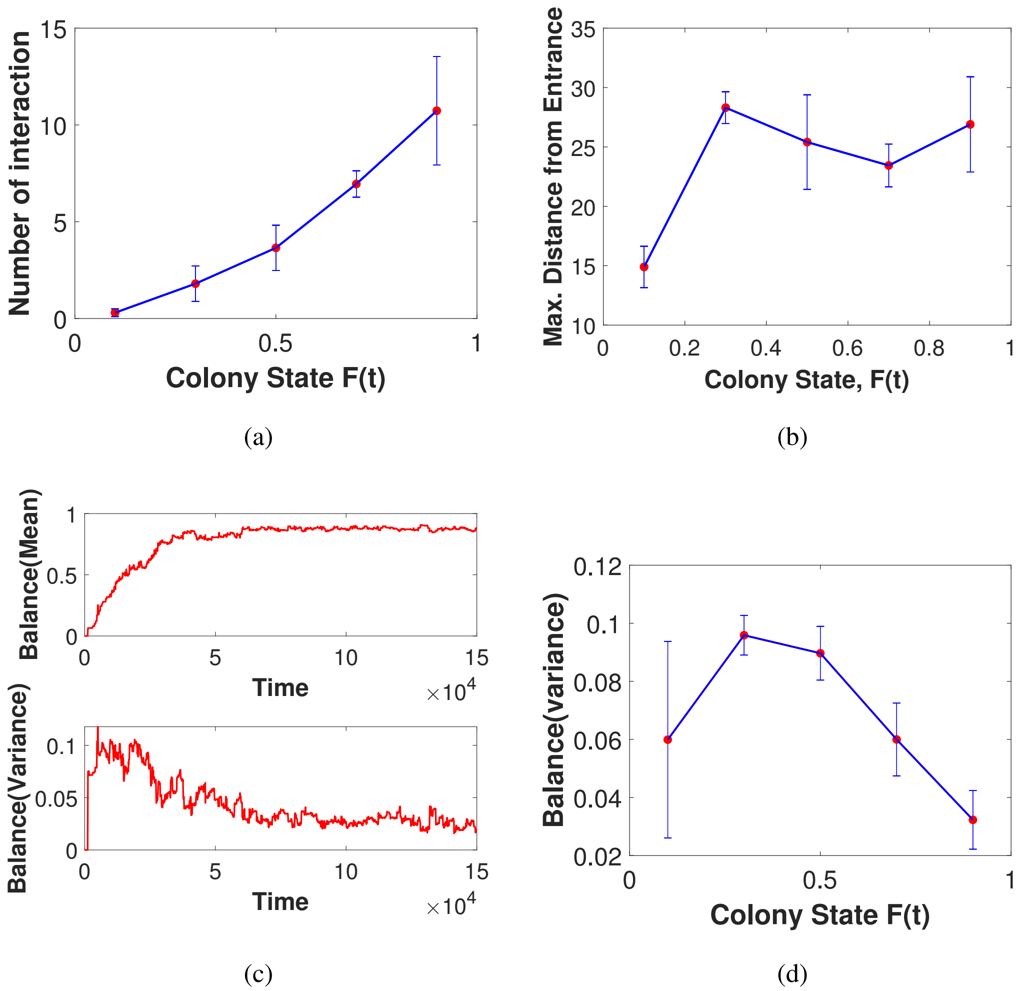


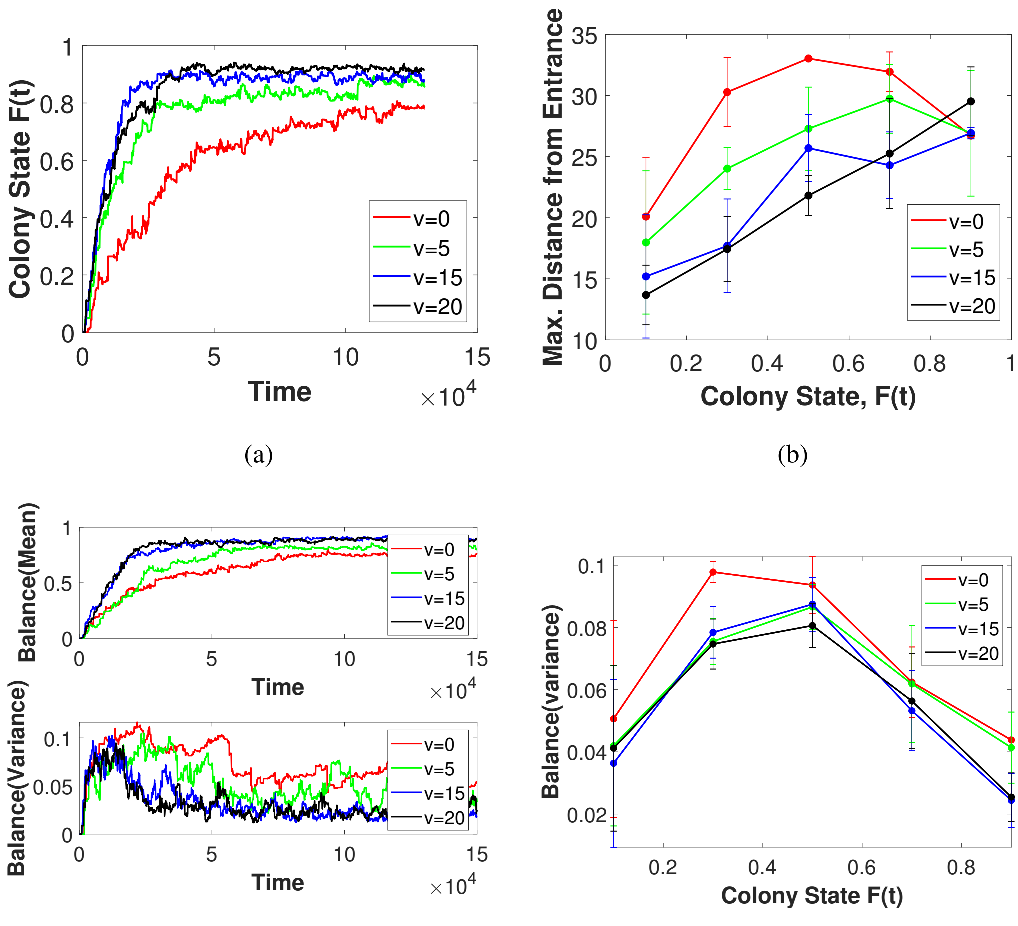

Figure 3.8: (a) Relationship between number of interactions and colony state. As the colony state $F(t)$ increases, the number of interactions linearly increases. (b) Relationship between maximum distance from the entrance and colony state. As the colony state $F(t)$ increases, the foragers enter the nest deeply. (c) Representation of mean and variance of non-foragers. Over time, the crop load of non-foragers is evenly distributed. (d) Balance is the relationship with the colony state. If all robots are full or empty, non-foragers' variance (balance) is 0. Except for this case, the graph shows the distribution of food that possessed by non-foragers.

distributed according to the feeding state of the non-forager. It clearly shows whether it is evenly distributed.

### 3.3.5 Relationship with mobility of non-forager

A system should be built according to its purpose. For example, it should be decided whether they should obtain the food sufficiently fast enough or not very well so that there are no robots that can not obtain food. We determined the characteristics of the system from these two perspectives. Among the various factors, the mobility of non-foragers greatly affects the system. The larger the non-forager mobility, the larger the dynamic element. The mobility of non-foragers in a nest is not large among real ants, and. even queen ants are at rest. We set parameters to create a similar situation, and have confirmed that a similar trend occurs. However, when implemented as a robot, mobility can be changed. Thus, we verified how the results change according to the mobility of non-foragers. Figure 3.9 reveals the changes in the colony state $F(t)$, balance, maximum distance from the entrance, and number of interactions caused by changes in non-forager mobility. The speed of the forager was set to 15, and the speed of the non-forager was changed. The speed of the non-forager was observed in four cases: a halt state, a state slower than a forager, a forager-like speed, and a speed faster than a forager. In the case of the colony state $F(t)$, the higher the velocity, the faster the decrease in variance. In other words, a high speed indicates that all robots are fed evenly. In a static state, because it is fed only by foragers, the variance cannot easily decrease unless the foragers exchange with all non-foragers. This can also been seen in Figure 3.9 (d). As a result, the higher the speed, the faster the colony state grows, and all the robots are distributed evenly as they receive food.

We also observed how the maximum distance from the entrance and the number of interactions affect the speed of non-foragers. Figure 3.9 (b) illustrates that the maximum distance from the entrance tends to be larger as the velocity decreases. If a non-forager is stopped, an exchange begins with non-foragers near the entrance. Because non-foragers can not exchange, non-foragers in the nest do not receive food if the forager does not come directly to deliver the food. Therefore, after the maximum amount of food for the non-forager nearest to the entrance is reached, the food is delivered to non-foragers inside the nest. This is confirmed by our results. Initially, it is filled from near the entrance, and when it is filled to a certain extent, it shows that it goes deeper into the nest. In contrast, as the speed increases, non-foragers near the entrance and non-foragers within the nest often change. Because non-foragers are exchanged, not all non-foragers near the entrance are fed; however, all non-foragers are fed as they are distributed to other non-foragers. For this reason, non-foragers near the entrance




(a)

(b)

(c)

(d)

Figure 3.9: Tests of four cases (v = 0, 5, 15, 20). (a) Colony state $F(t)$. As v is increases, the colony state quickly increases. (b) Mmaximum distance from the entrance. The smaller the speed, the larger the maximum distance from the entrance. (c) Having food evenly means that the variance is small. If v is big large, it becomes equalized quickly. (d) The higher the speed, the smaller the variance.

are in a state in which they can continue to feed. Therefore, even if foragers do not enter the nest deeply, they can continue to deliver food. For foragers to enter deeply, the non-foragers should be in a state in which they are nearly full and the foragers must be able to move around in the nest.

### 3.3.6 Relationship with the number of foragers

The role of the foragers is important for filling the colony state, $F(t)$. We observed how the number of foragers affected the system. In the case of real ants, there are not

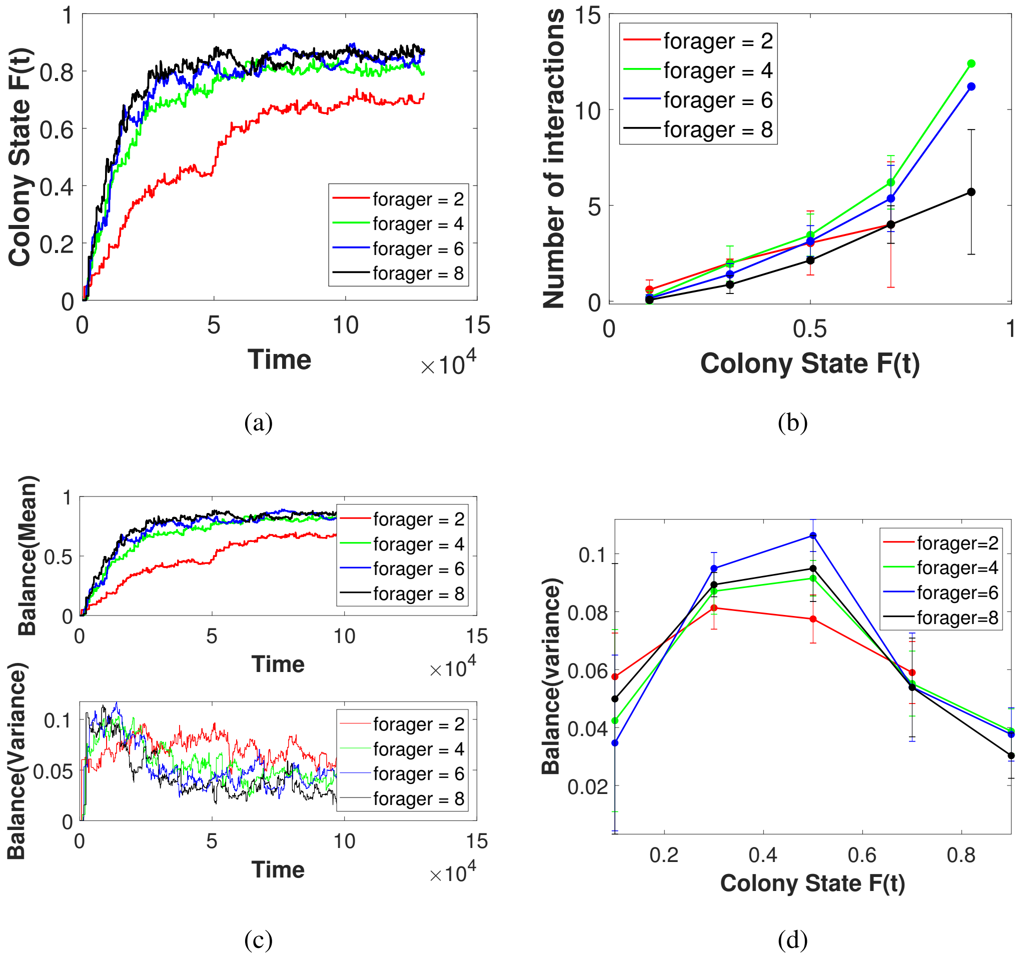



Figure 3.10: Results based on the number of foragers. (a) Colony state $F(t)$. As the number of foragers increases, the colony state quickly converges. However, the three cases (number of foragers = 4, 6, 8) are similar. (b) The larger the number of foragers, the smaller the number of interactions. (c) Graphs of the balance. The larger the number of foragers, the more evenly the food is distributed. The case with two foragers has a similar balance at all times. (d) displays the smallest variance when there are two foragers. The balance is seen according to the state of the non-forager.

many foragers relative to the total number of ants. We confirmed the reason for this in terms of the rate and balance of increasing the colony state. Figure 3.3.5 illustrates the number of graph changes for the forager. We set the speed of the non-forager to 3 to reduce the dynamic factor. Because the speed increases and the mobility of the non-forager increases, the effect of the number of foragers can not be determined because the situation changes with each execution of the simulation.

The colony state $F(t)$ increases as the number of foragers increases. However, when the number of foragers is greater than six, there is little difference. In addition, it can be seen that the larger the number of foragers, the more uniform the distribution is. Except for the case in which there are two foragers, the number of foragers generally reduces the number of interactions. When we examine each element, it can be seen that the better the number of foragers, the better the effect. However, the number of foragers does not increase among real ants. Thus, there appears to be a reason other than our theory. An ant swarm is heavily influenced by the queen ants, and the number of foragers is also significantly affected by the queen ant (Hee et al., 2000). Biologically, even if non-foragers distribute evenly, perform fewer interactions, and fill the nest quickly, ants do not choose this method for another reason. However, if absolute goals are related to this, it is theoretically better to increase the number of foragers. However, from the perspective of balance, it is favorable when the number of foragers is 2.

### 3.3.7 Diverse distribution

When a forager and a non-forager perform an exchanged, the amount delivered to the system, changes the system. We confirmed the type of patterns that appeared depending on the distribution. The distributions we applied were a uniform distribution, gaussian distribution, and exponential distribution. We compared it with the case in which the probability was not applied. $\lambda$ follows equation (3.6).

$$p(\nu|c) = \begin{cases} \lambda_c & (0 \leq \nu \leq \frac{1}{\lambda_c}) \\ 0 & otherwise \end{cases} \tag{3.7}$$

Equation (3.7) shows a uniform probability density function (PDF). We determined the maximum value according to $\lambda_c$ and the value of $\nu$ randomly between the values. The uniform distribution has the same probability of a small value and large.

$$p(\nu|c) = \frac{\lambda_c}{\sqrt{2\pi}} e^{-0.5(\lambda_c \nu)^2} \quad (\nu \geq 0) \tag{3.8}$$

Equation (3.8) presents a gaussian PDF. The mean is fixed at 0 and the variance depends on the value of $\lambda_c$. For example, if the amount of food fed by the relative robot is small, $\lambda_c$ decreases and the variance increases. The larger the variance, the higher the probability that a larger value will be obtained, which produces a larger amount of deliver food. In contrast, when the relative amount of food of the relative robot is large,

$\lambda_c$ increases and the dispersion decreases. The probability is then increased to a value close to zero, producing a small amount.

$$\nu = \frac{1}{\lambda_c} = \frac{C_0 - c}{\lambda_0} \tag{3.9}$$

Equation (3.9) is an equation that follows when robots instantly give deliver food without applying a probability. The value is determined according to the amount of remaining food of the opponent robot, and it delivers exactly that amount. In other words, it this signifies that the opponent robot knows exactly what it requires. Figure 3.11 illustrates the variation according to the distribution. The number of foragers is equal to 5, and the speed of non-foragers is set to 3 to reduce the dynamic factor. The position of the robot is also in the same state. In colony state $F(t)$, the uniform distribution is slower to fill the nest than in other cases.

However, the uniform distribution is more evenly distributed than the other cases. Because the uniform distribution is smaller than the average, the colony is smaller, but the variance is similar. This signifies that the colony state increases relatively slowly increases, but non-foragers are evenly distributed. Figure 3.11 (d) provides additional details. It indicates that the uniform distribution is the most evenly distributed, while the gaussian distribution is the most unevenly distributed.

A difference also appeared in the number of interactions. The case that no probability was applied was the smallest, while the second smallest was the number of the gaussian distribution. The gaussian distribution and exponential distribution cases filled the colony quickly but did not distribute it evenly. In the case of uniform distribution and no probability, the colony filled up slowly but evenly. However, in the case of the uniform distribution, the number of interactions was large, and when the probability was not applied, the number of interaction was generally small. However, balance and interaction numbers are influenced by non-forager exchanges, and we may thus not be able to reflect features directly in the distribution. Thus, in practice, non-foragers are perform exchanges, but non-foragers do not perform exchanges to determine the characteristics of the distribution.

There is a significant difference for each distribution when there is no exchange. In the case of the exponential distribution, the colony increases rapidly and the number of interactions is the smallest. However, it appears to be the most uneven distribution. In the uniform distribution, the colony increases slowly and the number of interactions is large; however, it is evenly distributed. This phenomenon is explained in the following

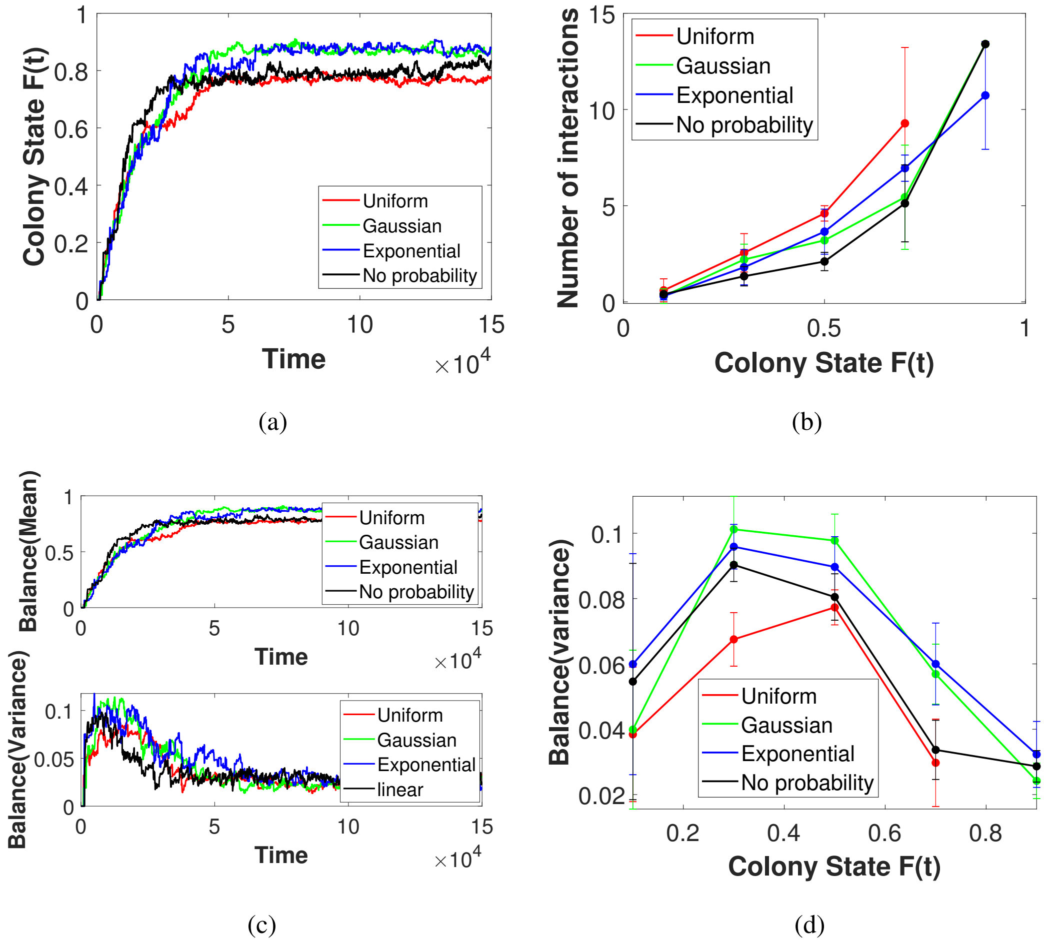
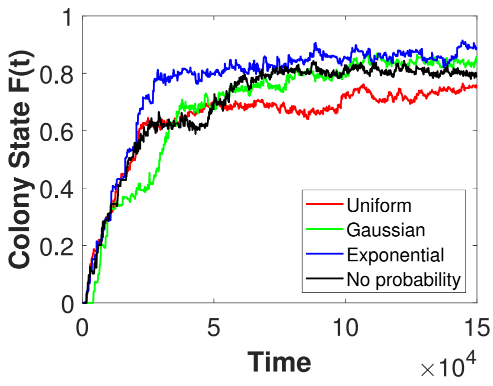
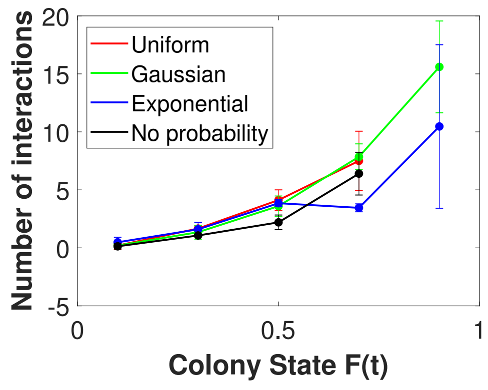


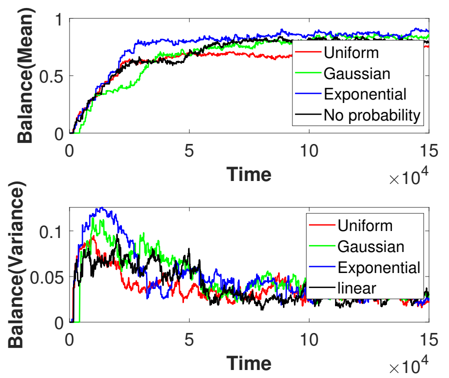

Figure 3.11: Results based on distribution when exchanging food with non-foragers. The distribution types is are uniform, gaussian, and exponential. 'No probability' signifies that the interaction volume applies without probability. uniform PDF reflects (equation 3.7), while gaussian PDF reflects (equation 3.8). exponential PDF reflects (equation 3.5), while 'No probability is reflects (equation 3.9). (a) Colony state $F(t)$. In a uniform distribution, the colony state $F(t)$ increases more slowly than in other cases. (b) The number of interactions is the smallest in the exponential distribution. 'No probability' is the smallest; however, a real ant does not know exactly what the opponent wants. Therefore, the ants select an exponential distribution because the colony is the fastest and the number of the interactions is smallest. (c) Graphs of balance. The type of distribution does not affect in the variance of all non-foragers. However, a uniform distribution has a smaller mean value than the other cases, but the variance values are similar. This signifies that the colony state is increased while being evenly distributed. (d) Uniform is distributed evenly while Gaussian is distributed the most unevenly.



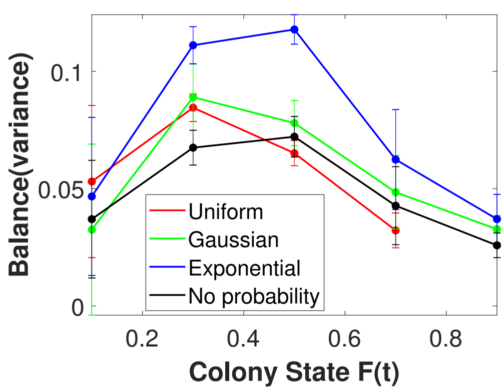

(a)




(b)




(c)




(d)

Figure 3.12: Results according to a distribution when there is no exchange between non-foragers. The distribution types is are uniform, gaussian, and exponential. 'No probability' signifies that interaction volume applies without probability. Uniform PDF represents (equation 3.7), while gaussian PDF represents equation (3.8). Exponential PDF represents (equation 3.5), while 'No probability is represents (equation 3.9). (a) Colony state $F(t)$. In a uniform distribution, the colony state $F(t)$ increases more slowly than different other cases. In contrast, the colony state F(t) is the fastest when it is exponential. (b) The number of interactions is the smallest in an exponential distribution among distributions. (c) The type of distribution does not affect in the variance of all non-foragers. However, the uniform distribution has a smaller mean value than the other cases, but the variance values are similar. This signifies that the colony state is increased while being evenly distributed. (d) The uniform distribution is evenly distributed. In contrast, the exponential distribution is the most unevenly distributed. 'No probability' is also evenly distributed.

section.

## 3.4 Discussion

We analyzed an ant trophallactic network system from various perspectives by implementing a robot simulation. The trophallactic network of real ants was analyzed in a previous paper Greenwald et al. (2018). Based on the results, we confirmed the validity of the robot simulation. We set the interaction volume $\nu$ as an exponential distribution. The graph of exponential distribution depends on $\lambda_c$, which is determined by the relative crop load. If the crop load is large, the value of the interaction volume $\nu$ is small. In contrast, if the crop load is small, the interaction volume $\nu$ is large value. The value of $\nu$ is related to colony state $F(t)$. When the value of colony state $F(t)$ increases, the value of $\nu$ decreases. A large value of colony state $F(t)$ signifies that most non-foragers have a lot large amount of food. In other words, the number of non-foragers that foragers can deliver to in large quantities is small. In addition, the interaction rate decreases. The interaction rate is the reciprocal of the time taken for the forager to feed non-foragers two consecutive times. A small value signifies that it takes a long time to deliver the food and deliver the food to the next non-forager. Because there are many non-foragers with sufficient food, it takes a long time to deliver the food.

The colony state $F(t)$ and time-average flows $\frac{df_i}{dt}$ drawn by the robot simulation showed similar tendencies to the actual ant trends of real ants, thus indicating that the robot simulation performed well. The graph of the number of interactions and the maximum distance from the entrance illustrates the situation of the robot simulation. We demonstrate how we can change features that can not be changed in real ant colonies and apply them these changes to real situations while proceeding with the robot simulation.

We verified how the system varies was affected with by changes in the number of non-forager mobility, number of foragers, and various distributions. In the case of non-forager mobility, the exchange between non-foragers is small when the speed is low. This means that the forager must be fed directly. Our results demonstrate that the larger the speed, the faster the colony state increases and the non-foragers become evenly fed. In addition, it does not go deep into the nest if it is relatively fast. A non-forager with no food approaches the entrance, or a fed non-forager gives food to other non-foragers

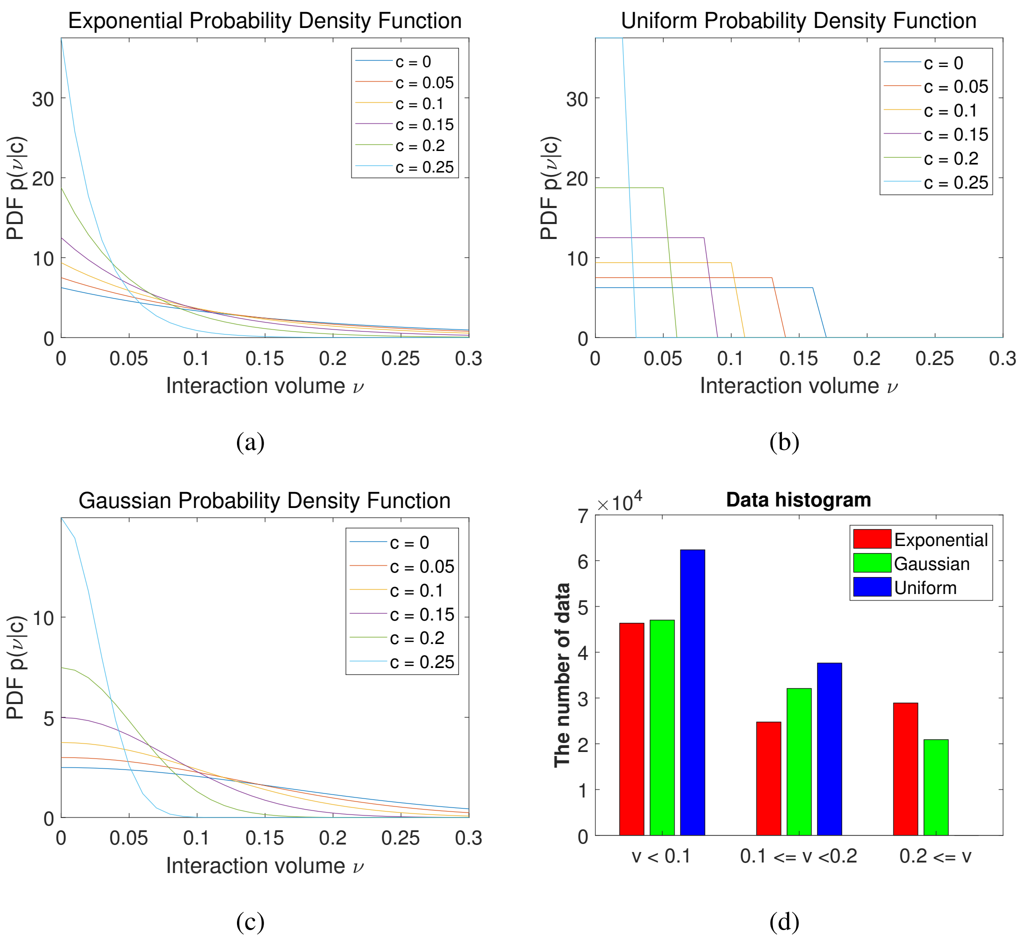



Figure 3.13: Probability density function according to the crop load. The larger the value of c, the higher the probability that the value of the interaction volume is small. (a) Exponential distribution. (b) Uniform distribution. (c) Gaussian distribution. (d) Histogram by size of generated data values when randomly generating 100,000 data points with the same lambda value. The gaussian distribution is more likely to have a smaller value, while the exponential distribution and uniform distributions are more likely to have a larger value. (Exponential, Gaussian, Uniform maximum value : (1.7372, 0.6948, 0.1333.)

so that they can continue to feed. Then, the forager's food is consumed quickly and can not be heard deeply. The number of foragers is related to the increase rate of the colony state. However, it does not significantly affect the balance. One way is to increase the number of foragers to quickly fill the colony state.

We observed how the system varies with the distribution. We have also confirmed that there is no exchange between non-foragers to better characterize the distribution. In the case of an exponential distribution, the colony increases rapidly, and the number of interactions is small but disproportionately distributed. Next, the gaussian distribution increases rapidly and distributes evenly; however, the number of interactions is high. Finally, the uniform distribution is the slowest, and the most interactions are distributed evenly. Figure 3.4 illustrates the reason for these results. Figure 3.4 (a)–,(b),(c) present the distribution according to crop load c. As the distribution changes, the value of the interaction volume $\nu$ is determined. Figure 3.4 (d) provides the reason for this. The exponential distribution and uniform distributions have a high probability of a relatively large value, while the gaussian distribution has a high probability of a small value. Therefore, the gaussian distribution is distributed evenly to non-foragers. However, even if you hand over a lot, the value is not sufficiently large to cause frequent exchanges; thus, the number of interaction seems appears to be increased. The exponential distribution leads to a large amount of food because it has a high probability of giving it often. Therefore, it has the characteristic that it can not be distributed evenly. Instead, it quickly fills the colony with a small number of interactions. However, the uniform distribution is likely to have a large value, but can not fill the colony quickly or evenly. The interaction is also large.

When comparing the maximum value of each distribution, the uniform distribution has the smallest value. This causes the colony to fail to fill quickly and increase the number of interactions. In other words, the uniform distribution is not very different from the gaussian distribution because it has a high probability of giving a large amount, but its value is not large. As a result, if the food is delivered at a large value, the colony can be quickly filled with a small number of interactions; however, it can not be evenly distributed. Nevertheless, when non-foragers exchange food with each other, the probability of being evenly distributed increases, and the variance decreases.

We were able to identify the the characteristics of the ants. In the paper Greenwald et al. (2018), ants followed the an exponential distribution, and. we could were able see the reason for determining the interaction volume by exponential distribution through robot experiments. In the case of ants, the goal is for the foragers to forage at once and quickly deliver the food to the colony at the food source. However, if one feeds a lot, it will not receive much; however, the fed non-foragers will exchange with other non-foragers to reduce the uneven distribution of food. The distribution of the exponential distribution displays the effect of feeding the colony quickly while determining the distribution. The robot simulation also reveals the possibility of using different distributions according to priority.

## 3.5 Summary of Chapter 3

In this chapter we observe the movement of ants by applying swarm robotics to a real trophallactic network system. In this paper, we apply the formula to determine the amount of food delivered by the forager, which is suggested in the paper (Greenwald et al., 2018). Ants determine the amount of food they deliver depending on each other's crop load during trophallaxis. The decision is then made randomly according to the exponential distribution applied. The value passed depends on the colony state. These two relationships represent a linear relationship with each other. Our robot simulation results also show the same results.

We propose that the exchange of non-foragers as well as foragers has the same rules. In the real ant exchange of forager and non-forager, the forager tends to feed more. Our simulation applies the same rules for non-forager exchanges. Although exchanged with the same rules, the results are similar to those obtained from actual ants. It acts only with local information, but shows the behavior of moving with knowing the total amount of food. We not only look at the foragers' movements with information, but also identify which criteria they want to gain from the group. Our point of view shows whether the speed of filling colony food is important or whether it is important to have food evenly. Basically, it's hard to have food evenly at first when it's been a while. As the colony is filled, it is balanced. We analyze the rate and balance of colony prey.

After all, each individual's ability is affected. Real ants have their own abilities. But in the case of robots, they can change their capabilities and adjust the number of robots appropriately. We analyze according to the speed of the non-forager, the number of foragers, and the distribution it delivers. Our result is that the non-forager speeds up the colony faster than the non-forager speeds. And more quickly balance the food. A velocity of zero corresponds to chapter 2 2.7. Because the foragers must deliver everything directly, it is difficult to fill the colony quickly and deep into the nest. We show the advantages and disadvantages of trophallaxis delivery through the speed of non-forager.

This time, it shows how the number of robots should be used as a forager when the total number of robots is set. If the number of foragers is large, they often go for their own food. So the colony fills up quickly. However, at some point, the rate at which colonies are filled does not vary significantly. When the number of foragers is too small, they cannot feed evenly. The problem of balance is also bad. The colony speeds up when the number of foragers is large, but it works best when the number is appropriate. The ant determines the amount of food to be delivered as a random variable for the exponential distribution. To confirm this reason, we applied a different distribution to observe the effect. The exponential distribution does not show much effect if the non-forager exchanges it. However, it is evident when non-foragers do not exchange. Forager's position shows a tendency to spread evenly. When the non-forager has no exchange, our results confirm that the colony fills quickly when using the exponential distribution, but everyone has a balanced diet. We tell you what criteria ants actually transmit trophallaxis and what should be considered when applying it to robots.

# Chapter 4

# Trophallaxis for recipient

In this chapter, we present our findings following the use of the trophallactic network system for purifying contaminated areas. Trophyallaxis is one of the ways insects obtain food. Insects bring some to their nests in search of food. Foragers can deliver food from mouth to mouth until it is passed round all the insects in the colony. This way, the amount of food required can be brought into the nest without prior knowledge of the exact quantity required. In a similar way, some robots are assigned to fetch the pollutants. These robots are referred to as the foragers, those that perform the clean-up in the clean areas are called the non-foragers or recipients. Foragers move out of the clean area to find and bring contaminants. The reason why only a few robots choose to search outside is that contaminated areas are defenseless against external threats.

This can reduce minimize loss. Furthermore, because the contaminated area is not deep, it is possible to quickly move contaminants out into the clean area. In Chapter 3, we presented the results of observing and analyzing the foragers' behavior. In this chapter, we present the details and analysis of how the recipients, not the foragers, receive information to determine their movements to perform their task. The recipients not only deliver pollutants by one-to-one exchange, they also share the location of pollutants among themselves to device clean-up strategies accordingly. Chapter 4 is presented in the following order: simulation, method, result, and discussion. Under simulation, we describe the behavior of the foragers and the recipients, and show the relationship between the location of the contaminant and the behavior of the recipient. Then we present the result, which is the evaluation of the success of the mission was done and the swarm movement of the recipient. Finally, under discussion, we explain the effects and analyze their implications.

## 4.1 Robot Simulation

Our simulation is based on deploying a swarm robotic strategy to purify contaminated areas. Our goal is to clean up all the selected contaminated areas. Using trophallaxis, foraging Insects insects search and find the amount of food they can bring with them. On their return, they pass the food from one non-forager's mouth to mouth to the other in the nest. This action is repeated until the food is duly passed round. We use simulate this principle to move pollutants from contaminated areas to clean areas. Accordingly, we call the robots that go to a the contaminated area to fetch the contaminantsa foragers, and those that receive and purify the contaminants from in the purification area are the non-foragers or recipients. We examine the success of this activity in two scenarios, with and without location information.

### 4.1.1 Scenario 1

In the first scenario, the contamination is restricted to a defined location. At the end of the search, it is assumed that the foragers know the contaminated area. The foragers go to the contaminated area, and return to the purification area with as much contaminant as they can fetch. These contaminants are delivered to nearby recipients returning to the purification area. These recipients randomly proceed from the area of contamination to other recipients; the recipients randomly proceed to a purification area where the contaminant is shed. Foragers help to remove the contaminants in other areas. In this set-up, we observe and analyze the difference made by the availability of location information.

### 4.1.2 Scenario 2

In the second scenario, the contaminated area is not defined. Unlike the first scenario, the foragers fetch randomly moving contaminants. Then, they bring the contaminants to clean areas, and deliver them to the non-foragers. Because the pollutants also come from random sources, when the foragers pass on the contaminants to the non-foragers,

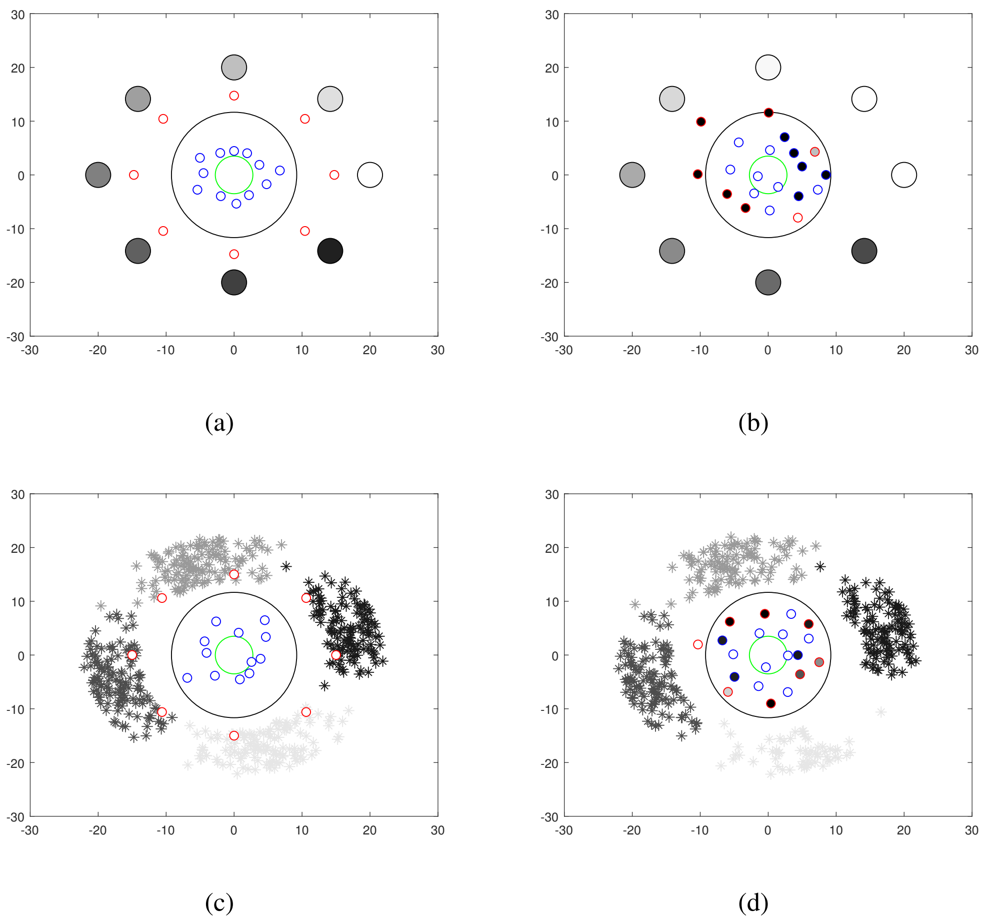



Figure 4.1: The experimental environment. The number of robot is 20 (forager = 8, non-forager = 12). Red robots are foragers and Blue robots are non-foragers. Contaminated areas are spaced at regular intervals on a circle with a radius of 20. The most contaminated areas are filled with black and the areas without any pollution are filled with white. The black circle in the center means clean area and the cyan circle means purification area. (a) The initial state in scenario 1. (b) Situation after some time in scenario 1. (c) The initial state in scenario 2. (d) Situation after some time in scenario 2.

they deliver not only the pollutants but also share the location they were obtained from. In this scenario, we consider the effects of receiving positional information as well as pollutant information.

### 4.1.3 Environment

The experimental set-up consists of a clean area and a purification area. The recipients only move within the clean area, where they wait to receive the contaminants. The purification area is the area where the robots can completely remove contaminants. The recipients move randomly through the clean area, and then pass through the purification area to completely shed the contaminants. The clean and purification areas are both circles with respective radii of 10 and 2. The forms of the contaminated areas vary across both scenarios. In Scenario 1, the pollutants are distributed in a specific area. Each region has a uniform distribution of eight regions on a circle with a radius of 20. Each region has a different level of pollution. Foragers bringing pollutants to the outside non-foragers act on the assumption already know these areas.

In Scenario 2, the contaminants are widely distributed. The contaminant particles are widely spread indistributed within a circle with a radius of 20. These particles form several groups. The degree of contamination of a particle varies according to the size of the group. Furthermore, these particles drift around randomly and have no fixed location. Air pollution pollutants, for instance, are carried by the wind. In Scenarios 1 and 2, the darker the color, the greater the contamination. This means that it takes a long time to purify all of them because of the limited amount of contaminants that can be fetched. The simulation is basically observed using a camera. It is assumed that the robot knows its position. Other than this, it is impossible to know any other information except by obtaining it directly through the sensor.

### 4.1.4 Robot

The robot is circular in with a radius 1. There are two types of robots: a forager that goes to a contaminated area and removes contaminants, and a recipient that receives the contaminants from the forager in a clean area, and sheds them in a specific area. The forager is represented by the red circle, and the recipient, by the blue circle. When there are no pollutants, the foragers fetch them from the polluted area. When a forager meets a recipient, it delivers the pollutant.

If it the pollutant cannot be delivered, the forager finds another recipient, and delivers it. The degree of contamination is expressed in various brightnesses from white to black. The recipient does not leave the clean area. Rather, it moves randomly inside, and receives the contaminants from the forager or other recipients. It may also deliver contaminants to other recipients. Then, passing through the purification area, it sheds the contaminants. Each robot has an infrared sensor by which it can sense its surroundings. Our robot is a flyable robot and that ignores collision avoidance when viewed in 2-dimension. This way, the effectiveness of swarm activity can be better observed. Therefore, infrared sensors are used to find robots to convey the pollutants and provide information or to detect the contaminated section in an area. The sensor has eight sensors receivers around the robot. The sensor distance range is 2 (m); among all the robots detected, information is exchanged with the closest. The behavior of the robot is described in detail in the following section.

## 4.2 Method

### 4.2.1 Pollutant Exchange

The trophallactic network system is based on delivering energy. Additional information is obtained as energy is delivered. Our system also uses this principle to eliminate pollutants. The amount of pollutants exchanged in the multi-robot system is based on the exponential distribution used in Chapter 3. The exponential distribution ensures that delivery to the recipients is swift, and all the recipients are evenly distributed.

$$p(\nu|c) = \lambda_c e^{-\lambda_c \nu} \tag{4.1}$$

The equation 4.1 represents the exponential density function that determines the amount of pollutant to be delivered. Probably randomly determines the value of $\nu$. In this case, “c” means represents the amount of pollutants possessed by the other robot. Therefore, when the amount of pollutants possessed by the other robot is determined, the amount of pollutants to be delivered at that time is determined. Thus, the probability changes according to the crop load, that is, the amount of pollutants possessed by the robot in the opposite category. The variable associated with the crop load is $\lambda_c$.

$$\lambda_c = \frac{\lambda_0}{C_0 - c} \tag{4.2}$$

Equation 4.2 is the $\lambda_c$ equation that determines the probability. $C_0$ represents the maximum value that can be possessed by the robot. Therefore, the more pollutants a robot has, the less $\lambda_c$. Consequently, the amount of pollutant to be delivered is regulated. In other words, if the amount of contaminant possessed by the recipient is large, fewer contaminants will be delivered.

### 4.2.2 Pollutant Position Information

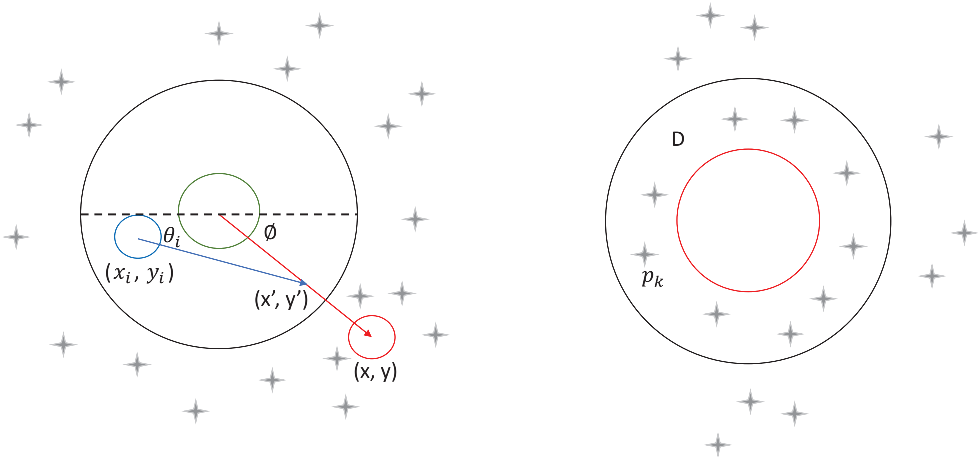
We demonstrate that robots can not only deliver pollutants but also deliver the location information necessary to perform their tasks more efficiently. Our robots do not know one another's situation except it is shared directly by robot-to-robot communication. When the forager obtains a pollutant, it memorizes the location coordinates of the area. When the forager meets the recipient, it delivers the location information together with the pollutant.

$$\begin{aligned}
&\phi_p = atan2(y_p,\ x_p)\ \ (-pi \le \phi \le pi\ ) \\
&x' = r * cos(\phi) \\
&y' = r * sin(\phi) \\
&(\ r\ is\ a\ random\ number\ along\ the\ uniform\ distribution.\ \ 2 \le r \le 10\ )
\end{aligned} \tag{4.3}$$

The equation 4.3 shows how the recipients process the pollutant location information delivered to them. The recipients store this information $(x_p, y_p)$ . It then finds a random area on the line between the purification area and the area from which the pollutant was taken. After moving to an area, they wander around the area for some time. This behavior continues as the robots constantly exchange objects and information. However, if there is no contaminant to be delivered, and no exchange occurs, the robots will randomly move to the clean areas.

$$\theta_i = atan2(y' - y_i,\ x' - x_i) \tag{4.4}$$

The equation 4.4 determines the robot's head direction. This location information is also delivered when the recipients interact. The location $(x_p, y_p)$ of the pollutant received from forager is delivered as it is. Therefore, the recipients move randomly and accidentally meet, acting on the location information of the pollutant delivered by the forager. Because this is a geographical exchange of information, it is only necessary to determine which region to move to by according to the information obtained through robot-to-robot communication.




(a) (b)

Figure 4.2: Recipient's movement using pollutant position information and forager's sensing. (a) Recipients receive location information (x, y) and use it to move to (x ', y'). (b) Show the situation when the forager finds contaminated particles. The forager collects only until the sum of the contamination levels $p_k$ is less than $C_0$. D stands for the sensor area of the robot.

### 4.2.3 Forager's Grasp

When the foragers move outside and find contaminants, they fetch as much as they can. In Scenario 1, as much as $C_0$ will be fetched if more than $C_0$, contaminants are available, depending on the degree of contamination in the contaminated area. In Scenario 2, on the other hand, when the robot finds a contaminant using a sensor, it returns with as many contaminants as possible. In other words, it does not always return to $C_0$ when you go back.

$$C_i = \sum_{k=1}^{N} p_k \quad (C_i \leq C_0 \text{ , } p_k \textit{ must be in region } D.) \tag{4.5}$$
$$\textit{Region D for the sensor area of the robot.}$$

Equation 4.5 represents the amount of contaminant particles taken by the forager. $C_i$ is the crop load of the forager, and $p_k$ is the contamination level of the contaminated particles. Because the maximum amount of crop load a robot can carry is $C_0$, the sum of all the contaminants cannot exceed $C_0$. In addition, these contaminant particles can only be taken if they are present in the area known to the robot. Figure 4.2 shows the situation described in the method. Figure 4.2 (a) shows the movement to the area where the location information of the pollutant is received. Figure 4.2 (b) shows a situation where a the forager brings contaminants when it moves to a contaminated area. The two cases represent Scenario 2, and in (b), even if the contaminants are located by the sensor, if they exceed the $C_0$, only part of them are not brought in. When the forager goes out again, it moves to the remembered contaminated area from which it commences its random search.

## 4.3 Simulation Results

We purified a contaminated area apply using a trophallactic network system. We have 20 robots in total, 8 foragers and 12 non-foragers. The foragers fetch pollutants from the polluted sections of each area, and deliver them to the recipients in the clean area. The amount of pollutant delivered to the recipient by the forager is determined by the equation 4.1. These recipients move randomly within the clean area, and then proceed to purification area.

We perform the task of eliminating the contaminants in two scenarios using the trophallactic network system. In the first scenario, there are eight sections in the contaminated area. In the second scenario, the contaminants are widely dispersed. Scenario 1 is a special situation, and Scenario 2 is a general situation. The default behavior of the forager is to deliver pollutants, deliver the location of the pollutant, bserve the recipients' movements, and analyze their effect on the mission.

### 4.3.1 Result of Scenario 1

The task in the first scenario is performed in a defined contaminated area. The location of each contaminated area and the degree of contamination vary. The foragers move to the polluted areas in their respective locations, and fetch the pollutants. At this point, if the amount of pollutant exceeds the maximum crop load, the maximum crop load is fetched. Once a given polluted area has been cleaned, the foragers move to other polluted areas to help fetch the pollutants. When the forager returns to the clean area, it meets the recipient and delivers the pollutant according to the equation 4.1. We compare this with when the location information of the pollutant was brought and when it was not.

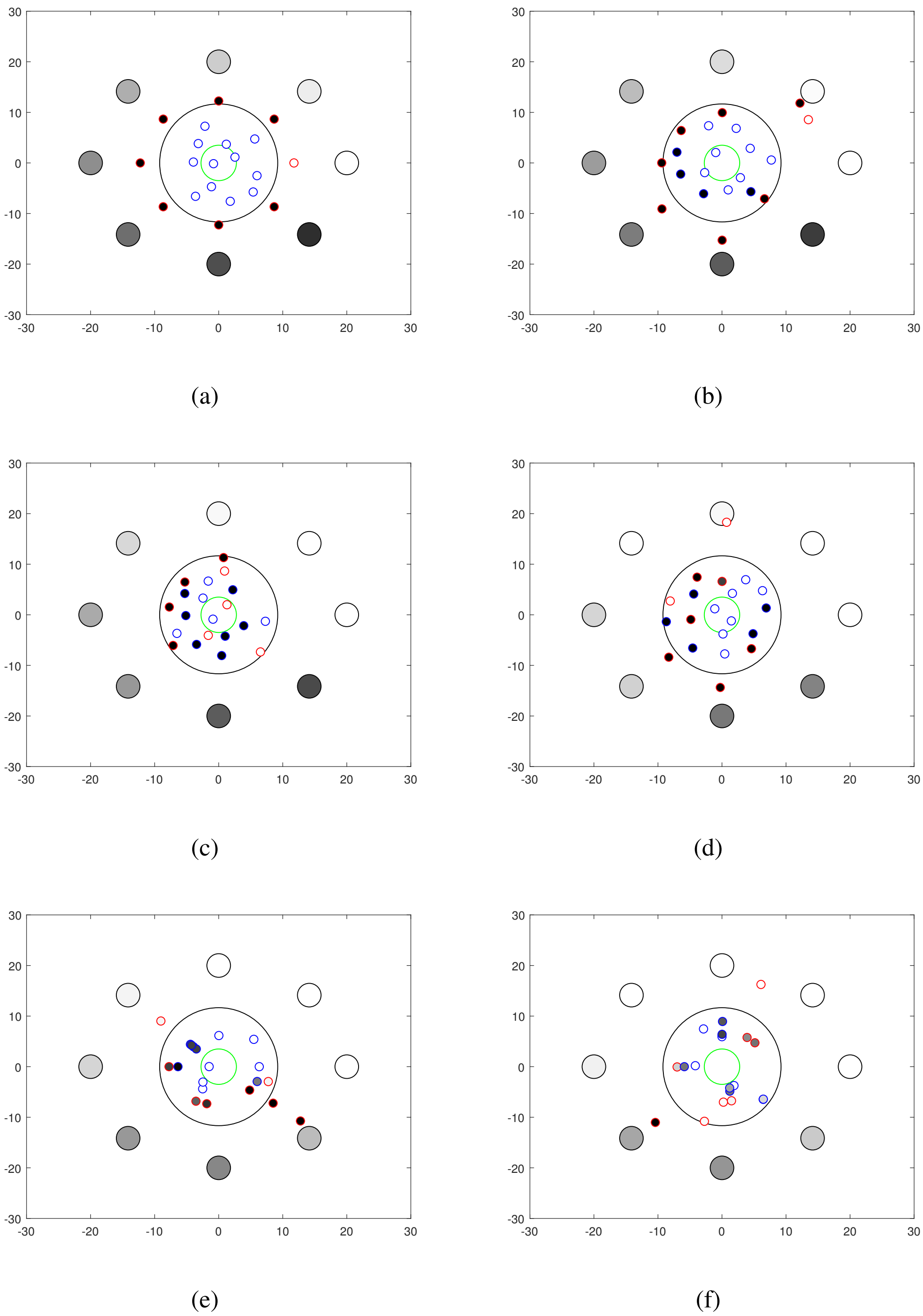



Figure 4.3: The squence of robot simulation. (a) and (b) are initial situation. (c) and (d) show when there is no location information, and (e) and (f) show when sending and receiving location information. (a) 100 (s) (b) 200 (s) (c) 400 (s) (d) 800 (s) (e) 2000 (s) (f) 2500 (s)

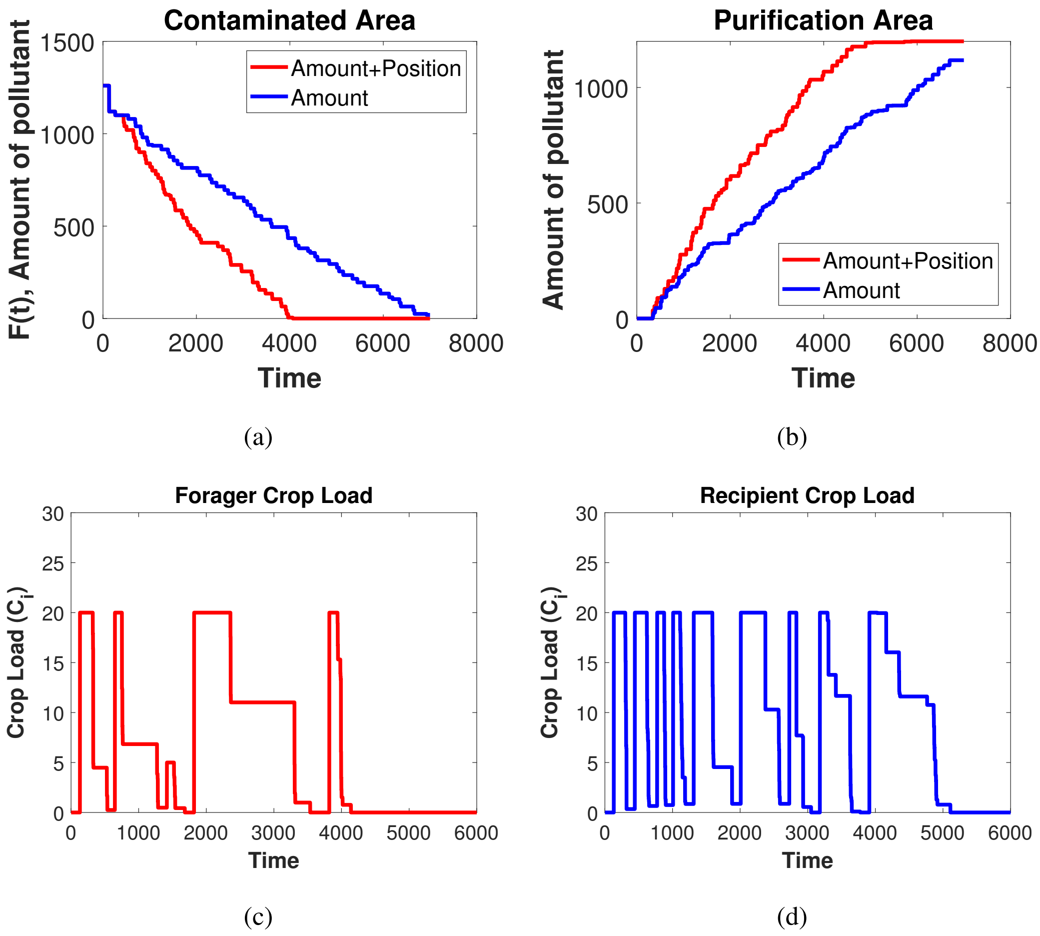
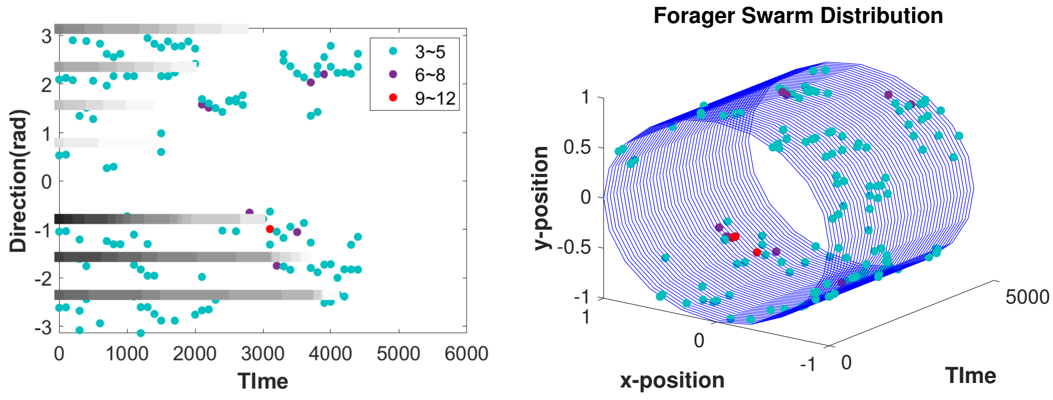



Figure 4.4: Change of pollutant. (a) shows that the contaminated area is being cleaned. It also shows that when pollutants as well as location information are shared, they are removed faster. (b) shows the amount of pollutant entering the purification area. Sharing pollutant and location information together will be faster. (c) shows the crop load of the forager. (d) shows the crop load of the non-forager.

Figure 4.3 shows the simulation set-up. Blue represents the recipient, and red, the forager. The green circle in the center represents the purification area where the robots can completely get rid of the contaminants. There are eight areas of contamination. Each contaminated area is contaminated to a different degree. The closer to black, the greater the degree of contamination. As shown in Figure 4.3, once a polluted area is completely cleaned up, the forager moves on to the next polluted area.

Over time, all the contaminated areas are clean. In addition, the recipients also shed the contaminants they received over time. Figure 4.4 shows the amount of contaminants in the contaminated area and the amount of contaminant in the purification area. We show

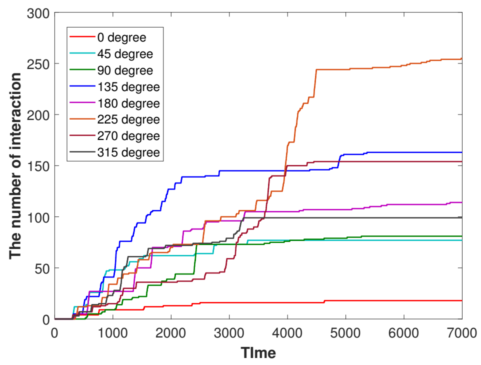



(a) (b)

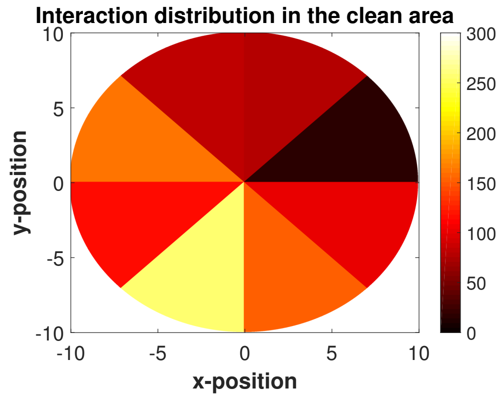
Figure 4.5: Movement of recipients and changes in pollutants by region. (a) shows the movement of a group of recipients. The color represents the number of recipients forming a herd. The direction of the center is expressed according to time. Contrasts indicate the level of pollution in each region. Black is the most polluted, and white means more purified. (b) Three-dimensional representation of the movement of recipients. This shows the actual movement in more detail.

the amount of contaminant removed and the amount of contaminant that is shed in the purification area, with and without location information. As shown in Figure 4.4, the availability of location information ensures that the pollutants are cleaned up quickly, because the foragers remove the pollutants and deliver them faster.

The crop load of the forager and the recipient. are shown in Figures 4.4 (c) and (d), respectively. An increase in the load of the forager, means that contaminants are obtained from the contaminated area. Then, the forager returns to the clean area to deliver the contaminants to the recipient. Figure 4.4 (c) shows the reduction in the forager's load. Thus, the load of foragers will rise to $C_0$ when they obtain contaminants from the contaminated area, and decrease when they get to the clean area. The recipients exchange the contaminants received from foragers or other recipients. The amount of pollutants exchanged is randomly determined along with the exponential distribution by the crop load of the particular robot. Thus, when the recipient encounters a forager or encounters a recipient with large amounts of contaminants, the crop load increases. On the other hand, if the recipient meets a recipient with less pollutants or moves to he purification area, its load reduceds. Following purification in the designated area, the crop load becomes 0.




(a)




(b)

Figure 4.6: The number of interactions by region. (a) shows interactions in each region. The area where the recipients gather generally has a large number of interactions. (b) The number of interactions in the clean area is represented by a heat map.

We show how the movement of the recipient changes if it receives location information in addition to contaminants. Recipients initially possess no information. Therefore, they moves randomly in the clean area. Then, when the foragers from the outside meet the recipient, they pass on the pollutant together with the pollutant location information. The recipients then move to a specific location in the clean area according to the equations 4.3 and 4.4. If another recipient is found, the location information from the forager is forwarded together. The recipient then moves to the specific location defined by the equations 4.3 and 4.4. If this process is repeated, the robots decide to act locally, however the effect is the appearance of global action.

Figure 4.5 shows the recipients moving in groups. Figure 4.5 (a) The angle of the direction of the location of the recipients over time. The eight contaminated areas are located at regular intervals from -pi to pi. Black represents a state of severe pollution, and the lighter the color, the more pollutants are being eliminated. Over time, the eight areas are being cleaned up. The recipients then appear to converge in the direction of contaminated area. Figure 4.5 (b) shows this in 3-dimension.

In the trophallactic network system, the interaction between robots is the most important factor. Therefore, the number of interactions correlates highly with the performance efficiency. Figure 4.6 is divided into eight regions to show where the number of interactions is highest. Figure 4.6 (a) shows the variation in the number of interactions over time. There are two types of interactions: the exchange between the forager and the recipient and the exchange between two recipients. Therefore, as the recipients gather in groups, the number of interactions in that area increases. As may be observed from Figure 4.5, the recipients usually move in groups between 180° and 315°. As a result, the interactions are represented as follows: 225°, 135° and 270°.

Figure 4.5 is a heat map of the number of interactions in the clean area. Compared to other regions, there are few interactions in the less polluted areas, 0°, 45° and 90°. The heavily polluted areas $-135°$, $-90°$ and $-45°$ show a relatively high number of interactions. We demonstrate that in heavily polluted areas, groups of recipients move closely, corresponding to a higher number of interactions.

## 4.4 Result of Scenario 2

We identify cases where contaminated areas are widespread rather than defined. In such cases, the strategy used is different from that of Scenario 1. Because no contaminated area is defined, foragers must search for the contaminants. Initially defined foragers set out in several directions as it is important that they do not gather in one location. The foragers search until they find contaminants. They move randomly. When contaminants are encountered, the foragers take the maximum amount that can be taken from the contaminants within the sensor range. The color varies according to the degree of contamination. Black depicts the worst pollution; the closer to white, the less the contamination.

Therefore, the closer the robot is to white, the greater the number of contaminants it carries, while the black ones carry smaller number of contaminants. The foragers carrying contaminants go to the clean area, and deliver them to the recipients. Foragers not only deliver contaminants, they also deliver the location information of the contaminants. By passing on the location information in this way, we observe whether the same effect as in Scenario 1 is achieved.

Scenario 2 is depicted in Figure 4.7. Figures 4.7 (a) and (b) show the initial situation. The foragers go out in different directions, and fetch the pollutants. If no contaminant is found immediately, a search is commenced. Figures 4.7 (c) and (d) depict the cases where the location information is not shared. The recipients that do not receive location information move randomly in the clean area. Thus, they can be observed to spread out widely. Recipients with location information, on the other hand, move in groups.

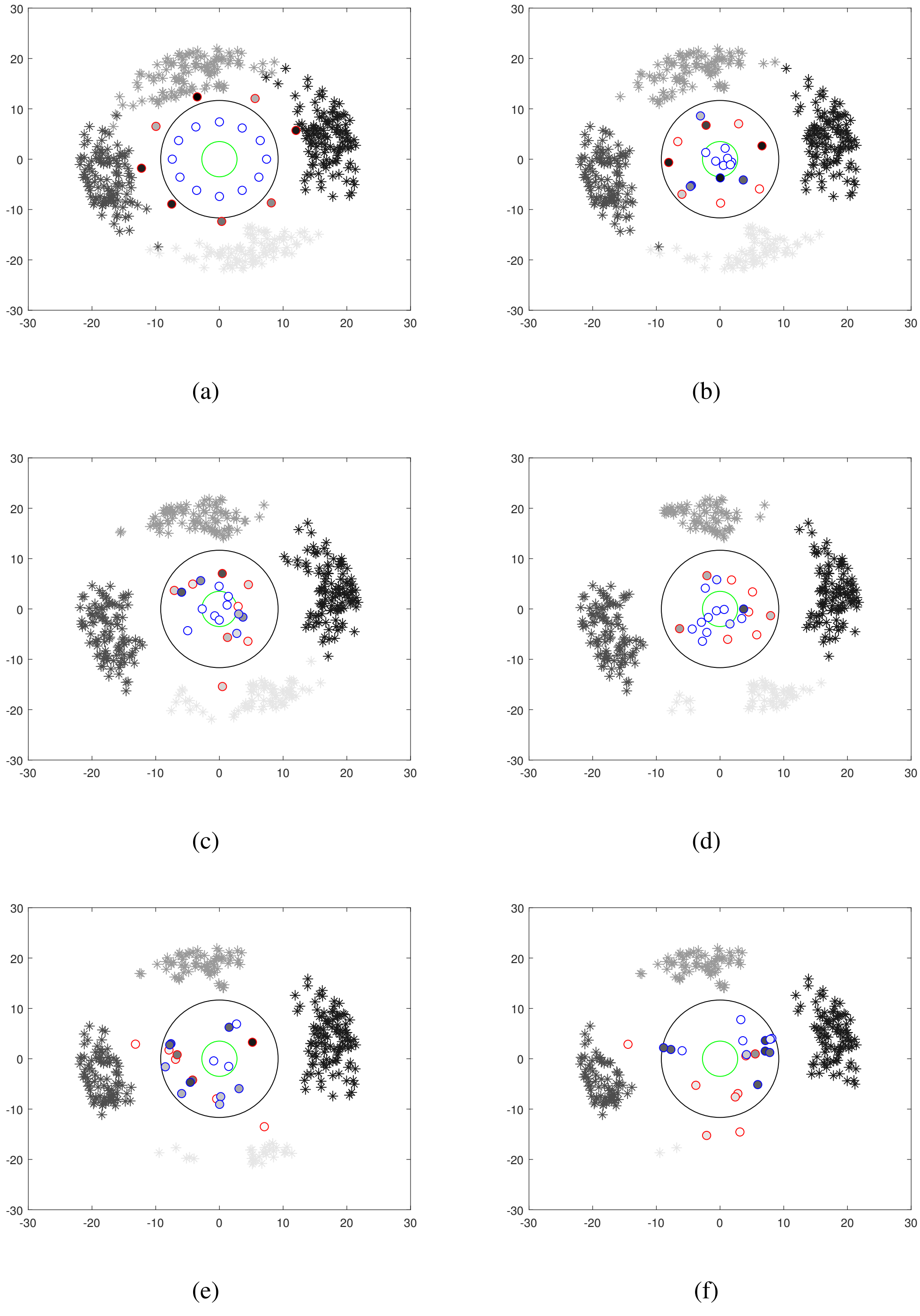



Figure 4.7: The environment for scenario 2. (a) and (b) show the initial situation. (c) and (d) show when only pollutants were shared without location information. Finally, (e) and (f) share location information and show recipients moving in groups. (a) 0 (s) (b) 200 (s) (c) 600 (s) (d) 1300 (s) (e) 700 (s) (f) 1000 (s)

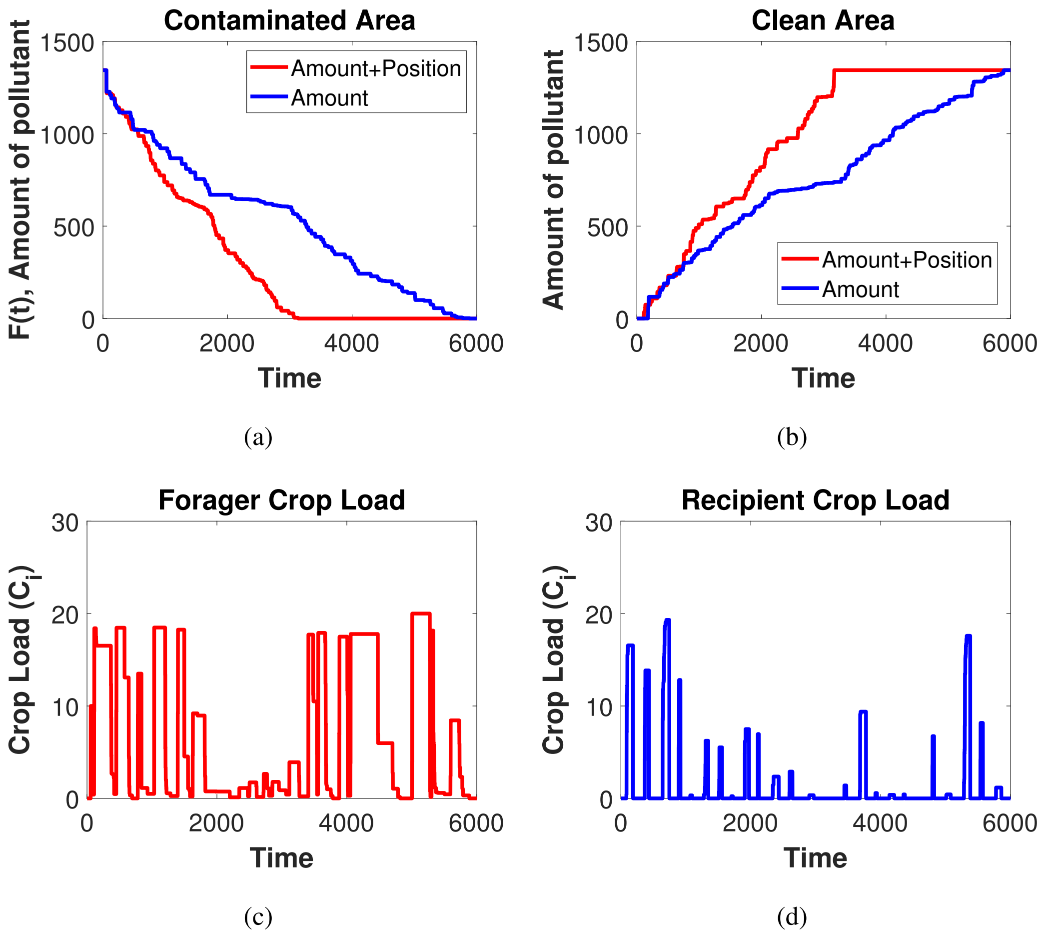
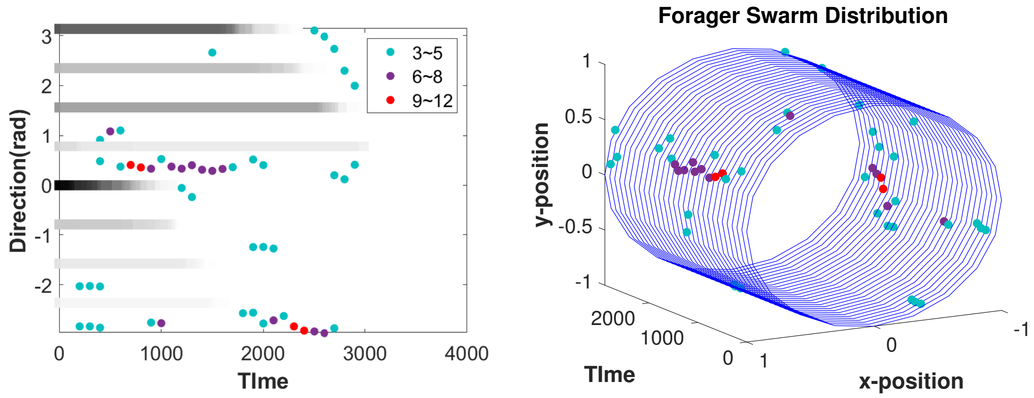



Figure 4.8: The results for scenario 2. (a) Removal from the contaminated area. If you share the location information is removed more quickly. (b) The results show the amount of pollutant delivered to the purification area. The case of sharing location information is shown to be delivered faster and cleaned up. (c) The crop load of the forager. (d) shows the crop load of the non-forager.

This indicates that they received the location information and adapted their actions accordingly.

From this, it can be deduced that the exchange of location information in this context has a better effect here, compared to Scenario 1. The speed with which the contaminant is moved from the contaminated area to the clean area is faster when location information is exchanged. Furthermore, purification is also faster. Although delivering only the contaminants makes it possible for all the robots to receive contaminants, the task is executed more efficiently when location information is shared. The findings are represented in the graph in Figures 4.8 (a) and (b). The foragers collect as many of

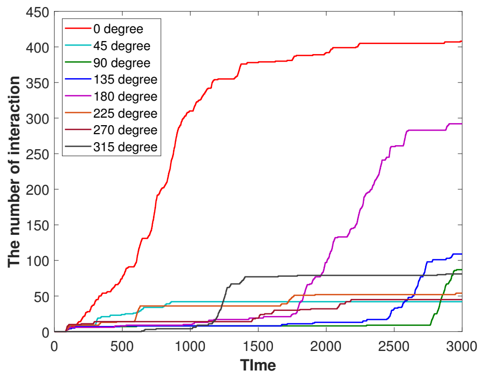



(a) (b)

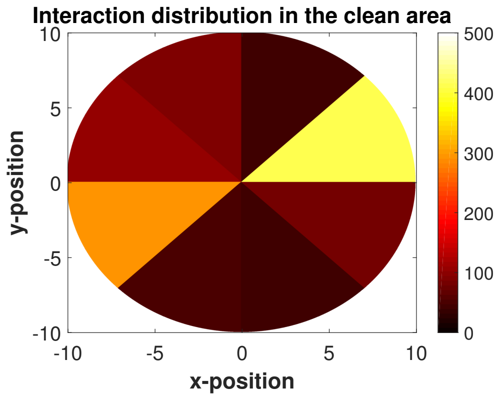
Figure 4.9: Movement of recipients and changes in pollutants by region. (a) As in scenario 1, show the movement of recipients. Color refers to the number of recipients and represents the center of the herd over time. The shaded bar graph shows the contaminant particles added by region. (b) Three-dimensional details of the movement of the recipient.

the contaminants as they can pick up. When the contaminants are obtained, they are delivered to the recipients in the clean area. Therefore, the amount brought into the clean area varies according to how much many contaminant particles are around. The recipients obtain contaminants when they encounter, foragers or other recipients with contaminants. At this point, the crop load increases, then decreases to zero as the robot passes through the purification area.

Figure 4.9 shows the location and the number of swarms in which the recipients move around. In Scenario 2, the pollutants are similarly distributed when the contaminated area is divided into eight regions, because the contaminants are widespread and the foragers move slowly. Therefore, the herd is divided into several parts. The pollutants are spread across the eight regions, and the degree of pollution of the regions are represented in black and white in contrast. Initially, the mark is based on the largest point. It shows the herd moving along in several directions. Figure 4.9 (b) shows a bunch of recipients in three-dimensions. It shows the locations where large clusters form over time, with many distributions observed in the 0° area and 180° area. This is also strongly related to the number of interactions. figure 4.9 (b) shows a bunch of recipients in three dimensions. It shows the locations that form large clusters over time, with many distributions in the 0° area and 180° area. This is also strongly related to the number of interactions.




(a)




(b)

Figure 4.10: The number of interactions by region. (a) The number of interactions over time divided into eight regions. This results in a large number of interactions where there are many recipients. (b) The clean area as a heat map for the number of interactions for each of the eight areas.

As in Scenario 1, we find that the number of interactions is related to the movement of the herd. In the present situation, the recipients are found to be distributed in the 0° area and 180° area. In this situation, the highest number of interactions also occurs in the 0° area and 180° area. These movements will have a good effect on the mission. We analyze the observations made in the experiment in the following section.

## 4.5 Discussion

### 4.5.1 Relationship between interaction number and swarm

Our results show a strong correlationship between the swarm movement of the recipients and the number of interactions. The location information of the pollutant delivered to the recipient allows the pollutant to move to the direction from which the foragers approach. This helps the foragers to deliver the pollutants quickly. And by delivering to other recipients, it helps to receive large amounts of contaminants. This may cause an increase in the number of interactions.

Figure 4.11 shows the recipients' movements and interactions when they do not exchange positional information. Figure 4.11 (a) and (b) depict Scenario 1, and (c) and (d) depict Scenario 2. The simulation results show that when the location information

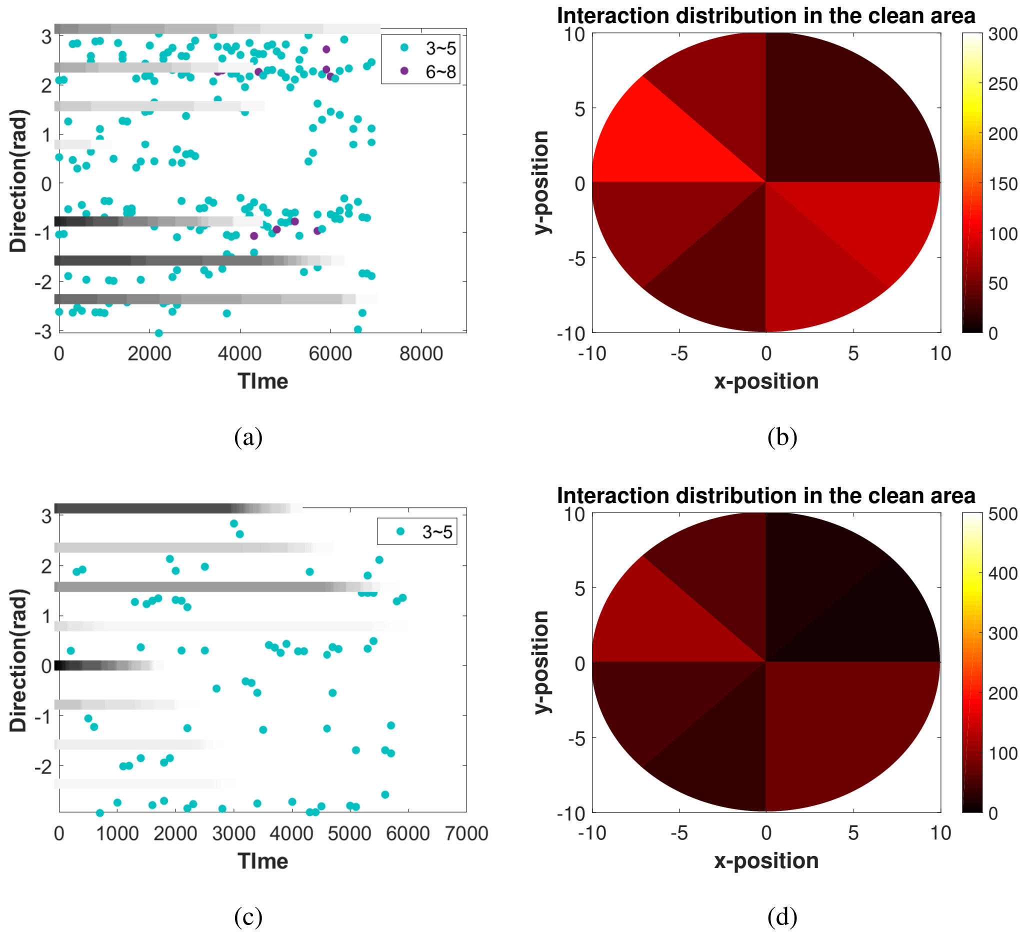
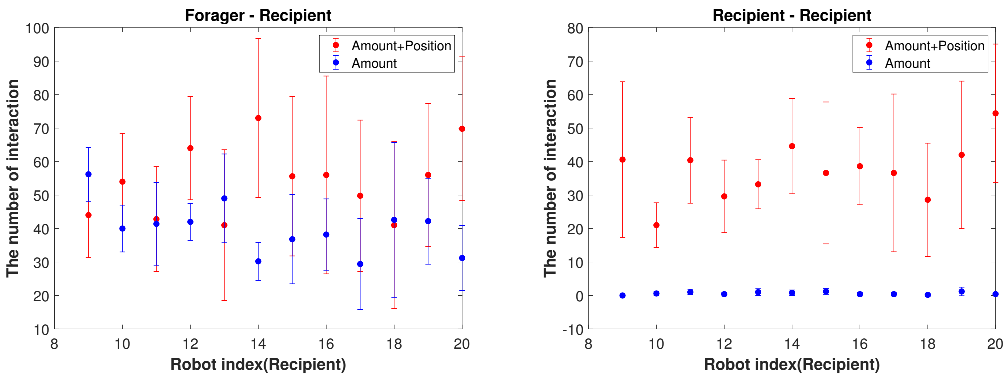



Figure 4.11: The number of interactions by region and movement of recipients. (a) The movement of the recipient when only contaminants are exchanged without location information in the scenario 1 situation. Recipients are widely spread regardless of the contaminated area. (b) The heat map of the number of interactions per region in scenario 1 situation. Overall, the number of interactions is small. (c) The movement of recipients for scenario 2. It is widely spread regardless of the contaminated area. (d) The number of interactions by region as a heat map. The overall number of interactions is low.

is shared, the recipients are widely distributed. Consequently, there are fewer number of interactions. If the recipients are spread out widely in an undefined manner, pollutants are received more slowly, compared to if the recipients move as a group to areas with high levels of contamination.

Figure 4.12 shows the number of exchanges among foragers and recipients and recipients. In general, the number of inteactions was higher when the location informa-




(a) (b)

Figure 4.12: The number of interactions of Forager and Recipient or Recipient and Recipient. Mean and standard deviation for five simulations are shown. (a) How recipients exchanged with foragers. Red is for sharing location information and blue is for delivering pollutants only. If the location information is shared together, the number of interactions is high. (b) The number of interactions by robot when the recipient exchanges with another recipient. Unique geolocation shows more interactions than simply contaminants.

tion was delivered in addition to the pollutants. However, the manner of exchanging the location information differs significantly between recipients, compared to between forager and recipients. This allows the recipients to quickly accept the contaminants brought by the foragers, and increases the probability that the recipients with contaminants will be purified in the purification area. In Chapter 3, we analyzed the advantages of receiving pollutants directly from foragers. Consequently, the effectiveness of the mission depends on the movement of the recipients. It is interesting to note that the higher number of interactions observed as the recipients move in groups improves the performance of the multi-robot system.

## 4.6 Summary of Chapter 4

We deploy the trophallactic network system in a multi-robot system designed to clean up a contaminated area. We demonstrate that the recipients' movement is an important factor in this process. Although transferring only pollutants is quite effective, sharing the location information of the contaminants obtained by the foragers has a better effect. All the robots do not possess global knowledge, because they do not have central control. Therefore, the chance encounter of two robots facilitates the exchange of local information, and the subsequent action of each robots is determined based solely on that information. In this way, information is shared, and decide behavior is determined. However, the recipients generally tend to move in groups.

Our results confirm that the convergence of the recipients ensures the contaminants are cleared quickly. Furthermore, the movement of this herd frequently corresponds with a higher number of interactions, which ensures that the pollutants brought by the foragers are passed on quickly. At the same time, the number of recipients with contaminants increases the probability of passing through the purification area, allowing for faster purification. Therefore, frequently sharing information in groups of recipients has a good effect on the swarm. In the next chapter, we present the details of applying the trophallactic network system using real robots in the same experimental set-up described in this chapter.

# Chapter 5

# Trophallactic network system using vibration motor robot

This chapter shows the implementation of the trophallactic network system in a real robot multi-robot system. Based on the set-up shown in Chapter 4, we run a real multi-robot. First, in Scenario 2, which is sprayed with external contaminants, it is used to remove the contaminants. Of the six robots, two are foragers, and the other four are recipients. The foragers simply collect the external contaminants, and return to the clean area to deliver the contaminants to the recipients. This process is repeated until all the contaminated particles are removed. We have demonstrated that the trophallactic method is effective for removing contaminants. However, when implemented using real robots, it is necessary to ascertain that our algorithm is functional. We use our own vibration motor robot to remove contaminants. We describe a set-up in which an actual robot is implemented. In addition, we describe the proposed algorithm and the components of the robot. The result shows how that the actual robot is effective in the environment described in Chapter 4. We then analyze our results and explain the observed phenomena. Finally, we summarize the contents of this chapter.

## 5.1 Multi-robot system

### 5.1.1 Scenario

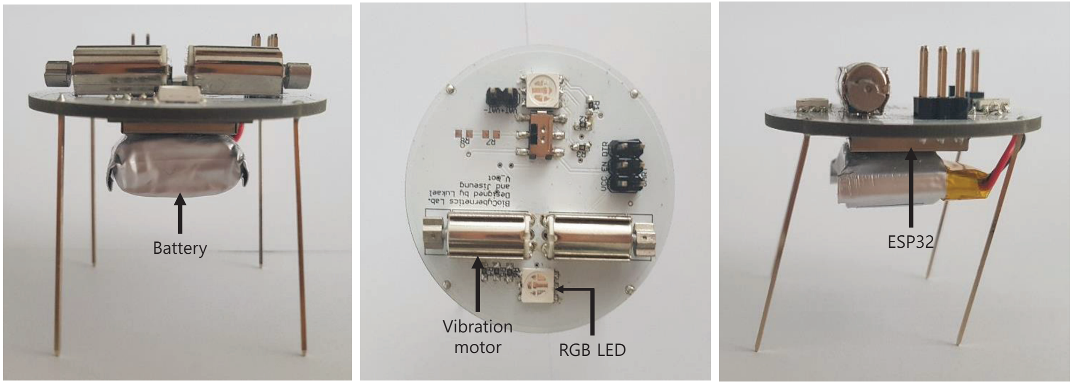
We use a real swarm robot to verify the simulation results detailed in Chapter 4. Our goal is to deliver external contaminants to the recipients. We virtually disperse the contaminated particles using image processing. When the foragers go outside and encounter the virtual contaminants, they collect them, and return to the clean area. Subsequently, when the recipients are encountered in the clean area, the foragers deliver the contaminant to them. This process is repeated until all the external contaminants have been removed. We ascertain that the foragers effectively fetch the contaminants and delivering them to the recipients.

Our strategy into is two-fold. The first part is simply delivering the contaminants; in the second, we consider the effect of sharing location information in addition to the pollutants on the behavior of the robots. We demonstrate that the same phenomena and results are achieved when real robots are deployed, thus corroborating the findings detailed in Chapter 4 with the robot simulation.

### 5.1.2 Environment

We control the robot using a camera installed overhead. We identify the color of the LED using the camera to obtain the head direction and location of the robot. We focused on robot-to-robot interaction with the aim of receiving contaminants using trophallaxis. Therefore, the information of the position and movement of the robot is obtained through the server computer. The forager moves randomly except when it moves outside and when it returns to the clean area, minimizing the form of control over the server computer.

The size of our image is 1280 x 720, and the camera is 2 m high. The clean area is a circle with a radius of 200 based on the center pixel of the image. We create and mark the contaminated particles virtually. The contaminated particles were made to form in a circle with a radius of 400 on the outside. The recipients receiving the pollutant particles place them in a circle with a radius of 100. The experiments were conducted in the a darkroom to enable LED tracking.




(a) (b) (c)

Figure 5.1: Swarm robot using vibration motor. (a) Front of robot (b) Top view of robot (c) Side of robot

## 5.2 Structure and Interface of Robot

### 5.2.1 Hardware

We utilize our custom cluster robot. The robots basically deploy vibration motors. In this study, two vibration motors are used, and each vibration motor can adjust the its intensity from -255 to 255. Therefore, the robot is controlled by the same model as the robot using two wheels. The robot's basic multipoint control unit (MCU) uses ESP32. The battery used is a lithium-polymer battery, and can run from 1 h to 1 h 30 min when fully charged.

We monitor the direction and position of the robot using the camera. To do this, two three-color leds LEDs were mounted. The front and back of the robot were identified according to the color of the light emitted by the three-color LEDs. In addition to this, the intensity of the light is controlled to express the amount of pollutants the robot carries. Therefore, the stronger the light intensity, the more pollutants the robot carries, and the weaker it means that there are no pollutants. We regulate the intensity of light according to eight different levels, ranging from 30 to 100, with 30 being the minimum intensity at which the camera can track light.

Figure 5.1 shows the structure of our robot. The robot is a 3 cm-high circle with a diameter of 3 cm. The robot has four legs made of solid material. To ensure smooth control, the legs are tilted approximately 10°, because when the motor vibrates, the robot's legs digs into the ground due to the vibration. Thus, if the legs are weak, they will not be inclined to attempt to move forward. Basically, to move forward, it is necessary to obtain the frictional force from the ground (Zhu and Kawamura, 2003).

### 5.2.2 Software

Our robot basically communicates with internet protocol (IP). Each robot becomes a server and the computer becomes a client, issuing commands to control movement. The robot is composed of c++ language, and computer proceeds with image processing, and controls the robot using MATLAB. The protocol type of the robot is the json form. The intensity of the motor and the color and intensity of the LEDs can be controlled by sending commands in json form from the computer.

The intensity of the motor and LED are controlled using pulse width modulation (PWM). The image processing algorithm for tracking the LED of the robot is based on the threshold method. The LEDs on the camera are more prominent in color further into the darkroom. Therefore, the high-speed threshold method is used to prevent the robot from moving. First, the robot moves based on the state of described in Figure 3.2 in Chapter 3. Therefore, the foragers go outside and fetch pollutants if there are none. If there are more than a certain amount of contaminants, the forager returns to the clean area. When it enters the robot's sensor range, it delivers contaminants to the robot. At this time, the intensity of the LED varies depending on the amount of pollutant.

## 5.3 Robot control method

We move the forager by selecting a target point, and receiving feedback. Figure 5.2 is a graphical representation of our model. The robot's speed and angular velocity are determined by the target point and the robot's current position. This value is entered into the robot motor.

$$\begin{aligned} \rho &= \sqrt{\Delta x^2 + \Delta y^2} \\ \alpha &= -\theta + atan2(\Delta y, \Delta x) \\ \beta &= -\theta - \alpha \end{aligned} \tag{5.1}$$

Equation 5.1 converts the coordinates of the robot and the target point into distance and angle. The basic robot pose, $(x, y, \theta)$, is changed to $(\rho, \alpha, \beta)$ to achieve control. This is

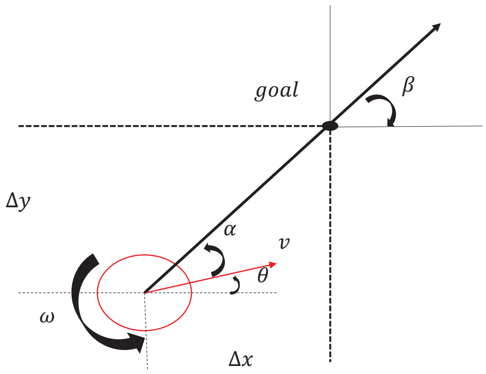



Figure 5.2: Our model for the vibrating robot is the same as the two wheel robot. Change the robot's pose to $(x, y, \theta)$ with $(\rho, \alpha, \beta)$ to model the control with velocity and angular velocity.

chosen to reduce the number of inputs to three. By reducing the inputs, the complex expressions can simply be computed.

$$\begin{aligned} \nu &= k_\rho \, \rho \\ \omega &= k_\alpha \alpha + k_\beta \beta \end{aligned} \tag{5.2}$$

Equation 5.2 matches the distance and angle variables with the robot's speed and angular velocity. We design a system to control the robot by reducing the input by matching the variables with speed and angular velocity. This reduces the complexity of calculations. $k_\rho$, $k_\alpha$, and $k_\beta$ are proportional control variables. This makes it possible to set priorities. We obtained the information about the angle directly from the camera.

$$\begin{aligned} \begin{bmatrix} \rho \\ \alpha \\ \beta \end{bmatrix}_g - \begin{bmatrix} \rho \\ \alpha \\ \beta \end{bmatrix}_R &= \begin{bmatrix} -cos(\alpha) & 0 \\ \frac{sin(\alpha)}{\rho} & -1 \\ -\frac{sin(\alpha)}{\rho} & 0 \end{bmatrix} \begin{bmatrix} \nu \\ \omega \end{bmatrix} \quad (-\frac{\pi}{2}, \frac{\pi}{2}] \\ \begin{bmatrix} \rho \\ \alpha \\ \beta \end{bmatrix}_g - \begin{bmatrix} \rho \\ \alpha \\ \beta \end{bmatrix}_R &= \begin{bmatrix} cos(\alpha) & 0 \\ -\frac{sin(\alpha)}{\rho} & -1 \\ \frac{sin(\alpha)}{\rho} & 0 \end{bmatrix} \begin{bmatrix} \nu \\ \omega \end{bmatrix} \quad (-\pi, -\frac{\pi}{2}] \cup (\frac{\pi}{2}, \pi] \end{aligned} \tag{5.3}$$

Equation 5.3 sets the distance and angle of the robot and the distance and angle of the target point to zero based on the feedback control system. However we do not do this continuously; rather, we partially truncate it, extract a fixed direction, obtain direct feedback from the camera, and move forward step- by- step. Because of the structural problems of the hardware and the influence of the ground, it is possible the robot makes the unwanted movements, which can cause it to deviate from its path. Therefore, to minimize this deviation, the control unit was set to move part by step by giving a stop signal.

## 5.4 Results

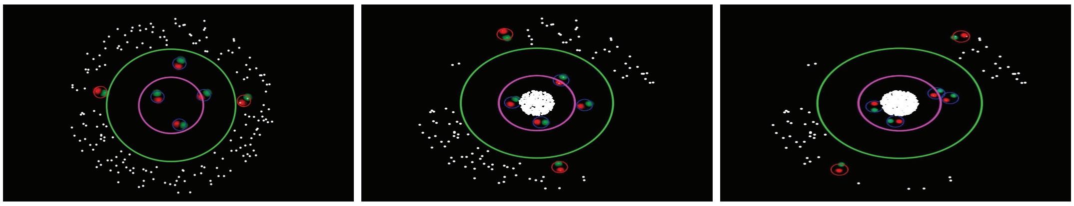
Our goal is to bring external contaminants to the clean area, and place them in the purification area inside(magenta circle). Our experimental environment is sprinkled with pollutants between 0° and 90°. We use two foragers to fetch the contaminants from the contaminated area. One forager goes in the direction of 0°. The other forager moves in the direction of 90°. Outward predators search the surroundings, and bring in as much as they can bring with them if there is a pollutant within the robot's sensor range. If no contaminant is found after searching the area, the foraeger that went in the direction of 0°, moves counterclockwise, and the one that went in the direction of 90°, moves clockwise to find pollutants. The recipients are initially separated at the same interval at points within a radius of 150, from 0° to 90°. Then, the recipients receive the pollutants they bring, and transfer them to the purification area(magenta circle). This depends on the information received by the recipients.

### 5.4.1 Strategy 1

Our first strategy is simply to deliver pollutants. We observe that we can succeed in our desired goal to clean up the contaminated area when we remove the pollutants by simply delivering them without any information. Figure 5.3 shows the movement behavior of the actual robot.

First, the pollutants are located more than 200 pixels away from the outside. The extent across which the contaminants are spread lies between 0° and 90°. The further you move from 0° to 90°, the greater the density of the pollutant. The two foragers start out at 0° and 90° respectively. There are four recipients between $-20°$ and 120°. When they receive the pollutants from the foragers, they move to the middle purification area. Upon arrival at the site, the recipients drop the pollutants they had.

After dropping the pollutants, the recipients return to their initial position and receive the pollutants from the forager as before. However, the foragers approach from a dif-



(a) (b) (c)

Figure 5.3: Movement of robot by time in Strategy 1. The robots share only pollutants with each other. (a) simulation time = 0(s) (b) simulation time = 100(s) (c) simulation time = 200(s)

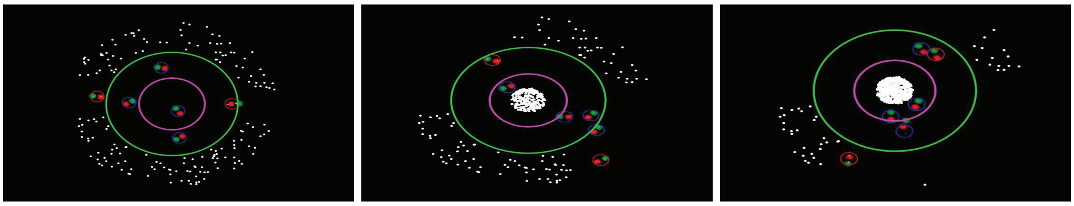
ferent direction with the pollutants. This results in some recipients being left in the lurch. Starting at 0°, the forager gradually changes its position, because there is less pollution in the contaminated area it is located. Thus, although the pollutants are initially brought in the direction of 0° around the clear area (green circle), over time, they will be brought in the direction close to 90°.

Similarly, foragers, starting out at 90°, will return to where the pollutants are clustered if the amount of pollutants are reduced. However, due to the large amount of contaminants at 90°, these foragers travel slower than those starting out at 0°. This makes it difficult for recipients starting out near 0° to receive contaminants from the foragers over time. The recipients starting near 90°, on the other hand, receive contaminants from the foragers relatively well.

### 5.4.2 Strategy 2

In the second strategy, the location information of the pollutants is shared as well. First, similar to the first strategy, the foragers search for contaminants at each point and fetch contaminants discovered there. The contaminants are then taken to the clean area (green circle) where they are delivered to the recipients. At this point, the information of the location from which the pollutant was fetched is shared as well. These recipients first drop the contaminants into the purification area (magenta circle). Then, when exiting the purification area (magenta circle), the recipient move in a direction detrermined based on the position information received. They move in that direction, and wait.

Figure 5.4 shows how the robot receives the location information, and removes the



(a) (b) (c)

Figure 5.4: Movement of robot by time in Strategy 2. The robots share pollutants and location information with each other. (a) simulation time = 0(s) (b) simulation time = 100(s) (c) simulation time = 200(s)

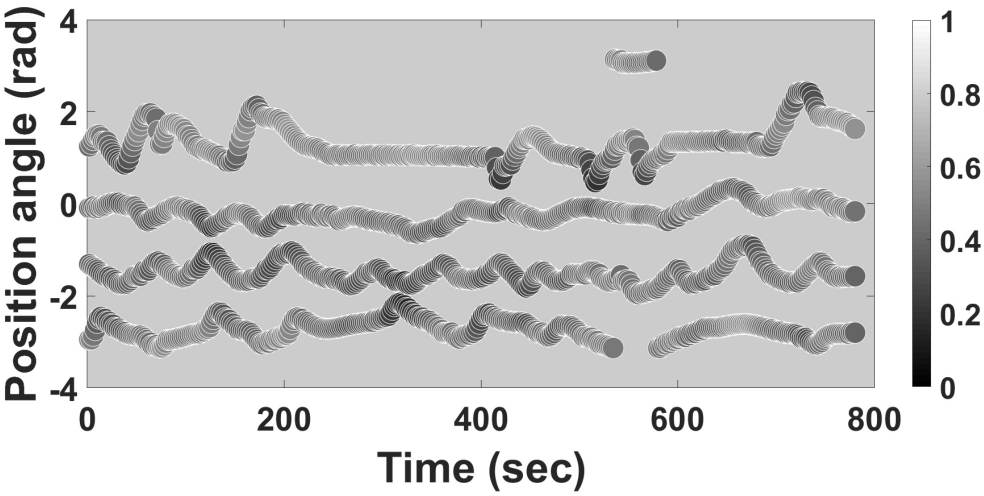
contaminants. When the location information is received, it can be observed that the forager moves to a similar point as it moves. Two phenomena are indicated by our results. First, as the forager moves, the recipients move in a similar direction with the forager. However, the recipients move in groups.

Each recipient simply determines its own path using the location information shared by the forager; however, when viewed as a whole, they appear to move in groups. The recipients also exchange the location information among themselves as they pass on the pollutants. Thus, the user can collect the recipients in a group. Over time, our goal of removing the contaminants is showing progress.

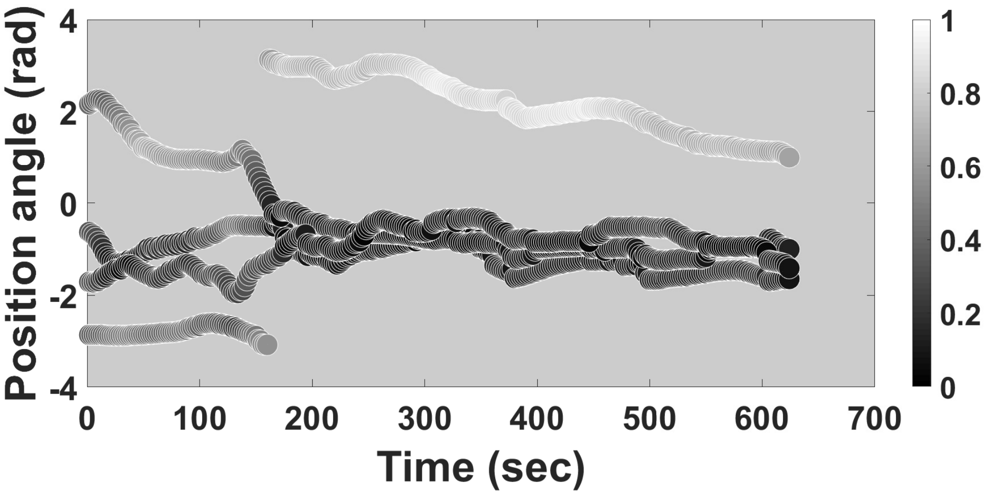
### 5.4.3 Flocking of recipients

Our experiment with an actual multi-robot system shows that when robots share pollutants with each other, the pollutant's location information is shared to determine their behavior, as a result of which they move in groups. Figure 5.5 shows the time-dependent direction of all the robots determined around the purification area (magenta circle). The contrast is closer to black as the robot gets closer, and closer to white as it moves away. The contrast is the ratio of the maximum angle difference to the minimum angle difference at a maximum of one. Figure 5.5 (a) shows the movement of the robots when only pollutants are delivered. As can be seen from the figurue, in this case, there is no tendency to move in groups. In other words, most of the recipients appear white.

On the other hand, when the sharing location information is shared in addition to the pollutants, the recipients tend to flock together. Figure 5.5 (b) shows that there are many




(a)




(b)

Figure 5.5: Recipient robot movements. The darker the black, the closer the robots are to each other. (a) How robots move in situations where only contaminants are delivered. The robots then move away from each other. (b) How robots move when they share location information as well as pollutants. Robots often move in groups rather than each other. It is common to gather together in part, but it is also common to partially gather.

points marked black. Even if all the robots are do not converge, some of them do. This phenomenon is not a command that can be directly issued; rather, each recipient simply determines its path using the location information of the pollutants. This corroborates the findings with the robot simulation detailed in Chapter 4.

Because we have noted in Chapter 4 that this phenomenon increases the number of interactions between the robots, which has a positive effect on the global task, the next result can be predicted.

### 5.4.4 Strategy comparison

Our results show that recipients are more effective in conveying not only the pollutants but also their location information. Figure 5.6 shows the amount of external contaminants and the amount of pollutants deposited in the purification area. The amount of contaminants on the outside shows that both pollutants and location information are delivered faster if they are shared simultaneously. However, the amount of pollutants delivered to the purification area is not different in the two cases. Nonetheless, our results show that if the external pollutants are brought into the clean area faster, they can also be used to transmit the location information of the contaminants. We examine each crop load to evaluate the effectiveness with which the foragers and recipients exchange the pollutants.

Figures 5.6 (c) and (d) show the crop load of foragers and recipients, respectively. The foragers only obtain pollutants from the outside. The rising portion of the load corresponds to when contaminants are obtained from the outside. On the other hand, the section where rope crop load decreases indicates that the foragers have met and delivered the pollutants to the recipients. As for recipients, their crop load increases when they receive the contaminant from the forager. On the other hand, when contaminants are transferred to other recipients or sprayed on the purged area, the crop load is reduced.

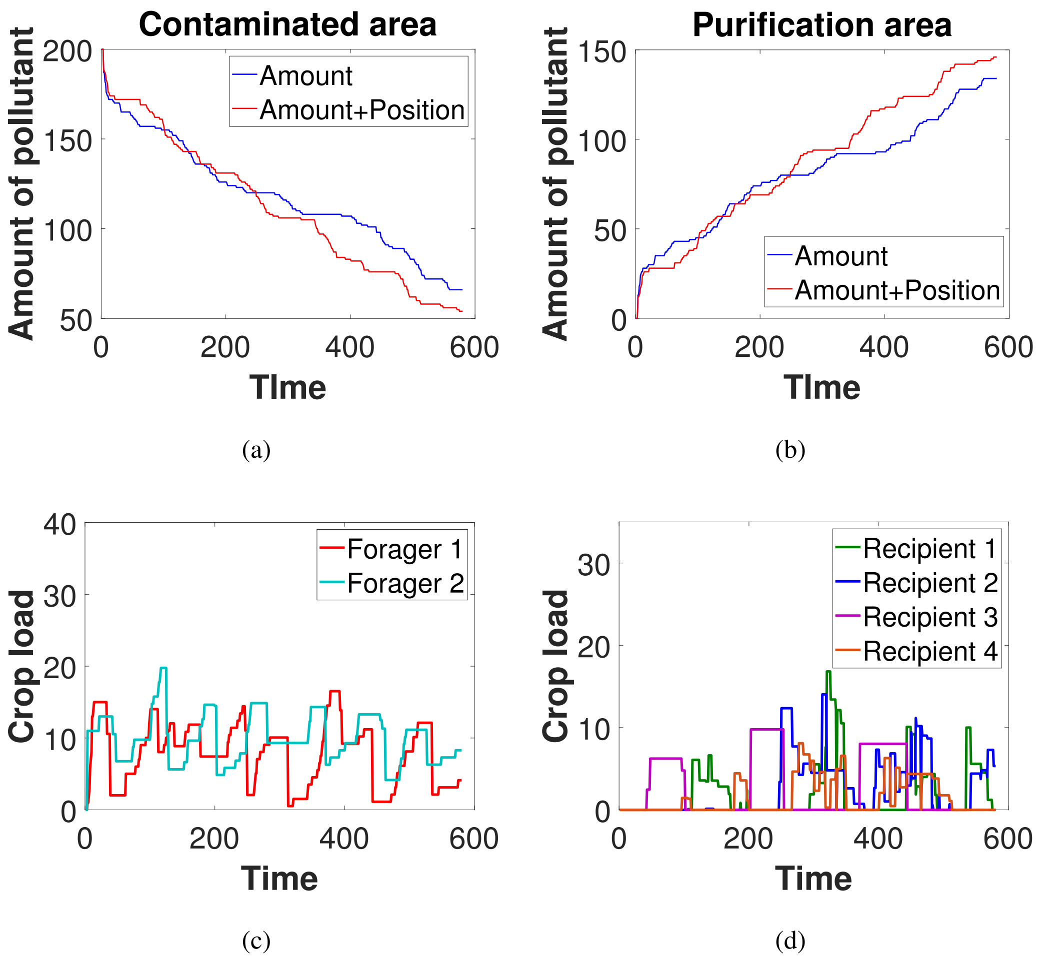



Figure 5.6: Change of pollutant. (a) shows time by time that external contaminants are removed in case only contaminants are transferred and not only pollutants are also transferred, but also location information. The transfer of location information together is eliminated more quickly. (b) The amount of contaminant delivered to the purification area. There is not much difference between the two cases. (c) Forager's crop load (d) Recipient's crop load

## 5.5 Discussion

### 5.5.1 Relationship between the Location Information of Pollutant and the Number of Interaction

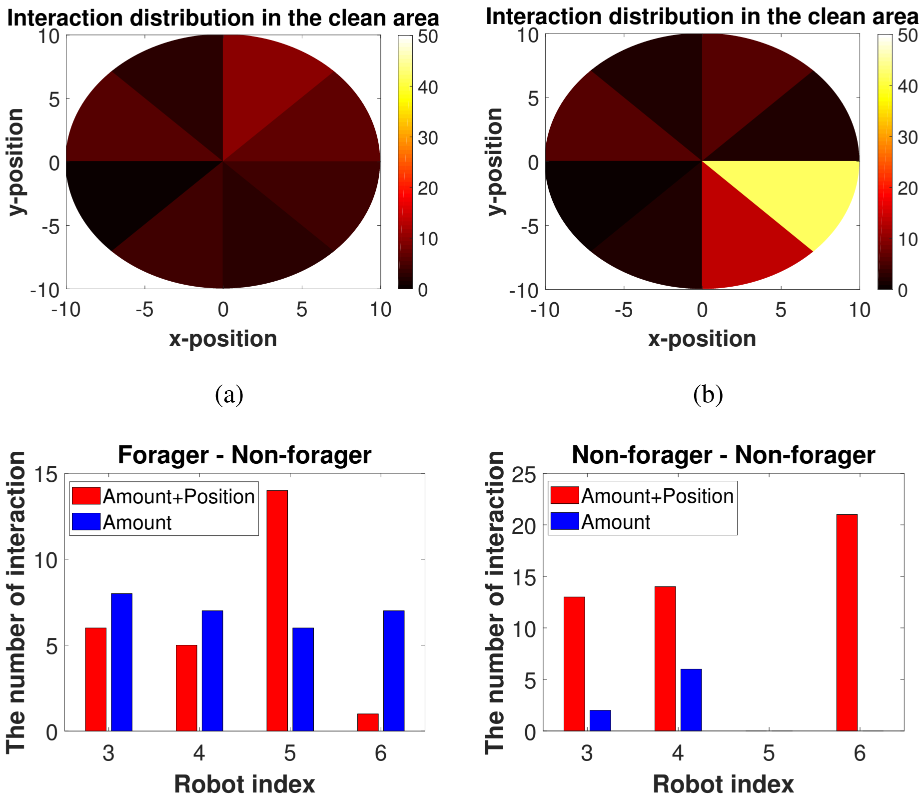
We are deeply concerned with the number of robot-to-robot interactions that occur as the share the pollutants and their location information is shared. By passing on the location of the pollutants, the recipients are spurred to head to the area. This is because the expectation is high that there will be a lot of pollutants in the area. Thus, each robot will move based on its own determined path; however, this movement will force the recipients to swarm. Thus, these groups are more likely to converge near areas where pollutants are present.

Figure 5.5 shows the information of the position of the recipients. In our experiment, the pollutant is spread between 0°, and 90°. The closer the contaminant is to 90°, the greater the density; and the closer it is to 0°, the smaller the density. Therefore, it can be observed that the recipients who were initially separated swarmed in the 0° area, after which the herd gradually moved the herd to the 90° area. In the middle, the herd splits into two groups of twos, because each of these influences is differ. The recipients responding to the foreger at 0°, will gather around it, and those responding to the foraeger at 90° will be visible. This is because, there are two recipients who are greatly affected by each forager;

thus, it can be observed that there are many cases where foraging is conducted in groups of two. However, for 0°, the density of the pollutant is small. Therefore, the amount of contaminated particles is small. Thus, the forager here will perform the task of removing faster than the forager at 90°, which will move up faster can be observed moving to the 90° area. As the recipients converge, the exchange between them becomes more active. At the same time, the recipients hold on to the pollutants for less time as they pass the their own contaminants to the other recipient their own contaminants.

Consequently, the pollutants are received faster from the forager. This occurs more than if the number of interactions with the forager and another number of interventions by patients only deliver contaminants. Thus, as shown in Figure 5.7, the number of interactions where the location of the contaminant is given along with the information of the location of the contaminant is higher, compared to when only the pollutant is delivered. Therefore, pollutants can be brought in and removed faster when location information is exchanged.

In figure 5.7, (a) and (b) show the number of interactions per area in the clean area. Overall, when you receive location information together, it shows more interaction. If only contaminants are exchanged, the recipients should exchange only at the designated location. It can be seen that the interaction occurred almost similarly in each area. However, if the location information is received together, the recipients use the




(a) (b)

(c) (d)

Figure 5.7: The number of interactions by region and by robot. (a) The number of interactions by area in clean area if only pollutants were exchanged. (b) The number of interactions by area in the clean area when the location information as well as the pollutants were exchanged. (c) The number of interactions between foragers and recipients when they have location information and when they do not. The presence of location information results in more interaction. (d)The number of interactions between recipients and recipients when they have and do not have location information. The presence of location information results in more interaction.

information to move to the area. We confirmed that the recipients are grouped between -2 (rad) and 0 (rad) through the figure 5.7. Therefore, it shows the highest number of interactions in the area. In figure 5.7, (c) and (d) show the number of interactions between robots. (c) shows the interactions of each recipient with the forager, and the number of interactions is almost the same when only pollutants are exchanged and when location information is exchanged together. However, the number of interactions between recipients in (d) can be seen clearly when the location information is exchanged together. Recipient 5 robots were separated from each other without clustering, so exchanges with foragers were greater than exchanges with recipients.

### 5.5.2 Limitations of vibration motor robot

This experiment was conducted using a vibrating robot. However, vibrating robots, they are greatly influenced by the angle of the robot's legs. Therefore, even though there are unnecessary movements, or the force received becomes random, and we attempt to deliver the same speed, different robots communicate differently. This results in errors and the desired point is reached slower than expected. Furthermore, it is necessary to consider the size of the robot because we deployed a terrestrial robot. The effect of movement in groups is less pronounced, compared to that observed with the robot simulation in Chapter 4. Therefore, our results verify the effectiveness of the proposed algorithm.

We verify the functionality of the algorithms derived from the robot simulations by replicating the situation described in Chapter 4 using a real robot. We have demonstrated that using the trophallactic network system, we can perform the desired global task that in the multi-robot system. However, the vibrations produced by the structural problems in the robot and affected by the robot's movement on the ground; thus, a tendency to deviate from the accurate path was observed. Rectifying this was rather time-consuming. Therefore, although the unexpected time-consuming nature of the task has does not affected its the performance of the multi-robot system, it is difficult to measure its temporal efficiency. Furthermore, our current system situation obtains data from the camera and inputs it into the computer to determine its behavior.

Although this is a problem that occurs because the robot's efficiency is significantly lower, our system simulates local movements as much as possible by allowing the robots to move randomly without using location information, except when exiting the clean area and externally, for central control. However, sharing real information is driven by robot-to-robot interaction, which works in line with the swarm intelligence concept to realize our desired goal.

### 5.5.3 Compare robot simulation result with real robot analysis result

We show that exchanging information in a trophallactic network system has a positive effect on a multi-robot system. We confirmed the algorithm through robot simulation and analyzed it with real robot to confirm the same effect. However, compared with robot simulation, the data obtained by real robots did not show a big difference. Compared to the rate of removal of contaminants for scenario 2, the robot simulation reduced the time by almost 50%. On the other hand, in real robots, the time was reduced by 17%. However, the overall effect is the same, and the behavior of recipients caused by exchanging positions is similar.

The difference in the reduction effect was that the robot simulation ignored the collision avoidance of the robot, and there was no long-term stay in one place because there was no obstacle in the robot's movement. On the other hand, a real robot cannot ignore the size of a robot unless it moves in the air like a drone. As a result, there is less freedom for movement. In other words, it takes a little longer to get out of an area. This phenomenon prevents the recipients from moving to the purification area to purify the contaminants and, if slowed down, takes longer to receive the contaminants from the forager. As a result, the effect obtained in the robot simulation is not obtained in the real robot.

## 5.6 Summary of Chapter 5

In this chapter, we present the findings of conducting the task of eliminating contaminants using a real multi-robot system. As noted in Chapters 3 and 4, the robots can perform the desired global task effectively using the trophallactic network system. The foragers determined the optimal behavior based on the experiment detailed in Chapter 3, and performed the task in the Scenario 2 situation described in Chapter 4. Our results show that the external contaminants are effectively collected and passed on to the recipients. We implemented the robot's motion shown in the robot simulation to realize a similar movement, and the recipients could be observed becoming more filled, similar to the trophallactic network system.

When all the recipients are full, the foragers are finished in a form that no longer runs out. Using real robots and ensuring that the movement of the robot simulations were well replicated, we obtain results similar to those detailed in Chapter 4. When the location information as well as the contaminants are shared, the recipients use this to determine their movements individually. Each individual recipient determines its own movements individually; however, because of the uniformity of the information, they all appear to move in groups. This results in frequent interactions, thus ensuring that pollutants are quickly removed from the outside.

# Chapter 6

# Conclusion

In this chapter we summarize the research in this paper. It also presents the contribution of the paper and provides future works.

## 6.1 The Action of Forager

As detailed in Chapter 3, we used robots to simulate the behavior of ants in a trophallactic network system. The foragers who bring food from outside are the agents of in the trophallactic network system. The results of the global task vary according to their actions. In the case of real ants, the foragers continue to fetch food from outside, and do not bring it from inside at any point. They act as if they know the amount of food required in the nest. Beyond merely delivering food, the foragers obtain information based on interaction volume and interaction rate.

This information is closely related to the amount of food that the nest requires to be filled up. First of all, the larger the amount of food in the nests, the lower these values are. These two values represent a near linear relationship with the amount of food that is in the nest. Thus, the interaction volume and interaction rate reduced steadily as the exchange continues. We made a similar observation in our robot simulations. Our robot simulations proceed to demonstrate why the amount of food the foraeger delivers is actually determined by the exponential distribution, the speed of the non-forager, and the ratio of forager to non-forager. First, we evaluate the choice of distribution in terms of the speed at which the nest is filled and how evenly each robot performs its role in passing on the object.

Our findings demonstrate that the fastest way to fill up the nest with food is through exponential distribution. Although the gaussian distribution fills it up at a similar rate, the food is not evenly distributed, compared to the external distribution. Besides filling up the nest, the objective of ants in the trophallactic network system is to feed one another evenly for the ultimate purpose of filling up the nest quickly. We will be able to efficiently deploy trophallaxis in multi-robot systems by selecting the nature distribution according to the purpose. This is related to the distribution of the amount of food selected by the probability. For exponential distribution, it is more likely that the amount of food delivered will be selected as a more significant value, compared to other distributions. In other words, when the nests start to fill up, they're going to deliver the maximum amount to the non-forgers, and then they're going to be delivered to the various robots in small quantities. Because this also affects the interaction rate, other distributions can convey information that is not suitable for the component distribution.

Furthermore, the interaction rate and speed of the foragers and non-foragers affect the global task. The effectiveness of our performance is dependent on maintaining an appropriate proportion. The slower the speed of the non-foragers, the more likely that food evenly distributed; however, it takes longer to fill the nests. Therefore, we can control the manner and speed in which food is delivered, and the role each individual plays for different purposes.

## 6.2 The Role of Non-foragers

The trophallactic network system cannot be solely dependent on the foreager, because it is necessary them to encounter non-foragers very often as they arrive at the nests so that it can be filled up quickly. This is highly dependent on the actions of the non-foragers. To examine this in detail, we set up an experiment in which the task was to clear a contaminated area and purify the contaminants; we focused specifically on to the impact of the movement behavior of the recipinents. Although the trophallactic network system is basically exists for the purpose of transporting a substance, the significance of communicating information at the same time cannot be undermined.

We examined the movement of the recipients following the robot-to-robot interactions in which the location information of the pollutant alongside the pollutants. Our results show that each recipient tends to determine its path according to the location information, resulting in the appearance of moving in groups. This phenomenon caused led to increased interactions. The increase in the number of such interactions sped up the delivery of the pollutants.

Thus, when the foragers brought in pollutants from the outside, the recipients were able to receive them quickly, and transfer them to the central purification area. Thus, the efficiency of the global task depends on the movement of recipients in response to the information shared by the foragers. Moving in groups made it possible for the robots to quickly remove pollutants from the outside and perform the task efficiently.

## 6.3 Verification through a Real Robot

It was necessary to verify the experimental situation in Chapter 4 using a real robot. Therefore, we created replicated the situation using a real robot, and verified that the same results were achieved. The findings in the set-up with the real robot corroborates the findings in the robot simulations. First, when the location information is supplied in addition to the object, the recipients were observed to swarm in the direction of the contaminants. Furthermore, when the forager moved to a location where contaminated particles were located, the recipients who received the pollutant from it swarmed in the same direction. As was the case with the robot simulations, this resulted in increased number of interactions, and hence, the rapid elimination of pollutants. On the other hand, real robots can collide, except if they are specially designed not to.

Therefore, the collision risk prevents them from clustering too closely, and prevents them from smoothing out pollutants. However, this difference did not cause the results to deviate too significantly from results of the simulated robots; however, it was observed that the objects were delivered faster to the purification area, and the robots a little faster when it was not crowded. It was a subtle difference, but the result showed that building a swarm somewhat impeded the robots' movement. However, we were able to demonstrate that barring the risk of collision and its impact on the real robots' movements, the the recipients' herd behavior has a positive effect on the effectiveness of the multi-robot swarmsystem.

## 6.4 Relationship between spreading information and flocking of robots

Our system operates based on diret communication between robots. In other words, when sharing information with each other, they move based on short-range communication. In many multi-robot systems, long distance communication is often impossible. Thus, short-range communication takes a long time for robots to share information. Updating in real time while the robot keeps exchanging information helps the robot perform its tasks. We show in chapters 4 and 5 that the behavior of a robot affects the number of times information is exchanged. In our system, robots exchange information through robot-to-robot interaction. Through the exchanged information, the robots move in groups, and the movements in groups often cause interaction between robots. The robots in a group share information obtained by each robot in real time, so that the task can proceed efficiently. In this paper, we show that grouping behaviors facilitate the sharing of information and have a positive effect by efficiently performing tasks through updated information in real time.

## 6.5 Contribution

Our goal is to enable robots in multi-robot systems to efficiently perform necessary tasks through robot-to-robot interactions. The efficiency of most multi-agent systems are limited to by environmental factors and the robots' physical factors. Therefore, there is a limit to which central control can be realized. Because centralized control has to process a large amount of information, the system is complex, and unwanted delays can occur during communication. Therefore, rather than performing a task with global information, we attempt to use local information.

We use the concept of swarm intelligence to solve this. The concept of swarm intelligence is based on simple entities making decisions through some local information without central control. Each robot's action is called self-orageanization; self-organization requires information to determine the action. Swarm intelligence uses local information rather than global information. Therefore, depending on the mission, each robot must use appropriate communication methods to obtain the necessary information. We introduce the trophallactic method, which is a robot-to-robot communication method.

Trophyallaxis is one of the indirect communication methods. The purpose of trophallaxis is to enable ants and bees share nutrients among themselves. Social insects not only obtain nutrients through this method but also obtain the information required for all insects to share nutrients efficiently. Such a system is called a trophallactic network system. Information is obtained and shared on a one-to-one. However, it has been observed that the foragers who bring food act as if they know how much food is required in the nest. Thus, although the exchange occurs locally, the entire group behavior appears to be based on global information. We extend the trophallactic network system to one-to-one information exchange and provide a solution to control multi-robot systems.

We deploy this system in a multi-robot system to perform a variety of tasks, with the aim of demonstrating that in spite of not possessing global information, it is possible to perform various tasks if the individual robots appropriately determined their action according to local information. Because trophallaxis is one-to-one communication, no complicated system is required. This is a useful method for simple multi-robot systems and for the goal of sharing certain objects. We introduce a contaminant removal system as an example, and demonstrated the usefulness of trophallaxis for such a task. In this study, we reveal the link between the location information supplied by foragers and the swarm behavior that is shaped by the information. The efficiency of the multi-robot system depends on identifying the optimal condition, passing on this information, and flocking swarming accordingly.

## 6.6 Future works

The trophallactic network system in this study is predicated on robot-to-robot communication. The inspiration was drawn social creatures who share nutrients among each other so that all individuals in the colony can receive energy from the outside. However, one of the problems with today's multi-robot systems is the battery. As many robots are involved, it is not easy to install sufficient charging stations for corresponding to their numbers. Besides, even if charging stations were sufficient, determining the required power is a challenge. Furthermore, it would be difficult to move the robots around in areas with many obstacles. In this case, charging can be achieved by replicating trophallaxis, and all the robots near each other will receive sufficient energy. In our future research, we will study the most effective way to perform this task by comparing the effectiveness of charging of all the robots in a multi-robot system at the charging station within a specific environment and that of the robots charging one another using the trophallactic network system. These studies will be of great help in suggesting ways to solve the battery problems in multi-robot systems.

In addition, we will study the positive effects of the trophallactic network system on the robots used in the courier logistics, depending on the environment and robot movement. Courier Logistics Transportation finds and delivers a number of desired couriers to people's areas. Because there are numerous parcels in this environment the space is small, and there are many obstacles impeding the movement of several robots. In operating various robots, in many cases, communication between several robots is impossible. It is a great advantage to be able to perform global tasks simply through robot-to-robot interaction. This will allow us to continue working on what information we need to share in order to proceed with the desired task and to be more efficient.

# Bibliography

Agrawal, D. and Karsai, I. (2016). The mechanisms of water exchange: the regulatory roles of multiple interactions in social wasps. *PloS one*, 11(1):e0145560.

Ahmed, H. and Glasgow, J. (2012). Swarm intelligence: concepts, models and applications. *School Of Computing, Queens University Technical Report*.

Beni, G. (1988). The concept of cellular robotic system. In *Proceedings IEEE International Symposium on Intelligent Control 1988*, pages 57–62. IEEE.

Beni, G. and Wang, J. (1993). Swarm intelligence in cellular robotic systems. In *Robots and biological systems: towards a new bionics?*, pages 703–712. Springer.

Bird, R. B., Bird, D. W., Smith, E. A., and Kushnick, G. C. (2002). Risk and reciprocity in meriam food sharing. *Evolution and Human Behavior*, 23(4):297–321.

Blanchard, G., Orledge, G., Reynolds, S., and Franks, N. (2000). Division of labour and seasonality in the ant leptothorax albipennis: worker corpulence and its influence on behaviour. *Animal Behaviour*, 59(4):723–738.

Blum, C. and Merkle, D. (2008). Swarm intelligence. *Swarm Intelligence in Optimization; Blum, C., Merkle, D., Eds*, pages 43–85.

Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41.

Cassill, D. L. and Tschinkel, W. R. (1995). Allocation of liquid food to larvae via trophallaxis in colonies of the fire ant, solenopsis invicta. *Animal Behaviour*, 50(3):801–813.

Charbonneau, D., Blonder, B., and Dornhaus, A. (2013). Social insects: a model system for network dynamics. In *Temporal networks*, pages 217–244. Springer.

Choi, J., Yu, S. M., and Kim, S.-L. (2013). Spreading information in mobile wireless networks. *arXiv preprint arXiv:1305.2275*.

Choi, J., Yu, S. M., and Kim, S.-L. (2014). Spreading information in mobile wireless networks. In *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, pages 1–5. IEEE.

Crailsheim, K. (1990). The protein balance of the honey bee worker. *Apidologie*, 21(5):417–429.

Crailsheim, K. (1992). The flow of jelly within a honeybee colony. *Journal of comparative physiology B*, 162(8):681–689.

Cushing, D. and Jones, F. H. (1968). Why do fish school? *Nature*, 218(5145):918.

De Marco, R. and Farina, W. (2001). Changes in food source profitability affect the trophallactic and dance behavior of forager honeybees (apis mellifera l.). *Behavioral Ecology and Sociobiology*, 50(5):441–449.

Dorigo, M., Bonabeau, E., and Theraulaz, G. (2000). Ant algorithms and stigmergy. *Future Generation Computer Systems*, 16(8):851–871.

Eberhart, R. and Kennedy, J. (1995). A new optimizer using particle swarm theory. In *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pages 39–43. Ieee.

Esch, H. E., Zhang, S., Srinivasan, M. V., and Tautz, J. (2001). Honeybee dances communicate distances measured by optic flow. *Nature*, 411(6837):581.

Farina, W. and Grüter, C. (2009). Trophallaxis: a mechanism of information transfer. *Food exploitation by social insects: ecological, behavioral, and theoretical approaches*, pages 183–197.

Farina, W. M. and Núñez, J. A. (1993). Trophallaxis in honey bees: transfer delay and daily modulation. *Animal behaviour*, 45(6):1227–1231.

Farina, W. M. and Núñez, J. A. (1995). Trophallaxis in apis mellifera: effects of sugar concentration and crop load on food distribution. *Journal of Apicultural Research*, 34(2):93–96.

Ferrante, E., Turgut, A. E., Duéñez-Guzmán, E., Dorigo, M., and Wenseleers, T.

(2015). Evolution of self-organized task specialization in robot swarms. *PLoS computational biology*, 11(8):e1004273.

Fewell, J. H. (2003). Social insect networks. *Science*, 301(5641):1867–1870.

Garnier, S., Gautrais, J., and Theraulaz, G. (2007). The biological principles of swarm intelligence. *Swarm intelligence*, 1(1):3–31.

Grassé, P.-P. (1959). La reconstruction du nid et les coordinations interindividuelles chezbellicositermes natalensis etcubitermes sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs. *Insectes sociaux*, 6(1):41–80.

Greenwald, E., Segre, E., and Feinerman, O. (2015). Ant trophallactic networks: simultaneous measurement of interaction patterns and food dissemination. *Scientific reports*, 5:12496.

Greenwald, E. E., Baltiansky, L., and Feinerman, O. (2018). Individual crop loads provide local control for collective food intake in ant colonies. *Elife*, 7:e31730.

Grüter, C. and Farina, W. M. (2009). The honeybee waggle dance: can we follow the steps? *Trends in Ecology & Evolution*, 24(5):242–247.

Hee, J. J., Holway, D. A., Suarez, A. V., and Case, T. J. (2000). Role of propagule size in the success of incipient colonies of the invasive argentine ant. *Conservation Biology*, 14(2):559–563.

Hölldobler, B. (1985). Liquid food transmission and antennation signals in ponerine ants. *Isr J Entomol*, 19:89–99.

Hong, Y., Kyung, Y., and Kim, S.-L. (2019). A multi-robot cooperative patrolling algorithm with sharing multiple cycles. In *2019 European Conference on Networks and Communications (EuCNC)*, pages 300–304. IEEE.

Isaac, G. (1978). The food-sharing behavior of protohuman hominids. *Scientific american*, 238(4):90–109.

Kennedy, J. (2010). Particle swarm optimization. *Encyclopedia of machine learning*, pages 760–766.

Kim, M. and Kim, D. (2019). Analysis of trophallaxis network system using multi-robot. *KIEE Transactions on Robotics and Automation(being prepared).*

King, A. J., Wilson, A. M., Wilshin, S. D., Lowe, J., Haddadi, H., Hailes, S., and Morton, A. J. (2012). Selfish-herd behaviour of sheep under threat. *Current Biology*, 22(14):R561–R562.

Korst, P. and Velthuis, H. (1982). The nature of trophallaxis in honeybees. *Insectes Sociaux*, 29(2):209–221.

Kubo, M. and Melhuish, C. (2004). Robot trophallaxis: Managing energy autonomy in multiple robots. *Proceedings of Towards Autonomous Robotic Systems*, pages 77–84.

Lee, W. and Kim, D. (2014). Adaptive division of labor in multi-robot system with minimum task switching. In *Artificial Life Conference Proceedings 14*, pages 750–756. MIT Press.

Lee, W. and Kim, D. (2017). Autonomous shepherding behaviors of multiple target steering robots. *Sensors*, 17(12):2729.

Lien, J.-M., Bayazit, O. B., Sowell, R. T., Rodriguez, S., and Amato, N. M. (2004). Shepherding behaviors. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 4, pages 4159–4164. IEEE.

Lien, J.-M., Rodriguez, S., Malric, J.-P., and Amato, N. M. (2005). Shepherding behaviors with multiple shepherds. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 3402–3407. IEEE.

Ma, H., Tovey, C., Sharon, G., Kumar, T. S., and Koenig, S. (2016). Multi-agent path finding with payload transfers and the package-exchange robot-routing problem. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Mlot, N. J., Tovey, C. A., and Hu, D. L. (2011). Fire ants self-assemble into waterproof rafts to survive floods. *Proceedings of the National Academy of Sciences*, 108(19):7669–7673.

Penders, J., Alboul, L., Witkowski, U., Naghsh, A., Saez-Pons, J., Herbrechtsmeier, S., and El-Habbal, M. (2011). A robot swarm assisting a human fire-fighter. *Advanced Robotics*, 25(1-2):93–117.

Ratnieks, F. L. and Anderson, C. (1999). Task partitioning in insect societies. *Insectes sociaux*, 46(2):95–108.

Ren, W. and Atkins, E. (2007). Distributed multi-vehicle coordinated control via local information exchange. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 17(10-11):1002–1033.

Ren, W. and Sorensen, N. (2008). Distributed coordination architecture for multi-robot formation control. *Robotics and Autonomous Systems*, 56(4):324–333.

Riley, J. R., Greggers, U., Smith, A. D., Reynolds, D. R., and Menzel, R. (2005). The flight paths of honeybees recruited by the waggle dance. *Nature*, 435(7039):205.

Robinson, E. J., Feinerman, O., and Franks, N. R. (2009). Flexible task allocation and the organization of work in ants. *Proceedings of the Royal Society B: Biological Sciences*, 276(1677):4373–4380.

Robson, S. K. and Traniello, J. F. (2002). Transient division of labor and behavioral specialization in the ant formica schaufussi. *Naturwissenschaften*, 89(3):128–131.

Rubenstein, M., Ahler, C., Hoff, N., Cabrera, A., and Nagpal, R. (2014). Kilobot: A low cost robot with scalable operations designed for collective behaviors. *Robotics and Autonomous Systems*, 62(7):966–975.

Rubenstein, M., Ahler, C., and Nagpal, R. (2012). Kilobot: A low cost scalable robot system for collective behaviors. In *2012 IEEE International Conference on Robotics and Automation*, pages 3293–3298. IEEE.

Rubenstein, M. and Nagpal, R. (2010). Kilobot: A robotic module for demonstrating behaviors in a large scale (\(2^{10}\) units) collective. In *Proceedings of the IEEE 2010 International Conference on Robotics and Automation Workshop, Modular Robotics: State of the Art*, volume 1, pages 364–369. Institute of Electrical and Electronics Engineers.

Şahin, E. (2004). Swarm robotics: From sources of inspiration to domains of application. In *International workshop on swarm robotics*, pages 10–20. Springer.

Saifullah, A., Xu, Y., Lu, C., and Chen, Y. (2014). End-to-end communication delay analysis in industrial wireless networks. *IEEE Transactions on Computers*, 64(5):1361–1374.

Schioler, H. and Ngo, T. D. (2008). Trophallaxis in robotic swarms-beyond energy autonomy. In *2008 10th International Conference on Control, Automation, Robotics and Vision*, pages 1526–1533. IEEE.

Schmickl, T. and Crailsheim, K. (2006). Trophallaxis among swarm-robots: A biologically inspired strategy for swarm robotics. In *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006.*, pages 377–382. IEEE.

Schmickl, T. and Crailsheim, K. (2008). Trophallaxis within a robotic swarm: bio-inspired communication among robots in a swarm. *Autonomous Robots*, 25(1-2):171–188.

Sendova-Franks, A. and Franks, N. R. (1993). Task allocation in ant colonies within variable environments (a study of temporal polyethism: experimental). *Bulletin of Mathematical Biology*, 55(1):75–96.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423.

Shehory, O. and Kraus, S. (1998). Methods for task allocation via agent coalition formation. *Artificial intelligence*, 101(1-2):165–200.

Shiqin, Y., Jianjun, J., and Guangxing, Y. (2009). A dolphin partner optimization. In *Intelligent Systems, 2009. GCIS'09. WRI Global Congress on*, volume 1, pages 124–128. IEEE.

Suárez, M. E. and Thorne, B. L. (2000). Rate, amount, and distribution pattern of alimentary fluid transfer via trophallaxis in three species of termites (isoptera: Rhinotermitidae, termopsidae). *Annals of the Entomological Society of America*, 93(1):145–155.

Sugiyama, H., Tsujioka, T., and Murata, M. (2006). Qos routing in a multi-robot network system for urban search and rescue. In *20th International Conference on Advanced Information Networking and Applications-Volume 1 (AINA'06)*, volume 2, pages 5–pp. IEEE.

Thom, C., Gilley, D. C., Hooper, J., and Esch, H. E. (2007). The scent of the waggle dance. *PLoS biology*, 5(9):e228.

Tuna, G., Gungor, V. C., and Gulez, K. (2014). An autonomous wireless sensor network deployment system using mobile robots for human existence detection in case of disasters. *Ad Hoc Networks*, 13:54–68.

Wainselboim, A. and Farina, W. (2000). Trophallaxis in filled-crop honeybees (apis mellifera l.): food-loading time affects unloading behaviour. *Naturwissenschaften*, 87(6):280–282.

Wang, X., Cheng, H., and Huang, M. (2013). Multi-robot navigation based qos routing in self-organizing networks. *Engineering Applications of Artificial Intelligence*, 26(1):262–272.

Waters, J. S. and Fewell, J. H. (2012). Information processing in social insect networks. *PLoS One*, 7(7):e40337.

Weihs, D. (1973). Hydromechanics of fish schooling. *Nature*, 241(5387):290.

Wenner, A. M. (1962). Sound production during the waggle dance of the honey bee. *Animal Behaviour*, 10(1-2):79–95.

Wilkinson, G. S. (1984). Reciprocal food sharing in the vampire bat. *Nature*, 308(5955):181.

Wilson, E. O. (1984). The relation between caste ratios and division of labor in the ant genus pheidole (hymenoptera: Formicidae). *Behavioral Ecology and Sociobiology*, 16(1):89–98.

Wilson, E. O. et al. (1971). The insect societies. *The insect societies.*

Yu, S. M. and Kim, S.-L. (2010). End-to-end delay in wireless random networks. *IEEE Communications Letters*, 14(2):109–111.

Zhu, C. and Kawamura, A. (2003). Walking principle analysis for biped robot with zmp concept, friction constraint, and inverted pendulum model. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 1, pages 364–369. IEEE.