Connectivity Maintenance for Robotic Swarms by Distributed Role Switching Algorithm

Kazuho Kobayashi
Graduate School of Engineering Science
Yokohama National University
Kanagawa, Japan
kobayashi-kazuho-dj@ynu.jp

Higuchi Takehiro
Faculty of Environment and Information Sciences
Yokohama National University
Kanagawa, Japan
higuchi@ynu.ac.jp

Seiya Ueno
Faculty of Environment and Information Sciences
Yokohama National University
Kanagawa, Japan
ueno-seiya-wk@ynu.ac.jp

***Abstract*—Swarm robotics requires a practical scheme to maintain operator supervision for the acceptability of autonomy of the systems by humans. For this purpose, this paper proposes a distributed algorithm for continuous connectivity between robots and a base station to maintain the controllability and transparency of the swarm. This algorithm forms network topology among the swarm members and deploys repeaters to maintain the connection to the base station by a role switching scheme. Preliminary simulations have shown that the revised acute angle test reduced the cost of the network formation with the Gabriel graph topology. Through the simulated patrol missions, the proposed algorithm successfully maintained the continuous connectivity between the base station and the swarm members without significant inequality in the computational cost among swarm members. Furthermore, as the number of robots increases, the computational cost per robot does not increase significantly. These results indicate the distributed nature and scalability of the proposed algorithms.**

***Index Terms*—Swarm robotics, Role switching, Decentralized algorithm, Connectivity maintenance**

I. Introduction

Swarm robotics is a way to operate multiple robots by local interactions among robots and surrounding environments. It has several advantages, such as robustness, scalability, and flexibility [1], and gaining attention in robotics technologies.

Despite these advantages, there are few cases where robotic swarms are deployed for real-world applications [2]. According to the literature, the concerns include individual robot failure, communication reliability, and system predictability. System predictability is a characteristic to what extent the system can be expected to behave as intended by human operators. This concern is a significant problem for whole robotics areas, especially in complicated situations and life-threatening applications, such as medicine, security, and rescue. In swarm robotics, this concern becomes more severe because of the characteristic that system behaviors emerge from robot local interactions, which are usually invisible from out of the system [2].

To relieve the concern, a practical scheme is required to ensure supervision from humans. Maintaining connectivity among swarms, including a base station (BS), is fundamental to realizing this requirement. In this context, BS corresponds to human operators or interface to their authority to supervise the system and make high-level decisions. Especially for tasks that should be solved collectively, it is essential to consolidate swarm members to a network frontier. Collected robots at the forefront of the network provide immediate engagement to those tasks according to instructions from the BS.

The contribution of this research is a proposal of a decentralized, scalable, and low-cost algorithm for continuous connectivity of robotic swarms, including fixed base stations. As an extension of the existing research [3], the algorithm employs role switching to retain the connectivity while the related processes depend on locally available information. The following sections review the existing research (Sect.II), describe the proposed algorithm (Sect.III), and show the result of simulations (Sect.IV), discussion (Sect.V), and conclusions (Sect.VI).

II. Related Works

Swarm connectivity is a topic that is widely researched. Amigoni et al. reviewed research on multi-robot exploration with various restrictions in terms of connectivity [4]. The review categorized those restrictions as either *event-based* or *continuous*.

Event-based connectivity is a concept that requires recovering the connection triggered by particular events. According to Banfi et al., this is further categorized as periodic and recurrent [5]. Periodic connectivity requires connections every after a past of a specific length of time [6] [7]. Recurrent connectivity, with more adaptivity, requires connections every after each robot gains new information. This timing usually corresponds to the deployments or arrivals of robots to their assigned area [5] [8] [9].

On the other hand, as this research's main point of interest, continuous connectivity requires connections among robots

This work was presented in part at the joint symposium of the 27th International Symposium on Artifcial Life and Robotics, the 7th International Symposium on BioComplexity, and the 5th International Symposium on Swarm Behavior and Bio-Inspired Robotics (Online, January 25-27, 2022).

The published version of this manuscript is available at: Kobayashi, K., Higuchi, T. Ueno, S. Connectivity maintenance for robotic swarms by distributed role switching algorithm. Artificial Life and Robotics (2022). https://doi.org/10.1007/s10015-022-00803-5

during whole missions. Although the event-based connectivity may be efficient, still in cases where real-time access to information is necessary, continuous connectivity is needed [10]. Arkin and Diaz studied three exploration algorithms with continuous connectivity, varied by degree of apriori knowledge on mission spaces [11]. Mukhija et al. proposed the exploration algorithm to maintain connections to fixed BS by propagating a tree network [12]. Nestmeyer et al. showed an exploration algorithm to arrive at multiple locations assigned to each robot by switching priority with maintaining connectivity among robots [13]. Hung et al. developed a network topology management scheme by hierarchical control to maintain continuous connectivity [14].

This paper proposes an algorithm to maintain continuous connectivity in a distributed and scalable manner. The novelty of this research is that the proposed algorithm can maintain connectivity without global coordination or high computational cost. Existing studies on continuous connectivity frequently depend on central coordination by a central agent. Other distributed methods, often with high computational costs, require the swarm member to infer the network topology or other global swarm status [15] [16] [17] [18]. This research builds on these efforts to develop a simpler algorithm for continuous connectivity by introducing role assignment and switching. The main part of the algorithm only requires behaviors based on adjacent robots' information. The following sections describe the detail of the proposed algorithm and its performance.

III. Methodologies

A. Problem Settings

This research considers a patrol mission for disaster areas, for example, to detect the objects of interest, such as victims and malfunctioning instruments. A robotic swarm R consisting of N homogeneous mobile robots: $R = \{r_1, r_2, ..., r_N\}$ are deployed for this mission to detect and report the objects of interest to the BS. In the current settings, the study regards the robots without failures or malfunctions. All variables related to r_i are shown with the suffix i. If not explicitly indicated, they are status at timestep t. Other assumptions for the robots are:

1) The robots can sense their accurate location $\boldsymbol{x}_i \in \mathbb{R}^2$, and inform the neighbors via communications.
2) The robots can identify other robots in range d_s.
3) The robots can keep connected and communicate with the others in range d_c.
4) Robots with connections can exchange information on their status. Multi-hop communications are also available, consuming a single timestep per hop.

1 is assumed to be achieved by some techniques, such as SLAM by external sensors (e.g., vision sensors) or dead reckoning by internal sensors (e.g., inertial measurement units). Similarly, 2 is assumed to be performed by external sensors (e.g., depth cameras) or near-range communication devices(e.g., RFID tags).

B. Role Switching for Continuous Connectivity

The robots behave according to their roles: $role_i = \{base\ station, repeater, local\ leader, explorer\}$. The base station is assigned to r_1 and fixed to the origin: $\boldsymbol{x}_1 = (0, 0)$ to be an interface to the central agency. Repeaters maintain the connection between the base station and a local leader. The local leader represents a subgroup of the swarm consisting of explorers. It also communicates with the base station if needed. Explorers are swarm members as the task detection and processing executor acting with the local leader. This study omits the detailed behavior of explorers except for moving because the study simplifies the simulated mission as an elementary test. A typical state of the swarm is shown in Fig.1. In the figure, the circles and their attached triangles represent the robots and their orientations, respectively. The solid lines between the circles indicate that the robots' communication is available (i.e., connected). A group of robots directly connected to r_i is denoted as A_i. For repeaters and local leaders, a robot $r \in A_i$ at the closer side to the base station is called a parent robot r_{p_i}, and similarly, a robot $r \in A_i$ at the closer side to the explorers is called a child robot r_{c_i}. As for local leaders, child robots include all explorers directly connected to the local leaders.

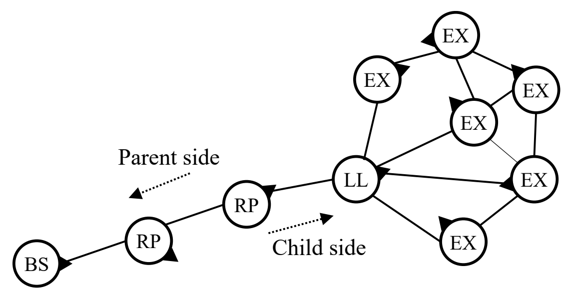


Fig. 1: A typical form of a swarm (BS: Base station, RP: Repeater, LL: Local leader, EX: Explorer). The circles, triangles, and solid lines correspond to the robots, their orientation, and communication connections between them, respectively

Each robot acts as one of these four roles and switches or requests others to switch roles based on local information to maintain a connection from a base station. The basic idea is to have the local leader switch its role to repeater or explorer and request an adjacent to become a local leader instead. These decisions are made based on the distances between the local leader and its parent robot as an indicator of network vulnerability towards the base station. The detailed process is shown in Algorithm 1, where $d_{i,j} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$, $|A_i|$ is the number of robots belonging to A_i. $\eta_a,\ \eta_b (0 < \eta_a,\ \eta_b < 1)$ are parameters for margins. The requests to change roles are made by the inter-robot communications. As Fig.1 indicates, there are several additional assumptions for connections between robots:

1) The base station connects to one repeater or local leader.
2) Repeaters connect with two robots, a parent side base station or repeater, and a child side repeater or local leader.

3) The local leader connects to a parent side base station or repeater and connects to multiple explorers.

The robots broadcast their state to other robots directly connected at every timestep. The number of this type of messages r_i receives corresponds to $|A_i|$, and each message contains the information on $r \in A_i$, such as locations and velocities required in algorithms described in the following sections.

Algorithm 1 Process to switch roles for connectivity

```
when controlling a local leader: r_i
if d_{i,p_i} > η_a * d_c then
    role_i ← repeater
    request r_{c_i} with max |A_{c_i}| to: role_{c_i} ← local leader
else if d_{i,p_i} < η_b * d_c then
    role_i ← explorer
    request r_{p_i} to: role_{p_i} ← local leader
end if
```

C. Robot Motion

Motions of each robot are controlled based on a target velocity $\boldsymbol{u}_i \in \mathbb{R}^2$ as a control input, reflected to the robot's linear and angular velocity $\boldsymbol{v}_i \in \mathbb{R}^2$ and, in sequence, to location $\boldsymbol{x}_i \in \mathbb{R}^2$ and heading $\theta_i \in \mathbb{R}$. The maximum linear and angular velocities are capped by *0.03*[unit length/timestep], *0.15*[rad/timestep] through all of the simulations in this paper.

As for repeaters, $\boldsymbol{u}_i$ keeps themselves in a straight line between their parent and child robots. $\boldsymbol{u}_i$ are calculated as Eq.1 if they are directly connected to the local leader and Eq.2; otherwise. k_1, k_2, and k_3 are parameters. The difference between the two equations is that Eq.1 keeps the repeater to the midpoint between the parent robot and the child robot (i.e., local leader). In contrast, Eq.2 keeps the distance to its parent robot. This difference is intended to prevent over-frequent deployment and collection of the repeater robots.

$$\boldsymbol{u}_i = k_1\{(\boldsymbol{x}_{p_i} - \boldsymbol{x}_i) + (\boldsymbol{x}_{c_i} - \boldsymbol{x}_i)\} \tag{1}$$

$$\begin{aligned}\boldsymbol{u}_i = k_2\{(\boldsymbol{x}_{p_i} - \boldsymbol{x}_i) - \eta_a * d_c * \frac{\boldsymbol{x}_{p_i} - \boldsymbol{x}_i}{\|\boldsymbol{x}_{p_i} - \boldsymbol{x}_i\|}\} \\ +k_3\{(\boldsymbol{x}_{p_i} - \boldsymbol{x}_i) + (\boldsymbol{x}_{c_i} - \boldsymbol{x}_i)\}\end{aligned} \tag{2}$$

Control input for the local leader guides the swarm to move on to directed destinations. The input $\boldsymbol{u}_i$ corresponds to the vector from the current position of the local leader to the goal location, multiplied by an appropriate scaling factor (Eq. 3).

$$\boldsymbol{u}_i = k_4(\boldsymbol{x}_{goal} - \boldsymbol{x}_i) \tag{3}$$

Explorers are directed by $\boldsymbol{u}_i$: a summation vector of velocity matching vector $\boldsymbol{u}_i^m$, inter-robot distance maintenance vector $\boldsymbol{u}_i^d$, and input $\boldsymbol{u}_i^L$ to follow the local leader in Eqs.4 - 7. The background idea of the $\boldsymbol{u}_i^m$ and $\boldsymbol{u}_i^d$ is from the flocking algorithm [19]. Each robot r_i acquires the control inputs for adjacent robots: $\boldsymbol{u}_j$ for $r_j \in A_i$ through inter-robot communications. Likewise, the control input for the local leader $\boldsymbol{u}_{LL}$ and its location $\boldsymbol{x}_{LL}$ propagates from the local leader to all subordinate explorers via multi-hop communication. Here, the propagation works under a message expiration model, forwarding the message to all adjacent robots other than its sender until its preset expiration period passes. This model enables all explorers to acquire $\boldsymbol{u}_{LL}$ and $\boldsymbol{x}_{LL}$ without knowledge of network routing by setting the expiration period long enough. d_n is a neutral distance between explorers, and $k_5 - k_8$ are parameters.

$$\boldsymbol{u}_i = \boldsymbol{u}_i^m + \boldsymbol{u}_i^d + \boldsymbol{u}_i^L \tag{4}$$

$$\boldsymbol{u}_i^m = \frac{k_5}{|A_i|} \sum_{r_j \in A_i} \boldsymbol{u}_j \tag{5}$$

$$\boldsymbol{u}_i^d = k_6 \sum_{r_j \in A_i} \{(\boldsymbol{x}_j - \boldsymbol{x}_i) - d_n \frac{\boldsymbol{x}_j - \boldsymbol{x}_i}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|}\} \tag{6}$$

$$\boldsymbol{u}_i^L = k_7 \boldsymbol{u}_{LL} + k_8(\boldsymbol{x}_{LL} - \boldsymbol{x}_i) \tag{7}$$

D. Connection Management

A scheme to restrict the number of connections for each explorer is also employed. Connecting to all other robots in the communication range will impair the responsiveness to other robots' motion due to the velocity matching (Eq.5). Furthermore, it results in a higher burden of inter-robot communication since a number of messages will rush into the robot. The algorithm introduces Acute Angle Test (AAT) [20] to determine which robot to connect or disconnect. In this method, a robot r_i connects to r_j if and only if all angles $\angle \boldsymbol{x}_i \boldsymbol{x}_k \boldsymbol{x}_j$ are acute for robots $R_k = \{r_k \mid k \in 1, 2, ..., N, d_{i,k} \leq d_s\}$. In other words, r_i and r_j are connected whenever no other robot is located inside the circle C_{ij}: the circle with diameter $\overline{\boldsymbol{x}_i \boldsymbol{x}_j}$. AAT is also a way to form a Gabriel graph [21] in a decentralized manner.Furthermore, to reduce the computational cost, AAT is conducted only if the following conditions are satisfied.

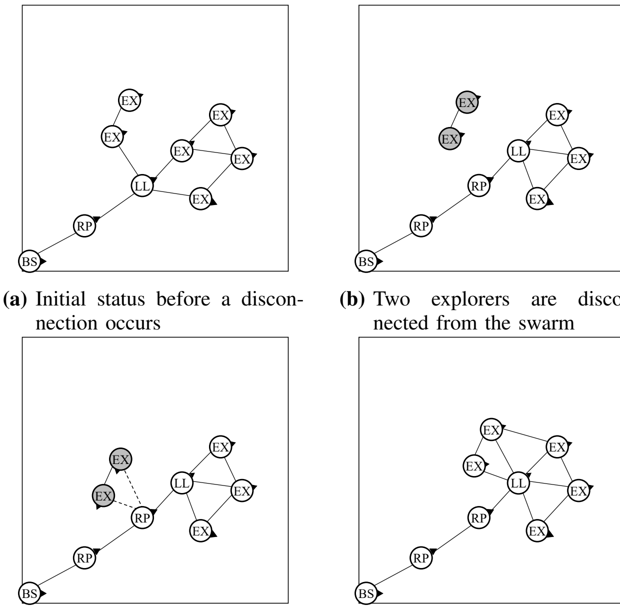
1) r_i has just changed its role to local leader or explorer.
2) r_i needs more connection to keep connectivity robust.
3) The condition of Gabriel graph is broken.

In other cases, the network topology is inherited to the next timestep without new connections or disconnections. The detailed process is shown in Algorithm 2. η_c in line 6 corresponds to the threshold of the number of connections to satisfy condition 2 in the list above. It should be kept in mind that all robots' roles are either local leaders or explorers in Algorithm 2. Additionally, r_i must explicitly request a new (dis)connection to the counterpart robot when r_i will newly (dis)connect to others. This is because the symmetric characteristic [20] of decisions on AAT is no longer reserved in Algorithm 2.By this scheme, although the computational cost of the AAT is theoretically expected as $O(N^2)$, the actual performance is significantly improved, as shown in the preliminary study (Sect.IV-A).

Algorithm 2 Process to refresh network topology among the local leader and explorers

when controlling r_i at timestep: t
$R_k = \{r_k \mid k \in \{1, 2, ..., N\}, d_{i,k} \leq d_s\}$
$R_l = \{r_l \mid l \in \{1, 2, ..., N\}, d_{i,l} \leq d_s\}$
if $role_i(t-1) \neq role_i(t)$ **then**
  $R_{AAT} = R_l$
else if $|A_i| < \eta_c$ **then**
  $R_{AAT} = R_l$
else
  for each $r \in A_i$ **do**
    for each r_l **do**
      if $\boldsymbol{x}_l \ is \ in \ C_{ij}$ **then**
        add $r \ \& \ r_l \ to \ R_{AAT}$
      end if
    end for
  end for
end if
if $R_{AAT} \neq \emptyset$ **then**
  for each $r \in R_{AAT}$ **do**
    if $\angle \boldsymbol{x}_i \boldsymbol{x}_k \boldsymbol{x} < 90°$ for all $r_k \in R_k$ **then**
      add r to R_{pass}
    else
      add r to R_{fail}
    end if
  end for
  $A_i(t+1) = R_{pass} \cup (A_i \cap \overline{R_{fail}})$
else
  $A_i(t+1) = A_i$
end if

This scheme also includes the connection recovery for the disconnected explorers. When an explorer detects disconnections, it moves towards the origin until it finds the other robots. The explorer immediately re-joins the swarm network if they find the other explorers or the local leader. Otherwise, the disconnected explorer continues moving towards the origin, and when it finds a repeater or the base station, it temporarily connects to the found robot. Subsequently, the explorer requests instruction on the child robot's location of the found robot. The explorer moves toward the location until it finds the local leader or the other explorers to re-join. If the child robot of the robot under temporal connection is another repeater, the explorer switches the temporary connection to the newly found repeater and repeats the process above. Fig.2 illustrates this procedure. A typical disconnection occurs when a local leader entrusts the role to another robot and switches its role to a repeater. A connection between explorers and the original local leader (Fig.2a) is disconnected because connections between repeaters and explorers are prohibited (Fig.2b). If the explorers cannot detect the new local leader or other explorers, they remain disconnected and start moving towards the origin (Fig.2b shows the disconnected explorers in gray). When the explorers find repeaters or the base station, they get the location of the child robot of the found robots (Dashed lines in Fig.2c show these temporary communications). The explorers move in the direction until they find the local leader or other explorers to connect with (Fig.2d).



(a) Initial status before a disconnection occurs
(b) Two explorers are disconnected from the swarm
(c) The explorers receive an instruction on the direction to recover the network
(d) The explorers recover the network connection

Fig. 2: A typical case of network detection and recovery

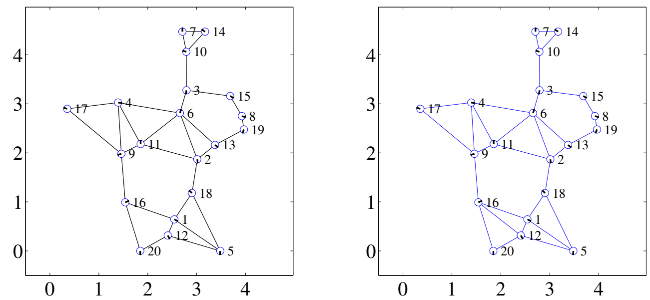
IV. Simulations

A. Preliminary Simulation

As a preliminary experiment, simulations evaluated the calculation costs to form a network topology among explorers implemented based on Algorithm 2.

a) Configurations: The simulations emulated the network formation among the moving N explorers and logged the program execution time as a computational cost. Each explorer moved according to a random walk with constant step length: $l = 1$, random moving direction: $0 \leq \theta < 2\pi$, sensor range: $d_c = d_s = 3$, and $\eta_c = 3$. The simulator configured the field size to keep the density of robots constant: $Field = \sqrt{N} * \sqrt{N}$. The configuration also included the base line condition that conducts the acute angle test to all detectable robots at every timestep. The simulation ran 30 trials per condition, with 1000 timesteps for each trial. As for initial deployment, the simulator distributed the explorers at random locations in the field. There were no other roles: base station, repeaters, and local leaders, in this preliminary simulation.

b) Results: The proposed and the baseline scheme have successfully formed the network with the Gabriel graph topology. Fig.3 shows the network topology by Algorithm 2 (Fig.3a) and by the baseline scheme (Fig.3b), with the identical locations of robots. Fig.3 shows that the two schemes formed almost identical Gabriel graph-like network topology, while the topology by the baseline has formed a slightly larger number of edges. Only the edge between r_{12} and r_{16} failed to form in proposed method since the conditions in Sect III-D are not satisfied. Fig.4 shows the computational cost for network formation. While the networks in Fig.3 resulted in almost identical topologies, the intermittent AAT execution by Algorithm 2 significantly decreased the computational cost. On the other hand, the smaller number of edges by the proposed scheme leads to the slightly lower algebraic connectivity for each N, shown in Fig.5.



(a) Network topology by the Algorithm 2

(b) Network topology by the baseline scheme

Fig. 3: Formed network topology for the identical robot distribution with the swarm size: $N = 20$. An edge between r_{12} and r_{16} is an only edge that failed to be formed by Algorithm 2 compared to the baseline

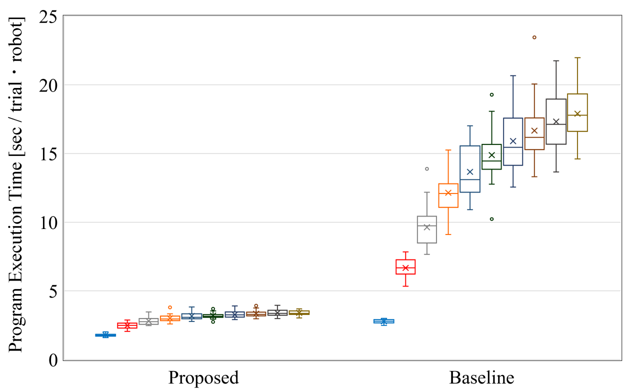


Fig. 4: Mean computational cost for each robot per trial, for the swarm size: $N = 10,\ 20,\ \ldots,\ 100$

B. Performance of the Proposed Algorithm

For the main proposal in this research, scalability and distributedness were evaluated through simulated patrol missions.

a) Configurations: A patrol in a two-dimensional area were emulated for algorithm evaluations. The simulation simplified the patrol missions described in the section III-A to the arrivals at the target locations of the local leader. To evaluate the scalability and distributedness of the proposed algorithms, the processing time and the number of messages were logged as indicators of busyness with different N during the mission.

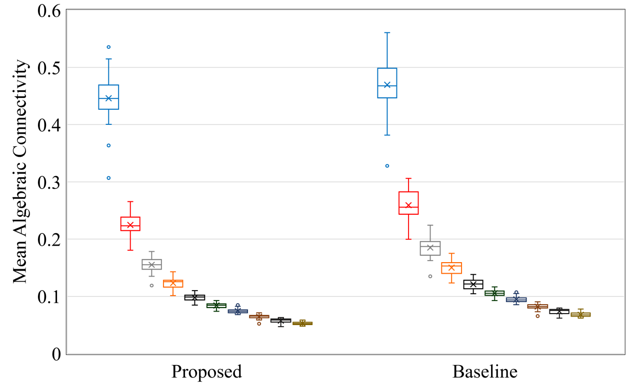
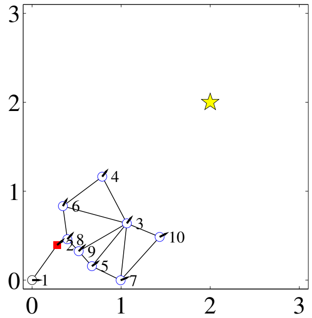


Fig. 5: Mean algebraic connectivity of the network, for the swarm size: $N = 10,\ 20,\ \ldots,\ 100$

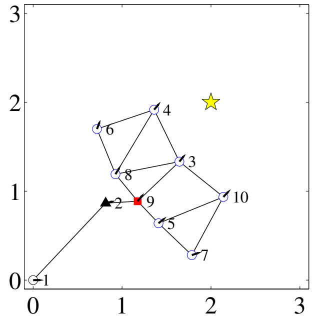
The detailed mission procedures were as follows: the simulator generated the target at a random location, and the swarm detected and reported it to the BS. When the new object was generated, its location is informed from the simulator to the base station and forwarded to the local leader. After the local leader arrived at the location, it reported the arrival to the base station, and then the simulator generated another target.

The simulation ran each trial with the following parameter settings. As for initial deployment and role assignment, the r_1 as a base station was located at the origin: $\boldsymbol{x}_1 = (0,\ 0)$, and the others: $r_2,\ \cdots\ r_N$ were located at random locations in the fan-shaped area: $\|\boldsymbol{x}_i\| \leq d_{init} = 0.6\sqrt{N}$. The robot closest to the BS became the initial local leader, and the others acted as explorers. The initial deployment will be redone if the network connectivity failed to establish without repeaters. The field size was large enough: $\|\boldsymbol{x}_i\| \leq 1000$ to eliminate the boundary effect, while the area of target generation was proportional to the number of swarm sizes: $0 \leq x_{target}, y_{target} \leq 1.8\sqrt{N}$. The simulation lasted 2000 timesteps per trial, and 30 trials were held for each $N = 10,\ 20,\ \ldots,\ 50$.

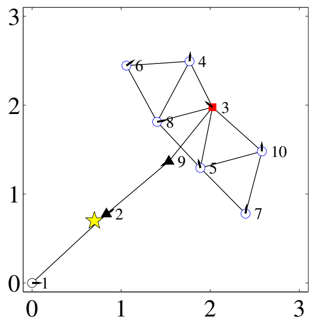
b) Results: Fig.6 describes the typical progress of the simulated patrol mission. The figure shows that the proposed role switching algorithm works successfully to maintain swarm connectivity. At the beginning of the task (Fig6a), the simulator generated a first target (plotted with a star-shaped symbol at $\boldsymbol{x}_{target} = (2, 2)$) and located the base station: r_1 at $\boldsymbol{x}_1 = (0, 0)$. The simulator also distributed the other robots at random locations and assigned the role: local leader to a robot closest to the base station (r_2 in this case). The rest of the robots act as explorers. The local leader and explorers are plotted in a solid red square and vacant blue circles, respectively. As the mission progressed and $d_{1,2}$ got longer than the threshold, r_2 entrusted its role: local leader to r_9 and switched its role to repeater, plotted in a solid black triangle(Fig.6b). After the arrival at the first target, the simulator generated a second target at $\boldsymbol{x}_{target} = (0.7, 0.7)$ (Fig.6c). In this situation, the swarm started to retract the network to reach the target, and r_3, the local leader at the moment, entrusted its role to r_9 and



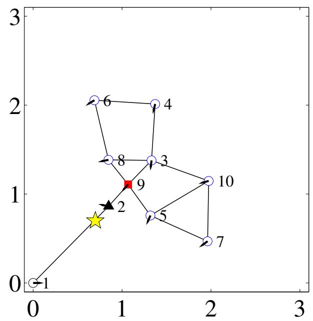
(a) the randomly located positions of the robots at the beginning of the mission



(b) local leader r_2 entrusted its role to r_9 and changed its role to the repeater

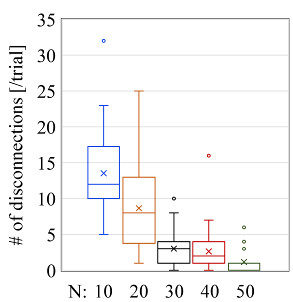


(c) the swarm start to retract the network to arrive the second target

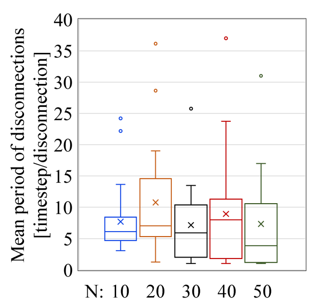


(d) local leader r_3 entrusted its role to r_9 and changed its role to the explorer

Fig. 6: Typical mission progress with $N = 10$



(a) The number of disconnections occurred per trial.



(b) Mean period of disconnections per trial

Fig. 7: Disconnections and recoveries in proposed algorithm

switched its role to explorer (Fig.6d). Fig.7 also shows the performances of the connectivity maintenance from the aspect of the disconnection recovery. Fig.7a shows the number of disconnections occurred during the missions. Fig.7b shows the mean length of timesteps in each trial required to recover the connections for each disconnection.

The simulation results showed that the proposed algorithm has distributed characteristics. As Fig.8 shows, there was no severe inequality in processing time among the robots in swarms with size: $N = 10$, indicating that the algorithm is distributed. This trend was identical when the N increased. On the other hand, from the viewpoint of robot roles, Fig.9 showed slight inequality in computational costs. The cost by the local leader and explorers tends to be larger than that of the base station and repeaters. Additionally, as mentioned in detail in the next paragraph, costs increased as the swarm size N increased.

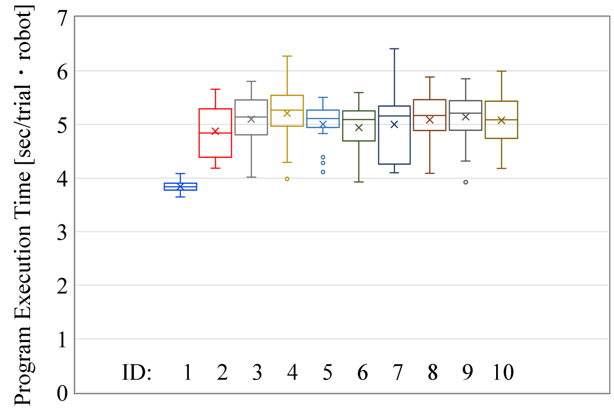


Fig. 8: The processing time per 2000 timestep missions at $N = 10$

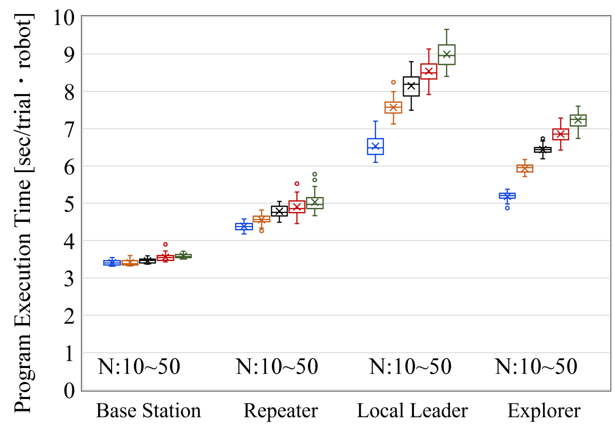


Fig. 9: The processing time per robot per 2000 timestep missions with each role (each box corresponds to $N = 10,\ 20,\ \ldots,\ 50$ robots

Furthermore, as shown in Fig.10 and Fig.11, the simulation showed the scalability of the proposed algorithms. Fig.10 plots the total processing time per robot per trial. While the processing time increases as the swarm size N increases, the processing time increases more gradually than the increase in N. Fig.11 shows the mean number of received messages per robot per timestep as an indicator of the busyness of each robot. The result indicates the same trend as the increase of the measured value was more gradual than the increase of the swarm size.

V. Discussions

The preliminary simulations in Sect.IV-A indicate that the method described in Sect.III-D successfully reduced the computational cost of the acute angle test. In Algorithm 2, each robot maintains its current connections and refreshes them

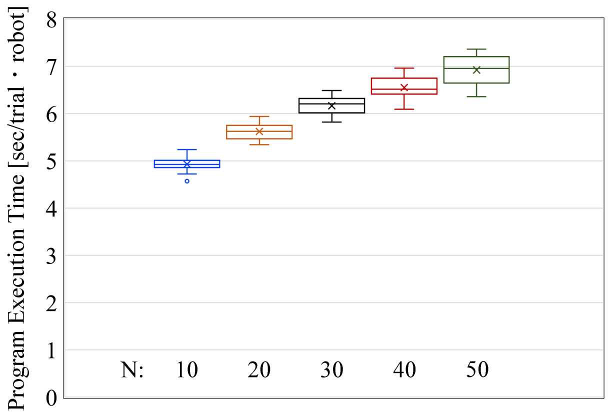


Fig. 10: The processing time per robot per 2000 timestep missions

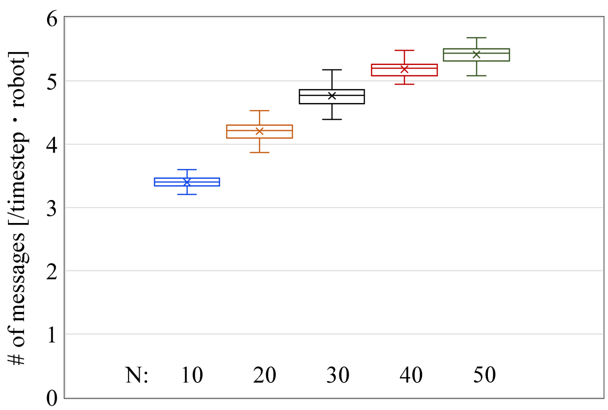


Fig. 11: The mean number of received messages per robot per single timestep

only when required. This scheme reduced the computational cost by approximately 35 - 80 %, with more reduction at larger N(Fig.4). Furthermore, in both conditions, the increase in computational cost was more gradual as N increased ($N \geq 50$) since the field size expanded proportionally to N, and the number of robots detectable by each robot converges to a constant. On the other hand, the robots under the Algorithm 2 would not attempt to gain further connections unless they satisfy several conditions. This feature decreased algebraic connectivity, approximately 5 - 20 % lower than the baseline when the N is identical, with a larger degree of decrease with larger N (Fig.5). In both cases, since the node degree of the Gabriel graph converges to constant [22], the algebraic connectivity decreases as the number of robots increases.

The main simulations described in Sect.IV-B successfully maintained the connectivity of the swarms with low computational cost. Fig.6 shows the deployment and collection of repeaters for the continuous connectivity. The local leader can decide to switch its role to repeater or explorer only by the distance between its neighbors, which is locally available information. As exemplified in Fig.2, temporary network disconnections may occur during the missions. Though the disconnections occur a few dozens times at most (Fig.7a), all disconnections are successfully recovered within a few to dozens of timesteps (Fig.7b). The networks of smaller swarms were disconnected more frequently. This was because the smaller number of robots leads to the smaller number of connections among the explorers, resulting in disconnections illustrated in Fig.2. The summation of the disconnection period is approximately 100 timesteps per trial, at most (mean value at $N = 10$), corresponds to 5% of the mission time. The acceptability of this length depends on the mission; however, many types of missions may be assumed that they can wait for the connection recovery. Increasing the degree of aggregation of the explorers (i.e., decrease d_n and increase k_6 in Eq.6) can relieve these risk of disconnections. The total processing time was 8 seconds at most per 2000 timesteps mission (i.e., ≤ 0.01 seconds per timestep). This low-cost algorithm maintains the feasibility of being implemented for small mobile robots with limited computational capabilities.

Fig.8 and Fig.9 show the distributed property of the proposed algorithm. There are no large inequalities in the processing times of the individual robots. This characteristic enables a robot to replace another robot when it malfunctions, and keep the system robust and prevent the existence of a single point of failure. On the other hand, robots' roles caused inequality in computational costs, and the costs for the local leader and explorers were higher than those of the base station and repeaters. Since the local leader and explorers detect more robots than robots with other roles, the computational costs for sensing and network formation tend to be larger. While the acceptability of this inequality depends on the robots' specification and mission configuration in principle, it is natural to assume that individual robots have enough computational capability since sensing and network formation are fundamental processes for mission performance. Furthermore, since each robot switches its role as the mission progresses, this inequality will be damped, as Fig.8 indicates. It is also worth noting that the base station's processing time is almost constant. Since the robots can decide their behavior distributedly, the main task of the base station is independent of the swarm size: directing the local leader to the next target location.

Fig.10 and Fig.11 show that the proposed algorithm has a certain degree of scalability in terms of computational cost and the busyness of each robot. As Fig.10 shows, while the computational cost increases as the swarm size N increases, the increase is more gradual than that of N. The cost increased by approximately 50%, while the swarm size increased five times larger. The proposed algorithm avoids the worst-case scenario where the cost increases exponentially with N, as well as the algorithm works at a low computational cost regardless of the swarm size, $\leq$ 0.01 second per timestep at most. Similarly, this trend was also the case with the number of received messages by each robot (Fig.11). These results prove the system's scalability, or in other words, the feasibility of adding more robots to the swarm under operations.

Future works may include further evaluations of the proposed algorithm in more practical situations or from different aspects. For instance, the current study does not regard robot failures. It is possible to assume that the adjacents can move and replace the failed robot's position and role, respectively, according to the latest message from the failed robot. However, assessing quantitative factors, such as the time required for those procedures, is essential for real-world deployment. Other future works with more fidelity may include detailed behaviors such as aggregation, pursuit, and multiple subgroups forming by robots encountering realistic tasks, either in 2-D or 3-D environments. These may also identify the required specifics of the base station to control robotic swarms with transparency. Existing studies, such as [23], have similar approaches by exemplifying requirements for base stations. Further research can clarify how the base station and accompanying human operators can interact with and support autonomous swarm robotic systems for real-world applications.

VI. Conclusions

This paper has proposed a decentralized algorithm to maintain the continuous connectivity of robot swarms for effective supervision of the systems. The algorithm is composed of role switching and network formation, depending on locally available information. The simulation studies showed that the proposed algorithm successfully maintained the connectivity of a moving swarm in distributed, scalable, and low computationally costed manner.

References

[1] E. Sahin, "Swarm robotics: From sources of inspiration to domains of application," *Swarm Robotics*, pp. 10–20, 2005.

[2] M. Schranz, M. Umlauft, M. Sende, and W. Elmenreich, "Swarm Robotic Behaviors and Current Applications," *Frontiers in Robotics and AI*, vol. 7, no. 36, 2020.

[3] K. Kobayashi, T. Higuchi, and S. Ueno, "Connectivity maintenance for robotic swarms by distributed role allocation algorithm," in *Proceedings of the Joint Symposium of AROB-ISBC-SWARM2022*. Oita, Japan: International Society of Artificial Life and Robotics, 2022, p. 897–902.

[4] F. Amigoni, J. Banfi, and N. Basilico, "Multirobot Exploration of Communication-Restricted Environments: A Survey," *IEEE Intelligent Systems*, vol. 32, no. 6, pp. 48–57, 2017.

[5] J. Banfi, A. Quattrini Li, I. Rekleitis, F. Amigoni, and N. Basilico, "Strategies for coordinated multirobot exploration with recurrent connectivity constraints," *Autonomous Robots*, vol. 42, no. 4, pp. 875–894, 2018.

[6] G. A. Hollinger and S. Singh, "Multirobot Coordination With Periodic Connectivity: Theory and Experiments," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 967–973, 2012.

[7] J. De Hoog, S. Cameron, and A. Visser, "Autonomous multi-robot exploration in communication-limited environments," in *Proceedings of the Conference on Towards Autonomous Robotic Systems*. Citeseer, 2010, pp. 68–75.

[8] E. Stump, N. Michael, V. Kumar, and V. Isler, "Visibility-based deployment of robot formations for communication maintenance," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 4498–4505.

[9] Y. Pei, M. W. Mutka, and N. Xi, "Connectivity and bandwidth-aware real-time exploration in mobile robot networks," *Wireless Communications and Mobile Computing*, vol. 13, no. 9, pp. 847–863, jun 2013.

[10] J. Banfi, A. Q. Li, N. Basilico, and F. Amigoni, "Communication-constrained multirobot exploration: Short taxonomy and comparative results," in *Proceedings of the IROS workshop on on-line decision-making in multi-robot coordination (DEMUR2015)*, 2015, pp. 1–8.

[11] R. C. Arkin and J. Diaz, "Line-of-sight constrained exploration for reactive multiagent robotic teams," in *7th International Workshop on Advanced Motion Control. Proceedings (Cat. No.02TH8623)*, 2002, pp. 455–461.

[12] P. Mukhija, K. M. Krishna, and V. Krishna, "A two phase recursive tree propagation based multi-robotic exploration framework with fixed base station constraint," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 4806–4811.

[13] T. Nestmeyer, P. Robuffo Giordano, H. H. Bülthoff, and A. Franchi, "Decentralized simultaneous multi-target exploration using a connected network of multiple robots," *Autonomous Robots*, vol. 41, no. 4, pp. 989–1011, 2017.

[14] P. D. Hung, T. Q. Vinh, and T. D. Ngo, "Hierarchical Distributed Control for Global Network Integrity Preservation in Multirobot Systems," *IEEE Transactions on Cybernetics*, vol. 50, no. 3, pp. 1278–1291, 2020.

[15] P. Yang, R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S. Srinivasa, and R. Sukthankar, "Decentralized estimation and control of graph connectivity for mobile sensor networks," *Automatica*, vol. 46, no. 2, pp. 390–396, 2010.

[16] N. Goddemeier, K. Daniel, and C. Wietfeld, "Role-Based Connectivity Management with Realistic Air-to-Ground Channels for Cooperative UAVs," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 5, pp. 951–963, 2012.

[17] L. Sabattini, N. Chopra, and C. Secchi, "Decentralized connectivity maintenance for cooperative control of mobile robotic systems," *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1411–1423, sep 2013.

[18] H. Sugiyama, T. Tsujioka, and M. Murata, "Real-time exploration of a multi-robot rescue system in disaster areas," *Advanced Robotics*, vol. 27, no. 17, pp. 1313–1323, dec 2013.

[19] C. W. Reynolds, "Flocks, Herds and Schools: A Distributed Behavioral Model," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: Association for Computing Machinery, 1987, pp. 25–34.

[20] B. Shucker and J. K. Bennett, "Virtual Spring Mesh Algorithms for Control of Distributed Robotic Macrosensors," *University of Colorado at Bulder, Technical Report CU-CS-996-05*, no. 930, 2005.

[21] K. R. Gabriel and R. R. Sokal, "A New Statistical Approach to Geographic Variation Analysis," *Systematic Biology*, vol. 18, no. 3, pp. 259–278, sep 1969.

[22] D. W. Matula and R. R. Sokal, "Properties of Gabriel Graphs Relevant to Geographic Variation Research and the Clustering of Points in the Plane," *Geographical Analysis*, vol. 12, no. 3, pp. 205–222, jul 1980.

[23] N. Dousse, G. Heitz, and D. Floreano, "Extension of a ground control interface for swarms of Small Drones," *Artificial Life and Robotics*, vol. 21, no. 3, pp. 308–316, 2016.