Distributed Connectivity Restoration and Improvement in Networked Robots

Atharva Sagale   Ramviyas Parasuraman

***Abstract*— Multi-robot coordination requires maintaining network connectivity, especially in critical operations such as search and rescue, where network robustness is paramount. In this paper, we study the Fast k-connectivity Restoration problem (FCR), which aims to minimize the maximum distance required to restore connectivity. Recent works have proposed scalable solutions, but they rely on a centralized architecture or a learning-based solution to transfer the observable policies from centralized to distributed variants. However, computing node connectivity in a distributed setting is challenging, and performing such operations in a deterministic, algorithmic manner is critical for generalizability and persistent deployment in diverse, unknown environments. We propose a distributed control approach for improving and restoring connectivity. Using local neighborhood interactions and information, our algorithm improves the degrees of individual robots by augmenting 1-hop edges based on their distances, achieving the desired level of node connectivity and yielding comparable performance to achieve sparse connectivity improvements and up to 58% reduction in movements required to achieve dense connectivity over the state-of-the-art algorithms.**

I. Introduction

Mobile robots play a crucial role in various real-world applications, such as urban search and rescue, mining, exploration, and environmental monitoring [1]. In scenarios like search and rescue missions in disaster-stricken areas, robots can be deployed to explore inaccessible environments, such as collapsed buildings, dense forests, or underwater cave systems, where human intervention is difficult or dangerous [2], [3]. Multi-robot teams benefit from collaboration among agents, extending their capabilities of spatial coverage and improving task performance and providing redundancy beyond single-robot systems [4]. A critical enabler of these capabilities is reliable inter-robot communication: limited wireless ranges demand that the team maintain a connected communication graph so that information can flow among all robots during task execution [5], [6]. With limited infrastructure in such scenarios, robot teams must rely on distributed approaches for cooperative tasks, as the lack of centralized coordination poses a major challenge [7], [8], [9].

In harsh and adversarial environments, robots can suffer sensor or actuator faults, battery depletion, or malicious attacks; the network can fragment unless its topology is designed for resilience [10], [11]. Therefore, the problem of improving connectivity through robot movements has received increasing attention in recent years [12], [13], [14], [15]. Specifically, we consider the problem of Fast k-Connectivity Restoration (FCR) [16], which arises when an initially connected multi-robot network becomes less than k-connected due to failures or dynamic changes, or if the connectivity is not up to a desired level due to sparsity in the connectivity graph. The objective is to improve k-connectivity by relocating robots with minimal movement. Fig. 1 provides an overview of the problem with our distributed solution.

School of Computing, University of Georgia, Athens, GA 30602, USA. Corresponding author email: ramviyas@uga.edu

This research is supported by the Army Research Laboratory and is accomplished under DCIST Cooperative Agreement W911NF-17-2-0181.

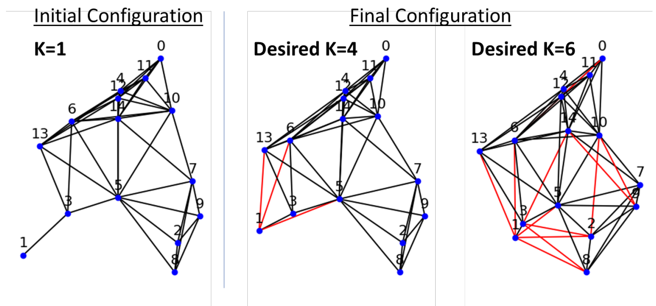


Fig. 1: Depiction of the FCR problem showing the initial configuration and the final configuration after augmented edges (links) to restore k-connectivity to K = 4 or 6.

In this context, k-connectivity is a strong graph-theoretic metric indicating the robustness of a graph: a k-connected graph remains connected upon removal of any $(k-1)$ nodes and guarantees k vertex-disjoint paths between every pair of surviving robots, thus bounding the worst-case impact of failures and eliminating single-node bottlenecks [17], [18]. Also, the k-connectedness plays an important role in wireless sensor networks, including networked robots [19], [20].

Recent works in the literature [21], [16] have demonstrated robot control algorithms using centralized algorithms as well as learning-based distributed variants [16], [22] to improve k-connectivity of a robotic network. Despite these advances, there remains a critical gap in designing fully distributed algorithmic solutions for k-connectivity restoration that (i) operate with only local neighbor information, (ii) scale to large teams, and (iii) minimize the worst-case robot movement without centralized coordination. In this paper, we address this important gap by introducing a novel distributed algorithm for FCR. Our key contributions are as follows:

- We propose the first fully distributed solution to the FCR problem that relies solely on local (neighbor) information, eliminating the need for centralized coordination and making it highly scalable and robust for large multi-robot networks and sparse networks.
- We introduce a distributed edge augmentation algorithm that restores k-connectivity by identifying edges to add through 1-hop local information, and a distributed relocation control algorithm that simultaneously increases

connectivity while significantly reducing the maximum travel distance made by the robots.

Extensive experiments, including real-world demonstrations, highlight that our algorithm performs significantly better than state-of-the-art baselines across diverse team sizes and k-connectivity needs. We also demonstrate our algorithm for connectivity restoration following failure events. The proposed algorithms are available in Github[1].

II. Related Work

Several approaches have been presented in the literature that focus on robust connectivity maintenance during task execution. The node connectivity improvement algorithms typically embed k-connectivity constraints directly into task-oriented controllers using Control Barrier Functions (CBFs) [23]. While these methods ensure minimal deviation from nominal tasks and maintain robustness in online settings, they rely on centralized (global) optimization frameworks, which are impractical for scalability and computational tractability. Similarly, bi-connectivity restoration (i.e., $k = 2$) has been demonstrated via approximation algorithms [15], but the results do not extend to general k-connectivity restoration. Furthermore, real-world applications impose additional requirements of fault tolerance. In fact, very early Work by Dolev [24] suggests a correlation between k-connectivity and the number of faulty nodes that can be tolerated by the system, formally expressed as $\kappa \geq 2f+1$, for f faulty nodes. This relationship motivates pursuing higher levels of k-connectivity in fault-tolerant networked multi-robot systems.

A key advancement in [21] presents scalable approximation strategies to achieve k-connectivity, demonstrating theoretical bounds and experimental results using large-scale simulations. Yet, this approach did not minimize the maximum distance traveled by robots to improve k-connectedness, thereby making the solution less energy-efficient.

Shi et al. [16] proposed a centralized algorithm that yields an optimal solution to the Quadratically Constrained Program (QCP) formulation of the FCR problem. In the first phase, termed the Edge Augmentation (EA) algorithm, operates on the network graph topology and identifies a set of additional edges that are necessary to achieve the desired level of k-connectivity. Here, we provide a brief outline of the process. For a graph $G(E, V)$, the algorithm begins by sorting the set of absent edges $\overline{E}$ and incrementally adding them until the required k-connectivity is attained. Redundant edges are subsequently pruned while maintaining node connectivity, and the remaining edges in $G' - G$ constitute the augmentation set. The second phase, Sequential Cascaded Relocation (SCR), addresses the controlled relocation of robots with the dual objectives of minimizing movement and maintaining existing connectivity. This is accomplished by iteratively traversing the augmentation set to establish new edges sequentially. During edge formation, the corresponding robot pair moves toward one another, while the remaining robots verify that their neighbor links remain intact. If any links are disrupted, corrective movements are performed to re-establish connectivity. However, this EA-SCR approach assumes a centralized optimization and planning and does not account for local distributed implementations. Moreover, the restoration process is not designed to execute incrementally or adaptively, which limits flexibility in practical settings. The authors also propose a distributed variant using a Graph Neural Network (GNN), which learns the policies by observing the EA-SCR outcomes. But, the learning-based solutions fail to generalize to new environments and suffer from scalability and unseen scenarios. In contrast, we propose an algorithmic solution that deterministically improves k-connectivity using only local (distributed) control laws.

[1]https://github.com/herolab-uga/distributed-fast-connectivity-restroration

III. Problem Formulation

A graph G with the vertex (robots) set V, and the edge (connectivity links) set E is *connected*, iff $\forall u, v \in V, \exists$ a path from u to v. A graph $G = (V, E)$ is k-connected iff $|V| > k$ and $\forall S \subseteq V, |S| < k$, the graph $G[V \setminus S]$ is connected [25]. A robot team of size N is deployed in an unknown environment. The set of robots $V = r_1, r_2, \ldots, r_n$, represented by the positions $X = x_1, x_2, \ldots, x_n \in \mathbb{R}^2$. Given robot positions X, and communication radius r, the *communication graph* $G(x)$ is defined as a geometric disk graph. If there is an edge between two robots, the euclidean distance is at most r. Note that we use the terms 'network' and 'graph' interchangeably to denote the same structural entity represented by the structure G.

FCR and Connectivity Improvement: Given the connectivity graph G, and the desired level of connectivity k (also referred to as *node connectivity*), the FCR problem aims to find the augmented edges, *i.e.,* the set of edges that must be added to achieve the desired k, and the maximum movement of the robots is minimized. The new positions represented by $X^* = x_1^*, x_2^*, \ldots, x_n^*$ form the graph $G(X^*)$, which is now $k-$connected, and $\min \max_{1 \leq i \leq n} \|X_i^* - X_i\|$.

IV. Methodology

In this section, we present our key contributions, the scalable Distributed Edge Augmentation (DEA) and Distributed Relocation (DR) algorithms. Our algorithms focus on the Graph Topology Optimization (GTO) problem [16], where the goal is to determine the edges to augment to make a graph k-connected in a distributed architecture. To arrive at a solution, we ground our principle on the key result of Menger's theorem (Lemma 1), which is supported by Whitney's theorem in graph theory [26].

Lemma 1 (Menger's Theorem [27]): Let G be a nontrivial graph (i.e., G has at least two vertices). Then the vertex k-connectivity $\kappa(G)$, edge connectivity $\lambda(G)$, and minimum degree $\delta_{min}(G)$ of G satisfy the following inequality:

$$\kappa(G) \leq \lambda(G) \leq \delta_{min}(G). \tag{1}$$

In a distributed setting, it is challenging for robots to determine the current k-connectivity of the entire network. Alternatively, it is more accessible for individual robots to perceive their degree $\delta_i = |\mathcal{N}_i|$, *i.e.*, the number of local

Algorithm 1: Distributed Edge-Augmentation

Input : Graph based on sensing range G; desired connectivity degree δ
Output: Augmentation set E_a

1 **Function** `DEA`(G, δ)**:**
2   **for** *every robot* $i \in V(G)$ **do**
3     $E_a^i = \emptyset$;
4     $\delta_i = \mathcal{N}_i$ # get number of neighbors;
5     $\gamma \leftarrow \delta - \delta_i$;
6   **if** $\gamma > 0$ **then**
7   **for** *every* $j \in \mathcal{N}_i$ **do**
8         $E_a^i \leftarrow (i, k) \mid k \in \mathcal{N}_j / i$;
9       $sort(E_a^i)$ # sort based on distance ;
10       $E_a^i \leftarrow \{E_i\}_{i=1}^{\gamma}$ # Get only first γ edges;
11 **return** $E_a^i \;\; \forall i \in V(G)$;

Algorithm 2: Distributed Restoration (DR)

Input : Input graph G, Sorted Edge Augmentation Set E_a, Robot positions X
Output: Augmented Graph G'

Function `dist_relocate`(G, E_a)**:**
  for *every robot* $i \in V(G)$ **do**
    $E^i \leftarrow$ current edges for i from G ;
    $E_a^i \leftarrow$ new edges for i from E_a ;
    $E^i \leftarrow E^i \cup E_a^i$;
    for *each edge* $(i, j) \in E^i$ **do**
      $d_{ij} \leftarrow \|x_j - x_i\|$;
      if $d_{ij} > h$ **then**
        $\dot{x_i} = \dot{x_i} + \eta \, d_{ij} \, \frac{x_j - x_i}{\|x_j - x_i\|}$;
    $x_i = f_{move}(x_i, \dot{x}_i)$;
return $G(X)$

neighbors of robot i, where $\mathcal{N}_i$ is the neighborhood set of robot i. For a non-trivial graph G, the minimum degree of any node $\delta_{min}(G) = \min_{1 \le i \le n} \delta_i$ and the vertex connectivity k are related in such a way that improving $\delta_{min}(G)$ will eventually result in improvement of k-connectivity, albeit not proportionally, as there are more disjoint paths connecting the robot to neighboring robots. By imposing the constraint on the degree of individual robot i, we can ensure that $\delta_{min}(G) \ge K$, where K is the desired k-connectivity, thereby improving the global connectivity.

Given the distributed setting of our approach, we make the following assumption: the input graph G is connected (not disjoint) and local neighbors can be sensed throughout the robot network. Furthermore, to simplify the computational model in our high-scale simulations, robots are abstracted as point entities, without collision-handling mechanisms.

We ground our comparisons against the centralized EA-SCR algorithm [16], which takes the robot positions as input and outputs the Edge Augmentation set, a list of edges to add to improve node connectivity. Because this algorithm has access to the global graph, it can add the most cost-effective edges. In the Sequential Cascaded Relocation phase, it takes the Edge Augmentation set as input and computes the necessary robot motions to realize the addition of each edge, orchestrated by a centralized planner. The algorithm also includes an edge restoration mechanism to address connectivity loss that may arise as robots reposition themselves to establish new edges.

We propose two distributed algorithmic variants to the FCR problem: Alg. 1 for Distributed Edge Augmentation (DEA) and Alg. 2 for Distributed Relocation (DR). The DEA algorithm, similar to the centralized variant, outputs an Edge Augmentation set for (and obtained by) each individual robot, as a list of edges that a particular robot needs to add to improve the degree of that robot. The DR algorithm proposes a parallelized approach to edge augmentation, wherein the graph is continuously updated and monitored during execution. This allows the algorithm to identify edges that have already been established as a by-product of other robot movements, thereby avoiding redundant repositioning. A notable consequence of this continuous neighbor monitoring is that edge disconnections are implicitly tracked throughout the process, eliminating the need for a separate restoration step. This process also enables fault-tolerant motion during restoration, assuming that faults in robots are reflected in network failures, which are captured in our DR algorithm.

A. Distributed Edge Augmentation (DEA)

In this section, we present the DEA algorithm (Alg. 1), which is a distributed transformation of the EA algorithm in [16]. We begin by introducing a distributed protocol, where each robot i decides and applies the algorithm onboard, possessing information limited only to itself, such as its position and the set of neighbors detected within its communication range. The robots in the network can exchange information with their immediate neighbors per the graph. We assume that the robots can locate themselves in a common frame of reference (e.g., GPS or via collaborative localization approaches), and have prior information about the desired K (objective). Each neighbor robot shares a list of their neighbors along with the neighbor locations, which the robots accumulate as a list of 1-hop neighbors.

Each robot begins by calculating its individual degree and computing the γ which represents the edge deficit, *i.e.*, the number of edges required to satisfy minimum degree connectivity constraints (Line 5). Next, a list E of all the 1-hop neighbor edges for i are computed, defined as $[(i, k) \mid k \in \mathcal{N}_j / i]$, where $j \in \mathcal{N}_i$ (Line 8). The candidate edges in E are then sorted in non-decreasing order of their distance, and the first γ edges are chosen to augment the graph (Lines 9, 10). Our key result guaranteeing an improvement in k-connectivity is presented in Theorem 1.

Theorem 1 (Distributed Edge Augmentation Convergence): Let $G = (V, E)$ be the initial sensing graph with a minimum degree $\delta_{min}(G) < \kappa$. Applying Alg. 1 (Distributed Edge Augmentation) produces an augmentation set E_a such that the augmented graph $G' = (V, E \cup E_a)$ satisfies $\delta_{min}(G') \ge \kappa$. Moreover, convergence is achieved in a synchronous round.

Proof: Each robot i computes its local degree deficit as:

$$\gamma = \kappa - \delta_{min}(G_i), \tag{2}$$

where G_i is the subgraph induced by robot i and its two-hop neighbors. If $\delta_{min}(G_i) \geq \kappa$, then $\gamma \leq 0$, and no edges are added. Otherwise, robot i selects the γ number of edges from a list of one-hop neighbors sorted based on the shortest distance (minimum-cost) candidate links to non-neighbors and proposes to form those connections.

The complete augmentation set is given by:

$$E_a = \bigcup_i \{(i,j) \mid j \in \mathcal{N}_i,\ j \text{ among top } \gamma_i \text{ by cost } \}. \tag{3}$$

This ensures that after a single update, all nodes meet the degree constraint: $\delta_{min}(G') \geq \kappa$. Hence, the graph improves its minimum degree δ_{min} in one synchronous round. Using Lemma 1, this increase in δ_{min} will eventually be captured as an increase in k-connectivity of the network. ■

B. Distributed Relocation

We introduce a Distributed Relocation (DR) algorithm in Alg. 2 to efficiently achieve edge augmentation. In this algorithm, the robots repeatedly adjust their positions based on the constantly updating neighborhood graph, defined by sensing radius and augmented edges. We design a persistent algorithm that ensures preservation of robot edges by enforcing a constraint on the robots to maintain their local neighborhood, and we achieve edge restoration. First, each robot determines a list of its distance-based neighbors and adds the candidate neighbors from E_a. The robots repeatedly check whether any of their edge lengths (actual and augmented) are exceeding the sensing threshold. If they do, it is implied that a new edge has been introduced or an existing edge has been severed. Next, a velocity vector is generated to pull the robots towards their corresponding neighbors. This mechanism serves the dual purpose of edge augmentation and implicitly restoring any severed edges while the robots reposition. All robots update their positions based on these velocities. This continuous adjustment continues until no neighbor pair exceeds the sensing range. In contrast to the baseline, this is a parallelized process, with multiple robots executing movements in a single step. By parallelizing the movements and tracking the changes, the algorithm accounts for real-time changes to the robot network.

It is worth noting that the movement algorithm in Alg. 2 incorporates a low-level motion planner in Step 10, which allows handling local obstacle avoidance, kinematic constraints, and other real-world complexities through local control frameworks [23].

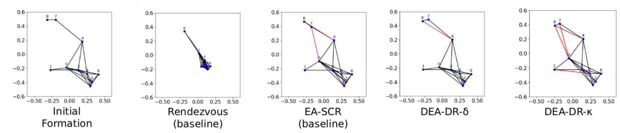
C. Optional verification of 'K' connectivity

In cases where ensuring node connectivity is critical and time-sensitive, simply improving the minimum degree is insufficient. Given the distributed nature of the DEA-DR algorithm, we focus on improving the minimum degree to enhance node connectivity, but as deduced from Eq. 1, improving the minimum degree does not ensure desired node

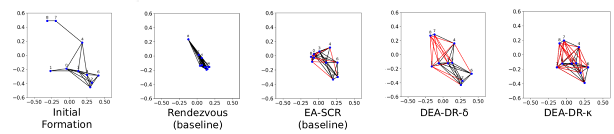
Algorithm 3: Verify Node Connectivity

Input : Graph based on sensing range G; desired connectivity degree K

```
ctr ← 0 ;
while not has_node_connectivity(G, K) do
    E_a ← DEA(G, K+ctr);
    G ← dist_relocate(G, E_a) ;
    ctr ← ctr + 1 ;
```



(a) N = 10, initial K=1 and target K = 2 in this experiment.



(b) N = 10, initial K=1 and target K = 6 in this experiment.

Fig. 2: Robot positions across different target K.

connectivity. To this end, we deploy a verification algorithm (Alg. 3) that checks global node connectivity (if available to all robots) and iteratively reruns the DEA-DR algorithms to increase the minimum degree by an additional level until k-connectivity is satisfied.

This algorithm serves as an additional safeguard when a global graph is available to all robots. We refer to our algorithmic approaches as **DEA-DR-δ and DEA-DR-k** to denote the versions without and with the k-connectivity check routine (Alg. 3), respectively. Note that the DEA-DR-δ variant does not guarantee achieving the desired node connectivity, but the DEA-DR-k variant does.

V. Experiments

In this section, we briefly explain our experimental setup and evaluation metrics. We empirically compare our two distributed variants primarily against the centralized baseline (as well as its distributed learned model) from [16], as well as a weighted rendezvous approach [28], which serves as a distributed baseline. We use a synthetic 2D environment dataset comprising randomly generated robot positions. In our experiments, the initial $k \geq 1$. The algorithms are implemented in Python 3.10 and evaluated on a machine equipped with an Intel Core i7-8700 CPU and 16 GB RAM.

The centralized EA-SCR algorithm serves as a meaningful reference point for evaluating the relative performance and the impact of the proposed distributed edge augmentation algorithms. The distributed rendezvous approach serves as an ablation study to evaluate the impact of strategically selecting edges, rather than simply converging towards a common point, on node connectivity.

The experiment setup involves a robot network with a known initial K-connectivity of 1, with different target K and number of robots, in a $1m \times 1m$ virtual testbed. The robots form a disk graph with the sensing radius predefined to be 0.5 m. Any pair of robots within sensing range of each other is treated as a bi-directional communication edge. In the first experiment, Fig. 3, we test the algorithms in teams of sizes N = [10, 20, 30], the robots start with the same 1-connected graph and augment to increasing levels of connectivity. The data points represent the mean and deviation across 10 distinct seeded trials. In the second experiment, we set the same target K = 10, but vary the initial levels of K-connectivity. As observed in Fig 4, we test robot networks of size N = 20 with varying initial densities, from sparse to dense. Additionally, we also perform a scalability test on the robots, depicted in Fig. 5, the experiment setup involved trials on teams of N = [10, 20, 30], initially 1-connected graphs, with target K = N/2. We use the following metrics to evaluate the proposed algorithms:

Max-Min movement: This metric represents the maximum among the minimized per-robot movement distances, capturing the largest displacement required by any robot to achieve edge augmentation. Since the problem objective is to minimize overall motion, lower max–min distances are indicative of superior algorithm performance.

Total movement: This is the sum of $\|X_i^* - X_i\| \forall i \in N$, where X are the initial robot-positions and X^* are the augmented positions *i.e,* the sum of all robot-movements. We use Total movement and Max-Min movement (also referred to as 'Max Move' in the figures) to compare the performance of the approaches. This is an equally important metric as it captures the reposition behavior of the entire network. Approaches that achieve a lower Max move may nonetheless exhibit a higher Total move, owing to a higher number of robots contributing to the cumulative displacement.

Area of convex hull: This is the total area covered by the formation. In applications with secondary objectives such as coverage or formation control, maintaining the shape of the formation is critical. Lower area of formation generally indicates that the structural integrity of the formation has been compromised, with the robots collapsing into a fully-connected configuration, analogous to that obtained from a rudimentary rendezvous approach.

Added Edges: We keep track of the number of added edges to monitor the trade-off between adding low-cost edges and adding relevant edges. For instance, augmenting edges between two densely connected nodes solely due to low edge cost might be inefficient.

VI. RESULTS

In the first experiment, as described earlier, the robots begin with the same formation and are tasked with achieving varying levels of node connectivity. The Max Move is a key metric to monitor, as it is the optimization problem's objective in EA-SCR. Figs 2 depict the working of the proposed approaches and baselines. The figures illustrate the final positions of the robots after edge augmentation and displacement. The edges in red represent the augmented edges. In Fig. 2a, we can see the distinct edges chosen by the centralized and distributed algorithms, signifying the varying edge selection strategies by these algorithms. In Fig. 2b it is clear that the area covered by DEA-DR-k is superior to the centralized baselines.

As observed in Fig. 3, the proposed approach, DEA-DR-κ, outperforms the baseline as the target K increases. This effect can be observed for different levels of N and also in adjacent experiments in Figs. 4 and 5. Furthermore, this is also evident in the Total Move metric, which is highly correlated and exhibits a similar trend.

Our observations reveal a consistent pattern in the results. As the "connectivity gap" between the initial and target node connectivity increases, the performance advantage of our approach becomes more pronounced. This 'gap' will be referred to throughout the subsequent discussion, as it is a significant factor influencing the performance of the algorithms under evaluation. This is observed in almost all metrics and figures. When the gap is sufficiently small, the EA-SCR baseline outperforms the other approaches, as its global edge augmentation strategy selects the most cost-efficient edges while minimizing the maximum robot movement. However, as the connectivity gap increases, the number of edges required to reach the target connectivity increases, as indicated by the experimental results.

Since the baseline first computes the full Edge Augmentation set based on the initial state of the graph and then adds edges by displacing robots sequentially, the pre-computed edge selections do not account for the structural changes that occur as robots reposition themselves during execution, such as changes caused by robots trying to account for severed edges. As a result, although the augmentation set is globally optimal with respect to the initial graph, the sequential robot movements may alter the graph structure, making later edge additions suboptimal. This effect becomes increasingly pronounced as the number of required edges grows with the gap.

Our distributed approach, while constrained to locally available edges, benefits from parallel execution that collectively accounts for broader connectivity requirements, mitigating the sub-optimality introduced by sequential decision-making. Furthermore, this trend is observed in the weighted Rendezvous baseline, which also relies on local information, thereby further supporting the argument that local edge selection becomes increasingly advantageous as the connectivity gap widens. Notably, our proposed approach outperforms both the centralized baseline and the Rendezvous algorithm, demonstrating that the combination of distributed execution and strategic edge selection yields superior performance as connectivity requirements become more strict.

We further examine the influence of the gap on the performance of dealing with the FCR problem. Based on the experimental results, one might intuitively conclude that the baseline performs better when the target connectivity corresponds to a sparse graph, while our approach becomes increasingly advantageous as the target graph grows denser. To validate this hypothesis, an additional experiment is conducted. Fig. 6 depicts an experiment on a team of N=20

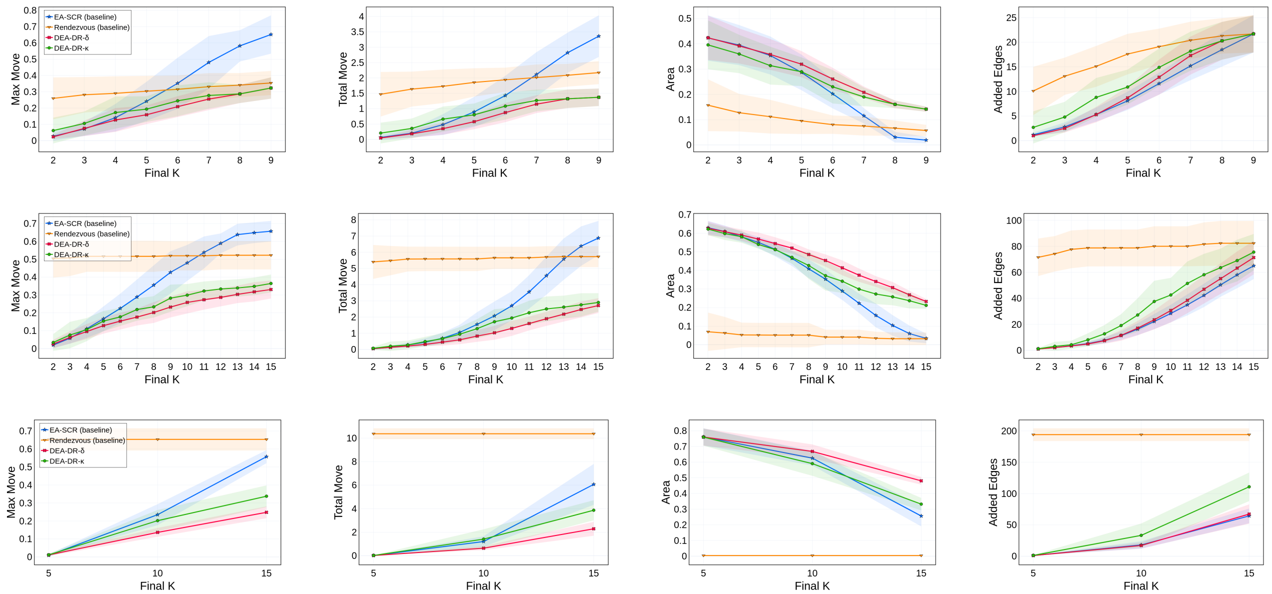


Fig. 3: Final k-connectivity for various team sizes. Top row (N=10), middle row (N=20), and the bottom row (N=30). All the approaches are initialized with the same 1-connected formation.

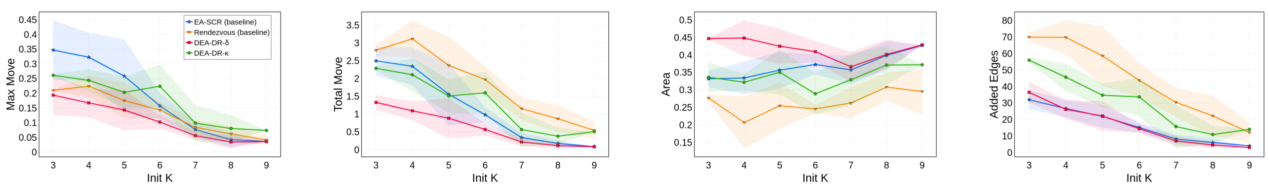


Fig. 4: Experiment on a team of N = 20 robots, with target K = 10 and varying levels of initial node connectivity.

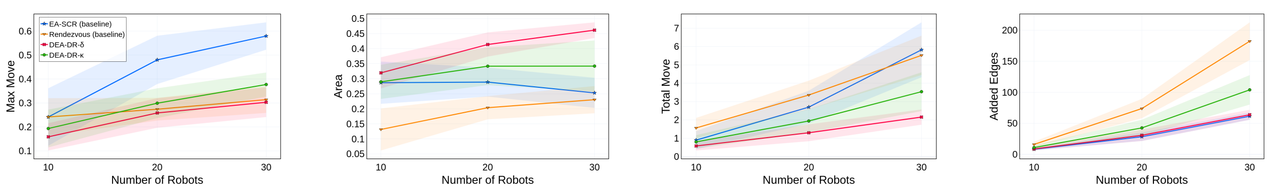


Fig. 5: Experiment on a team of N = [10, 20, 30] robots, with target K = [5, 10, 15] *(target K = N/2)*.

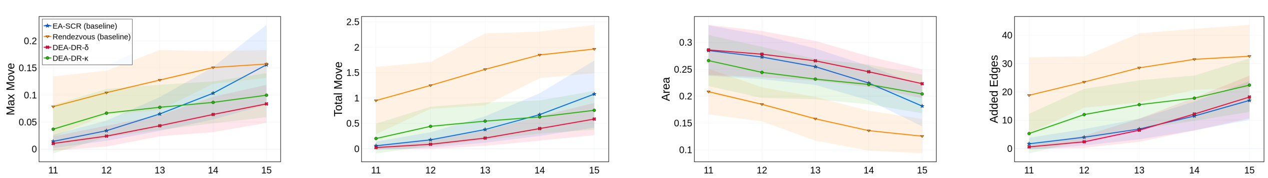


Fig. 6: Experiment on a team of N = 20 robots, with initial K = 10 and varying levels of final node connectivity.

robots, with initial connectivity K=10, which is a relatively dense initial setting. However, contrary to what one might anticipate, the baseline performs better when the gap is small and gradually loses performance. This implies that the density of the initial graph has no effect on the performance, and the 'gap' is also indifferent to the initial graph structure. However, based on empirical data, there appears to be a threshold in this gap beyond which the rendezvous baseline and our distributed approach achieve improved performance. This threshold may be related to the team-size of robots. Further empirical analysis may provide insights into this threshold, which can be useful for determining which algorithm best suits the use case.

The Added Edges metric provides useful insights into network dynamics. As the gap increases, our algorithm improves performance on the movement metrics, but requires more edge additions than the EA-SCR baseline. This observation further reinforces the notion that although the baseline adds fewer edges, the associated robot movements are significantly more costly, highlighting the sub-optimality of the sequential globally-planned approach under larger connectivity gaps.

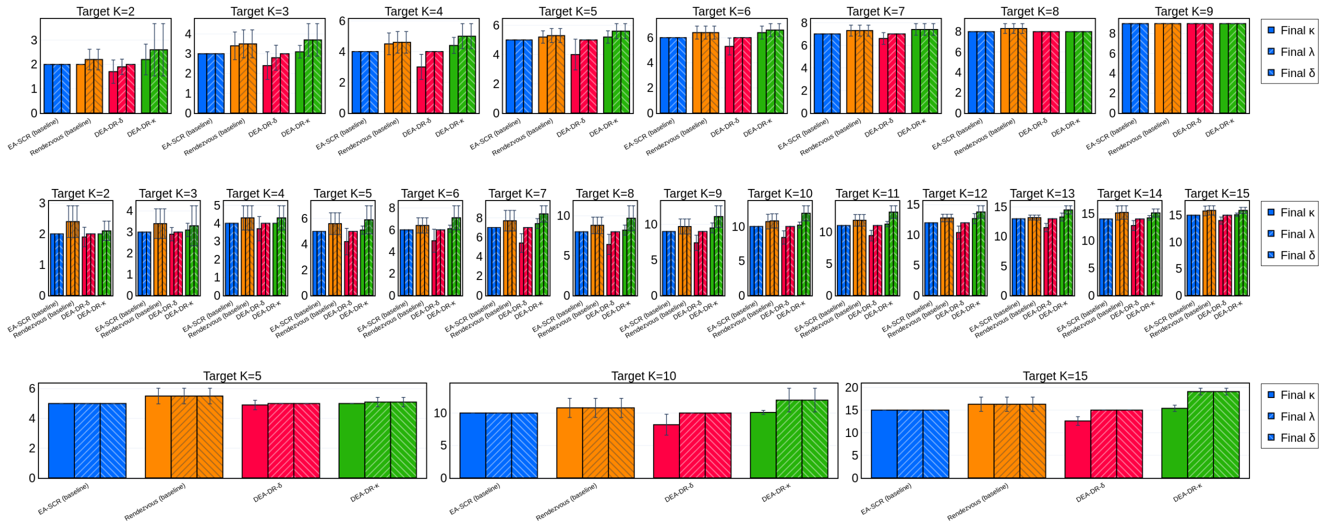


Fig. 7: Final K, λ and δ_{min} representing the final node connectivity, edge connectivity, and minimum degree for robots in different team sizes (*from Fig. 3 experiment*). Top row (N=10), middle row (N=20), and the bottom row (N=30). Initial connectivity level $K = 1$ in all the experiments.

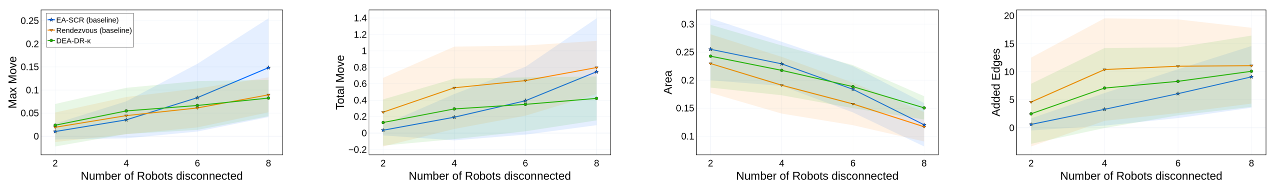


Fig. 8: Experiment on a team of N = 20 robots, with initial K = 10, with simulated failures in n = [2, 4, 6, 8] robots.

TABLE I: Normalized maximum movement for number of robots N and target connectivity K.

Approach	N=20, K=2	N=20, K=3	N=20, K=4	N=30, K=2	N=30, K=3	N=30, K=4	N=40, K=2	N=40, K=3	N=40, K=4
GNN-EASCR [16]	**1.130**	1.110	1.050	**1.060**	1.130	1.070	**1.180**	1.140	1.210
DEA-DR-δ	1.306	**1.081**	0.860	1.343	1.090	0.971	1.271	0.979	0.901
DEA-DR-κ	1.761	1.310	**0.957**	1.343	**1.090**	**0.971**	1.419	**0.979**	**0.901**

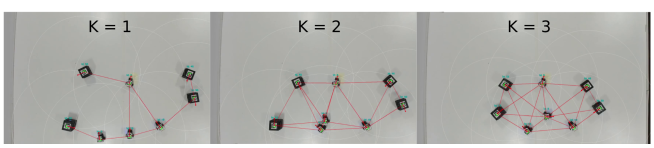


Fig. 9: Real-word experiment showcasing DEA-DR-κ on our in-house swarm robotics testbed with a team of N = 8 robots, sensing radius = 0.5 m.

The final values of node connectivity, edge connectivity, and minimum degree in Fig 7 depict the pattern envisioned by Lemma 1 referring to the relation between these three values. As expected, the DEA-DR-δ variant, which lacks the node connectivity verification step (Alg. 3), has a lower success rate in achieving the desired k-connectivity, but it still comes close. We also observe that the proposed approach utilizes Alg. 3 to rerun the calculations with a higher degree requirement until the target K is achieved.

Alongside this, Shi *et al.* present an aggregation GNN-based distributed variant and compare it against their centralized baseline. Results in Table I present the Max Move metric normalized against the centralized baseline. GNN-EASCR is designed to learn from the centralized algorithm and mimic its behavior using local measurements, but the algorithm does not guarantee success and limits generalizability. In contrast, our approach is purely algorithmic, rendering it deployable on any arbitrary non-trivial graph without the need for task-specific training or adaptation. The results are consistent with the network's observed behavior, and our approach improves performance and surpasses the centralized baseline for larger connectivity gaps.

Formation Restoration Experiment: Preserving the geometric structure of the formation during edge augmentation is a critical performance consideration. Allowing the robots to substantially distort or abandon the formation in restoring connectivity would be detrimental to the intended application. Therefore, it is essential to maintain the overall shape of the formation, in particular, its occupied area, throughout the augmentation process.

To test the restorative capabilities of the network, we perform an additional experiment on a team of N = 20 robots, K = 10, and randomly disconnect [2, 4, 6, 8] number of robots from the network (while ensuring the graph stays connected), the algorithms are tasked with restoring the target K back to 10. The effect of the connectivity gap is once again apparent in the results depicted in Fig. 8. When fewer robots are disconnected from the network, the baseline achieves better performance; however, as the number of failures increases and the resulting connectivity gap widens, our proposed approach demonstrates an increasing performance advantage over the baseline. Note that the DEA-DR-δ variant does not guarantee the desired k-connectedness, as it increases the graph's degree; we exclude this variant from this set of experiments, which focuses on k-connectivity restoration.

A. Real-world robot demonstrations

We also present a hardware demonstration of our proposed approach, implemented to increase the node connectivity of a network. As presented in Fig. 9, a team of 8 robots is set up in a 1-connected formation. The experiment is run on an in-house swarm robotics testbed spanning an area of $3m \times 3m$. The team consists of heterogeneous robots, each with a sensing radius of 0.5 m. As illustrated, the proposed distributed algorithm achieves a 2-connected graph, after which the robots maintain connectivity and further increase it to the target 3-connectedness.

VII. Conclusion

In this paper, we present the first fully distributed solution to the Fast k-Connectivity Restoration (FCR) problem that relies solely on local neighborhood information, removing the need for centralized coordination, making it scalable and robust for large or sparse multi-robot networks. We propose the algorithm *DEA-DR-κ*, which restores k-connectivity in a single step, achieving up to a 58% reduction in total movements compared to centralized methods. We also introduce comparative variants that improve connectivity, based on the minimum degree. This makes the approach closer to a truly distributed scenario, in which knowledge of global node connectivity is unavailable. Extensive simulations show that our methods match or outperform centralized baselines in terms of total travel distance, min-max movement, number of added edges, and formation area, especially as the need for connectivity improvement increases.

References

[1] Y. Rizk, M. Awad, and E. W. Tunstel, "Cooperative heterogeneous multi-robot systems: A survey," *ACM Computing Surveys (CSUR)*, vol. 52, no. 2, pp. 1–31, 2019.

[2] R. R. Murphy, *Disaster robotics*. MIT press, 2017.

[3] I. Kruijff-Korbayová, F. Colas, M. Gianni, F. Pirri, J. de Greeff, K. Hindriks, M. Neerincx, P. Ögren, T. Svoboda, and R. Worst, "Tradr project: Long-term human-robot teaming for robot assisted disaster response," *KI-Künstliche Intelligenz*, vol. 29, pp. 193–201, 2015.

[4] Q. Yang and R. Parasuraman, "Needs-driven heterogeneous multi-robot cooperation in rescue missions," in *2020 ieee international symposium on safety, security, and rescue robotics (ssrr)*. IEEE, 2020, pp. 252–259.

[5] L. Guerrero-Bonilla, A. Prorok, and V. Kumar, "Formations for resilient robot teams," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 841–848, Apr. 2017.

[6] Y. Wang and H. Ishii, "Resilient consensus through event-based communication," *IEEE Control Systems Letters*, vol. 7, no. 1, pp. 471–482, Mar. 2020.

[7] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, "Resilient asymptotic consensus in robust networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 4, pp. 766–781, Apr 2013.

[8] A. Wiktor and S. Rock, "Collaborative multi-robot localization in natural terrain*," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4529–4535.

[9] L. Sabattini, C. Secchi, N. Chopra, and A. Gasparri, "Distributed control of multirobot systems with global connectivity maintenance," *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1326–1332, 2013.

[10] H. Park and S. A. Hutchinson, "Fault-tolerant rendezvous of multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 565–582, Jun. 2017.

[11] A. Sagale, T. K. Tasooji, and R. Parasuraman, "Dcl-sparse: Distributed range-only cooperative localization of multi-robots in noisy and sparse sensing graphs," *arXiv preprint arXiv:2412.14793*, 2024.

[12] N. Maalel, M. Kellil, P. Roux, and A. Bouabdallah, "Fast restoration of connectivity for wireless sensor networks," in *Conference on Internet of Things and Smart Spaces*. Springer, 2012, pp. 401–412.

[13] Z. Mi, Y. Yang, and J. Y. Yang, "Restoring connectivity of mobile robotic sensor networks while avoiding obstacles," *IEEE Sensors Journal*, vol. 15, no. 8, pp. 4640–4650, 2015.

[14] S. Caccamo, R. Parasuraman, L. Freda, M. Gianni, and P. Ögren, "Rcamp: A resilient communication-aware motion planner for mobile robots with autonomous repair of wireless connectivity," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 2010–2017.

[15] M. Ishat-E-Rabban, G. Shi, G. Bonner, and P. Tokekar, "Fast k-connectivity restoration in multi-robot systems for robust communication maintenance," in *International Symposium on Distributed Autonomous Robotic Systems*. Springer, 2024, pp. 553–568.

[16] G. Shi, G. Bonner, P. Tokekar *et al.*, "Fast k-connectivity restoration in multi-robot systems for robust communication maintenance: algorithmic and learning-based solutions," *Autonomous Robots*, vol. 49, no. 4, pp. 1–18, 2025.

[17] Z. A. Dagdeviren, V. K. Akram, O. Dagdeviren, B. Tavli, and H. Yanikomeroglu, "k-connectivity in wireless sensor networks: Overview and future research directions," *IEEE Network*, vol. 37, no. 3, pp. 140–145, May 2023.

[18] P. R. Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff, "A passivity-based decentralized strategy for generalized connectivity maintenance," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 299–323, 2013.

[19] V. K. Akram, O. Dagdeviren, and B. Tavli, "Distributed k-connectivity restoration for fault tolerant wireless sensor and actuator networks: Algorithm design and experimental evaluations," *IEEE Transactions on Reliability*, 2020, early access.

[20] Y. Yuan, H. Yuan, D. W. C. Ho, and L. Guo, "Resilient control of wireless networked control system under denial-of-service attacks: A cross-layer design approach," *IEEE Transactions on Cybernetics*, vol. 50, no. 1, pp. 48–60, Jan. 2020.

[21] K. S. Engin and V. Isler, "Establishing fault-tolerant connectivity of mobile robot networks," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 2, pp. 667–677, Jun. 2021.

[22] J. Li, P. Yi, T. Duan, Z. Zhang, J. Li, Y. Wang, and J. Yu, "Fast connectivity restoration of uav communication networks based on distributed hybrid maddpg and apf algorithm," *Ad Hoc Networks*, vol. 171, p. 103785, 2025.

[23] W. Luo, N. Chakraborty, and K. Sycara, "Minimally disruptive connectivity enhancement for resilient multi-robot teams," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA, 2020, pp. 11 809–11 816.

[24] D. Dolev, "The byzantine generals strike again," *Journal of Algorithms*, vol. 3, no. 1, pp. 14–30, 1982. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0196677482900049

[25] D. B. West *et al.*, *Introduction to graph theory*. Prentice hall Upper Saddle River, 2001, vol. 2.

[26] O. R. Oellermann, L. Beineke, and R. Wilson, "Menger's theorem," in *Topics in Structural Graph Theory*. Cambridge University Press, 2013, pp. 13–39.

[27] B. Bollobas, *Modern Graph Theory*, ser. Graduate Texts in Mathematics. Springer New York, 2013.

[28] R. Parasuraman and B.-C. Min, "Consensus control of distributed robots using direction of arrival of wireless signals," in *Distributed Autonomous Robotic Systems: The 14th International Symposium*. Springer, 2019, pp. 17–34.