Distributed Multi-Robot Exploration Approach With Connectivity Maintenance

Hazem Chaabi, Nathalie Mitton

Distributed Multi-Robot Exploration Approach With Connectivity Maintenance

Hazem Chaabi
Inria, France
hazem.chaabi@inria.fr

Nathalie Mitton
Inria, France
nathalie.mitton@inria.fr

***Abstract*—In this paper, we propose a novel distributed multi-robot exploration algorithm capable of efficiently exploring unknown environments while ensuring uninterrupted communication with each other and with a base station, including multi-hop scenarios when direct communication is not feasible. The system is applicable in scenarios of emergencies like search and rescue operations where time and communication reliability is of utmost importance. Our approach introduces two distinct roles, explorers and supporters, to enhance efficiency and connectivity based on live assessments of network connectivity and exploration needs. Explorers perform frontier-based exploration keeping primarily focus on maximization of information gain through unexplored parts. Supporters adopt a flocking-inspired approach for their positioning, acting as relay points to ensure strong communication links. Decision making is distributed, with local information sharing so that each robot can autonomously come to a decision in the absence of centralized control. Results of our simulation indicate that the algorithm has reduced time required for exploration by 80% and time spent on redundant exploration by 74% as compared to Random walk exploration.**

***Index Terms*—multi-robot systems, distributed exploration, connectivity maintenance, frontier-based exploration, relative neighborhood graph**

I. Introduction

Multi-robot systems are vital for search and rescue, disaster response, and environmental monitoring. In emergency scenarios, efficient exploration and reliable communication are paramount to locate survivors and assess hazardous conditions quickly. Efficient exploration requires not only the rapid coverage of unexplored areas but also the maintenance of reliable communication links among robots and with a base station. Achieving these objectives simultaneously is challenging because of the inherent trade-offs between exploration efficiency and network connectivity. Most existing methods either focus on maximizing exploration speed at the expense of connectivity or maintaining connectivity but suffering from a reduced exploration efficiency, particularly in environments with dynamic conditions [1].

Traditional exploration algorithms often rely on centralized control mechanisms or assume global knowledge of the environment, which may not be feasible in real-world scenarios where communication is constrained or infrastructure is unavailable [2]. Moreover, methods that focus solely on exploration efficiency may neglect the importance of maintaining network connectivity, leading to isolated robots [3].

To address these challenges, we propose a distributed multi-robot exploration algorithm that enables robots to dynamically switch between two roles: explorers and supporters. This dynamic role assignment allows the system to balance the competing objectives of maximizing exploration efficiency and maintaining network connectivity. By default, all robots start as *explorers*, focusing on discovering new areas using a frontier-based exploration strategy. Frontiers, defined as the boundaries between explored and unexplored spaces, guide explorers to regions that maximize information gain. However, in multi-robot settings, frontier-based methods can lead to redundant paths and require coordination to ensure efficient area coverage [4]. *Supporter* robots are crucial for connectivity maintenance. Inspired by flocking behavior observed in nature, supporters position themselves to act as relay nodes, enabling the formation of a communication chain back to the base station. This flocking-inspired positioning enables the network to adapt dynamically to changes, ensuring that explorers remain connected even as they venture further into unknown territories [5], thanks to the use of Relative Neighborhood Graph (RNG) [6]. Our algorithm operates on distributed decision-making and local information sharing principles. Each robot makes decisions based on its local observations and interactions with neighboring robots within communication range. Robots share essential information such as position, velocity, current target, role, and local map data. This decentralized approach reduces the reliance on global knowledge and enhances the system's robustness to individual robot failures and communication interruptions [7].

The main contributions of this work are:

- A dynamic distributed role-switching mechanism that balances exploration and connectivity by allowing robots to adapt their roles based on real-time network conditions and exploration needs.
- A distributed flocking-inspired approach for supporter positioning, enhancing network connectivity through decentralized and adaptive relay placement.
- A simple yet powerful connectivity strategy leveraging the RNG, wherein each robot ensures continuous connectivity with minimal overhead and guaranteeing a globally connected network by design.

We evaluate the performance of our proposed algorithm through comprehensive simulations, comparing it with existing multi-robot exploration methods. In our simulations, the proposed method reduces the exploration time by up to 80%

and the time spent on redundant exploration by 74% relative to purely random exploration approaches, while maintaining overall connectivity. In addition, it achieves performance comparable to that of centralized frontier-based methods, thus providing a good balance between speed, connectivity, and decentralized implementation.

The remainder of this paper is organized as follows: Section II reviews literature on multi-robot exploration and connectivity maintenance, Section III introduces the system model and background, Section IV presents the proposed method, which performance analysis is presented in Section V. Section VI concludes and outlines directions for future research.

II. Related work

A. Connectivity Maintenance in Multi-Robot Systems

Connectivity maintenance is a fundamental requirement for effective multi-robot collaboration. Sabattini et al. [8] proposed a decentralized control strategy that uses algebraic connectivity to ensure network robustness, focusing control efforts on critical robots—nodes whose disconnection could partition the network. Cai et al. [9] extended this with an adaptive connectivity maintenance framework that dynamically adjusts communication links according to environmental and task specific requirements. Recent advancements have further refined these strategies. Luo et al. [10] introduced a minimally disruptive connectivity enhancement approach that optimizes robot movement to strengthen team connectivity while minimizing disruptions to ongoing tasks. Similarly, Ramachandran and Pierpaoli [11] proposed a resilient monitoring method for heterogeneous multi-robot systems. Their framework allows for adaptive network reconfiguration to effectively counter node failures in dynamic environments.

Robustness in the face of real-world uncertainties is another critical focus. Panerati et al. [12] proposed a controller capable of maintaining connectivity even under fault conditions to address the impact of hardware and software failures in swarm robotics. Luo and Sycara [13] tackled the challenge of robust connectivity by ensuring k-connectivity, allowing the network to withstand the failure of up to k-1 robots. Their k-Connected Minimum Constraints Subgraph (k-CMCS) algorithm provides robust connectivity with minimal interference to the robots' primary exploration tasks.

In summary, current research on maintaining connectivity in multi-robot systems offers robust solutions, yet most adopt either highly centralized architectures or rely on static assumptions regarding communication links. While these strategies ensure stable network configurations, they often constrain individual robots' ability to explore freely or adapt in real time. Our approach builds on the concept of decentralized connectivity control but introduces dynamic role-switching to achieve a more flexible balance between exploration and communication.

B. Exploration Strategies with Connectivity Awareness

Connectivity-aware exploration strategies aim to balance the competing objectives of efficient exploration and robust communication. Pei et al. [1] developed Connectivity and Bandwidth Aware Exploration (CBAX), which optimizes relay node placement and routing paths to manage bandwidth while ensuring network connectivity. Later enhancements demonstrated significant reductions in exploration time, particularly in dense environments [14].

Nestmeyer et al. [15] proposed a decentralized multi-target exploration approach that dynamically assigns tasks to robots while maintaining a connected network. However, their method does not address dynamic role-switching between exploration and connectivity maintenance, which is a key aspect of our approach.

Mahdoui et al. [4] introduced a frontier-based exploration strategy that reduces communication overhead by exchanging only frontier points between robots instead of full maps. This approach enables efficient coverage while conserving bandwidth and maintaining connectivity.

Recent studies have introduced more sophisticated methods to improve connectivity-aware exploration. Lin et al. [16] developed an online connectivity-aware dynamic deployment strategy for heterogeneous multi-robot systems. This method allows robots to adaptively redistribute themselves to maintain connectivity while optimizing exploration efficiency in dynamic environments.

Yang et al. [17] extended the role of line-of-sight constraints in multi-robot exploration. Their minimally constrained multi-robot coordination framework ensures connectivity while enabling robots to explore spatially separated regions efficiently.

Connectivity-aware exploration methods often combine frontier-based coverage with strategies to keep robots in range of one another. While effective, these solutions can struggle with slow adaptation or high communication overhead. In contrast, our framework builds on their frontier-focused principles but empowers robots to switch roles on the fly based on local exploration and connectivity demands, thus maintaining efficient coverage and dependable communication links.

C. Machine Learning for Exploration and Connectivity

Recent advancements have leveraged machine learning to improve multi-robot exploration. Zhang et al. [5] introduced Hierarchical-Hops Graph Neural Networks (H2GNN) for multi-robot systems. This approach uses multi-agent reinforcement learning (MARL) to optimize collaborative strategies, enhancing both exploration efficiency and network maintenance.

Li et al. [18] explored reinforcement learning for decentralized connectivity maintenance, embedding connectivity constraints within the learning framework. Their approach demonstrated adaptability across various simulated and real-world scenarios.

Machine learning techniques have shown promise in improving robot coordination. However, these methods typically require significant training and may not adapt well to highly dynamic environments. In contrast, our method avoids such dependencies by using a decentralized, real-time decision-making framework. This approach allows for greater flexibility and robustness in unknown or changing scenarios, where pre-trained models might not perform well.

III. System Model and Background

A. System model

We consider a fleet of N mobile robots, $R = \{r_1, r_2, \ldots, r_N\}$, operating autonomously in an unknown 2D environment. A fixed *base station* B is located at a known position $\mathbf{p}_B \in \mathbb{R}^2$, serving as both the primary communication hub and the starting point for exploration. Each robot r_i makes decisions based on local observations and interactions with neighbors within communication range.

By default, all robots start as *explorer* within communication range R_c of the base station B, thus focusing on discovering new areas via frontier-based exploration strategies. Each robot r_i has:

1) *Position and velocity:* $\mathbf{p}_i(t), \mathbf{v}_i(t) \in \mathbb{R}^2$ at time t.
2) *Role:* $\rho_i(t) \in \{\text{explorer}, \text{supporter}\}$.
3) *Sensing range:* $R_s > 0$, within which it can observe or map the environment.
4) *Communication range:* $R_c > 0$, within which it can exchange data with other robots or the base station.

$d_{ij}(t) = \|\mathbf{p}_i(t) - \mathbf{p}_j(t)\|$ is the Euclidean distance between two robots r_i and r_j at time t. Additional parameters include:

- $\alpha \in [0, 1] \subset \mathbb{R}$ is a scaling factor used so that a robot switches to supporter role preemptively to avoid losing communication, thus maintaining a stronger link margin.
- $\gamma \in [0, 1] \subset \mathbb{R}$ is a weighting factor balancing the influence of supporters and explorers on the supporters robots movement.
- A boolean $\text{Cn}(i)$ indicating whether robot r_i is connected to B via supporters,
- $M_i(t)$ as the local map of r_i, and $F_i(t)$ as the set of frontier points for r_i,
- $N_i(t)$ as the set of RNG neighbors for r_i,
- $L_i(t)$ as the set of all robots in communication range of r_i,
- $E_i(t)$ and $S_i(t)$ as sets of directly connected explorers and supporters, respectively.

B. Relative Neighborhood Graph (RNG) Background

The Relative Neighborhood Graph (RNG) is a classical concept in computational geometry for connecting points in a plane (or higher-dimensional space) based on their relative proximity. Given a set of points $\{p_1, p_2, \ldots, p_n\}$ in a metric space with distance function $d(\cdot, \cdot)$, an edge (p_i, p_j) is included in the RNG if and only if no third point p_k is strictly closer to both p_i and p_j than they are to each other. Formally:

$$d(p_i, p_j) < \max\{d(p_i, p_k),\ d(p_j, p_k)\} \quad \forall\, p_k \neq p_i, p_j. \quad (1)$$

Because the RNG is typically sparse, and has the ability to be locally computed, each point p_i only has a limited number of neighbors, thereby reducing communication overhead and computational complexity. Nonetheless, its design still preserves sufficient connectivity to perform tasks such as multi-robot coordination, decentralized networking, and distributed sensor coverage.

IV. Our Proposed Method DRBECM

In this section, we introduce the "Dynamic Role-Based Exploration with Connectivity Maintenance (DRBECM)" algorithm designed to efficiently navigate in unknown environments while maintaining continuous communication with a fixed base station. The key innovation of our approach lies in its dynamic role-switching mechanism, which allows robots to adaptively balance between exploration efficiency and network connectivity. A major advantage of it is the added flexibility it provides. By enabling each robot to switch roles in response to changing conditions, the approach strikes a balance between swiftly covering unexplored regions and reliably maintaining communication links. Additionally, we incorporate collision avoidance and stagnation detection mechanisms to enhance the robustness and safety of the system.

A. Dynamic Role Assignment

To balance exploration and connectivity, robots dynamically switch between explorer and supporter roles based on local network conditions and exploration needs [19]. This adaptability ensures efficient coverage of the environment while maintaining robust communication links back to the base station. The role update rule for each robot $r_i \in R$ at time t is defined as:

$$\rho_i(t+1) = \begin{cases} \text{supporter}, & \text{if } \rho_i(t) = \text{explorer},\ E_i(t) \neq \emptyset, \\ & \forall r_j \in S_i(t) \cup B,\ d_{ij}(t) > \alpha R_c, \\ \text{explorer}, & \text{if } \rho_i(t) = \text{supporter},\ \text{Cn}(i) = \text{True}, \\ & E_i(t) = \emptyset, |S_i(t)| = 1, \forall r_j \in L_i(t), \\ & \text{Cn}(j) = \text{True}, \exists r_k \in L_i(t), \\ & \rho_k(t) = \text{supporter}, d_{kB}(t) < d_{iB}(t), \\ \rho_i(t), & \text{otherwise.} \end{cases} \quad (2)$$

By continuously reallocating roles through local decision-making, the system mitigates single points of failure, preserves a coherent multi-hop network under dynamic conditions, and accelerates overall coverage without relying on a fixed or centralized strategy.

B. Efficient Neighbor Selection

To reduce computational complexity and enhance scalability, we employ the RNG for neighbor selection. Each robot constructs its local RNG graph solely from local observations of nearby robots, which aligns naturally with our distributed architecture and minimizes global information requirements.

RNG neighbors $N_i(t)$ for robot r_i are defined as:

$$N_i(t) = \left\{ r_j \in L_i(t) \;\middle|\; \begin{array}{l} \mathbf{d}_{ij}(t) < \max\{\mathbf{d}_{ik}(t),\, \mathbf{d}_{jk}(t)\}, \\ \forall r_k \in L_i(t) \cap L_j(t) \end{array} \right\}, \tag{3}$$

This selective communication ensures that robots maintain only essential connections without unnecessary messaging. The RNG facilitates more efficient and localized neighbor selection compared to considering all possible neighbors. A key property of the RNG is its sparsity; research has shown that a node in a two-dimensional RNG has, on average, between two and three neighbors [20]. This sparsity provides more degrees of freedom for explorers. By restricting each robot's connections to these local RNG neighbors, the entire graph remains connected, which is crucial for effective exploration and information sharing in a fully distributed manner.

C. Frontier-Based Exploration

Explorer robots utilize a frontier-based strategy to maximize information gain by focusing on unexplored regions. Frontiers are defined as the boundaries between known and unknown spaces [21]. Each explorer selects frontiers that are within communication range of a supporter or the base station to ensure continuous connectivity. The set of safe frontier points $F_{\text{safe}}(t)$ for robot r_i is determined by:

$$F_{\text{safe}}(t) = \left\{ f \in F_i(t) \;\middle|\; \begin{array}{l} \exists r_j \in N_i(t), \rho_j(t) = \text{supporter}, \\ \|\mathbf{f} - \mathbf{p}_j(t)\| \le R_c \end{array} \right\} \tag{4}$$

where, $\mathbf{f}$ is the position of a frontier point.

By selecting frontiers that maintain communication links, explorers can safely expand the explored area without becoming isolated from the network.

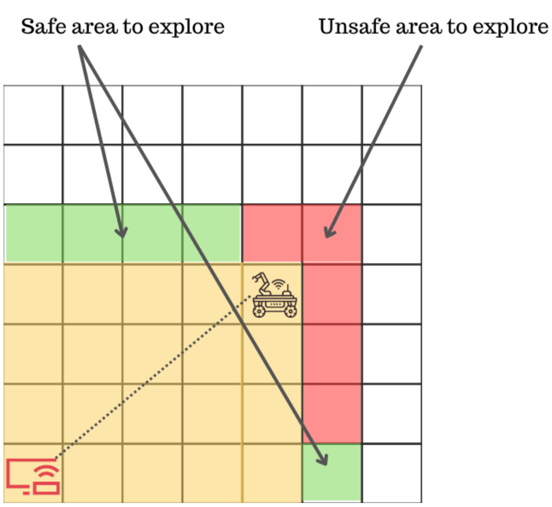

Fig. 1. Illustration of safe and unsafe areas for exploration based on communication range.

Figure 1 visually demonstrates the concept of safe frontiers. The grid represents the environment, with yellow cells indicating explored areas, white cells unexplored areas, and the robot and base station shown in their respective positions. The green cells represent the safe area to explore, which are frontier cells within the communication range ($R_c = 5$) of the robot. The red cells, in contrast, represent an unsafe area to explore, as they exceed the maximum communication range of 5 units from the robot, thus risking breaking the connectivity.

D. Flocking-Inspired Supporter Positioning

Supporter robots play a crucial role in maintaining network connectivity by acting as relay nodes between explorers. Inspired by flocking behavior observed in nature, supporter robot r_i adjusts its positions based on the movements of neighboring robots [22] as follows:

$$\mathbf{p}_i(t+1) = \gamma \mathbf{p}_{i,s}(t) + (1-\gamma)\mathbf{p}_{i,e}(t) \tag{5}$$

where:

- $\gamma \in [0,1] \subset \mathbb{R}$ is a weighting factor balancing the influence of supporters and explorers.
- $\mathbf{p}_{i,s}(t)$ is the position influenced by neighboring supporters and the base station:

$$\mathbf{p}_{i,s}(t) = \mathbf{p}_i(t) + \beta_1 \frac{1}{|N_{\text{supporters}}(t)| + 1} \Bigg(\sum_{r_j \in N_{\text{supporters}}(t)} \big(\mathbf{p}_j(t) - \mathbf{p}_i(t)\big) + \big(\mathbf{p}_B - \mathbf{p}_i(t)\big) \Bigg) \tag{6}$$

- $\mathbf{p}_{i,e}(t)$ is the position influenced by neighboring explorers:

$$\mathbf{p}_{i,e}(t) = \mathbf{p}_i(t) + \beta_2 \frac{1}{|N_{\text{explorers}}(t)|} \Bigg(\sum_{r_k \in N_{\text{explorers}}(t)} \big(\mathbf{p}_k(t) + \mathbf{v}_k(t) - \mathbf{p}_i(t)\big) \Bigg) \tag{7}$$

- β_1 and β_2 are scaling constants.
- $N_{\text{supporters}}(t)$ and $N_{\text{explorers}}(t)$ are the sets of neighboring supporters and explorers, resp.
- $\mathbf{v}_k(t)$ is the velocity of explorer r_k at time t.
- $\mathbf{p}_B$ is the position of the base station.

This approach enables supporters to dynamically position themselves to maintain robust communication links as explorers move further into unexplored regions.

E. Collision Avoidance Mechanism

Each robot uses collision avoidance to keep a safe minimum distance from its RNG neighbors. By focusing on RNG neighbors, the robots can efficiently manage collision avoidance without excessive computational overhead.

The acceleration $\mathbf{a}_i(t)$ of robot r_i is adjusted as:

$$\mathbf{a}_i(t) = \mathbf{a}_i(t) + \sum_{r_j \in N_i(t)} \phi(d_{ij}(t)) \frac{\mathbf{p}_i(t) - \mathbf{p}_j(t)}{d_{ij}(t)} \tag{8}$$

where:

- $\phi(d)$ is a repulsion function defined as:

$$\phi(d) = \begin{cases} k_{\text{avoid}} \left(\dfrac{R_{\text{avoid}} - d}{R_{\text{avoid}}} \right), & \text{if } d < R_{\text{avoid}}, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

- R_{avoid} is the collision avoidance range,
- $k_{\text{avoid}} \in (0,1)$ is a scaling factor for the repulsion force,

This mechanism operates between RNG neighbors, ensuring that robots maintain a safe distance from nearby robots and avoid potential collisions without needing to consider all robots in the environment.

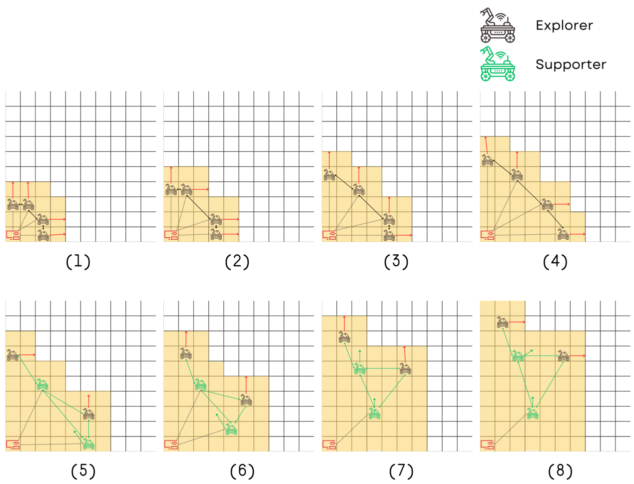

Fig. 2. **Progression of DRBECM.** Illustration of an exploration scenario with $N = 4$ robots: **(1)** Explorer robots are initially deployed near the base station; **(2)–(4)** Early exploration, robots disperse and expand the explored area until they reach the base station's communication limit; **(5)** Dynamic role assignment, as two robots switch to "supporter" roles; **(6)** Supporters move to maintain strong communication links; **(7)** "Support supporting," where one supporter aids another to reach farther areas; **(8)** Flocking-inspired support mechanism: supporter robots strategically position themselves to maintain connectivity.

F. Stagnation Detection and Recovery

Explorer robots may become stagnant due to obstacles or challenging terrain, hindering the overall exploration efficiency. To address this, each explorer robot monitors its recent movements to detect stagnation and if so to return back to a previous exploration point. A robot is considered *stagnant* if the maximum displacement over the last T_{stagnant} time steps is less than a threshold D_{stagnant}:

$$\max_{t - T_{\text{stagnant}} \le \tau \le t} \|\mathbf{p}_i(t) - \mathbf{p}_i(\tau)\| \le D_{\text{stagnant}} \tag{10}$$

where:

- T_{stagnant} is the chosen stagnation detection time window.
- D_{stagnant} is the chosen stagnation distance threshold.

Upon detecting stagnation, the explorer initiates a recovery behavior by moving towards the nearest supporter or the base station to reset its position and resume exploration:

$$\mathbf{p}_i(t+1) = \mathbf{p}_i(t) + \gamma_{\text{rec}} \big(\mathbf{p}_{\text{target}} - \mathbf{p}_i(t) \big) \tag{11}$$

where:

- $\mathbf{p}_{\text{target}}$ is the position of the closest supporter or the base station.
- γ_{rec} is a movement scaling factor for recovery.

This mechanism enables explorers to escape from local minima and keeps on effectively exploring.

G. Information Sharing and Map Updating

Robots share local information and update their maps based on sensor data and information received from their neighbors.

The map and frontier updates for robot r_i are given by:

$$M_i(t+1) = M_i(t) \cup \text{SensorData}_i(t) \cup \bigcup_{r_j \in L_i(t)} M_j(t) \tag{12}$$

$$F_i(t+1) = \left(F_i(t) \cup \bigcup_{r_j \in L_i(t)} F_j(t) \right) \setminus M_i(t+1) \tag{13}$$

where:

- $\text{SensorData}_i(t)$ is the new sensor data collected by robot r_i.

By sharing the local maps and frontiers with neighbors in the communication range, we improve the visibility of the environment for the robots, thus making better choices to explore unknown areas.

This decentralized strategy enables effective exploration by ensuring that robots have up-to-date local information for decision-making without being overwhelmed by data. By focusing on essential information from relevant neighbors, robots can make timely and informed decisions, enhancing performance and scalability of the multi-robot system.

Implementation Considerations: The proposed method operates under the principles of distributed decision making and local interactions. Robots rely solely on local information and communication with nearby robots, without the need for centralized control. This decentralization improves scalability and robustness to individual robot failures or communication disruptions.

H. Main Algorithm

To consolidate the components of our proposed method, we present Algorithm 1, which outlines the main steps executed by each robot r_i during the exploration process.

Algorithm 1 DRBECM, run at each robot r_i

Input: Constants: R_c, R_s

Output: $\mathbf{p}_i$, $\mathbf{v}_i$, ρ_i, M_i, N_i, F_i

$\mathbf{p}_i = \mathbf{p}_{\text{initial}}$ {Initial deployment position }

$\mathbf{v}_i = \mathbf{0}$ {Initial velocity }

$\rho_i = \text{explorer}$ {Set initial role }

$M_i = \{\mathbf{c} \in [0, X_{\max}] \times [0, Y_{\max}] \mid \|\mathbf{p}_i - \mathbf{c}\| \le R_s\}$ {Initially sensed area }

$F_i = \partial M_i$ {Boundary of initially sensed area }

while $F_i \neq \emptyset$ **do**

  $N_i \leftarrow \text{GetRNGNeighbors}(\mathbf{p}_i)$ {Eq. (3) }

  $\rho_i \leftarrow \text{NewRole}(N_i, \rho_i)$ {Eq. (2) }

  $\mathbf{p}_i, \mathbf{v}_i \leftarrow \text{MoveRobot}(N_i, \rho_i, R_c)$ {Eq. (4)–(11) }

  $M_i, F_i \leftarrow \text{MapUpdatingShareInformation}(N_i, M_i, F_i, R_s)$ {Eq. (12), (13) }

end while

Algorithm 1 presents the proposed distributed approach for multi-robot exploration inspired by flocking behavior. Each robot r_i maintains its state (position $\mathbf{p}_i$, velocity $\mathbf{v}_i$, role ρ_i), a local map M_i, and a frontier set F_i. The algorithm initializes these variables (lines 1–5) and then enters its main exploration loop (lines 6–10).

The exploration process is driven by the frontier set F_i, which represents unexplored areas. In each iteration, the robot determines its Relative Neighborhood Graph (RNG) neighbors N_i (line 7, Equation (3)). This step is crucial, as it defines the local network topology that informs subsequent decisions.

Based on this network information, the robot updates its role ρ_i (line 8, Equation (2)). This dynamic role assignment allows the system to adapt to the current exploration state, balancing between active exploration and network maintenance. Specifically, if an explorer robot detects that its movement might break network connectivity, it can switch to a supporter role, acting as a relay to maintain the communication link and further support the other explorers, as shown in the example scenario depicted in Fig. 2 —specifically sub-figure (5)— where two robots switch to the supporter role. Conversely, if a supporter robot determines that it's no longer needed as a relay (e.g., when explorers have moved closer to the base or other supporters), it can switch back to an explorer role to continue active exploration.

The robot's movement is then computed based on its current role and the positions of its neighbors (line 9, Equations (4)-(11)). Explorers move towards frontier points, while supporters adjust their positions to maintain network connectivity, ensuring a cohesive exploration effort. As illustrated in Fig. 2 —sub-figures (6) and (7)— the two explorer robots advance toward the frontiers, aided by supporters who maintain connectivity back to the base station.

Finally, the robot updates its local map and frontier set based on new sensor data and information shared with neighbors (line 10, Equations (12)-(13)). This step is critical as it integrates new information, potentially revealing new frontiers or closing existing ones, which directly influences the next iteration of the algorithm.

This cycle continues until the frontier set is empty, indicating complete exploration or the robots cannot move further due to the communications constraints.

V. Performance Evaluation

A. Experimental setting

All simulations were conducted with Python scripts using a grid-based simulation. The source code and configuration files for replicating these experiments are available at https://github.com/HazemCHAABI/DRBECM.

We conducted a series of simulations to evaluate the performance of our proposed method against existing multirobot exploration algorithms. The experiments were carried out on a 120×120 grid map, with a sensing range of 4 units and a communication range of 20 units. Each algorithm was tested with varying numbers of robots, ranging from 11 to 15, on 100 runs. We stop the simulation after 3000 steps if it is not complete. The following methods were compared:

- **DRBECM (Proposed Method)**: A decentralized method inspired by the flocking behavior, focusing on efficient exploration and connectivity maintenance.
- **Random Walk** [23]: A baseline decentralized approach where robots move randomly without coordination. In this method, each robot independently selects a random direction and moves in that direction for a predetermined number of steps before choosing a new random direction.
- **HCETIIC (Hybrid Cheetah Exploration Technique with Intelligent Initial Configuration)** [24]: Utilizes a central planner to coordinate robot movements and exploration tasks, considering the critical impact of initial robot positions. The algorithm aims to maximize exploration efficiency across different start configurations, including uniform, centralized, random, perimeter, clustered, and strategic positions.
- **Frontier Exploration (No Map Sharing)** [25]: A decentralized method in which robots individually explore frontiers without sharing information. Each robot maintains its own map of the environment and identifies frontier cells (unexplored areas at the boundary of known space) independently. Robots select frontiers to explore based on criteria such as distance and potential information gain, without coordinating their choices with other robots.
- **Frontier Exploration (Map Sharing)** [25]: A centralized solution that enables robots to share maps and coordinate their exploration. In this approach, robots periodically communicate their local maps to a central server. The shared information is used to construct a global map of the environment, allowing for more informed decision-making when selecting frontiers to explore.

B. Exploration Time

The exploration time, defined as the time required to achieve 100% coverage of the map, was analyzed across the exploration methods. Figures 3 and 4 depict the distribution of exploration times for the proposed method and baseline algorithms. The results demonstrate that the proposed method achieves exploration times comparable to the centralized Frontier with Map Sharing method. This is particularly notable given the decentralized nature of the proposed approach. The exploration time is significantly reduced for the proposed method compared to decentralized baselines such as Random Walk and Frontier without Map Sharing. Our proposed method achieves a median exploration time that is 21.49% faster than Frontier without Map Sharing and 80.26% faster than Random Walk. Whilst, the centralized approach Frontier with Map Sharing, achieves the lowest median exploration time but requires global map sharing.

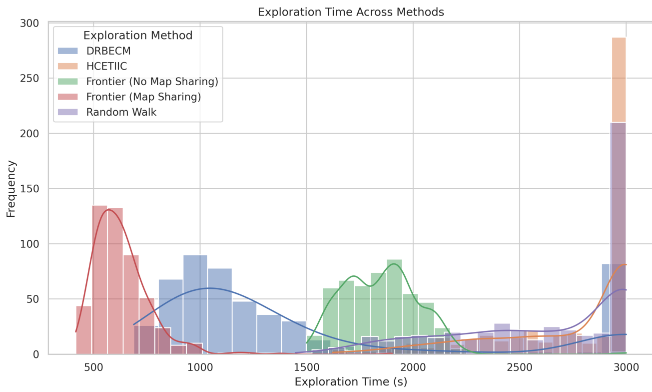

Fig. 3. Exploration time distributions for the different exploration methods.

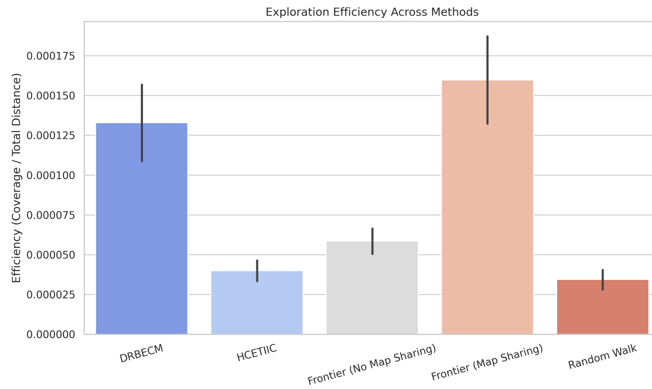

Fig. 5. Exploration efficiency across different methods.

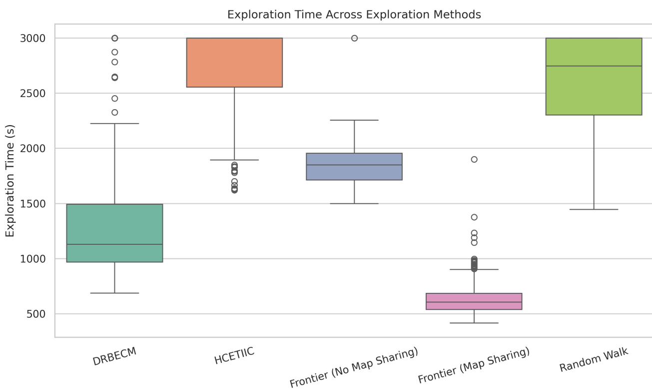

Fig. 4. Mean Exploration time across varying numbers of robots.

C. Exploration Efficiency

Exploration efficiency, measured as the ratio of coverage to the total distance traveled by all robots, was used as a key metric to evaluate the performance of different methods. As shown in Fig. 5, DRBECM presents a significantly higher efficiency compared to decentralized baselines. Specifically, it outperforms Frontier without Map Sharing and Random Walk by 126.8% and 291.2%, respectively. Although Frontier with Map Sharing achieves the highest efficiency, this comes at the cost of requiring centralized communication.

D. Redundant Exploration

Redundant exploration, measured as the total overlap of explored areas among robots, was evaluated to understand the efficiency of coverage coordination. As shown in Fig. 6, the DRBECM method achieves a significant reduction in redundant exploration compared to both centralized and decentralized baselines. Redundancy levels are 55.93% lower than Frontier without Map Sharing and 74.10% lower than Random Walk, highlighting the inefficiencies of these baselines in coordinating exploration. Centralized approaches, such as Frontier with Map Sharing, achieve the lowest redundancy overall due to their ability to globally coordinate robot movements. However, the proposed decentralized method effectively minimizes overlap without requiring global map sharing, demonstrating its ability to optimize coverage efficiently.

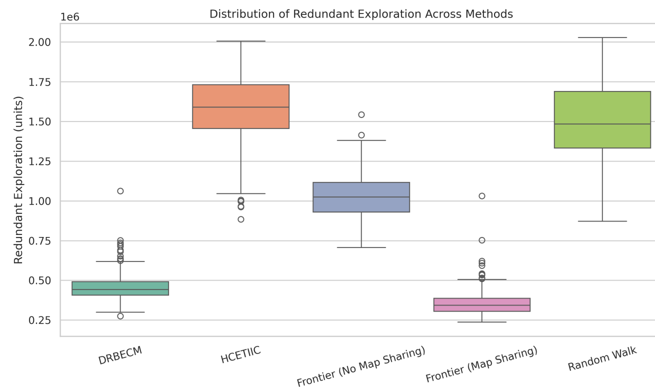

Fig. 6. Redundant exploration across the different methods.

E. Discussion

The results show the efficacy of the DRBECM method in achieving efficient and robust multi robot exploration. Unlike centralized approaches such as HCETIIC and Frontier with Map Sharing, our method does not rely on a central controller, making it more suitable for real-world applications where communication infrastructure is limited. Additionally, the dynamic role-switching mechanism allows our system to adapt to changing conditions, ensuring that robots maintain connectivity while exploring unknown regions.

By outperforming decentralized baselines in exploration efficiency and redundancy, and achieving competitive performance in exploration time, our proposed method strikes an effective balance between scalability, efficiency, and connectivity maintenance.

While our results show notable improvements in both coverage and connectivity, several limitations warrant closer examination. First, our role-switching mechanism can occasionally lead to multiple robots remaining in the supporter role if local connectivity metrics become overly conservative, thus reducing the number of active explorers and delaying full coverage. Additionally, our approach assumes relatively stable communication, which may not hold in cluttered or noisy settings. In reality, intermittent link failures, variable bandwidth, and communication delays could abruptly shift the network topology in ways our current switching rules may not fully accommodate. Maintaining flocking-inspired behavior among supporters also becomes more challenging in obstructed environments, where line-of-sight may be frequently lost. Moving forward, we plan to strengthen our algorithm by introducing a machine learning model that predicts the future QoS before the movement of the robots happen. We will also conduct extensive real-world experiments to validate our method's resilience under practical constraints.

VI. Conclusion

In this paper, we presented a novel distributed multi-robot exploration algorithm that dynamically balances exploration efficiency with network connectivity maintenance. By introducing a dynamic role-switching mechanism, robots adaptively assume explorer or supporter roles based on real-time assessments of network conditions and exploration demands. Explorers utilize a frontier-based exploration strategy to maximize information gain, while supporters employ a flocking-inspired approach to maintain robust communication links to the base station. Our method operates on the principles of distributed decision-making and local interactions, reducing reliance on centralized control and enhancing scalability and robustness. The incorporation of collision avoidance and stagnation detection mechanisms further improve the safety and efficiency of the multi-robot system.

Extensive simulations demonstrated that our approach achieves exploration times comparable to centralized methods while maintaining the advantages of a decentralized system. The results showed significant improvements in exploration efficiency and reduced redundancy compared to both centralized and decentralized baselines. Our algorithm effectively balances the trade-offs between exploration and connectivity, making it suitable for real-world applications where communication infrastructure may be limited or unreliable, such as in emergency response and disaster relief operations.

Future work will focus on extending the algorithm to more complex and dynamic environments, incorporating obstacles and varying communication ranges. Additionally, we plan to explore the integration of machine learning techniques to enhance decision-making and adaptability, as well as conducting real-world experiments to validate the practical applicability of our approach.

Acknowledgment

This work has been partially supported by the European Union's Horizon Europe research and innovation project under grant agreement No. 101070487 (NEPHELE).

References

[1] Y. Pei, M. Mutka, and N. Xi, "Coordinated multi-robot real-time exploration with connectivity and bandwidth awareness," in *EEE International Conference on Robotics and Automation*, 2010.

[2] A. Pal, R. Tiwari, and A. Shukla, "Multi-robot exploration in wireless environments," *Cognitive Computation*, vol. 4, pp. 526–542, 2012.

[3] K. Masaba and A. Quattrini Li, "Gvgexp: Communication-constrained multi-robot exploration system based on generalized voronoi graphs," in *IEEE Int. Symp. on Multi-Robot and Multi-Agent Systems (MRS)*, 2021.

[4] N. Mahdoui, V. Fremont, and E. Natalizio, "Cooperative frontier-based exploration strategy for multi-robot system," in *IEEE 13th Annual Conference on System of Systems Engineering (SoSE)*, 2018.

[5] H. Zhang, J. Cheng, L. Zhang *et al.*, "H2gnn: Hierarchical-hops graph neural networks for multi-robot exploration in unknown environments," *IEEE Robotics and Automation Letters*, vol. PP, 2022.

[6] G. T. Toussaint, "The relative neighbourhood graph of a finite planar set," *Pattern Recognit.*, vol. 12, pp. 261–268, 1980.

[7] W. Tang, C. Xue, C. Li, and Q. Zhu, "Towards coordinated multi-robot exploration under bandwidth-constrained conditions," in *IEEE Int. Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2022.

[8] L. Sabattini, C. Secchi, N. Chopra, and A. Gasparri, "Distributed control of multirobot systems with global connectivity maintenance," in *IEEE Int. Conference on Robotics and Automation (ICRA)*, 2011.

[9] D. Cai, S. Wu, and J. Deng, "Distributed global connectivity maintenance and control of multi-robot networks," *IEEE Access*, 2017.

[10] W. Luo, N. Chakraborty, and K. Sycara, "Minimally disruptive connectivity enhancement for resilient multi-robot teams," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[11] R. K. Ramachandran, P. Pierpaoli, M. Egerstedt, and G. S. Sukhatme, "Resilient monitoring in heterogeneous multi-robot systems through network reconfiguration," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 126–138, 2022.

[12] J. Panerati, M. Minelli, C. Ghedini, L. Meyer, M. Kaufmann, L. Sabattini, and G. Beltrame, "Robust connectivity maintenance for fallible robots," *Autonomous Robots*, vol. 43, pp. 769–787, 2018.

[13] W. Luo and K. Sycara, "Minimum k-connectivity maintenance for robust multi-robot systems," in *IEEE Int. Conference on Intelligent Robots and Systems (IROS)*, 2019.

[14] Y. Pei, M. Mutka, and N. Xi, "Connectivity and bandwidth-aware real-time exploration in mobile robot networks," *Wireless Communications and Mobile Computing*, vol. 13, pp. 847–863, 2013.

[15] T. Nestmeyer, P. Giordano, H. Bülthoff, and A. Franchi, "Decentralized simultaneous multi-target exploration using a connected network of multiple robots," *Autonomous Robots*, vol. 41, pp. 989–1011, 2015.

[16] C. Lin, W. Luo, and K. Sycara, "Online connectivity-aware dynamic deployment for heterogeneous multi-robot systems," in *IEEE Int. Conference on Robotics and Automation (ICRA)*, 2021.

[17] Y. Yang, Y. Lyu, and W. Luo, "Minimally constrained multi-robot coordination with line-of-sight connectivity maintenance," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[18] M. Li, Y. Jie, Y. Kong, and H. Cheng, "Decentralized global connectivity maintenance for multi-robot navigation: A reinforcement learning approach," in *Int. Conference on Robotics and Automation (ICRA)*, 2022.

[19] Z. Yan, N. Jouandeau, and A. Cherif, "A survey and analysis of multi-robot coordination," *International Journal of Advanced Robotic Systems*, vol. 10, p. 1, 01 2013.

[20] O. Melchert, "Percolation thresholds on planar euclidean relative-neighborhood graphs," *Physical Review E*, 2013.

[21] B. Yamauchi, "A frontier-based approach for autonomous exploration," *IEEE Int. Symposium on Computational Intelligence in Robotics and Automation (ICRA). 'Towards New Computational Principles for Robotics and Automation'*, 1997.

[22] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[23] F. Xia, J. Liu, H. Nie, Y. Fu, L. Wan, and X. Kong, "Random walks: A review of algorithms and applications," *IEEE Transactions on Emerging Topics in Computational Intelligence*, no. 2, p. 95–107, Apr. 2020.

[24] A. El Romeh and S. Mirjalili, "Theoretical framework and practical considerations for achieving superior multi-robot exploration: Hybrid cheetah optimization with intelligent initial configurations," *MDPI Mathematics*, vol. 11, no. 20, 2023.

[25] U. Jain, R. Tiwari, and W. W. Godfrey, "Comparative study of frontier based exploration methods," in *Conference on Information and Communication Technology (CICT)*, 2017.