SHORTER PATH GENERATION SCHEME FOR ADAPTABLE SELF-ORGANIZED ROBOT SWARMS

SOE SOE HLAING

M.C.Tech. **FEBRUARY 2020**

SHORTER PATH GENERATION SCHEME FOR ADAPTABLE SELF-ORGANIZED ROBOT SWARMS

BY

SOE SOE HLAING

(B.C.Tech.)

A Dissertation Submitted in Partial Fulfilment of the Requirements for the Degree of

Master of Computer Technology

(M.C.Tech.)

University of Computer Studies, Yangon

FEBRUARY 2020

ACKNOWLEDGEMENTS

First of all, the author would like to express the deepest gratitude and sincere appreciation to the following persons who have contributed directly or indirectly towards the completion of this thesis and helped her to maintain this dissertation possible.

The author would like to express her respectful thanks to Dr. Mie Mie Thet Thwin, Rector of the University of Computer Studies, Yangon, for her kind permission to submit this thesis.

The author wholeheartedly expresses her sincere gratitude to her supervisor, Professor Dr. Khin Than Mya, Dean of Faculty of Computer Systems and Technologies, Head of Embedded System Lab, University of Computer Studies, Yangon (UCSY), for her support, valuable suggestions, ideas, motivation and effective encouragement. It was a great honor to work under her supervision and also immense experiences working in this research.

The author would like to give her sincere thanks to Professor Dr. Thet Thet Khin, Faculty of Computer Systems and Technologies, University of Computer Studies, Yangon, for her administrative supports as the Dean of Master Course throughout this thesis.

The author would like to express her respectful thanks to Dr. Wynn Htay, Principal of Japan IT and Business College, for his kindness support.

The author also would like to extend her extreme, sincere, gratitude and appreciation to lecturer, Daw Sandar, English Department, University of Computer Studies, Yangon, for editing this thesis writing from the language point of view.

The author would like to give her sincere thanks and regards to her supervisor, Associate Professor Geunho Lee, Ph.D., Graduate School of Engineering, University of Miyazaki, for his generous support, kindness, suggestions, encouragement, ideas, optimism and tremendous knowledge.

Furthermore, the author would like to convey her grateful thanks to co-supervisors, Professor Kikuhito Kawasue, Ph.D., and Associate Professor Yokomichi Masahiro, Ph.D., Graduate School of Engineering, University of Miyazaki (UoM), for their valuable suggestions, ideas, motivation and encouraging help in UOM.

The author would like to express her deepest thanks and sincere appreciation to Professor Dr. Myint Myint Sein, Head of Geographic Information System Lab, and Professor Dr. Thi Thi Soe Nyunt, Dean of Faculty of Computer Science, Head of Computer Graphics and Visualization Lab, University of Computer Studies, Yangon, for their kindness, precious help, comprehensive advices and motivation and valuable ideas.

The author would like to express her deepest thanks and sincere appreciation to Professor Dr. Thi Thi Zin, Graduate School of Engineering, University of Miyazaki, for her kindness, precious help, comprehensive advices and motivation and valuable ideas. Moreover, the author also would like to extend her extreme, sincere, gratitude and appreciation to Professor Dr. Pyke Tin, former Rector of University of Computer Studies, Yangon, for his endless kindness and generous advice and extreme knowledge.

Finally, the author would like to extend her special thanks to all teachers who tookpart in teaching with their valuable ideas throughout the master's degree course and also to her friends, for their cooperation. The author would like to give thanks to her beloved parents, all of her colleagues, and friends, for their encouragement and help during this thesis.

ABSTRACT

Swarm robotics is an approach to the collective robotics that takes inspiration from the self-organized behaviors of social animals. Through simple rules and local interactions, swarm robotics aims at designing robust, scalable and flexible collective behaviors for the coordination of large numbers of robots. Recently, there has been a growing interest in multi-robot systems. This interest is motivated by the fact that inherent parallelism and redundancy make multi-robot systems more robust than single robot systems. Furthermore, a multi-robot system can be versatile enough to generate the different structures and functions required to undertake missions in unknown environmental conditions.

Robots are used in many fields, such as cleaning, medical treatment, space exploration, disaster relief and so on. The proposed work introduces path communication scheme from the self-organized robot swarms. The main purpose is to work swarm robots with minimal requirements, such as reducing the time required for finding the path between swarm robots from its self-organized network without long-lived state information. The important features include self-organization, self-healing, topology adaptable and the shorter path selecting from its self-organized neighbors. The proposed work can also be proved that the system can also reduce the processing time by comparing with another path finding algorithm.

CONTENTS

LIST OF FIGURES

FIGURE **Page**

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Nowadays, there has been more interest in swarm robot. Swarm robot provides an interesting alternative to more classical approaches such as artificial intelligence [1-3]. Artificial intelligence (AI) is a computer science for intelligent machine. The main problem of AI includes programming computer such as: problem solving, planning, manipulation and ability to move objects. A core part of AI is knowledge engineering. Robot is one of the major field related with AI [4]. Robot requires intelligence to do tasks such as motion planning, mapping, navigation and object manipulation. Studying the design of robot, controlling behaviors and physical bodies are swarm robots.

Swarm robot is the study of developing and controlling large groups of simple robots [5]. Swarm robot can be seen as a research area that the field of swarm intelligence, whereas swarm intelligence describes a subfield of artificial intelligence. Swarm intelligence (SI) is the collection of behavior of self-organized systems, decentralized and artificial of natural. SI consists simple agents and interacts locally with another and their environment. The inspiration of swarm robots come from nature. The agent follows very simple rules. Although swarm robot is no centralized control dictating how each agent should behave, local and interaction between agents lead to emergence of intelligence behavior without knowing to individual agents [6-8].

In the nature, examples of swarm includes ants, fish schooling ,bird flocking, bacterial growth, hawks hunting, animal herding and microbial intelligence [9]. Last few years, swarm robotics have been developed. In swarm robot, cost and miniaturization are key factors. These are the main constraint for building group of robots. Therefore, simplicity of individual robot should be emphasized. This can motivate swarm intelligence to get meaningful behavior for swarms. Swarm robots are different in some aspects such as sensing capability, power consumption and locomotion strategy [10]. The sensing capability of a robot influences the type of experiments one can perform. Swarm robots that are low-cost and functional is a challenging task [11].

There are many advantages over more complex individual robot using many robots instead of just one. This is possible by simple design of the robot modules because they are cheap and easier to build [12]. Example applications of swarm robot are as shown in Figure 1.1. Swarm robots are able to cover more area than a single robot [13]. Distributed search algorithms are able to cover different parts of a search space at once.

Swarm robots are fault tolerant because swarm robotic algorithms without require robot to depend on another. If a robot fails, rest of swarms can perform its actions [14]. Individual robot may be worthless when there is a failure in its component [15]. Robustness is an important feature in complex environments. Algorithms for swarms are well and without depend on the robots.

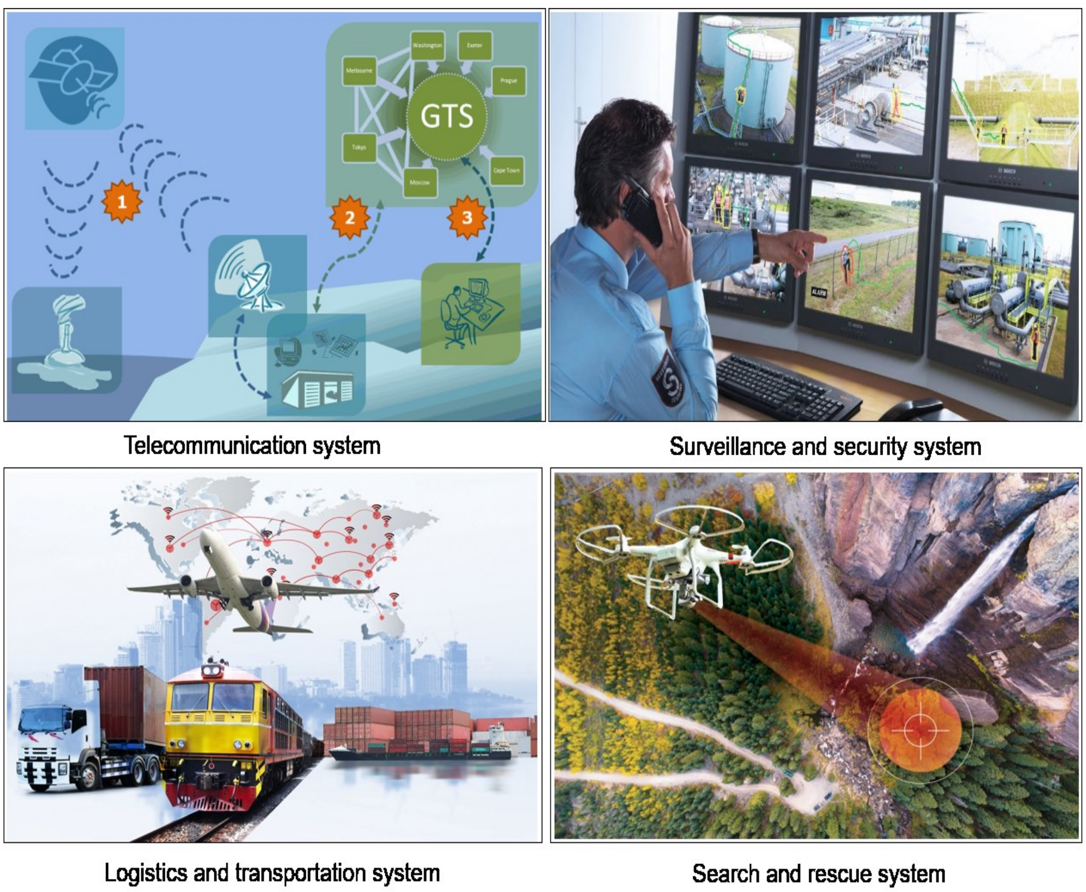


Figure 1.1 Application Examples Using Robot Swarms

1.1 Objective of Thesis

The objectives of this thesis are as follows:

- To provide locally communicative interaction between swarm robots.
- To work swarm robot with minimal requirements.
- To work with the important features including self-organization, self-healing and topology adaptation.

1.2 Overview of the Proposed System

The proposed solution, shorter communication-path generation scheme is suitable for many applications such as path finding and sensing applications, military of logistics and transportation, telecommunication systems, surveillance and security, traffic patterns for transportation systems, hazard detection and exploration [16]. From the motivated issues, this system addressed the path planning problem for autonomous robots. There are challenges about how to communicate and interact with some conditions such as no requirements of long-lived state information, minimal locality [17].

Based on the minimal conditions, the proposed the shorter communication-path generation scheme, for reducing the processing time and enabling robot self-organizing their network adapting to topology changes due to robot failures and movements [18-20]. This section describes an overview of research for path communication and self-organized robot swarm with minimum requirements. With the advancement of robotic technology, much attention has been paid to increase availability and use of a large-scale swarm of robots relatively simple capability in the fields of swarm robotics [21]. Nowadays, swarm robots are popular according to their self-organized communication. Swarm robots are also useful for path planning.

1.3 Organization of the Thesis

The organization of the thesis is as follows:

Chapter 1 describes the introduction of swarm robot. Moreover, objectives and thesis organization are also expressed in this chapter. Chapter 2 presents the background of the thesis. Swarms found in nature and characteristics of swarms are also explained in this chapter. Chapter 3 discusses the problem statement including the explanations and notations of robots. Problem definitions are also discussed in this chapter. Chapter 4 presents the shorter communication path generation scheme. The self-organization and adaptation are also described. The steps of the algorithms are also explained in this chapter. Chapter 5 includes the simulation results and discussions. How the simulation results are got is also described in this chapter. In conclusion, advantages and disadvantages, further extensions are presented in Chapter 6.

CHAPTER 2

BACKGROUND THEORY

2.1 Swarm Robots

Swarm robot is the study of how to coordinate large groups of simple robots by the use of local rules [22]. It takes its inspiration from insects such as bees, ants and they can perform tasks beyond the capabilities of the individuals. Swarm robot has its origin in swarm intelligence. The main focus of swarm robotic research was to study biological research [23]. Swarm in nature are as shown in Figure 2.1. Swarm robot studies how to operate without relying on any structure and any form of centralized control.

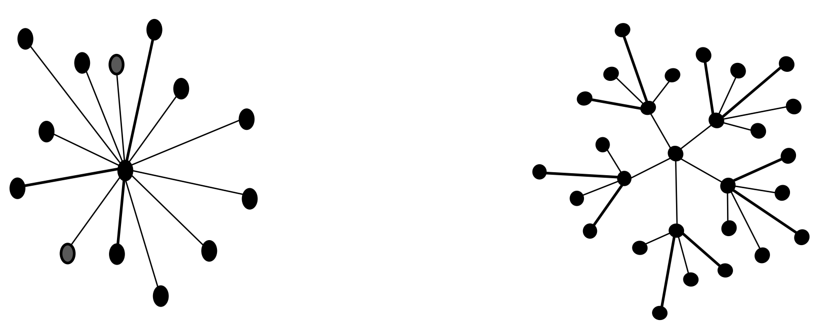
Figure 2.1 Example of Natural Swarms

Collaboration between biologists and roboticists helped swarm robotics research. The focus of swarm robotics has been shifting: from a bio-inspired field of robotics. Swarm robotic becomes an engineering field whose focus is on the development of methods to solve real problems. Cooperation is the main concept in multi robot systems for performing complex tasks [24]. In swarm robotic, a self-organized cooperation is applied, where robots with limited intelligence interact and cooperate locally to build up the desired global behavior.

In swarm robots, the collective behavior of the robots results from local interactions between robots and the environment in which they act [25]. The design of robot swarm is gained by using the principles of swarm intelligence. These principles promote the realization of systems that are flexible, fault tolerant and scalable. Swarm robotics appear to be a promising approach when performing different activities concurrently and when it is infeasible to set up the structure required to control the robot in a decentralized or centralized [26]. Swarm robot has some special characters, which are found in insects, that is, lack of synchronization and decentralized control.

2.2 Centralized and Decentralized System

A centralized system is a central controller and control over the lower level components of the system directly and through the use of a power hierarchy (such as instructing a middle level component to instruct a lower level component). Centralized and decentralized network are shown in Figure 2.2.



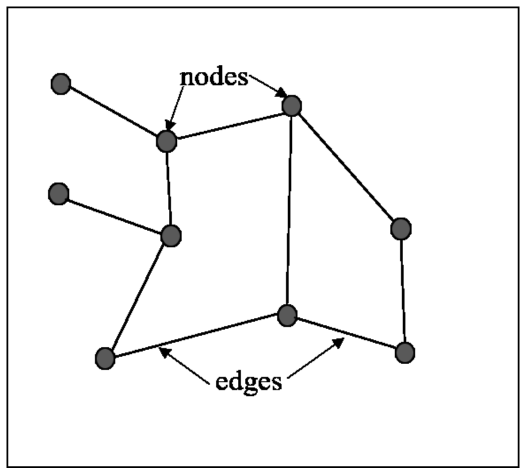
(a) Centralized Network (b) Decentralized Network

Figure 2.2 Centralized and Decentralized Network

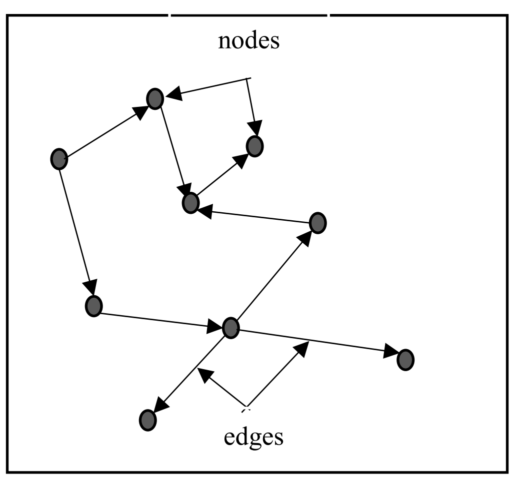
A decentralized system emerges through the lower level components operating on local information. The complex behavior is not exhibited by this decentralized system. The result of the central controller's control over lower level components in decentralized system, including the active supervision of the lower level components instructions of any commanding influence [27].

2.3 Undirected and Directed Network

Undirected and directed network are shown in Figure 2.3.



(a) Undirected Network



(b) Directed Network

Figure 2.3 Undirected and Directed Network

A collection of connected objects is simply called a network. Objects are defined as vertices or nodes. Connection between nodes are called edges.

2.4 Characteristics of Swarm Robots

Swarm robot is a self-organization system characterized by high redundancy. The robot's communication and sensing capabilities are local. Swarm robots do not have access to the global information. The behavior of swarm emerges from the interactions of each individual robot [28]. Characteristics of swarm robots are to promote systems that are scalable, fault tolerant and flexible. Swarm does not rely on any centralized control entity, leaders, or any individual robot playing a predefined role. Figure 2.4 shows the general model of swarm robot. Fault tolerance is abled by high redundancy of the swarm.

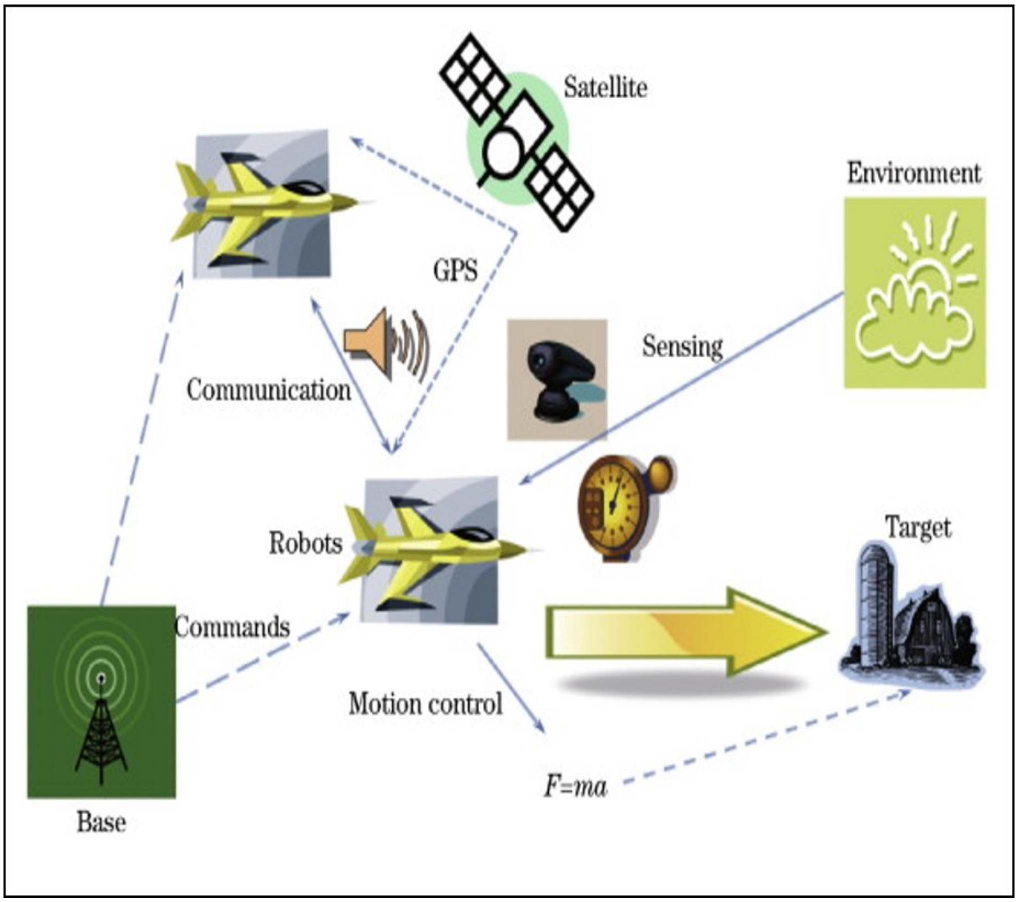


Figure 2.4 General Model of Swarm Robotics

Studying the design of group of simple robot, physical body and controlling behaviors of simple robots are swarm robots. In swarm robots, individual robots are simple and low cost. The size of each robots are also small in swarm robots. For swarm robots, there are many applications. Swarm robot performs tasks that demand for low cost designs, for example, agricultural foraging tasks, mining, etc. In addition, military robots can also form an autonomous army. Using robots in the military, robots would save soldier's lives by removing serving of soldiers. Military robots are also designed for military applications such as attack, rescue and search. In swarms, individual robots are simpler and smaller than a single robot. When compared in battery size, the energy cost of swarm robot is lower than a single robot. So, the lifetime of swarm robots are enlarged.

Swarm robot can also be applied to many problems involving spaces, certain danger may also exit in the environments. Swarm robots can perform the tasks completely, through cooperative behaviors that emerged from individual robot while a single robot hardly adapts the situation. In last decade, swarm robot has become an also important in the research field. Robot swarms can offer many advantages in fault-tolerance, cost per system, efficiency and generality over a single high performance robot. Swarm robots are able to cope when changes in their group size.

Structure of single robot and swarm robot can be found in Figure 2.5. Finally, swarm robotics promotes the development of system that are able to deal with a broad spectrum of environments and operating conditions. Flexibility is enabled by the distributed and self-organized nature of a robot swarm.

(a) Single High Performance Robot

(b) A Robot Swarm with Simple Capability

Figure 2.5 General Structure of Single Robot and Swarm Robot

Swarm robot dynamically allocate themselves to different tasks to match the requirements of the specific environment and operating conditions. Robot can operate on the basis of local sensing and communication and do not rely on pre-existing infrastructure or on any form of global information. Swarm robotic is the study of how large number of relatively simple physically embodied agents can be designed such that a desired collective behavior emerges from the local interactions among agents and between the agents and the environment.

CHAPTER 3

PROBLEM STATEMENT

3.1 Model Definitions and Notations

This section describes the explanations and notations of mobile robots swarm. Considers a swarm is composed of p robotic $n_1, n_2, n_3, \dots, n_i, \dots, n_p$ (for simplicity, a robot n_i, afterwards). Each robot has its own recognition. At initial state, no need to assign roles such as sink, gateway and leader. In graph, sink is node without exiting edges. Gateway allows data that flows from one network to another. A node or robot who leads or commands a group, is called a leader. Definition and notations of robot model can be seen in Figure 3.1.

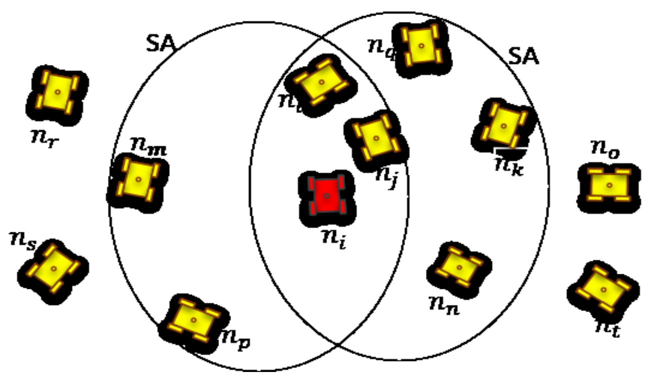


Figure 3.1 Model Definition and Notations of Robot

All robots can work without depending on another robots. Robots can be executed on the same algorithm without occurring or living in a stable conditions, but asynchronously act from other robots.In particularly, a robot can send its data or information within limited submission area range through broadcasting.

Inversely, robots can also get or received transmitted data from another robots. According to n_i communicates to adjacent robot n_j indirectly or directly, communication between n_j and n_i can be defined into one-hop communication and two-hop communication. One-hop communication is a state when n_i and n_j can communicate each other, directly. The one-hop communication with respect to robot $\mathrm{n_i}$ are called $\mathrm{n_i}$'s one-hop robots. In Figure 3.1, $\mathrm{n_i}$'s 1-hop robots are $\mathrm{n_j}, \mathrm{n_p}$,$\mathrm{n_m}$ and $\mathrm{n_l}$. A set of $\mathrm{n_i}$'s one hop robot is represented as 1- hop submission area $\mathrm{SA_i^1}$. If $\mathrm{n_i}$ wants a connection for other robot $\mathrm{n_s}$, it can connect indirectly, so, $\mathrm{n_s}$ is the 2-hop of $\mathrm{n_i}$. The 2- hop robots of $\mathrm{n_i}$ represented as submission area, $\mathrm{SA_i^2}$.

To get the communication states, predefined the waiting time, $\mathrm{T_w}$. Assume data are sending from robot $\mathrm{n_i}$ to $\mathrm{n_j}$, $\mathrm{n_i}$ needed to check its waiting time. If there is any reply from $\mathrm{n_j}$ during $\mathrm{T_w}$, the connection is existing between $\mathrm{n_j}$ and $\mathrm{n_i}$. Time requirements from sending to replying is specified $\mathrm{T_{ij}}$. There is a connection between $\mathrm{n_i}$ and $\mathrm{n_j}$, if $\mathrm{T_{ij}} \leq \mathrm{T_w}$,

Local network can be formed base on the linking conditions. In Figure 3.1, for the local communication distributions, can be described as undirected graph $\mathrm{G_m} = \{\mathrm{V_m}, \mathrm{E_m}\}$. A set of vertices is $\mathrm{V_m} = \{\mathrm{v_1}, \mathrm{v_2}, \dots, \mathrm{v_m}\}$ and set of edges among vertices, $\mathrm{E_m} = \{(\mathrm{v_m}, \mathrm{v_n}) | \mathrm{v_m}, \mathrm{v_n} \in \mathrm{V_m}\}$. Suppose, there is no loops between themselves. The communication is based on the three broadcasting. If robot $\mathrm{n_i}$ wants to connect $\mathrm{n_j}$, firstly, $\mathrm{n_i}$ must notified $\mathrm{n_j}$ is existed or not by broadcasting hello message $\mathrm{hel_i}$ to its neighbors. At the time of receiving any information from$\mathrm{n_j}$, $\mathrm{n_i}$ need to reply its acknowledgement $\mathrm{ack_j}$. Secondly, output message $\mathrm{out_i}$ of $\mathrm{n_i}$ is send to $\mathrm{n_j}$ again. $\mathrm{n_j}$ gives its back acknowledges notice $\mathrm{ack_j}$. Finally, $\mathrm{n_i}$ sends its answer $\mathrm{ans_i}$ whenever $\mathrm{n_i}$ is needed any data $\mathrm{req_i}$ from $\mathrm{n_j}$ through transmission.

3.2 Problem Definition

The system addresses the path planning scheme for a swarm of *n* autonomous robots. This address problem search the shorter communication path planning based on locally broadcasting. Each robot builds local networks from the distribution of neighbor robots by removing redundant communication links. Then, collecting the local networks also allows the robots to reach self-organization of the overall networks.

Although the networks are initially generated, the network is easily changeable by the movements of robot. Therefore, robots partially update the network configuration according to changing situations. Consequently, the addressed problem can be solved by offering a self-healing, network adaptation and self-organization.

Path planning is important for autonomous mobile robots. Path planning allows robots to find the between two points. In addition, paths minimize the amount of turning, and whatever a specific application requires. Algorithms to find a shortest path are important not only in robotics, but also in network routing, video games and gene sequencing.

CHAPTER 4

SHORTER COMMUNICATION-PATH GENERATION

4.1 Self-Organized Local Network

In the nature, social insects activities are based on a self-organizing process. Self-organizing is the process in which the global level of system emerges from numerous interactions process of lower level of the system. Figure 4.1 shows the self-organizing thousand-robot swarm.

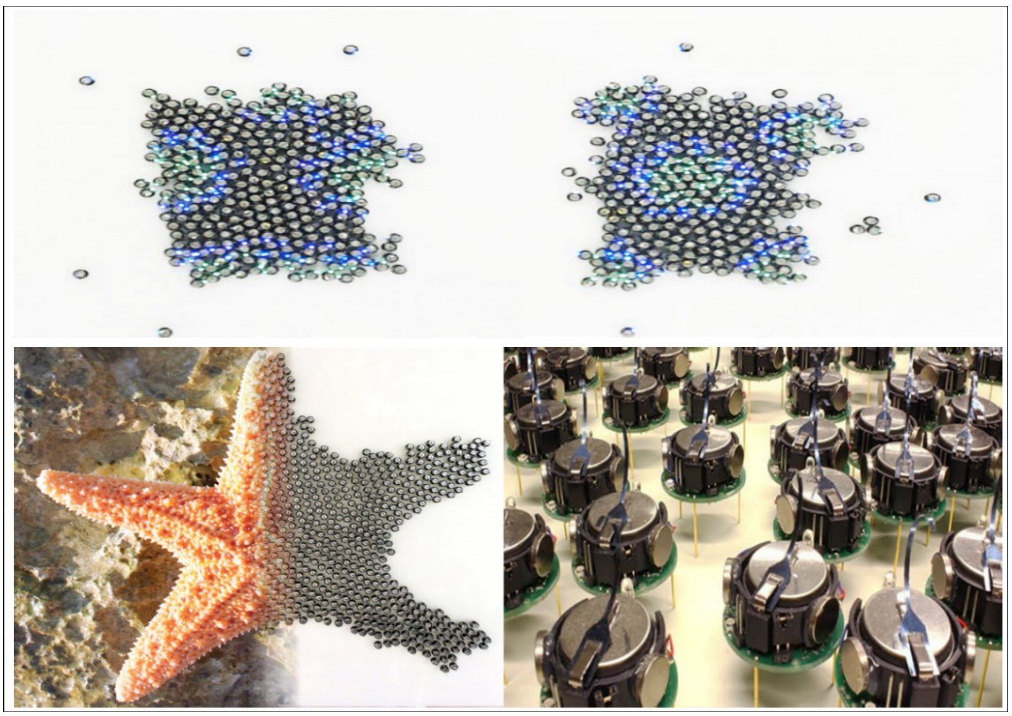


Figure 4.1 Self-organizing Thousand-robot Swarm

In a self-organizing system, there is neither a leader that drives the activities of the group, nor are the individual ants informed about a global recipe to be executed. Each single ant acts autonomously following simple rules and locally interacting with other ants. As a consequence of the numerous interactions among individuals, a coherent behavior can be observed at the colony level.

In the proposed solution, shorter path generation scheme is composed with three main sequential functions: collecting information from the local distribution of neighbor robot (Algorithm 1), selecting neighbors based on the collected information (Algorithm 2) and local shorter path generation (Algorithm 3).

4.1.1 Local Distribution Acquisition

The first function (Algorithm 1) is to investigate the local configuration of adjacent robots by broadcasting and receiving including overhearing. Steps of Algorithm 1 can be seen in Table 4.1.

Table 4.1 Algorithm 1

Algorithm 1: Local Distribution Acquisition

Input : (msg_q from n_q, $\mathrm{T_{iq}}$, $\mathrm{T_w}$)

1. If (new msg_q or msg_q with $\mathrm{T_{iq}} \leq \mathrm{T_w}$) then
2. $\mathrm{SA^1_{i,k}} \leftarrow \mathrm{SA^1_{i,k-1}} \cup \{n_q\}$
3. if ($msg_q = \mathrm{SA^1_{q,k}}$) then

$$\left(\bigcup_{\mathrm{j}\in\mathrm{SA^1_{i,k}}} \mathrm{SA^1_{j,k}}\right) \leftarrow \left(\bigcup_{\mathrm{j}\in\mathrm{SA^1_{i,k}}} \mathrm{SA^1_{j,k-1}}\right) \cup \mathrm{SA^1_{q,k}}$$

$$\mathrm{SA^2_{i,k}} \leftarrow \left(\bigcup_{\mathrm{j}\in\mathrm{SA^1_{i,k}}} \mathrm{SA^1_{j,k}}\right) - \mathrm{SA^1_{i,k}} - \{\mathrm{n_i}\}$$

4. else ($msg_q \neq \mathrm{SA^1_{q,k}}$) then

$$\left(\bigcup_{\mathrm{j}\in\mathrm{SA^1_{i,k}}} \mathrm{SA^1_{j,k}}\right) \leftarrow \left(\bigcup_{j\in SA^1_{i,k}} \mathrm{SA^1_{j,k-1}}\right)$$

$$\mathrm{SA^2_{i,k}} \leftarrow \mathrm{SA^2_{i,k-1}}$$

5. else if (no msg_q within $\mathrm{T_w}$) then

$$\mathrm{SA^1_{i,k}} \leftarrow \mathrm{SA^1_{i,k-1}}$$

$$\left(\bigcup_{\mathrm{j}\in\mathrm{SA^1_{i,k}}} \mathrm{SA^1_{j,k}}\right) \leftarrow \left(\bigcup_{\mathrm{j}\in\mathrm{SA^1_{i,k}}} \mathrm{SA^1_{j,k-1}}\right)$$

$$\mathrm{SA^2_{i,k}} \leftarrow \left(\bigcup_{\mathrm{j}\in\mathrm{SA^1_{i,k}}} \mathrm{SA^1_{j,k}}\right) - \mathrm{SA^1_{i,k}} - \{\mathrm{n_i}\} \cup \{\mathrm{n_q}\}$$

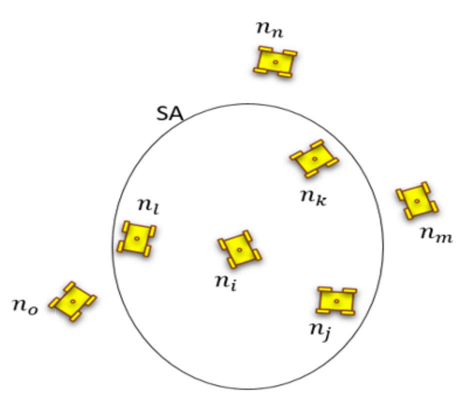
Output: $\left(\mathrm{SA^1_{i,k}},\ \mathrm{SA^2_{i,k}}\right)$

To generate the shorter communication-path, the initial step is to examine the local configuration of neighbor robots around n_i by receiving and broadcasting including overhearing. In Algorithm 1, the input is $msg_{j,k-1}$, and outputs are $SA^1_{i,k}$ and $SA^2_{i,k}$. Robot, n_i computes its outputs $SA^1_{i,k}$ and $SA^2_{i,k}$ based on the input message $msg_{j,k-1}$. Respectively, the inputs and outputs obtained through communications are at the time of k-1 and k. The system omits the notation of k and k-1 afterwards.

At the beginning state, robot n_i broadcasts or send hel_i to adjacent robot n_j. Then, robot n_i waits to get answer their ack_j from robot r_j. Based on the receiving ack_j, n_i computes SA^1_i and asks 1-hop communication of n_j, SA^1_i for their own SA^1_j. Robot r_i builds a local configuration table L_i, for each element of SA^1_i after receiving SA^1_j from its adjacent robot, n_j as shown in Figure (b). Collections of all elements of SA^1_j, ($\cup_{j \in SA^1_i} SA^1_j$) is represented by L_i and shows the direct mapping for individual elements of SA^1_i.

$$SA^2_i = \left(\bigcup_{j \in SA^1_i} SA^1_j \right) - SA^1_i - \{n_i\} \tag{1}$$

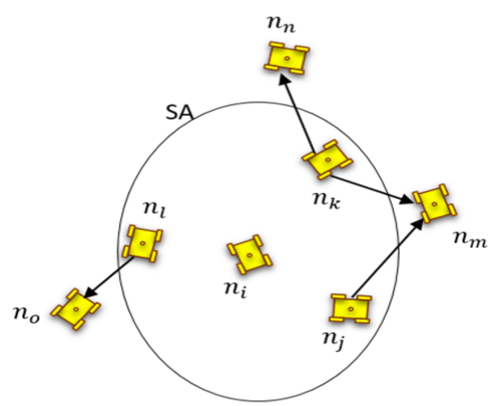
Robot n_i can obtain information located outside of its submission area, SA by computing SA^2_i. n_i can estimate an extended network configuration by SA^1_i and SA^2_i. By the configuration of robot in SA^2_i, n_i can count of communication links between robots and also figure out the configuration of their topology. When algorithm1 is processed in the system, local distribution of robots can be found in Figure 4.2 (a).



(a) Local Distribution

One hop submission area (SA_i^1)	Two hop submission area (SA_i^2)
n_j	n_i,n_k,n_m
n_k	n_i,n_j,n_m,n_n
n_l	n_i,n_o

(b) Local Configuration



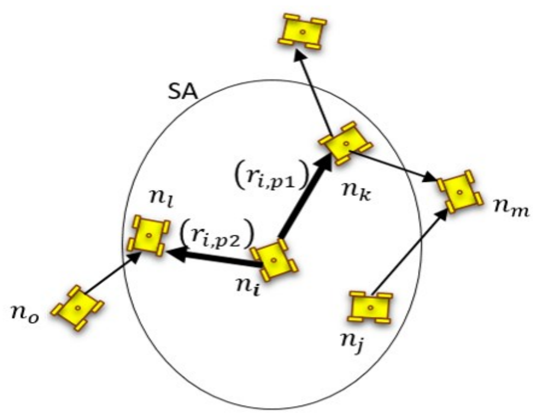
(c) Communication Function from SA_i^1 to SA_i^2

One hop submission area (SA_i^1)	Two hop submission area (SA_i^2)
n_j	n_m
n_k	n_m,n_n
n_l	n_o

(d) Representation of $f_{i,12}$: $T_{i,12}$

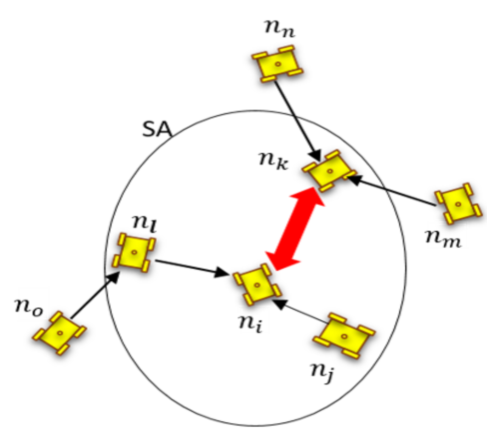
Two hop submission area (SA_i^2)	One hop submission area (SA_i^1)
n_m	n_j, n_k
n_n	n_k
n_o	n_l

(e) Representation of $f_{i,21}$: $T_{i,21}$



(f) Determination of Neighbor n_i

Figure 4.2 Illustration of Shorter Communication-Path Generation



(a) Partially-connected Mesh Network

ID	N_i
n_i	$\{n_k\}$
n_j	$\{n_i\}$
n_k	$\{n_i\}$
n_l	$\{n_i\}$
n_m	$\{n_k\}$
n_n	$\{n_k\}$
n_o	$\{n_l\}$

(b) Neighbor List of Individual Robot in Network

Figure 4.3 Partially Connected Mesh Network and Neighbor List of Individual Robot in Network

4.1.2 Neighbor Determination

For neighbor robot selection, the system defines two communication functions $f_{i,12}$ and $f_{i,21}$ by allowing n_i to examine its communicative mappings between n_j. And then, $f_{i,12}$ on SA_i^1 into SA_i^2 is defined:

$$f_{i,12} : SA_i^1 \rightarrow SA_i^2 \quad (2)$$

$T_{i,12}$ is the representation of $f_{i,12}$. Figure 4.2(c) and (d) is the illustration of $f_{i,12}$ and $T_{i,12}$. $f_{i,21}$ on SA_i^2 into SA_i^1 is defined as :

$$f_{i,21} : SA_i^2 \rightarrow SA_i^1 \quad (3)$$

Similarity, $T_{i,21}$ denotes the representation of $f_{i,21}$. A function from SA_i^1 through SA_i^2 is defined as the composition of $f_{i,12}$ and $f_{i,21}$ and is represented :

$$f_{i,21} \circ f_{i,12} \quad (4)$$

Table 4.2 Algorithm 2

Algorithm 2: Neighbor Determination

Input : $SA^1_{i,k}$, $SA^2_{i,k}$

1. $SA^1_i \leftarrow SA^1_{i,k}$; $SA^2_i \leftarrow SA^2_{i,k}$
2. $N_{i,k} \leftarrow N_{i,k-1}$; $j \leftarrow 1$
3. While $\left(SA^2_i \neq \emptyset\right)$ do
4. $f_{i,12} := SA^1_i \rightarrow SA^2_i$; $f_{i,21} := SA^2_i \rightarrow SA^1_i$
5. $n_{i,pj} \leftarrow \max_{n \in SA^2_i} \left[\text{freq}\left(f_{i,21}(n)\right)\right]$
6. $SA^1_i \leftarrow SA^1_i - \{n_{i,pj}\}$; $SA^2_i \leftarrow SA^2_{i,k} - \{f_{i,12}\left(n_{i,pj}\right)\}$
7. $N_{i,k} \leftarrow N_{i,k} \cup \{n_{i,pj}\}$; $j \leftarrow j+1$

Output: $N_{i,k}$

---

n_i can also estimate the connected state of local network by using composition of $f_{i,12}$ and $f_{i,21}$. In Algorithm 2, outputs of Algorithm 1, SA^1_i and SA^2_i are worked as inputs and its output is N_i. First, n_i also examines the most mapped element of SA^1_i from robots, SA^2_i. n_i can investigate a robot of SA^1_i with the most mapping frequency by applying $f_{i,21}$ to each robot of SA^2_i (to obtain $SA^1_i = f_{i,21}(r)$ where $r \in SA^2_i$).

The most mapping element is selected as the first neighbor $n_{i,p1}$. And then, $n_{i,p1}$ of SA^1_i and directly associated robots of SA^2_i are deleted from SA^1_i and SA^2_i . After expulsion from SA^1_i and SA^2_i , each complementary sets are also defined as $SA^1_{i,(1)}$ and $SA^2_{i,(1)}$, respectively. Same to the process above, n_i finds the second neighbor $n_{i,p2}$ with the most mapping from the elements of $SA^2_{i,(1)}$.

After determination of $n_{i,p2}$, each complementary sets can be defined as $SA^1_{i,(2)}$ and $SA^2_{i,(2)}$. By doing this process, repeatedly until $SA^2_i = \emptyset$, n_i can select its $n_{i,pj}$ in SA^1_i. A set of $n_{i,pj}$ selected by the robot n_i is defined as N_i.

4.1.3 Local Shorter communication-Path Generation

Table 4.3 Algorithm 3

Algorithm 3: Local Shorter Communication Path Generation

Input : $N_{i,k} = \{n_{i,pj} | 1 \leq j \leq m\}$

1. $N_{i,k} \leftarrow \min\{dist(N_{i,k-1})\}$
2. if $(dist(N_{i,pj}) = dist(N_{i,p(j+1)}))$

   $N_{i,k} \leftarrow dist(N_{n_{i,pj}})$
3. $v_{i,i} \leftarrow N_i$, $V_i \leftarrow \{v_{i,i}\}$; $j \leftarrow 1$
4. for ($j \leq m$) do
5. $v_{i,nj} \leftarrow N_{i,k}$; $e_{ij} \leftarrow (v_{i,i}, v_{i,pj})$
6. $V_i \leftarrow V_i \cup \{v_{i,nj}\}$; $E_i \leftarrow E_i \cup \{e_{ij}\}$;
7. $j \leftarrow j+1$;

Output : $G_{i,k} = (V_{i,k}, E_{i,k})$

The input is N_i, and its output is $G_i = (V_i, E_i)$. n_i and the selected $n_{i,pj}$ are considered as individual vertices $v_{i,i}, v_{i,n1}, v_{i,n2}, \ldots, v_{i,nj}, \ldots, v_{i,nm}$ and set of vertices is defined as V_i. Individual edge between $v_{i,i}$ and $v_{i,nj}$ is defined as $e_{ij} = (v_{i,i}, v_{i,pj})$, and E_i denotes $\{e_{ij} | 1 \leq j \leq m\}$. Next, $G_i = (V_i, E_i)$ is formed with respect to n_i.

Figure 4.2 (f) shows the generated local network of n_i where n_k and n_l is selected as $n_{i,p1}$ and $n_{i,p2}$, respectively. Similarly, $G_j = (V_j, E_j)$ can independently built with the same process. After completion of $G_i = (V_i, E_i)$, r_i exchanges SA_i^1 and N_i with its n_j as out_i by broadcasting. Since n_i is connected to $n_{i,pj}$ of N_i like a V_i to V_j connection can be found in Figure 4.2 (f), this can be defined as the star network topology. The collection of $G_i = (V_i, E_i)$ can globally reach self-organization for network G without using centralized control scheme.

When the local star networks overlap each other, the overall network can have the partially connected mesh network topology as shown in Figure 4.3. From the main point of network topology, the scheme proposed to take the advantage of network redundancy through being connected to as many neighbor robots as possible. If individual robots agree on the mutual neighbor, *E* becomes a central communication path.

4.2 Adaptable Self-Organized Communication

The interaction of the swarm is local, by allowing the individuals to join the task at any time without interrupting the whole swarm. The swarm can adapt to the change in population through implicit task re-allocating schemes without the need of any external operation. This also indicates that the system is adaptable for different sizes of population without any modification of the software or hardware which is very useful for real-life application.

Once a network G is generated, its network topology varies by robot movements and/or robot failures. To promptly react to changing situations, it is desirable to modify a part of generated network, G corresponding to the changes than its overall regeneration by all of robots. Under shorter communication-path generation, for n_i to adapt to network changes by partially updating SA_i^1 , SA_i^2 and N_i by the overhearing of adjacent robots.

Assume, there are two network changes in SA_i^1 and SA_j^1. Firstly, we need to consider the situation when a robot n_p approaches the robot n_j of SA_i^1. After n_j notices its local network is changed due to msg_p, n_j examines who is changed in robots of SA_j^1 and SA_j^2. To update the network changes depending on msg_p, n_j partially modifies SA_j^1 and SA_j^2 under Algorithm 1, examines if N_j by Algorithm 2 is changed and determines G_j.

Then, SA_j^1 and N_j are broadcasted to the robots of SA_j^1. Meanwhile, n_i updates SA_i^1 and SA_i^2 based on the received SA_j^1 and N_i are broadcasted. Secondly, a robot n_k disappears within submission area of n_i as it moves far away from n_i. Once n_i cannot receive any msg_k, SA_i^1 is updated and Algorithm 2 is performed. Similarity, n_i broadcasts

SA_i^1 and N_i. Ultimately, shorter communication-path scheme allows n_i to adapt to topological changes by partial modification.

CHAPTER 5

SIMULATION RESULTS AND DISCUSSIONS

5.1 Simulation Results

Shorter communication- path generation scheme is already explained in Chapter 4. When executing the three main sequential functions: collecting information from the local distribution (Algorithm 1), selecting neighbors (Algorithm 2) and local shorter path generation (Algorithm 3), shorter communication path is got. To show the simulation results, user interface is designed as shown in Figure 5.1. To show the validity of shorter communication-path generation scheme, visual studio 2017 is used and performed a series of simulations to show the features of topological adaptation, self-organization and self-healing or robustness of the system.

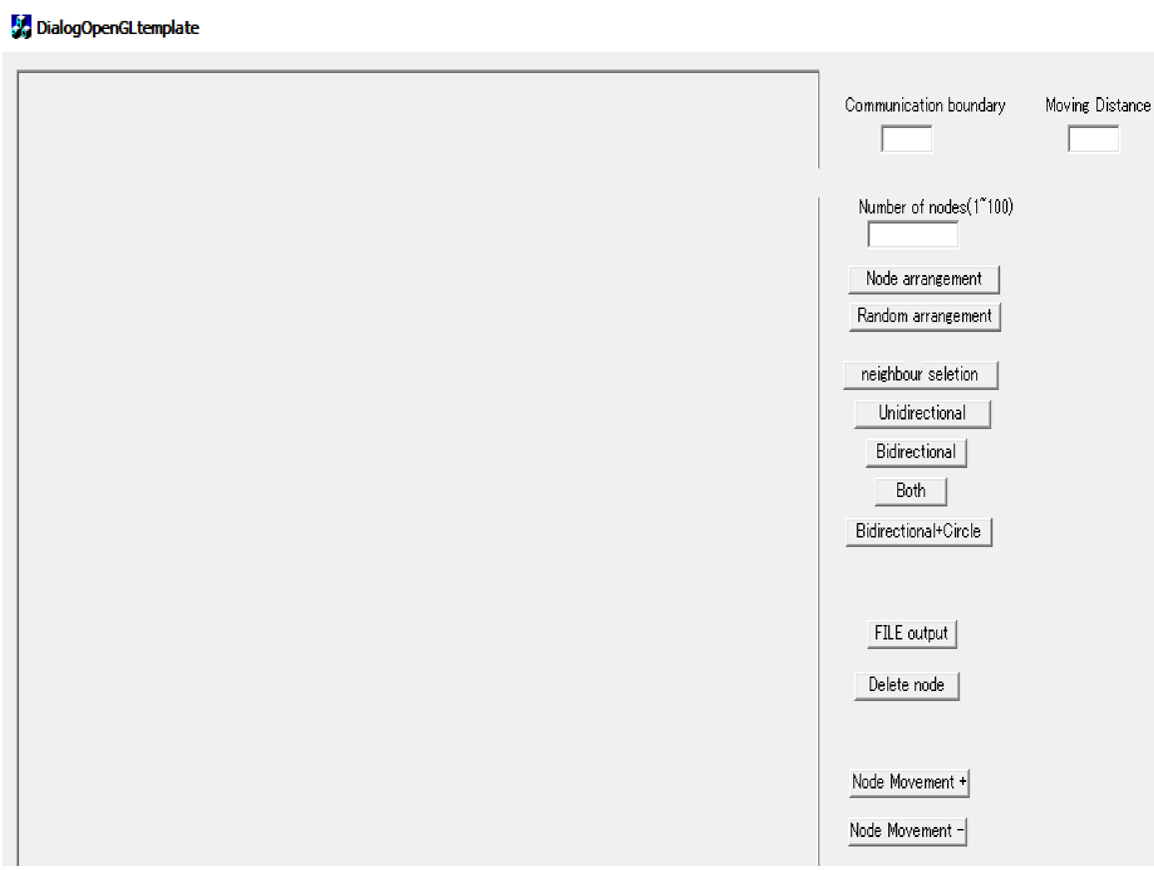
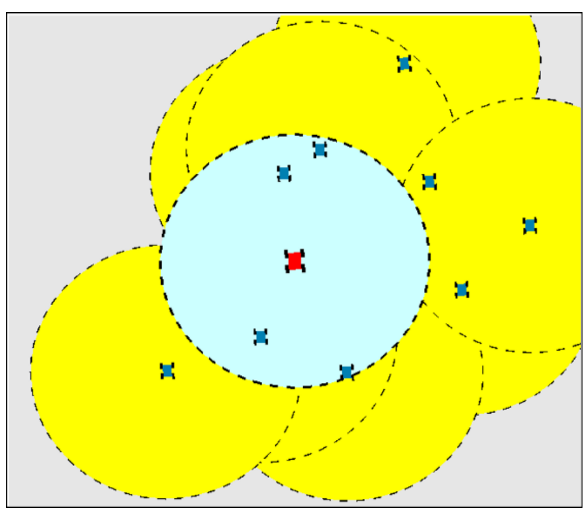


Figure 5.1 Dialog Template for User Interface

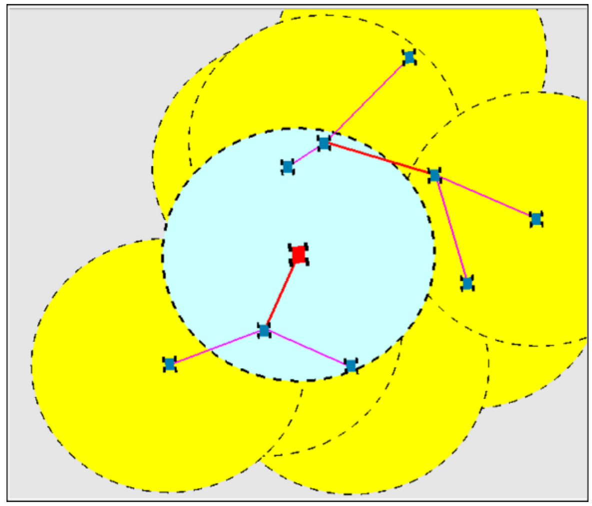
Firstly, for simulations, dialog template can be formed. In Figure 5.1, communication boundary is the submission area for each robot. Moving distance is how much robot want to move. Number of nodes is the number of robot to work or simulate. Specifically, in the simulations, modeled random communication delays during local communications between robots based on the defined packets. By clicking neighbor selection button, the system worked by using the shorter communication-path generation scheme. Unidirectional and bidirectional means the directed edges to neighbors and individual robots agreed on the mutual neighbor after the network generation, respectively.

Both button is for seeing unidirectional and bidirectional. File output button is for showing data in excel file. Delete button is for self-healing. When clicks delete button, the system worked for showing self-healing properties without failing the whole network. Next, the movement node + and movement node - are for robot moving direction.

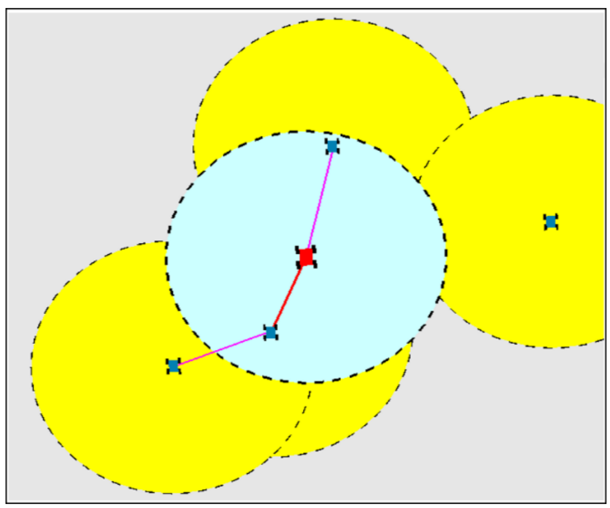
Figure 5.2 shows the simulation result of the generated network by 10 robots. Firstly, sets submission area range to 75 and number of robots for working is 10. User can change any submission area range and number of robots as the user needs. After setting the necessary data, the result can be seen in Figure 5.2(a) as initial stage of the system. Then, the system process according to ALGORITHM step by step, then can see their connection in as shown in Figure 5.2(b). The connection between robots are represented with black and red line.



(a) Initial Distribution



(b) Network Generation



(c) Loss of 5 Robots in 10 Robots Network

Figure 5.2 Simulation Result for Network Generation of 10 Robots

The pink line is the unidirectional and is defined edges to n_{i,n_j} from n_i. The red line shows the bidirectional connection between robots, respectively. Robots can organize the overall network G by the collection of local networks G_i. If robots are failed or disappeared from the network, the system can work without affecting the network.

In Figure 5.2(c), 5 robots are unexpectedly failed or disappeared from Figure 5.2(b), but the system can process. By broadcasting and overhearing information, n_i checks the existence of adjacent robots within submission area range. If neighbor robots disappeared around n_i, the proposed scheme allowed individual robot to partially restore their local networks adapting to topological changes. Table 5.1 shows neighbors ID and distance. In Table 5.1, there are 10 nodes from 0 to 9. For node 0, it has two neighbors, node ID 3 and 4. The distance from node ID 0 to 3 is 47 and node ID 0 to 4 is 68 as shown in Table 5.1. According to shorter path generation scheme, node 0 chooses more shorter distance from its neighbor.

Table 5.1 Neighbors ID and Distance

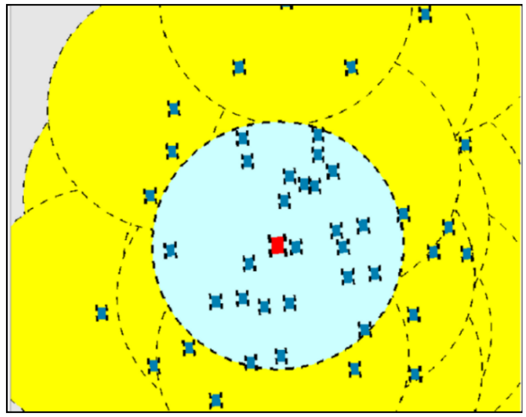
Node ID	Neighbors			
	Node ID		Distance between node and its neighbors	
0	3	4	47	68
1	3	-	57	-
2	9	-	61	-
3	0	-	47	-
4	0	9	68	63
5	0	4	52	24
6	0	3	73	53
7	9	-	65	-
8	4	9	68	71
9	4	-	63	-

Table 5.1 got by the processing of ALGORITHM 1 and ALGORITHM 2. From the result of Table 5.1, ALGORITHM 3 solved to get shorter path as shown in Table 5.2. Simulation result of Figure 5.2(b) are got from the result of Table 5.2. In Table 5.2, node ID 0 chooses the shorter distance from the results (node ID 3) of Table 5.1.

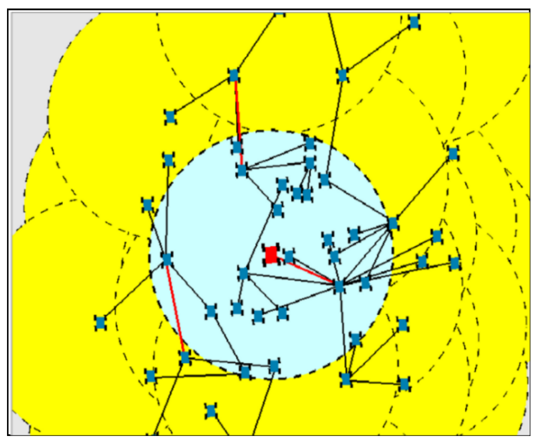
Table 5.2 Neighbor with Shorter Distance

Node ID	Neighbor ID with Shorter Distance
0	3
1	3
2	9
3	0
4	9
5	4
6	3
7	9
8	4
9	4

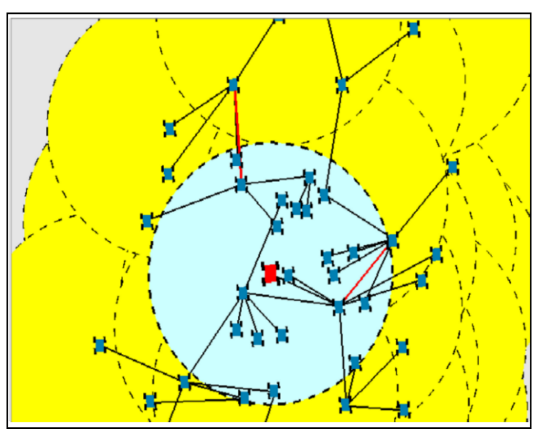
Simulation results for 50 robots can be seen in Figure 5.3. In figure 5.3, submission area sets 75 and number of robot is 50. After setting the necessary data, the system worked by using the propose algorithm. Visual studio 2017 is used for simulation.



(a) Initial Distribution



(b) Network Generation



(c) Loss of 5 Robots in 50 Robots Network

Figure 5.3 Simulation Result for Network Generation of 50 Robots

Simulation results for topological adaptation when 5 red robots move arbitrarily is shown in Figure 5.4. In the generated network ***G***, to make the topological changes, 5 red robot moves arbitrarily and simultaneously. Each local network can carry out the topological changes according to network variation. In Figure 5.4, 5 red robots move and easily adaptable when the topology changed. Figure 5.4 shows the graph for the simultion results of 5 robot moving.

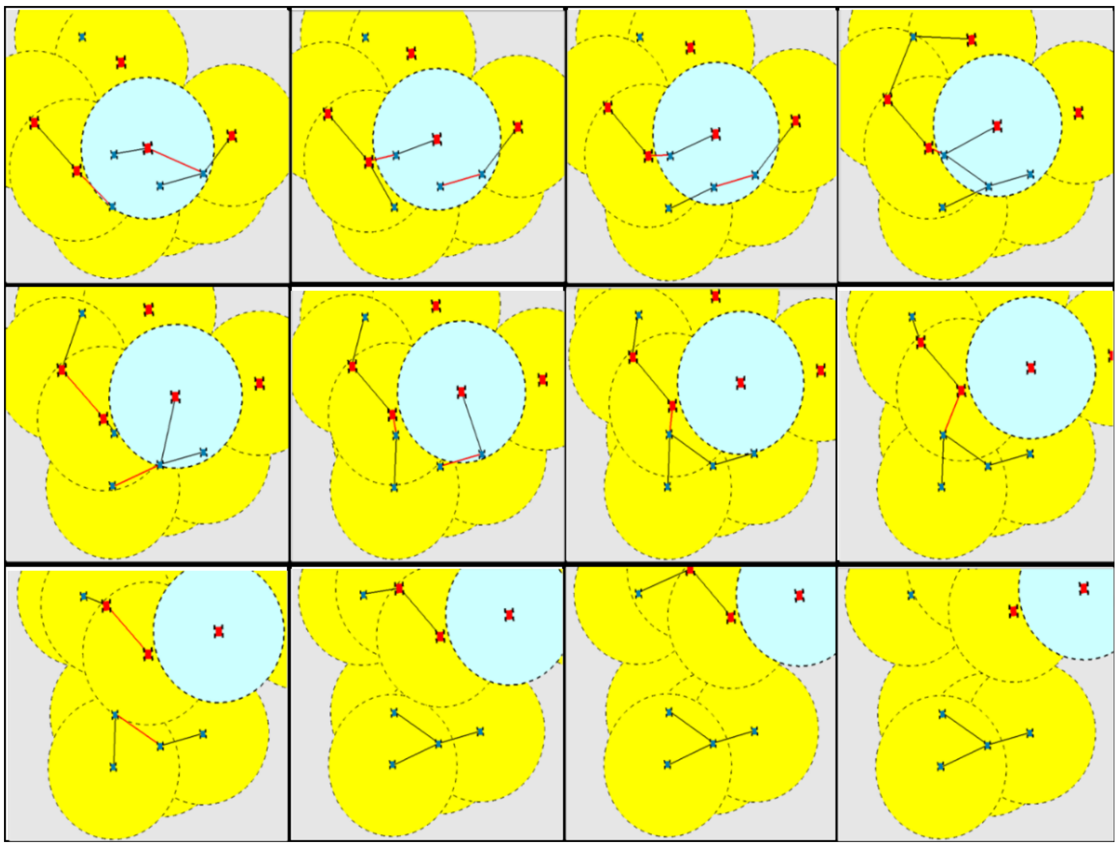


Figure 5.4 Simulation Results for Topological Adaptation When 5 Red Robots Move Arbitrarily

Figure 5.5 shows five robots moving from Figure 5.4.

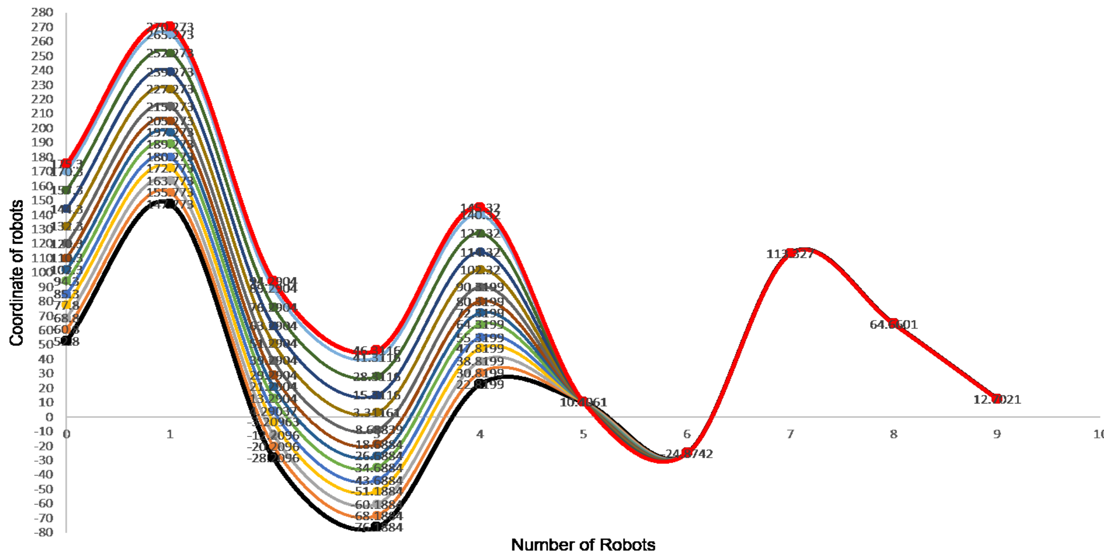
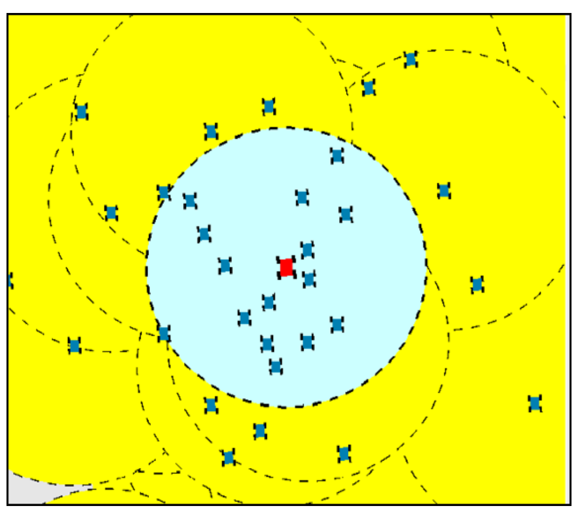
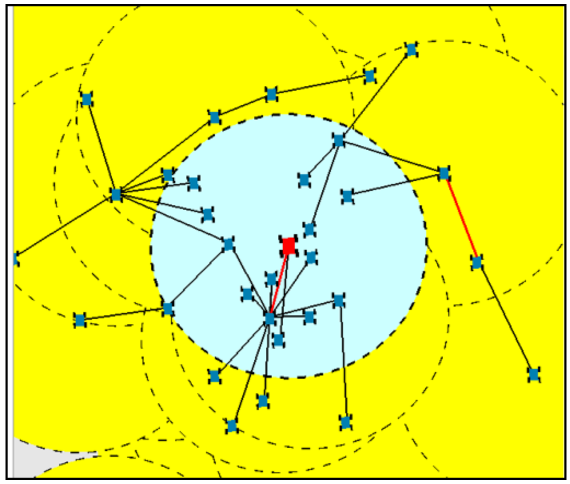


Figure 5.5 Five Robots Moving from Figure 5.4

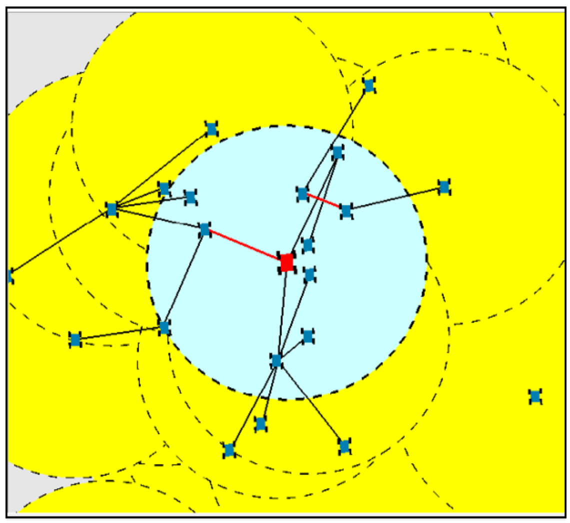
In figure 5.5, the black line is the initial condition without moving. The red line is the final condition after they moved. Robot 0 to 4 are moving but robots 5 to 9 are same initail and final conditions. From the graph and simulation results, the scheme is easily adaptable from their self-organized network. Figure 5.6 shows the simulation result for network generation of 35 robots.



(a) Initial Distribution



(b) Network Generation



(c) Loss of 10 Robot

Figure 5.6 Simulation Result for Network Generation of 35 Robots

For showing self-healing again, sets the submission area to 80 and number of robot is 35 as shown in Figure 5.6. After 10 robots fail from the network, but the system can maintain without failing the whole network.

5.2 Shortest Path Problem

For graphs, shortest path can be defined as directed, undirected or mixed. Finding a path between two nodes such that the sum of the weights of its constituent edges are minimized in graph theory. Shortest path algorithms are mostly applied to find directions for physical locations. For minimal number of moves, shortest path algorithms are used as a solution. In telecommunications or networking, shortest path problem also called min-delay path problem.

Single source shortest path is determining from one node to another and all pair shortest path problem is finding the shortest distance between pairs of nodes or vertices. To compare the system, they have the same point. In shorter communication path generation algorithm, the graph is undirected. For shortest path, the system can also use undirected. The inputs are also the same. The shorter communication path generation scheme has three steps as described in Chapter 4. Figure 5.7 shows the dialog template for user interface when simulating with shortest path algorithm.

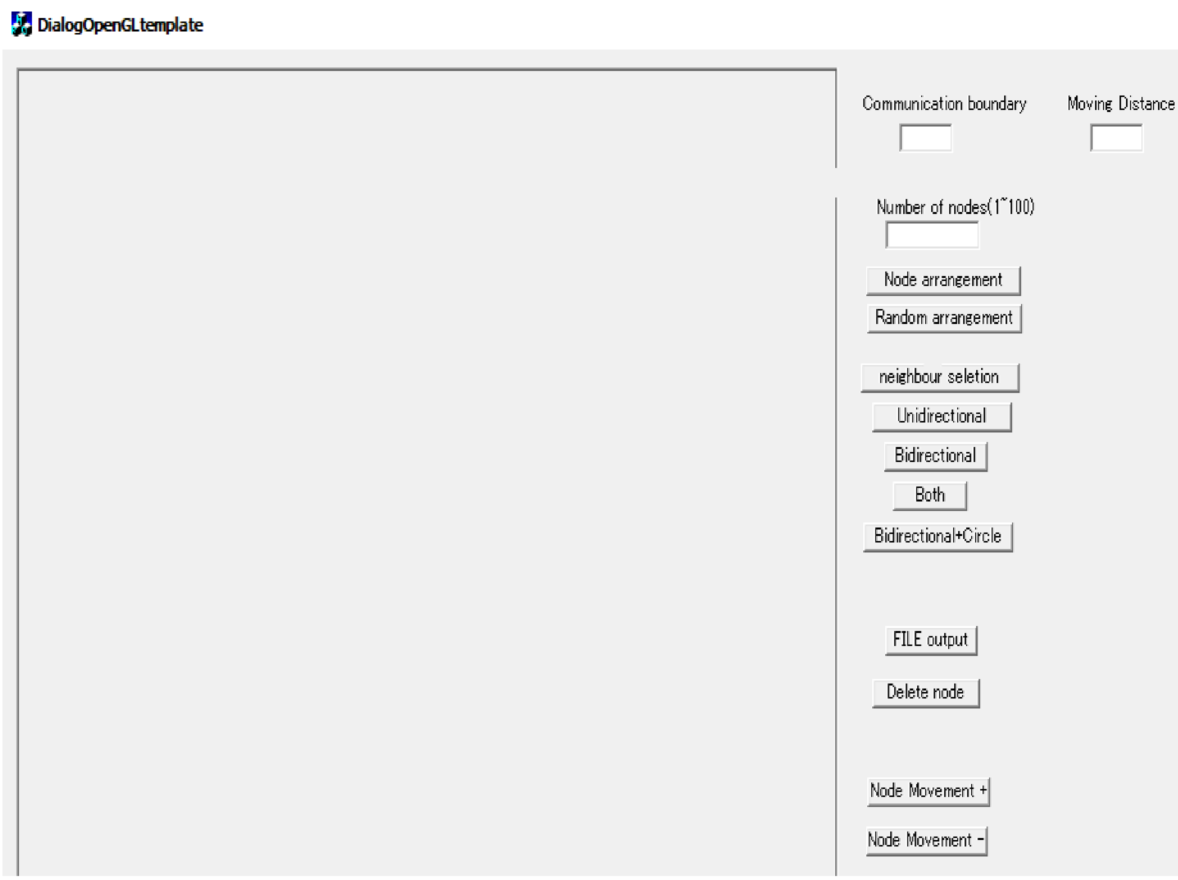
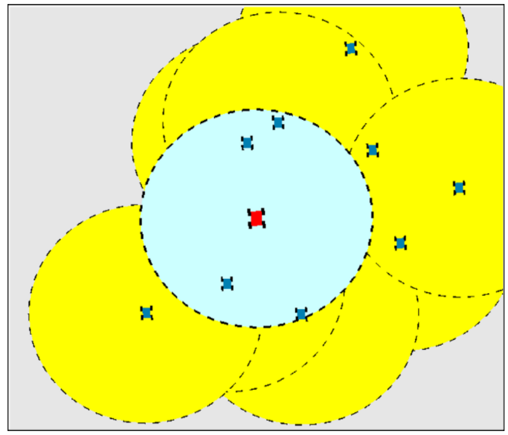


Figure 5.7 Dialog Template for User Interface

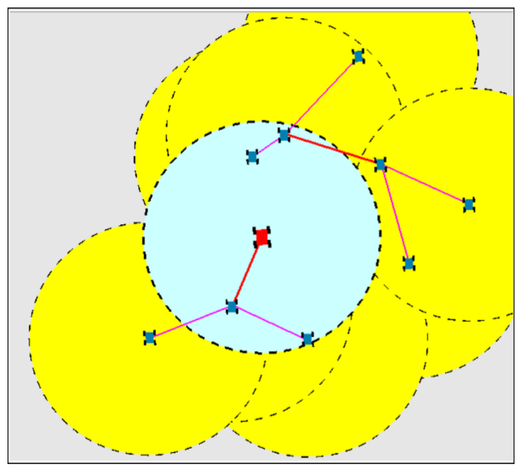
To show the validity of shortest path scheme, visual studio 2017 is used to perform a series of simulation and to show the features of topological adaptation, self-organization and self-healing of the system. In the system, the proposed scheme name is shorter path generation scheme, so the name can be confused with famous shortest path finding algorithm such as A* algorithm, Dijkstra's shortest path planning algorithm and etc. Shortest path algorithms are also used for time reducing when we move from one place to another. In our system, A* algorithm is used when choosing the neighbor of each robot. A* algorithm finds the shortest path from all of the path. Assume that, there are 10 nodes in network. Firstly, node ID 0 finds the shortest path from all of the nodes. At the same time, other node ID 1 to 9 also find the shortest path. User interface for shortest path is also designed as the same with the shorter communication path generation scheme.

In Figure 5.7, communication boundary is the submission area for each robot. Moving distance is how much robot want to move. Number of nodes is the number of robot to work or simulate. Specifically, in the simulations, the system modeled random communication delays during local communications between robots based on the defined packets. By clicking neighbor selection button, the system worked by using the shortest path algorithm.

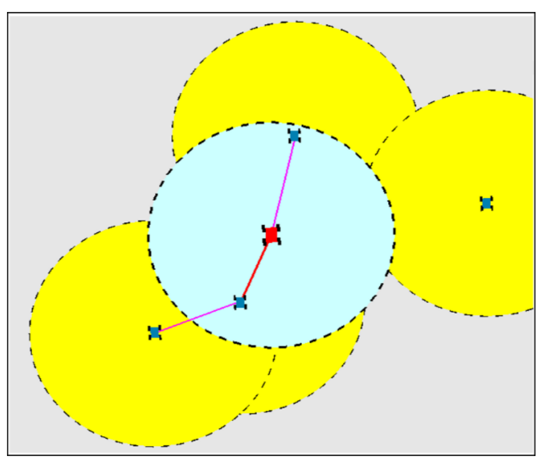
Unidirectional and bidirectional means the directed edges to neighbors and individual robots agreed on the mutual neighbor after the network generation, respectively. Both button is for seeing unidirectional and bidirectional. Delete button is for self-healing. When clicks delete button, the system can work for showing self-healing properties without failing the whole network. Next, the movement node + and movement node - are for robot moving direction. Simulation result for network generation of 10 robots using shortest path is shown in Figure 5.8.



(a) Initial Distribution



(b) Network Generation



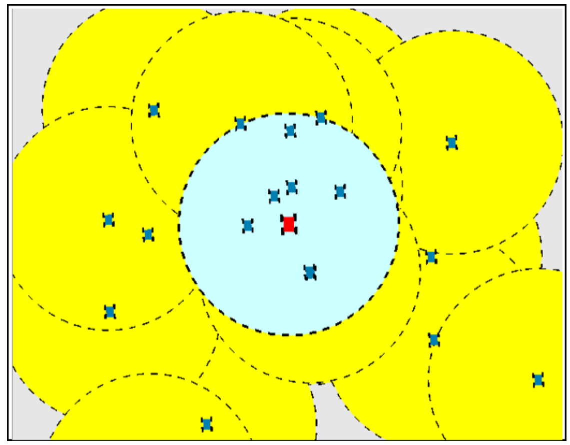
(c) Loss of 5 Robots in 10 Robots Network

Figure 5.8 Simulation Result for Network Generation of 10 Robots Using Shortest Path

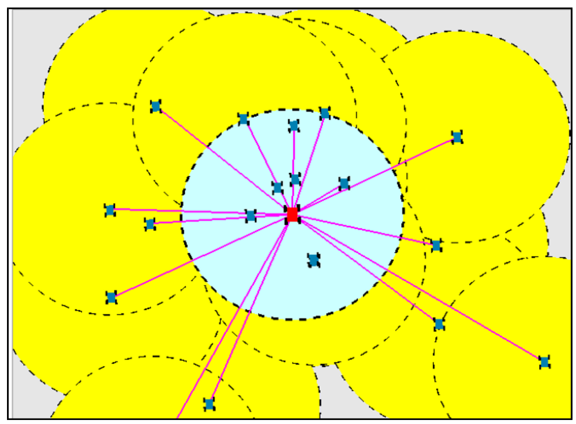
In the proposed system, A* algorithm is used when finding the neighbors. And then, A* algorithm finds the shortest path from all of the network. For example, there are 5 nodes in network, the system needed to find all path form node 0 to 5. So, searching all path takes more time. If there are 50 nodes in the network, it needs to find the path from node 0 to 49. So, finding path between all pair of nodes can take more processing time. When the working objects become more and more, the processing time will be longer. So, shorter communication path generation scheme can be reduced the processing time by comparing with shortest path finding algorithm. When comparing with the shortest path algorithm, the system can be worked by interacting with each other. So, the proposed scheme also reduce the processing time. Figure 5.8: shows the simulation result of the generated network by 10 robots. Firstly, sets submission area range to 75 and number of robots for working is 10. The user can also change any submission area range and number of robots as we need. After setting the necessary data, the results can be seen in Figure 5.8(a) as initial stage of the system.

Then, the system process according to algorithm step by step, then, the connection of each robot is shown in Figure 5.8(b). The connection between robots are represented with pink and red line. The pink line is the unidirectional and is defined edges to n_{i,n_j} from n_i. The red line shows the bidirectional connection between robots, respectively. Robots can organize the overall network G by the collection of local networks $\mathrm{G_i}$. If robots are failed or disappeared from the network, the system can work without affecting the network.

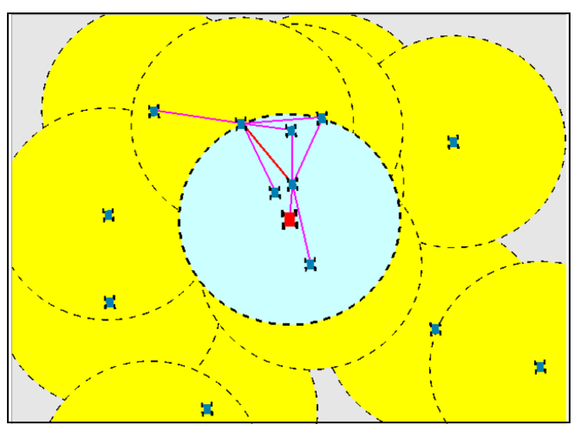
In Figure 5.8(c), 5 robots are unexpectedly failed or disappeared from Figure 5.8(b), but the system can process. By broadcasting and overhearing information, n_i checks the existence of adjacent robots within submission area range. If neighbor robots disappeared around n_i, the proposed scheme allowed individual robot to partially restore their local networks adapting to topological changes. Figure 5.9 shows the simulation result for network generation of 20 robots by using shortest path. In Figure 5.9, communication boundary is 75 and number of robots are 20. After setting the data, the system simulated using shortest path scheme.



(a) Initial Distribution



(b) Network Generation



(c) Loss of 5 Robots

Figure 5.9 Simulation Result for Network Generation of 20 Robots Using Shortest Path

Simulation results for 50 robots can be seen in Figure 5.10. Simulation results in Figure 5.10, topological adaptation of robots where one red robots move arbitrarily. In the generated network ***G***, to make the topological changes, red robot moves arbitrarily and simultaneously.

Each local network can carry out the topological changes according to network variation. Red robot moves and easily adaptable when the topology changed. From simulation results, the proposed scheme is easily adaptable from their self-organized network.

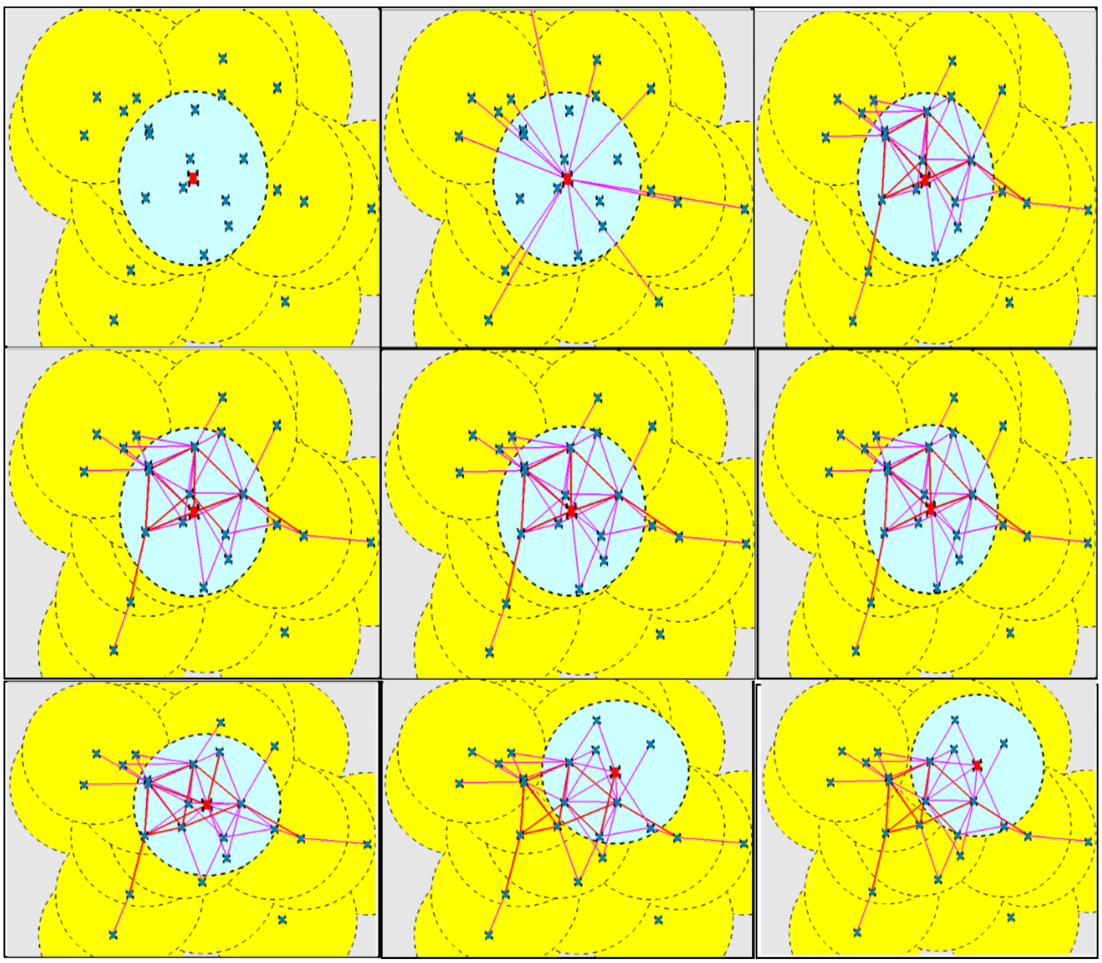


Figure 5.10 Simulation Results for Topological Adaptation when One Red Robot Moves Arbitrarily

When 10 robots are processing with shortest path and shorter communication path, persons can also see the processing time as shown in Figure 5.11 and Figure 5.12.

Figure 5.11 Processing time of 10 Robots with Shorter Communication Path

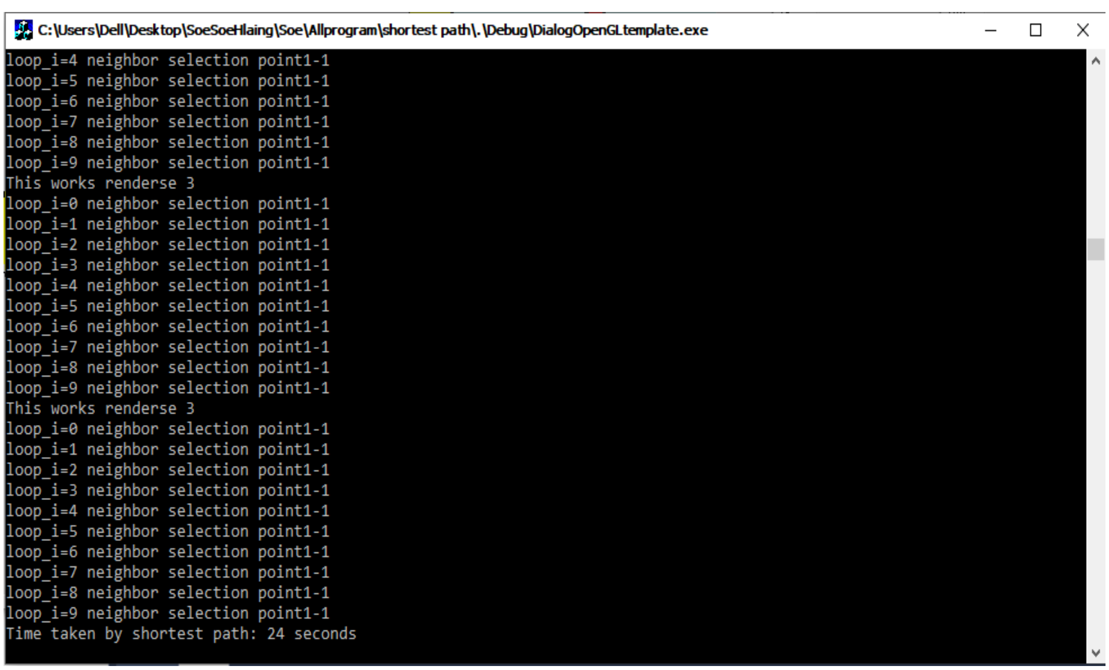


Figure 5.12 Processing time of 10 Robots with Shortest Path

Processing time of 20 robots can also see in Figure 5.13 and Figure 5.14.

Figure 5.13 Processing time of 20 Robots with Shorer Communication Path

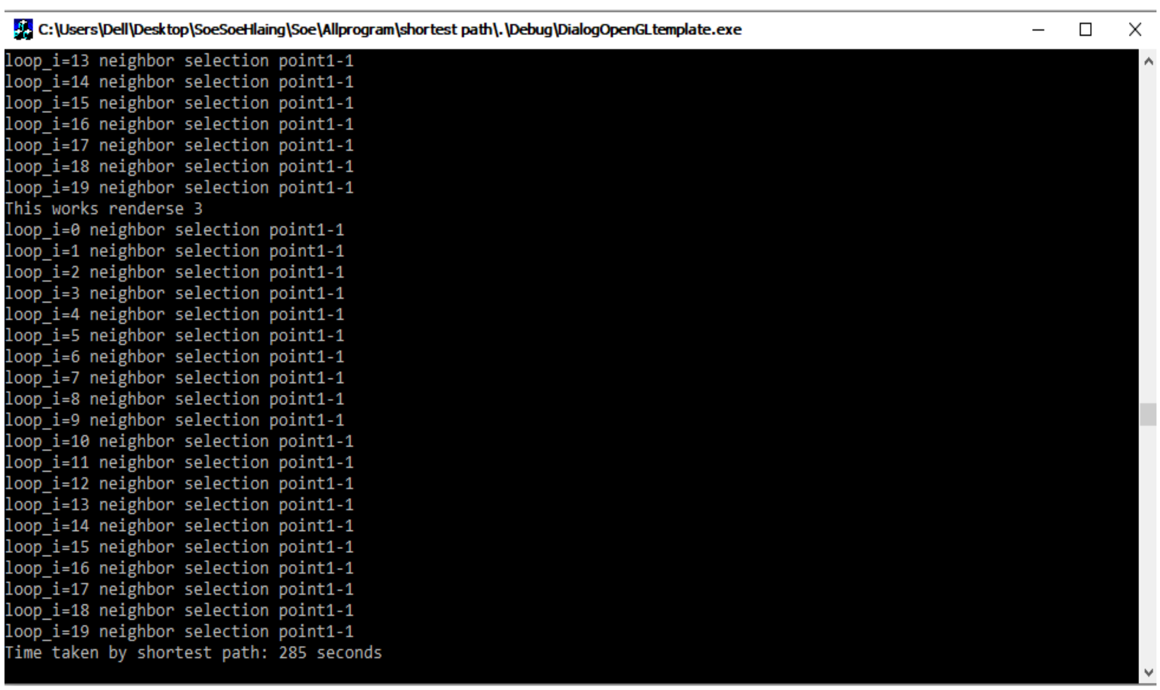


Figure 5.14 Processing time of 20 Robots with Shortest Path

Table 5.1 shows the comparison of processing time from Figure 5.11 to Figure 5.14.

Table 5.3 Processing Time of Shorter Communication Path and Shortest Path

Number of Node	Shorter Communication Path	Shortest Path
10	22 seconds	59 seconds
20	24 seconds	285 seconds

CHAPTER 6

CONCLUSION AND FURTHER EXTENSIONS

The proposed system is studied the self-organized swarm robots with minimum functions requirements based on two-dimensional plane. All robots are allowed for dynamically select and interact. The addressed scheme is verified by showing extensive simulation results. The proposed solution focus to find shorter path from the neighbors obtained from local network configuration by adapting the rapid change of topology. When each robot established a communication to its neighbor robots, it becomes a local network. When combine all of the partly connected local networks, swarm robots may be self-organized.

6.1 Benefits of the System

Shorter communication path generation scheme can be performed with minimal requirements and also reduced the processing time. The system can be effectively and successfully worked on simulation. It can be very useful in real time. Swarm robotic systems are often used to solve tasks that might be inherently impossible or too complex for a single robot to tackle, such as, collective transport, self-assembly, task allocation, chain formation collective exploration. Most of path planning methods take long time to process. But, if the person uses the proposed scheme for transporting an object by robots, namely pushing, pulling and caging, the person can save time to work in real world.

6.2 Limitations of the System

Although the proposed system has several advantages, it has some limitations of the system. The first limitation of the system is that the system can be processed on a two-dimensional plane. The second limitation is moving direction. The system is mainly considered the self-organization of swarms. So, robot can move only upward and downward directions.

6.3 Further Extensions of the System

A number of further extensions are possible to the developed system. The system will also consider to further improving the system fusing more features of useful information. The implemented system used two-dimensional plane. Other features such as three-dimensional plane can also be used to improve the simulation design. Moving direction of robots can also be considered as many directions. The shorter path generation scheme not considered about overlapping case when robots moved. Overlapping can be considered to get more better results.

REFERENCES

[1] Brambilla, Ferrante, Dorigo, “Swarm Robotics: A Review from the Swarm Engineering Perspective”, Swarm Intelligence, 2013.

[2] Bayındır, Sahin, “A Review of Studies in Swarm Robotics”, Turkish Journal of Electrical Engineering and Computer Sciences, 2007.

[3] Bayındır, “A Review of Swarm Robotics Tasks”, Journal of Neurocomputing 2015.

[4] M. Wahde, “Introduction to Autonomous Robots: Lecture notes in Autonomous Agents”, Chalmers University of Technology, Sweden, 2012.

[5] Z. Hendzel, J. Wiech, “Robotic Swarm Self-Organisation Control,” al. Powstańców Warszawy 12, 35-959 Rzeszów, Poland, received 26 June 2018, revised 25 June 2019, accepted 28 June 2019.

[6] H. C. H. Chiu, W.M. Shen, “Anchor Self-configuring Robotic Network”, IEEE International Conference on Robotics and Automation, pp.53755380, 2011.

[7] K. Tatara, G. Lee, H. Ono, N.Y. Chong, “Locally Communicative Interaction Framework for Adaptively Self-organizing Mobile Sensor Networks”, 2012 IEEE International Conference on Automation Science and Engineering (CASE) Seoul, Korea, August 20-24, 2012.

[8] G. Lee, N. Y. Chong, “A Geometric Approach to Deploying Robot Swarms,” Ann. Mathematics and Artificial Intelligence”, vol.52, no.2-4, pp.257-280, 2008.

[9] C. R. Kube and E. Bonabeau, “Cooperative Transport by Ants and Robots”, International Robotics and Autonomous Systems, pp. 85–101, 2000.

[10] B. Shucker, T. D. Murphey, and J. K. Bennett, “Convergence Preserving Switching for Topology-dependent Decentralized systems”, IEEE Transaction on Robotics, vol.24, no.6, pp.1405-1415, 2008.

[11] Trianni, Nolfi, "Engineering the Evolution of Self-organizing Behaviors in Swarm Robotics", Artificial Life, 2011.

[12] Camazine, Deneubourg, Franks, Sneyd, Theraulaz, "Self-organization in Biological Systems", Princeton University Press, Princeton, NJ, 2003.

[13] Dorigo, Trianni, Sahin, GroB, Baldassarre, "Evolving Self-organizing Behaviors for a Swarm-bot", Autonomous Robots 2004.

[14] M.Wahde, "A General-purpose Method for Decision-making in Autonomous Robots", International Next-Generation Applied Intelligence, Springer, pp. 1–10, 2009.

[15] Cully A, Clune J, Tarapore D, Mouret JB, "Robots that Can Adapt Like Animals", Nature 2015, 521 (7553):503–507.doi: 10.1038/nature14422 PMID: 26017452.

[16] M. A. Hsieh, A. Cowley, V. Kumar, and C. J. Taylor, "Maintaining Network Connectivity and Performance in Robot Teams," Journal of Field Robotics, vol.25, no.1-2, pp.111-131, 2008.

[17] Sheng X N, Xie W W, "Real-time Path Planning for Mobile Robots based on Quantum Evolutionary Algorithm", Computer Science, 40(5):229–232,2013.

[18] C. R. Kube and E. Bonabeau, "Cooperative Transport by Ants and Robots", International Robotics and Autonomous Systems, pp. 85–101, 2000.

[19] E.Tuci, "Cooperation Through Self-assembly in Multi-robot Systems", International ACM Transactions on Autonomous and Adaptive Systems (TAAS) 1.2 pp. 115–150,2006.

[20] M. Wahde, "A General-purpose Method for Decision-making in Autonomous Robots", International Next-Generation Applied Intelligence, Springer, pp. 1–10, 2009.

[21] M. Wahde. “Introduction to Autonomous Robots: Lecture notes in Autonomous Agents”, Chalmers University of Technology, Sweden, 2012.

[22] Saska, M. Kasl, Z. Preucil, “Motion Planning and Control of Formations of Micro Aerial Vehicles”, In Proceedings of the 19th World Congress of the International Federation of Automatic Control, 2014.

[23] Brambilla, Ferrante , Birattari M, Dorigo M, “Swarm Robotics: A Review from the Swarm Engineering Perspective”, Swarm Intelligence, 2013.

[24] Bayındır L, Şahin E, “A Review of Studies in Swarm Robotics”, Turkish Journal of Electrical Engineering and Computer Sciences, 2007.

[25] Bayındır L, “A Review of Swarm Robotics Tasks”, Journal of Neurocomputing, Volume 172, 8 January 2016.

[26] Dorigo M, TrianniV, Şahin E, GroB R, Labella TH, Baldassarre G, et al, “Evolving Self-organizing Behaviors for a Swarm-bot”, Autonomous Robots 2004.

[27] Cully A, Clune J, Tarapore D, Mouret JB, “Robots that Can Adapt Like Animals”, Nature 2015, 521 (7553):503–507.doi: 10.1038/nature14422 PMID: 26017452.

[28] Camazine S, Deneubourg JL, Franks NR, Sneyd J, Theraulaz G, Bonabeau E, “Self-organization in Biological Systems”, Princeton University Press, Princeton, NJ, 2003.

[29] Baldassarre G, Trianni V, Bonani M, Mondada F, Dorigo M, Nolfi S, “Self-Organized Coordinated Motion in Groups of Physically Connected Robots”, IEEE Transactions on Systems, Man and Cybernetics, 2007.

[30] Sperati V, Trianni V, Nolfi S, “Self-organised Path Formation in a Swarm of Robots”, Swarm Intelligence, 2011.

[31] Trianni V, Nolfi S, “Engineering the Evolution of Self-organizing Behaviors in Swarm Robotics”, Artificial Life, 2011.

LIST OF PUBLICATIONS

[1] Soe Soe Hlaing, Shu Takaira, Geunho Lee, "Communicative Interaction Scheme of Mobile Robot Swarms", 24th International symposium on Artificial Life and Robotics (AROB 2019).

[2] Soe Soe Hlaing, Geunho Lee, Takeshi Okuno, "Shorter Communication-Path Generation Scheme for Robot swarms", 2019 IEEE 8th Global Conference on Consumer Electronics (GCCE 2019).