This paper was presented at AROB-SWARM-ISBC2022, Online, Jan 25-27, 2022

Connectivity Maintenance for Robotic Swarms by Distributed Role Allocation Algorithm

Kazuho Kobayashi[1 †], Takehiro Higuchi[2], and Seiya Ueno[3]

[1]Graduate School of Engineering Science, Yokohama National University, Kanagawa, Japan
(Tel : +81-45-339-4240; E-mail: kobayashi-kazuho-dj@ynu.jp)
[2]Faculty of Environment and Information Sciences, Yokohama National University, Kanagawa, Japan
(Tel : +81-45-339-3697; E-mail: higuch@ynu.ac.jp)
[3]Faculty of Environment and Information Sciences, Yokohama National University, Kanagawa, Japan
(Tel : +81-45-339-4242; E-mail: ueno-seiya-wk@ynu.ac.jp)

Abstract: Swarm robotics requires a practical scheme to maintain supervision by human operators or managers, especially in complicated or life-threatening situations. For this purpose, this paper proposes an algorithm to maintain connectivity between robot swarm and fixed base station during missions. The main idea of the algorithm is maintaining connectivity by role allocation and switching among robots without centralized control by the base station. Our simulation studies have shown no significant inequality of computational cost among robots over the emulated patrol missions. Furthermore, as the total number of robots in the swarm increase, computational cost per robot does not increase significantly. These results have shown the distributed nature and scalability of the proposed algorithms.

Keywords: Swarm robotics, Role allocation, Decentralized algorithm, Connectivity maintenance

1. INTRODUCTION

Swarm robotics is a technology to control multiple robots by their local interaction among robots and surrounding environments, and it has several advantages such as robustness, scalability, and flexibility [1].

Despite these advantages, there are few cases that robot swarms are deployed for real-world application due to several practical concerns [2]. These concerns include transparency and predictability from humans, or in other words, concerns on how the relationship between humans and robots should be [3]. Though these concerns are also applicable to robotic systems other than swarms, in swarm robotics, these concerns get more severe because of the characteristic that system's behaviors emerge from robots' local interactions, which are usually invisible from out of the system [2].

To relieve these concerns, a practical scheme is required to ensure supervision from humans by maintaining connectivity among swarms, including a base station (BS). In this context, BS corresponds to human operators or interface to their authority to make high-level decisions. Especially for tasks that should be solved collectively, it is essential to consolidate swarm members to a frontier of the network while maintaining connectivity to the BS so that they can immediately be engaged to those tasks according to authorization from the BS.

Regarding swarm connectivity, Amigoni et al. [4] reviewed researches on multi-robot exploration with various kinds of restrictions on communication. In the review, the restrictions are categorized as either *event-based* or *continuous*.

Event-based connectivity is a concept that requires recovering the connection triggered by a particular event. According to Banfi et al. [5], this is further categorized as *periodic* and *recurrent*. Periodic connectivity requires connections every 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 deployment of robots to their assigned area [5, 8-9].

As the main point of interest in our research, continuous connectivity, on the other hand, requires swarm members to connect with each other during whole missions. Arkin and Diaz [10] studied exploration performances by three algorithms varied by degree of *a priori* knowledge on mission spaces. Mukhija et al. [11-12] proposed the exploration algorithm to maintain connections between robots and fixed BS by propagating a tree network. Nestmeyer et al. [13] showed an exploration algorithm to arrive at multiple locations assigned to each robot by switching priority. A Robot with higher priority can move towards their assignment while others work to maintain connectivity.

Although these *event-based* connectivities may provide efficient strategies, still in cases that real-time access to information such as video streaming is necessary, continuous connectivity is needed [14]. As for continuous connectivity, existing researches frequently depend on coordination by a central agent or the availability of global knowledge, combined with exploration strategies.

Furthermore, existing cases are not designed to gather members at the frontier of the network so that they can be engaged in tasks immediately after receiving BS's permissions. This characteristic is essential for swarms involved in missions to be solved cooperatively, such as cooperative transport [15].

In this paper, from these backgrounds, we propose a decentralized algorithm for continuous connectivity of swarms including base stations. The algorithm employs role allocation and switching to maintain the connectivity while all related processes are distributed. The following sections describe the proposed algorithm (section 2), show the result of simulations (section3) and conclusions (section 4).

† Kazuho Kobayashi is the presenter of this paper.

2. METHODOLOGIES

2.1 Problem settings and assumptions

Consider a patrol mission by a robot swarm R consisting of N homogenous mobile robots. Each robot is assigned their ID: $i \in \{1, \dots, N\}$ And is written as r_i . Accordingly, all variables related to r_i 's state are shown with suffix i, and if not explicitly indicated, they are states at timestep t. Other assumptions for the robots are:

(1) The robots can sense its accurate location $\boldsymbol{x}_i \in \mathbb{R}^2$

(2) The robots can sense the others' ID, role, and relative position by their omnidirectional sensors in their range d_s

(3) The robots can connect and communicate to the others in their range d_c.

(4) Robots with connections can exchange information on their state by multi-hop communications in a single timestep per hop.

(1) is assumed to be achieved by some technique such as SLAM or dead reckoning. Similarly, (2) is supposed to be performed by devices such as vision sensors or RFID tags. In this study, it is assumed that $d_s \leq d_c$.

2.2 Role allocation for continuous connectivity

The robots move according to their assigned roles: base station, repeater, local leader, explorer, indicated as $role_i$. The base station is assigned to r_1 and fixed to the origin point (i.e., $\boldsymbol{x}_1 = (0, 0)$) to make high-level decisions such as where the swarm should patrol next. Repeaters maintain the connection between the base station and a local leader. The local leader represents a sub-group of swarms consisting of explorers and communicates with the base station if needed. A typical state of the swarm is shown in Fig. 1. A robot directly connected to r_i at a side nearer to the base station is called a *parent*: r_{p_i}, and at a side nearer to the explorers is called a *child*: r_{c_i}, respectively. The set of robots directly connected to r_i is written as R_{adj_i}.

To maintain a connection from a base station, robots occupy one of these four roles and switch or request others to switch roles based on local information. The basic idea is to have the local leader switch to repeater or explorer and request an adjacent robot to become a local leader instead, based on a distance between the local leader and its parent side adjacent robot. This process is shown in pseudocode as Algorithm 1, while $d_{i,j} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$, and η_a , η_b $(0 < \eta_a$, $\eta_b < 1)$ are parameters to ensure margins of connection maintenance. The requests to change roles (lines 4, 7 in Algorithm 1) are made by the inter-robot communications.

As Fig.1 indicates, there are several additional assumptions for connections between robots:

- The base station connects one repeater or local leader.
- Repeaters connect two robots, parent side base station or repeater, and child side repeater or local leader.
- The local leader connects to one parent side base station or repeater, and explorers in range d_c.
- Explorers connect to other explorers based on the scheme explained in section 2.4.

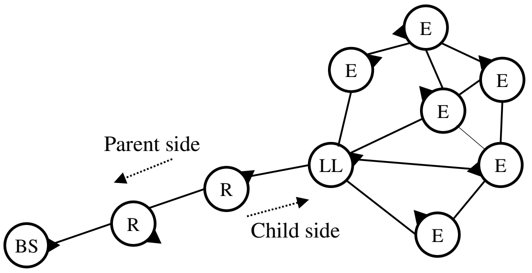


Fig. 1. A typical form of a robot swarm with N = 10. (BS: Base Station, R: Repeater, LL: Local Leader, E: Explorer) Solid lines indicate network connections, and black triangles indicate headings of robots.

Algorithm 1: Process to switch roles for connectivity

```
1 when controlling a local leader: $r_i$
2   if $d_{i,p\_i} > \eta_a * d_c$ then
3     $role_i \leftarrow Repeater$
4     request nearest $r_{c\_i}$ to: $role_{c\_i} \leftarrow local\ leader$
5   else if $d_{i,c\_i} < \eta_b * d_c$ then
6     $role_i \leftarrow explorer$
7     request $r_{p\_i}$ to: $role_{p\_i} \leftarrow local\ leader$
```

2.3 Robot motion by virtual forces

Motions of each robot are controlled based on a virtual force vector $\boldsymbol{f}_i \in \mathbb{R}^2$, reflected immediately to the robot's linear and angular velocity $\boldsymbol{u}_i \in \mathbb{R}^2$ and in sequence, to location $\boldsymbol{x}_i$ and heading $\theta_i \in \mathbb{R}$.

As for repeaters, $\boldsymbol{f}_i$ keep them in a straight line between their parent and child robots. $\boldsymbol{f}_i$ are calculated as Eq. (1) if they directly connect to the local leader and Eq. (2) otherwise, while $\alpha, \beta,$ and γ are parameters.

$$\boldsymbol{f}_i = \alpha\{(\boldsymbol{x}_{p\_i} - \boldsymbol{x}_i) + (\boldsymbol{x}_{c\_i} - \boldsymbol{x}_i)\} \tag{1}$$

$$\boldsymbol{f}_i = \beta\left\{(\boldsymbol{x}_{p\_i} - \boldsymbol{x}_i) - \eta_a * d_c * \frac{\boldsymbol{x}_{p\_i} - \boldsymbol{x}_i}{\|\boldsymbol{x}_{p\_i} - \boldsymbol{x}_i\|}\right\} + \gamma\{(\boldsymbol{x}_{p\_i} - \boldsymbol{x}_i) + (\boldsymbol{x}_{c\_i} - \boldsymbol{x}_i)\} \tag{2}$$

$\boldsymbol{f}_i$ for the local leader guides the swarm to move on to directed destinations. In this study, the base station directs the local leader to the destinations via repeaters.

Explorers are directed by $\boldsymbol{f}_i$ which is a resultant vector of force matching $\boldsymbol{f}_i^m$ and inter-robot distance maintenance $\boldsymbol{f}_i^d$ in Eqs. (3) to (5), inspired by the flocking algorithm [16]. The virtual forces loaded on adjacent robots are acquired through inter-robot communications. Here, $|R_{adj\_i}|$ is the number of robots belongs to R_{adj_i}, d_n is a neutral distance between explorers, and δ, ε are parameters.

2.4 Connectivity management

While the methodologies in sections 2.2 and 2.3 maintain continuous connectivity, a scheme to restrict the number of connections for each explorer is employed. Connections of

$$\boldsymbol{f}_i = \boldsymbol{f}_i^m + \boldsymbol{f}_i^d \tag{3}$$

$$\boldsymbol{f}_i^m = \frac{\delta}{\left|R_{adj\_i}\right|} \sum_{r_j \in R_{adj\_i}} \boldsymbol{f}_j \tag{4}$$

$$\boldsymbol{f}_i^d = \varepsilon \sum_{r_j \in R_{adj\_i}} \left\{ \left(\boldsymbol{x}_j - \boldsymbol{x}_i\right) - d_n \frac{\left(\boldsymbol{x}_j - \boldsymbol{x}_i\right)}{\left\|\boldsymbol{x}_j - \boldsymbol{x}_i\right\|} \right\} \tag{5}$$

explorers to all other robots in the range d_c will impair the responsiveness to other robots' motion due to the force matching (equation (4)) and the higher burden of inter-robot communication. To relieve this disadvantage, the algorithm introduces Acute Angle Test (AAT) by Shucker et al. [17] to determine which robot to connect. By this method, a robot r_i connects to r_j if and only if for robots $R_k = \{r_k \mid k \in \{1,\dots,N\},\ d_{i,k} \le d_c\}$, all angles $\angle \boldsymbol{x}_i \boldsymbol{x}_k \boldsymbol{x}_j$ are acute. In other words, r_i and r_j is connected whenever no other robot is located inside of the circle C_{ij}: the circle with diameter $\overline{\boldsymbol{x}_i \boldsymbol{x}_j}$. AAT is also a way to get a Gabriel graph [18] in a decentralized manner.

Furthermore, to reduce the computational load, AAT is conducted only if several conditions are satisfied, while network topology is inherited to the next timestep otherwise. The detailed process is shown in Algorithm 2, but briefly, the conditions are as follows:

- r_i has just changed its role to explorer
- r_i needs more connection to maintain connectivity
- The condition of Gabriel graph is broken

It should be kept in mind that all robots related here occupy the role of explorer. A local leader is not in the scope of this scheme since the larger number of connections between the local leader and explorers enhances the obedience of explores to the local leader's moving directions. Furthermore, in refreshing the network topology, it also should be kept in mind that in refreshing network topology, r_i must explicitly request a new (dis)connection to the counterpart robot because the symmetric characteristic of decisions on connections [17] is no longer reserved.

By this scheme, although the calculation time of the AAT is theoretically expected as $O(N^2)$, the actual performance is significantly improved as the process was approximately ten times faster than the usual AAT at preliminary experiments.

3. SIMULATIONS

3.1 Configurations

In this research, simulated patrol missions in a two-dimensional area were conducted for algorithm evaluations. The missions were simplified as arrivals at target locations of the local leader. To evaluate the scalability and distributedness of the proposed algorithms, the processing time and the number of messages received were measured as indicators of busyness with different *N* during the mission.

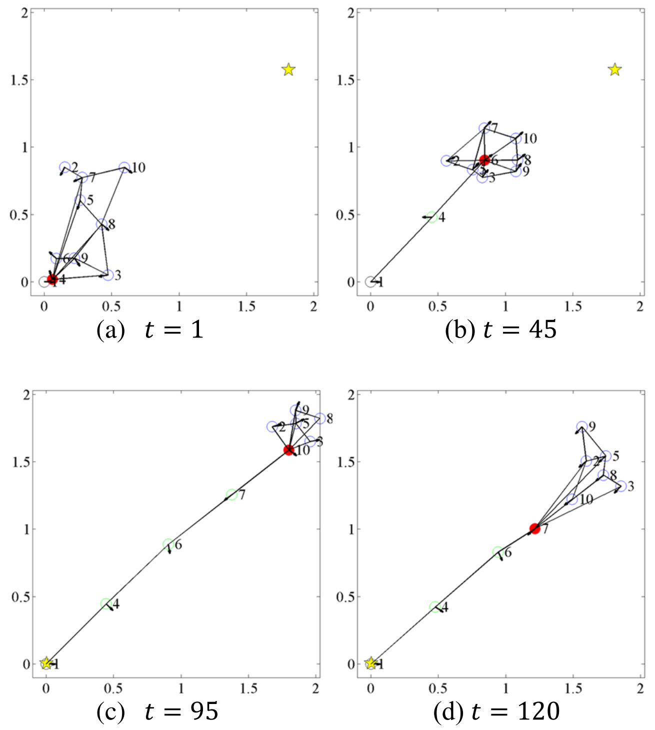
A target location to patrol is informed from the simulator to the base station and forwarded to the local leader to arrive at the target. After the local leader arrives at the location, it reports the arrival to the base station, and then another target location is generated by the simulator. To confine the effects of boundaries, the size of the field was set relatively large: $\|\boldsymbol{x}_i\| \le 1000$, while the area of target generation is set as proportional to the number of swarm size: $0 \le x_{target}, y_{target} \le 1.6\sqrt{N}$. The simulation lasts 1000 timesteps per trial, and 10 trials were run for each $N = 10, 20, \dots, 100$.

Algorithm 2: Process to refresh network topology

```
when controlling explorer: $r_i$ at timestep: $t$
    $R_k, R_l = \{r_k, r_l \mid k, l \in \{1,\dots,N\}, d_{i,k}, d_{i,l} \le d_s\}$
    if $role_i(t-1) \ne explorer$ then
        $R_{AAT} = R_l$
    elseif $|R_{adj\_i}| < 2$ then
        $R_{AAT} = R_l$
    else
        for each $r_{adj\_i} \in R_{adj\_i}$
            for each $r_l \in R_l$
                if $\boldsymbol{x}_l \in C_{ij}$
                    add $r_{adj\_i}$ & $r_l$ to $R_{AAT}$
    if $R_{AAT} \ne \emptyset$
        for each $r_{AAT} \in R_{AAT}$
            run AAT for $R_k$ to determine whether $r_{AAT}$
            should be connected to $r_i$
            if $r_{AAT}$ passed AAT then
                add $r_{AAT}$ to $R_{Pass}$
            else
                add $r_{AAT}$ to $R_{Fail}$
        $R_{adj\_i}(t+1) = R_{Pass} \cup (R_{adj\_i} \cap \overline{R_{Fail}})$
    else
        $R_{adj\_i}(t+1) = R_{adj\_i}$
```

3.2 Results

Fig.2 describes the typical progress of the simulated patrol mission, showing that the proposed role allocation and switching works appropriately according to Algorithm 1. At the beginning of the task ($t = 1$, Fig.2(a)), the simulator generated a first target (plotted with star shape symbol) and located r_1 at (0, 0) and assigned the role base station. The other robots have been distributed at random locations, and the one nearest to the base station (r_4 in this case) has been assigned the role local leader and the others have been assigned the role explorer. The local leader and explorers are plotted in red solid circle and blue vacant circle respectively. As the mission progresses and $d_{1,4}$ got larger than the threshold, the role local leader was entrusted from r_4 to r_6 and in turn, r_4 switched its role to repeater ($t = 45$, Fig.2(b)). When the swarm start to move to retract the network since the second target was generated near to the base station ($t = 95$, Fig.2(c)), the role local leader was entrusted from r_{10} to r_7 and in turn, r_{10} switched its role to explorer when $d_{7,10}$ got smaller than the threshold ($t = 120$, Fig.2(d)). During the missions, the connections between the base station and the other part of the swarm has been maintained.



(a) $t = 1$ (b) $t = 45$

(c) $t = 95$ (d) $t = 120$

Fig.2. Enlarged figures on typical mission progress with $N = 10$. (a): the random located positions of the robots at the beginning of the mission, (b) local leader: r_4 entrust its role to r_6 (c): the swarm start to move to retract the network, (d): local leader: r_{10} entrust its role to r_7.

As Fig.3 shows, there was no severe inequality in processing time during 1000 timesteps among the robots at $N = 10$, indicating that the algorithm is distributed. This trend was identical when the *N* increased.

On the other hand, robots' roles showed inequality in computational costs, and processing time at the local leader was the longest among the four roles (Fig.4). This is because the number of connected robots at the local leader is larger than robots with the other roles since the local leader connects to all explorers in range d_c. The proposed algorithm can mitigate this inequality since the local leader would entrust its role to adjacents and switch to other roles as the mission progresses. As for the results in Fig.3 and 4, it is worth keeping in mind that comparing only for $r_2, r_3, \ldots, r_N$ (i.e., robots with role other than the base station) may be appropriate since the base station is a relatively special role, with no role switching, virtual force, or connection management. Still, results on the base station in Fig.4 show the algorism's scalability as the base station's computational cost does not increase when *N* increases, since the proposed algorithm does not require the base station to manage the other robots' affairs, such as local motions and communications.

Fig.5 and Fig.6 show each robot's computational cost and busyness as the swarm size *N* increases, indicating that the proposed algorithm has a certain degree of scalability. Fig.5 plots the total processing time per robot during a trial. While the time increases as *N*, the processing time increases only approximately two to three times larger though the *N*

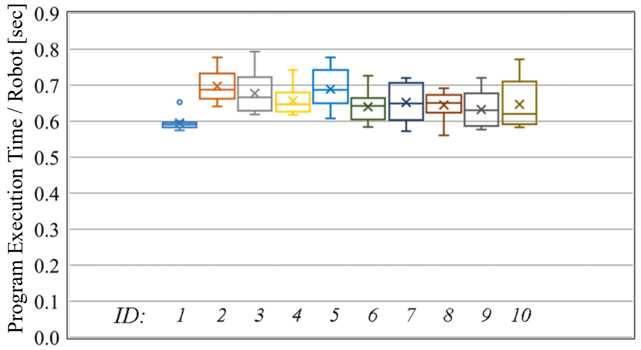


Fig. 3. The processing time during missions ($N = 10$).

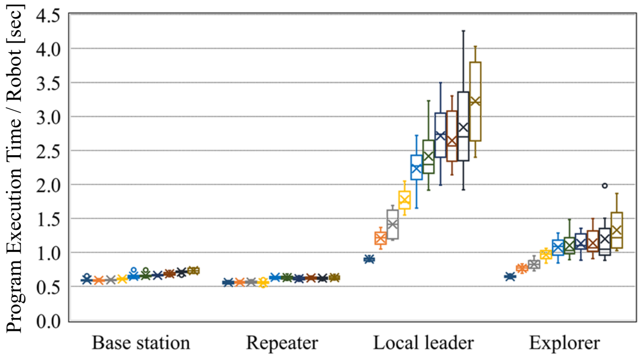


Fig. 4. The processing time per robot with four roles. (each box corresponds to *N* = 10, 20, …, 100 robots)

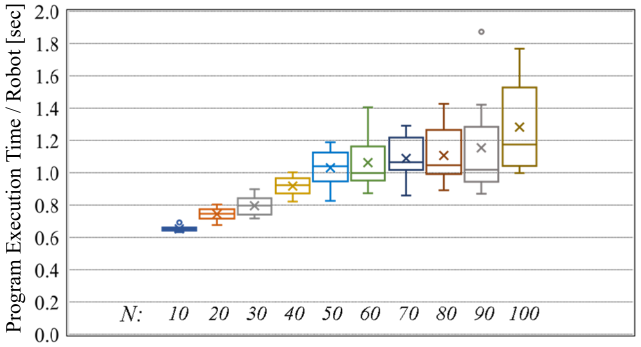


Fig. 5. The processing time per robot during missions.

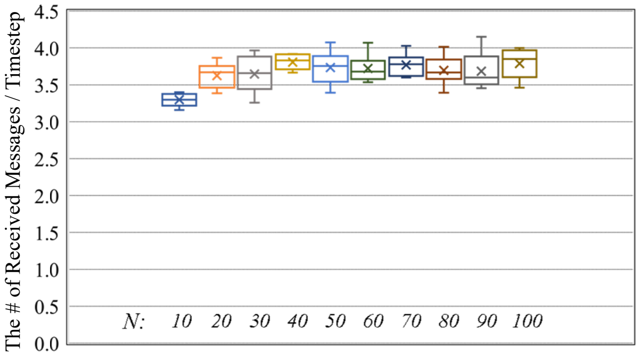


Fig. 6. The number of received messages per robot per single timestep.

increases ten times. The result indicates that adding robots to the swarm under operation is feasible in the aspect of computational cost on each robot. The increase of processing time results from the larger processing time on sensing other robots and connection management because, for each robot, the number of robots in the sensing and communication ranges tends to increase. As for the local leader, the larger number of explorers connected also contributes to the increase. Furthermore, the samples may include outliers, such as $N = 90$ in Fig.5, since the processing time, especially in Algorithm 2, is affected by the placement of robots. Fig.6 shows the number of received messages per robot per timestep as an indicator of the busyness of each robot. The messages, other than a message on the target location, are on the internal state (e.g., virtual forces loaded on itself) of adjacent robots broadcasted by them, therefore the number of the messages per robot strongly corresponds to the number of connections at the robot. The mean number of messages per robot is constant since the number of connections at each explorer is restricted by the connection management scheme (section 2.4). This result shows that the busyness of each robot is constant for the size of the swarm N, indicating the scalability of the algorithm.

4. CONCLUSIONS

This paper has proposed a decentralized algorithm to maintain the continuous connectivity of robot swarms. The algorithm is designed to maintain the connectivity by role allocation and switching (section 2.2) and flocking motion (section 2.3), while both methods can work in a decentralized manner. For further scalability, connectivity management (section 2.4) is introduced to restrict the number of connections between robots.

The main contribution of this research is the maintenance of connectivity including a fixed base station, by role allocation and switching (section 2.2, Fig.2). This enables robot swarms to arrange member robots for immediate, collective task solutions under continuous supervision by the base station. Furthermore, the algorithm's scope is only for connectivity maintenance and is independent of existing exploration or patrol strategies. The local leader and subordinate explorers can be considered a single unit and fit those strategies to the proposed algorithm. The multiple robot exploration/patrol strategies may also be applied when multiple local leaders organize the sub-swarms.

The simulations showed that the algorithm is distributed (Fig.3) and scalable (Fig.5, Fig.6). While a robot assigned as a local leader tends to be relatively busy, its role switches to the others as a mission progresses. The mechanism relieves inequality at total processing time among robots during the whole mission.

On the other hand, it should be noted that there are several disadvantages to requiring continuous connectivity. For example, the requirement sometimes results in insufficient coverage of the patrol area because the robots have to restrict their exploration not to go far from the base station. To offset this disadvantage, operators may have to deploy more robots which means higher cost and likelihood of failure [19].

For future work, we plan to evaluate the proposed algorithm in more practical situations to assess the utility of continuous connectivity.

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, et al., Swarm Robotic Behaviors and Current Applications, Frontiers in Robotics and AI, vol. 7, no. 36, 2020.

[3] D. Carrillo-Zapata, E. Milner, J. Hird, et al., Mutual Shaping in Swarm Robotics: User Studies in Fire and Rescue, Storage Organization, and Bridge Inspection, Frontiers in Robotics and AI, vol. 7, no. 53, 2020.

[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, et al., 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, Proceedings of the Conference on Towards Autonomous Robotic Systems, pp. 68–75, 2010.

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

[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, 2013.

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

[11] P. Mukhija, R. Sawhney, and K. M. Krishna, Multi Robotic Exploration with Communication Requirement to a Fixed Base Station, Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1, pp. 1515–1516, 2010.

[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, 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4806–4811,

2010.

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

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

[15] E. Tuci, M. H. M. Alkilabi, and O. Akanyeti, Cooperative object transport in multi-robot systems: A review of the state-of-the-art, Frontiers in Robotics and AI, vol. 5, no. 59, 2018.

[16] C. W. Reynolds, Flocks, Herds and Schools: A Distributed Behavioral Model, Proceedings of the 14th annual conference on Computer graphics and interactive techniques, pp. 25–34, 1987.

[17] 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.

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

[19] M. N. Rooker and A. Birk, Multi-robot exploration under the constraints of wireless networking, Control Engineering Practice, vol. 15, no. 4, pp. 435–445, 2007.