

Fast k -connectivity Restoration in Multi-Robot Systems for Robust Communication Maintenance

Md Ishat-E-Rabban¹, Guangyao Shi², Griffin Bonner¹, and Pratap Tokekar¹

¹ University of Maryland, College Park MD 20742, USA

ier,gbonner1,tokekar@umd.edu

² University of Southern California, Los Angeles, CA 90007, USA

shig@usc.edu

Abstract. Maintaining a robust communication network plays an important role in the success of a multi-robot team jointly performing an optimization task. A key characteristic of a robust cooperative multi-robot system is the ability to repair the communication topology in the case of robot failure. In this paper, we focus on the *Fast k -connectivity Restoration* (FCR) problem, which aims to repair a network to make it k -connected with minimum robot movement. Here, a k -connected network refers to a communication topology that cannot be disconnected by removing $k - 1$ nodes. We develop a Quadratically Constrained Program (QCP) formulation of the FCR problem, which provides a way to optimally solve the problem, but cannot handle large instances due to high computational overhead. We therefore present a scalable algorithm, called EA-SCR, for the FCR problem using graph theoretic concepts. By conducting empirical studies, we demonstrate that the EA-SCR algorithm performs within 10% of the optimal while being orders of magnitude faster. We also show that EA-SCR outperforms existing solutions by 30% in terms of the FCR distance metric.

Keywords: multi-robot systems, graph theory, connectivity maintenance

1 Introduction

There is a growing trend in assigning a team of robots to perform coverage, exploration, patrolling, and similar cooperative optimization tasks [1–5]. Compared to a single robot, a multi-robot system can extend the range of executable tasks, enhance task performance, and naturally provide redundancy for the system. Communication plays a key role in the successful deployment of a cooperative multi-robot system. To improve task performance, it is important to maintain uninterrupted communication among the robots. To ensure continuous information sharing and interaction, the robots in a team must always stay close to one another and maintain a connected communication topology among themselves.

However, maintaining a connected communication network may not always prove adequate for multi-robot systems, because robots may fail, e.g., experience sensor or actuator malfunction, or undergo an adversarial attack [6]. Any

such issue may cause disconnections in the network, and consequently hurt task performance. Most existing works [16–18, 21–25] on preventing such communication failures focus on maintaining a k -connected network among the robots. A k -vertex-connected network is a topology that cannot be disconnected by removal of any combination of $k - 1$ robots. Here, a high value of k translates to enhanced robustness of the communication network, and vice versa.

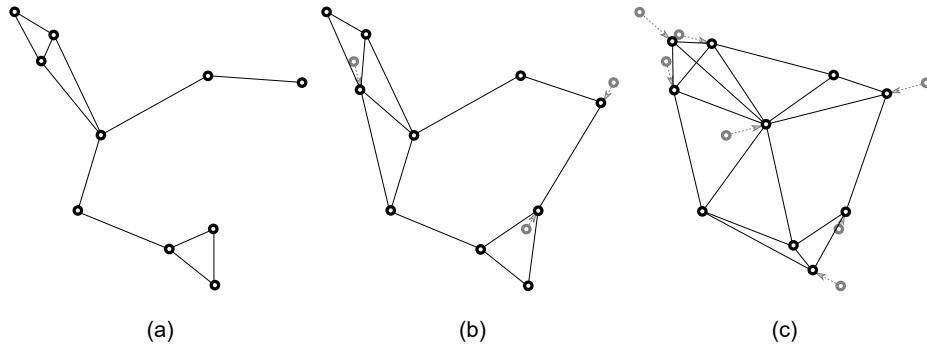


Fig. 1. (a) Initial 1-connected configuration. (b) Robot positions for $k = 2$. (c) Robot positions for $k = 3$. Dashed gray arrows show robot movements from initial positions.

In this paper, we investigate the problem of making a network k -connected while minimizing robot movements. The input to this problem is a set of robot positions and the value of k . Here, we assume a disk communication model with communication radius h . The goal is to find new positions of the robots which form a k -connected topology while minimizing the maximum distance between the previous and new positions of the robots, i.e., the *minmax distance*. We call this problem the *Fast k -connectivity Restoration* (FCR) problem. An example of the FCR problem instance is shown in Fig. 1. The FCR problem and its variants have been extensively studied in literature [16–18, 21, 23]. Some of the above-mentioned variants of the FCR problem minimize different metrics, such as sum of robot movements, number of robots that need to be moved, etc., and some variants are only applicable for specific values of k .

We develop a *Quadratically Constrained Program* (QCP) formulation of the FCR problem, which is based on the idea of multi-commodity network flow [28]. The QCP formulation enables us to solve the FCR problem optimally using a QCP solver. However, the QCP formulation can be used to solve only small instances of the FCR problem due to high computational overhead.

We therefore propose a scalable algorithm, EA-SCR, to solve the FCR problem which requires less computational time than the optimal QCP-based algorithm. The EA-SCR algorithm solves the FCR problem by dividing it into two phases. In the first phase, we find a set of edges that, if augmented to the input network, make the network k -connected, and the maximum cost of the augmented edges is minimized. Here the cost of an edge is proportional to the

distance between the two corresponding robots. We call this *Graph Topology Optimization* (GTO) problem. We solve the GTO problem by first sorting the non-existent edges of the communication graph in increasing order of weight, and then keep adding the edges in order until the graph becomes k -connected. In the second phase, we move the robots in such a way that the edges obtained from the first phase is established, and the maximum distance traveled by a robot is minimized. We call this *Movement Minimization* (MM) problem. To solve the MM problem, we propose a Breadth First Search (BFS) based idea called *cascaded relocation*, which establishes the edges obtained from the first phase without removing any existing edges.

We conduct extensive experimentation to evaluate the performance of our proposed algorithm. The empirical results show that the EA-SCR algorithm generates solutions which are within 10% of the optimal solution and gives 30% lower minmax distance than all existing algorithms [17, 21]. We also test the EA-SCR algorithm in action by performing a hardware experiment using 6 drones.

In summary, we make the following contributions:

- We propose a scalable heuristic algorithm, EA-SCR, for the FCR problem.
- We develop the first QCP formulation of the FCR problem which can be used to solve the FCR problem optimally.
- We conduct extensive experiments which demonstrate that the EA-SCR algorithm performs within 10% of the optimal solution and outperforms all existing solutions by 30% in terms of the minmax distance metric.

2 Related Work

In a broader sense, this work is related to the connectivity problem in networked multi-robot systems. A large portion of research in this field uses algebraic graph theory to maintain the connectivity of the robotic network [7–12]. These works are based on the fact that the second smallest eigenvalue of the Laplacian matrix of a graph, λ_2 , is a measure of how well-connected a graph is. λ_2 is called *algebraic connectivity* of a graph, which is 0 if the graph is not connected, and positive otherwise, with higher values representing better connectivity. Thus a gradient descent based controller can be used to move the robots in a direction that increases λ_2 , and thus enhance the connectivity of a graph. However, algebraic connectivity is not the same as vertex connectivity [13], and hence these methods cannot be used to guarantee k -vertex-connectivity.

A number of works [14–16] study how to minimize movements to form 1-connected networks from arbitrary deployments. Demaine et al. [14] considered the case in which the robots are placed on the vertices of a graph and they can move only along the edges. Anari et al. [15] proposed an algorithm with approximation ratio linear in the number of robots, n . Engin et al. [16] considered the case in which the initial positions of the robots are chosen uniformly at random and they presented an algorithm with $O(\sqrt{n})$ -approximation ratio.

Another body of work [17–20] studies different variants of the 2-connectivity restoration problem. The work most closely related to the FCR problem was

conducted by Basu et al. [17]. In this work, the authors proposed a Block Cut Tree based algorithm to make a 1-connected network 2-connected and their objective was to minimize the sum of the movements of all the robots. Abbasi et al. [18] considered the case in which 2-connectivity is lost due to the removal of one robot and proposed an algorithm that moves minimum number of robots to restore 2-connectivity. Lee et al. [19] considered the same objective as us, but they assumed that additional robots can be used to facilitate the transition to a 2-connected topology. Ramanathan et al. [20] addressed a variant in which the goal is to achieve a 2-connected network while minimizing the maximum level of power assignment. These works are only applicable for 2-connecting a network.

k -connectivity maintenance and restoration in multi-robot systems has been extensively studied in literature [21–25]. This paper is most closely related to the work of Engin et al. [21], who proposed an algorithm for the FCR problem which first forms a $k+1$ -clique at the center of the data-space, and incrementally adds the rest of the robots without compromising k -connectivity. In [22, 23], an FCR variant was considered which allows introducing new robots in the network, and the goal was to achieve k -connectivity by deploying the smallest number of additional robots. Luo et al. [24, 25] considered the problem of minimizing the sum of deviations from a prescribed controller while maintaining k -connectivity.

Distributed algorithms [18, 26, 27] to maintain and restore k -connectivity have been well-studied by researchers. Cornejo et al. [26] and Akram et al. [27] propose distributed algorithms for k -connectivity maintenance. In these methods, the robots rely on local information from up to 2-hop neighbors to test for k -connectivity. Consequently these algorithms require less computation than global and centralized solutions. But, these methods are not guaranteed to return a correct solution and may result in false positives. Abbasi et al. [18] proposed a distributed algorithm to restore 2-connectivity, but their algorithm can only handle the cases in which 2-connectivity is lost due to failure of a single node.

3 Problem Formulation

First, we introduce some graph-theoretic concepts. A graph G is a tuple (V_G, E_G) , where V_G and E_G are the set of vertices and edges of G respectively. A graph G is *connected*, if there exists a path between each pair of vertices in G . A graph G is *k -connected* (or *k -vertex-connected*), if for each $v \subseteq V_G$ with $|v| = k - 1$, the graph $G - v$ is connected. Here, $|.|$ denotes set cardinality and $G - v$ is the graph obtained by removing from G the vertex v and all edges incident on v .

We assume that n robots are operating in an unobstructed 2D or 3D environment. We denote the i^{th} robot by r_i , and the position of r_i by x_i , where x_i is a 2D or 3D point. R represents the set of robots, $R = \{r_1, r_2, \dots, r_n\}$, and X represents the set of positions of the robots, $X = \{x_1, x_2, \dots, x_n\}$.

Given the robot positions X and the communication radius h , we define the *communication graph* $G(X)$ as follows. $G(X)$ is an unweighted graph which has one vertex corresponding to each robot and there exists an edge between a pair

of vertices if the corresponding two robots are within the communication range of each other, i.e., if the Euclidean distance of the two robots is at most h .

Problem 1 (Fast k -connectivity Restoration): We are given a set of positions X of n robots, the communication radius h , and the degree of connectivity k . The FCR problem aims to find new positions $X^* = \{x_1^*, x_2^*, \dots, x_n^*\}$ of the robots such that $G(X^*)$ is k -connected and the maximum movement of the robots is minimized. Mathematically,

$$\min \max_{1 \leq i \leq n} \|x_i^* - x_i\| \quad s.t. \quad G(X^*) \text{ is } k\text{-connected} \quad (1)$$

Here, $\|\cdot\|$ denotes euclidean distance. We assume that the robots have single-integrator dynamics and operate at constant speeds. Therefore, the time a robot takes to transit between two points is proportional to the corresponding distance. It should be noted that we abstract away collision avoidance from the formulation due to the fact that the communication radius of robots is usually much larger than the size of robots [32, 33]. In such cases, the influence of collision avoidance on the controllers can be ignored.

Now we outline the organization of the rest of the paper. In Section 4, we present the QCP formulation of the FCR problem. In Section 5, we describe the EA-SCR algorithm. In Section 6, we present the experimental findings.

4 QCP Formulation

In this section, we present a Quadratically Constrained Program (QCP) formulation, which can be used to find the optimal solution to the FCR problem. The key idea behind the QCP formulation is based on multi-commodity network flow [28]. In the QCP formulation, we use the fact that if a graph is k -connected, there exists at least k vertex-disjoint paths between each pair of vertices. We use the following variables in the QCP formulation.

- Binary variable $e_{i,j}$, where $1 \leq i, j \leq n$, indicates if there exists an edge between robots x_i^* and x_j^* in the communication graph of X^* with respect to the communication radius h , i.e., if x_i^* and x_j^* are within distance h of each other.
- Binary variable $z_{s,d,i,j}$, where $1 \leq s, d, i, j \leq n$, $s \neq d$, indicates if the edge between robots r_s and r_d is included in a vertex-disjoint path from source robot r_s to destination robot r_d .
- x_i^* , where $1 \leq i \leq n$, represents a tuple of real-valued variables (x, y, and optionally z coordinates), which indicates the new position of the i^{th} robot.
- Real valued variable z^* is used to find the maximum movement of the robots.

The QCP formulation is presented below. We use the objective function and the set of constraints in Equation (2) to minimize the maximum movement of the robots. Constraints (3) ensure that if a pair of robots are within the communication radius of each other, the corresponding e variable is set to 1, and vice versa. Here, M is a large positive constant. Note that, we manually set

$e_{i,i} = 0$, for $1 \leq i \leq n$, to indicate that there are no self edges. Constraints (4) enforce that only valid edges (i.e., edges between robots within each other's communication radius) are used to form source-destination paths. Constraints (5) are used to maintain flow conservation. In other words, for each source-destination pair, the outgoing flow should be k greater than the incoming flow for the source vertex, the outgoing flow should be k less than the incoming flow for the destination vertex, and incoming and outgoing flow should be equal for all the other vertices. Constraints (6) ensure vertex-disjointedness, i.e., for each source-destination pair, an internal node is included in at most one path.

$$\min z^* \quad (2)$$

$$\text{s.t. } \|x_i^* - x_i\|^2 \leq z^* \quad \forall 1 \leq i \leq n \quad (2)$$

$$-M e_{i,j} \leq \|x_i^* - x_j^*\|^2 - h^2 \leq M(1 - e_{i,j}) \quad \forall 1 \leq i \neq j \leq n \quad (3)$$

$$z_{s,d,i,j} \leq e_{i,j} \quad \forall 1 \leq s, d, i, j \leq n \quad (4)$$

$\forall 1 \leq s, d, i \leq n$, such that, $s \neq d$,

$$\sum_{1 \leq j \leq n} z_{s,d,i,j} - \sum_{1 \leq j \leq n} z_{s,d,j,i} = \begin{cases} k, & \text{if } i = s \\ -k, & \text{if } i = d \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

$\forall 1 \leq s, d, i \leq n$, such that, $i \neq s, i \neq d, s \neq d$,

$$\sum_{1 \leq j \leq n} z_{s,d,i,j} \leq 1 \quad (6)$$

Note that, the QCP formulation has $O(n^4)$ binary variables. Hence it can be used to solve only very small instances of the FCR problem. To this end, in the following section, we present an algorithm that finds approximate solutions to the FCR problem with less computational overhead.

5 Scalable Algorithm: EA-SCR

In this section, we describe our main contribution, the EA-SCR algorithm, which provides a more scalable alternative compared to a QCP-based solution for the FCR problem. We divide the FCR problem into two sub-problems, namely, the Graph Topology Optimization (GTO) problem, and the Movement Minimization (MM) problem. The GTO problem is to determine which edges to augment to make a graph k -connected and the MM problem is to find out how to realize those edges such that maximum robot movement is minimized.

Now we provide a high-level overview of the EA-SCR algorithm, which has two steps. First we solve the GTO problem using the Edge Augmentation (EA) algorithm (described in Section 5.1) to determine the set of edges E_a to be augmented. Next we solve the MM problem using the Sequential Cascaded Relocation (SCR) algorithm (described in Section 5.2) to determine where to move

Algorithm 1: EA-SCR

Input : Set of robot positions X , Communication radius h ,
Degree of connectivity k

Output: New set of robot positions X^*

```

1 Algorithm EASCR( $X, h, k$ )
2    $E_a \leftarrow \text{EA}(X, h, k)$ 
3   Return SCR( $X, h, E_a$ )

```

the robots such that the edges obtained from the EA algorithm are established while the maximum robot movement is minimized. The pseudo-code of the EA-SCR algorithm is provided in Algorithm 1 above.

5.1 Graph Topology Optimization

In the GTO problem, we are given a set of robot positions X , the communication radius h , and the degree of connectivity k . The GTO problem aims to determine a set of edges E_a which are not present in $G(X)$, i.e. $E_a \subseteq \overline{E_{G(X)}}$, such that the graph $(V_{G(X)}, E_a \cup E_{G(X)})$ is k -connected and the weight of the most costly edge in E_a is minimized. Here, \overline{A} denotes the complement of set A and the weight of an edge between robots r_i and r_j in $\overline{E_{G(X)}}$ is $w(i, j) = \|x_i - x_j\|$. In other words, the GTO problem aims to find a set of edges, E_a , such that augmenting the edges of E_a makes $G(X)$ k -connected and the weight of the costliest edge in E_a is minimized. We call E_a the *Augmentation Set*.

Edge Augmentation Algorithm In this section, we present the *Edge Augmentation* (EA) algorithm (Algorithm 2) which determines the augmentation set. In the EA algorithm, first we sort the edges not present in $G(X)$ in increasing order of weight (Lines 2-4). Next we keep adding the edges in the sorted order one after another to $G(X)$ until $G(X)$ becomes k -connected (Lines 5-9). Finally, we remove those edges from $G(X)$ whose removal does not compromise the k -connectivity of $G(X)$ (Lines 11-17). The proof of correctness of Algorithm 2 follows by construction, as the for loop in Lines 11-17 ensures that all edges contributing to k -connectivity is inserted into the returned edge set E_a .

The running time of the EA algorithm depends on the value of k , because the time complexity of testing k -connectivity of a graph (in Line 6 and Line 13) depends on the value of k . For $k = 1$ and $k = 2$, k -connectivity can be tested in linear time [29] by performing a Depth First Search (DFS). For higher values of k , we can test for k -connectivity using tests for $k - 1$ -connectivity. For example, in the case of $k = 3$, we remove one vertex v from the graph and test for 2-connectivity of the remaining graph using the DFS based method. If, for all v , the graph remains 2-connected after removal of v , the graph is 3-connected and vice versa. Also, generalized k -connectivity test can be performed using a maximum flow based algorithm [30].

Algorithm 2: Edge Augmentation

Input : Set of robot positions X , Communication radius h ,
Degree of connectivity k

Output: Augmentation set E_a

```

1 Algorithm EA( $X, h, k$ )
2   compute  $G(X)$ 
3    $E \leftarrow \overline{E_{G(X)}}$ 
4   sort( $E$ )
5    $count \leftarrow 0$ 
6   while  $G(X)$  is not  $k$ -connected do
7      $G(X)$ .addEdge( $E[count]$ )
8      $count \leftarrow count + 1$ 
9   end
10   $E_a \leftarrow \emptyset$ 
11  for  $i \leftarrow 0$  to  $count - 1$  do
12     $G(X)$ .removeEdge( $E[i]$ )
13    if  $G(X)$  is not  $k$ -connected then
14       $G(X)$ .addEdge( $E[i]$ )
15       $E_a.insert(E[i])$ 
16    end
17  end
18  Return  $E_a$ 
```

5.2 Movement Minimization

In the MM problem, we are given a set of robot positions X , the communication radius h , and the augmentation set E_a . The goal of the MM problem is to find new positions $X^* = \{x_1^*, x_2^*, \dots, x_n^*\}$ of the robots such that $E_a \cup E_{G(X)} \subseteq E_{G(X^*)}$ and the maximum movement of the robots is minimized. Mathematically,

$$\min \max_{1 \leq i \leq n} \|x_i^* - x_i\| \quad \text{s.t. } \|x_i^* - x_j^*\| \leq h, \quad \forall (i, j) \in E_a \cup E_{G(X)}.$$

In other words, the MM problem aims to move the robots in such a way that the existing edges of $G(X)$ are retained, and additionally, each pair of robots in E_a comes within the communication radius of each other, while the maximum movement of the robots is minimized.

Sequential Cascaded Relocation Algorithm In this section, we present the *Sequential Cascaded Relocation* (SCR) algorithm (Algorithm 3) for the MM problem. The SCR algorithm is based on the idea of *Cascaded Relocation* (CR). A CR refers to moving one vertex of a graph to a new location while retaining the existing edges of the graph. Note that, moving only one vertex may disconnect some of the existing edges of the graph. To ensure that the existing edges are retained, a series of relocations may become necessary, hence the process is named cascaded relocation.

The CR process is described in the Procedure *relocate* in Algorithm 3. Here $\text{relocate}(X, h, i, j, d)$ refers to the movement of the i^{th} robot towards the direction of the j^{th} robot by a distance of d without disconnecting any existing edge of $G(X)$, where X and h have their usual meaning. To achieve this, first we perform a Breadth-First Search on $G(X)$ rooted at the i^{th} robot, r_i (Line 9). Performing BFS provides the graph distance of all the other robots from r_i and the parent of all the other robots on the BFS-tree rooted at r_i . The parent of robot k is denoted as $\text{par}(k)$. Next we move robot i to its goal location (Line 10). Then we consider all the other robots in increasing order of graph distance from r_i (Line 11). If a robot gets disconnects from its parent due to the relocation of the parent, we move the robot minimally towards the parent such that the disconnected edge is reestablished (Line 12-14).

Algorithm 3: Sequential Cascaded Relocation

Input : Set of robot positions X , Communication radius h ,
Augmentation set E_a
Output: New set of robot positions X^*

```

1 Algorithm SCR( $X, h, E_a$ )
2   for  $(i, j) \in E_a$  do
3      $d \leftarrow \|x_i - x_j\| - h$ 
4     relocate( $X, h, i, j, \frac{d}{2}$ )
5     relocate( $X, h, j, i, \frac{d}{2}$ )
6   end
7   Return  $X$ 
8 Procedure relocate( $X, h, i, j, d$ )
9    $G(X).\text{BFS}(i)$ 
10   $x_i = x_i + d \frac{x_j - x_i}{\|x_j - x_i\|}$ 
11  for  $k \in \{1, \dots, n\} \setminus \{i\}$  in increasing order of graph distance from  $i$  do
12     $d' \leftarrow \|x_k - x_{\text{par}(k)}\| - h$ 
13    if  $d' > 0$  then
14       $x_k = x_k + d' \frac{x_{\text{par}(k)} - x_k}{\|x_{\text{par}(k)} - x_k\|}$ 
15    end
16  end

```

Now we describe the SCR algorithm (Algorithm 3), where we augment the edges in the augmentation set to the communication graph by performing a sequence of CRs. In the SCR algorithm, we perform two CRs to connect each edge of the augmentation set. For each edge (i, j) in E_a , first we compute the minimum movement d required to connect r_i and r_j (Line 3). Then we move r_i and r_j towards each other by half the distance of d using the *relocate* method (Line 4-5), and thus establish the edge (i, j) . A demonstration of how one edge is established by the SCR method is shown in Figure 2. The proof of correctness of the SCR algorithm follows by construction. The running time of the *relocate*

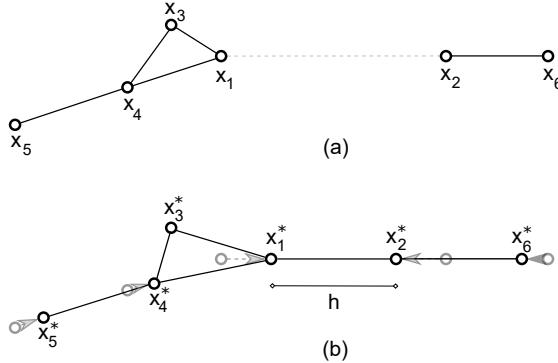


Fig. 2. Before (a) and after (b) connecting edge (1,2). First, r_1 moves towards r_2 by distance $\frac{d}{2}$, where $d = \|x_1 - x_2\| - h$. This movement disconnects edge (1,4), hence r_4 moves minimally towards r_1 to reconnect the edge. Subsequently edge (4,5) gets disconnected and r_5 moves minimally towards r_4 to reconnect the edge. Finally, r_2 moves towards r_1 by distance $\frac{d}{2}$ connecting (1,2). This disconnects edge (2,6), hence r_6 moves towards r_2 to reconnect the edge. Gray arrows in (b) show robot movements.

procedure is linear as BFS takes linear time. Thus the time complexity of the SCR algorithm is $O(|E_a|(n + m))$, where m is the number of edges in $G(X)$.

6 Experiments

In this section, first we describe the experimental setup (Section 6.1), next we present the empirical results (Section 6.2), and finally we discuss a hardware experiment deploying the EA-SCR algorithm (Section 6.3).

6.1 Experimental Setup

Compared Algorithms: We empirically compare the performance of two algorithms proposed in this paper (OPT and EA-SCR) and two existing algorithms (NB [21], and BT [17]). We use a commercial QCP solver, Gurobi [34], to optimally solve the QCP presented in Section 4, which we call the OPT algorithm. The EA-SCR algorithm is described in Section 5.

Engin et al. [21] proposed the *Net-BUILDER* (NB) algorithm to solve the FCR problem. In the NB algorithm, first $k + 1$ robots located closest to the center of the data space are moved using a geometric transformation called homothety to form a $k + 1$ -clique. In the next phase, the remaining robots are added to the clique in increasing order of distance such that k -connectivity is preserved.

The *Block Translation* (BT) algorithm proposed by Basu et al. [17] is based on Block Cut Trees and is applicable for $k = 2$. In each iteration of this algorithm, the leaf blocks of the Block Cut Tree are translated towards its parent node such that the leaf block merges with the parent block. This process is repeated until there is one block left, i.e., the graph becomes 2-connected.

Dataset: We use two types of 2D synthetic datasets: uniform and GMM (Gaussian Mixture Model [31]). First, we generate random 2D points representing the positions of the robots uniformly (or, according to a GMM) for uniform (or, GMM) dataset. Next, we construct the communication graph of the generated points using $h = 1m$. If the communication graph is connected, we add the set of points to our dataset. We do not add disconnected topologies to our dataset, because the BT algorithm requires the graph to be connected. Our experiments show that the performance of the compared algorithms are similar for both datasets. Hence, for brevity, we show the results only for uniform dataset.

Evaluation Metric: We use three metrics to empirically evaluate our proposed algorithms: minmax distance, total distance, and running time. The objective of the FCR problem is to minimize the maximum distance between the previous and new positions of the robots. We call this metric the minmax distance. The total distance is the sum of the movements of all the robots.

Platform: The algorithms are implemented using C++. The experiments are conducted on a core-i7 2.7GHz PC with 16GB RAM, running Microsoft Windows 11.

6.2 Empirical Results

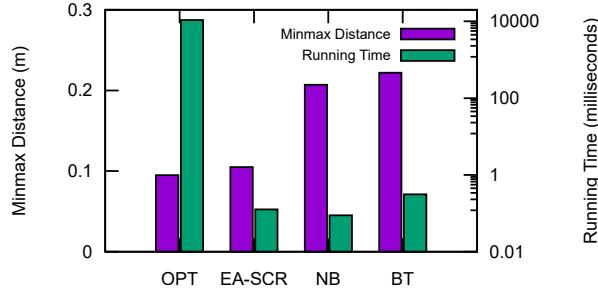


Fig. 3. Comparison with OPT algorithm using $n = 8$ and $k = 2$.

Comparison with OPT: In this experiment, we compare the minmax distance and running time of the discussed algorithms. As the OPT algorithm can be used to solve only small instances of the FCR problem, we use 8 robots in this experiment and the value of k is set to 2. The experimental results in Figure 3 show that the OPT algorithm requires 5 orders of magnitude higher running time than the other algorithms (because of the high complexity of the QCP formulation). In terms of minmax distance, our proposed EA-SCR algorithm performs within 10% of the OPT algorithm and almost 50% better than the existing algorithms (NB and BT). We do not report results for the total distance metric because our QCP formulation optimizes the minmax distance; not the

total distance. In all the experiments, we use a communication radius of $1m$. We perform each experiment 100 times and report the average.

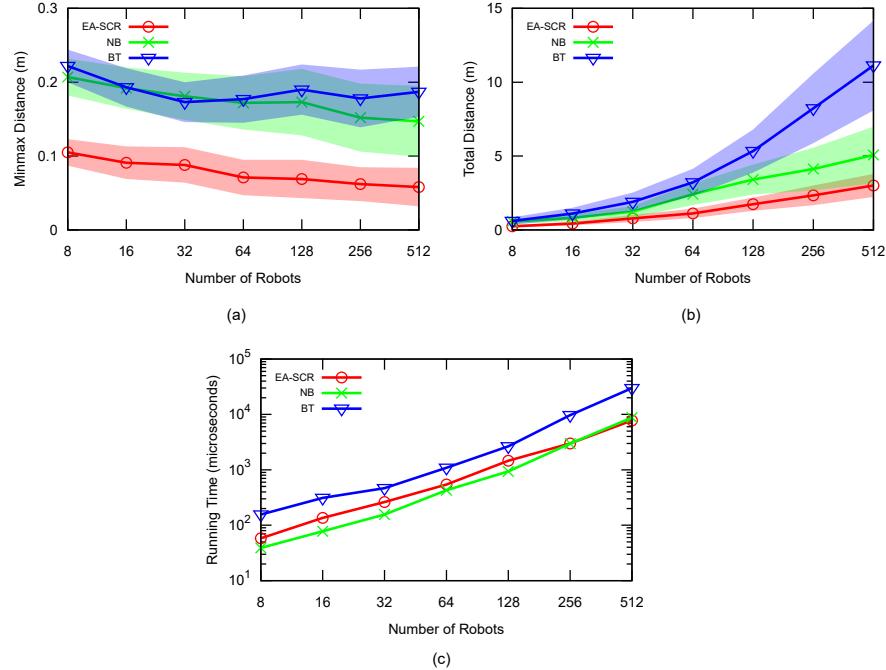


Fig. 4. Comparison of EA-SCR, NB, and BT algorithms for $k = 2$ in terms of (a) minmax distance, (b) total distance, and (c) running time.

Results for $k=2$: In this experiment, we compare the discussed algorithms (except for OPT) in terms of minmax distance, total distance, and running time for $k = 2$. We vary the number of robots from 8 to 512 in increments by a factor of 2. The experimental results in Figure 4 show that our proposed EA-SCR algorithm generates solutions with 50% lower minmax distance and 30% lower total distance compared to the NB and BT algorithms. The performance of NB and BT are equivalent according to minmax distance, and NB performs better than BT in terms of total distance. Here, the shades show standard deviation.

The running time of the EA-SCR algorithm is slightly higher than that of NB for lower values of n and EA-SCR catches up with NB as n grows. Both algorithms perform better than BT in terms of running time.

Results for $k=3$ and $k=4$: In this experiment, we compare EA-SCR and NB algorithms in terms of minmax distance and running time for $k = 3$ and $k = 4$. The experimental results are shown in Figure 5. Similar to the case of $k = 2$, EA-SCR algorithm performs better than NB algorithm in terms of minmax distance. The running time of the EA-SCR algorithm is higher than NB,

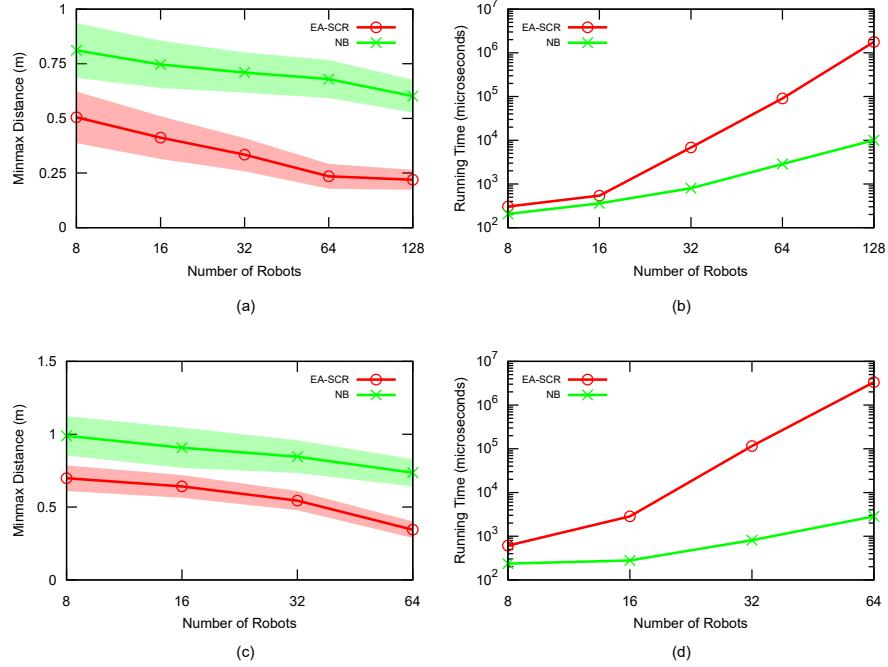


Fig. 5. Comparison of EA-SCR and NB algorithms. (a) $k = 3$. metric: minmax distance. (b) $k = 3$. metric: running time. (c) $k = 4$. metric: minmax distance. (d) $k = 4$. metric: running time.

but still less than a few seconds for teams with 128 robots. The reason behind EA-SCR having a high running time is that testing for k -connectivity for high values of k is computationally expensive as already explained in Section 5.1. Although EA-SCR has a higher running time than NB, EA-SCR gives 30% lower minmax distance compared to NB, thus providing a nice trade-off between running time and solution quality for higher values of k .

In this experiment, we do not compare with BT algorithm as it is only applicable for $k = 2$. Also, for the sake of brevity, we do not present results for the total distance metric. EA-SCR performs better than NB in terms of the total distance metric similar to the results for $k = 2$.

6.3 Hardware Experiments

We test the EA-SCR algorithm deployed in hardware for the Persistent Monitoring (PM) task [35] using a setup similar to [36]. Persistent monitoring is a patrolling task which involves repeatedly visiting of the unoccupied space in the environment. We use six Crazyswarm nano-quadrotors [37] to monitor a $4m \times 4m$ space with $h = 1m$ and $k = 2$. As shown in Fig. 6, the CrazyFlie robots maintain a 2-connected graph while executing the PM task (Fig. 6a). When a robot fails

resulting in loss of 2-connectivity (Fig. 6b), the robots move towards new positions (calculated using EA-SCR algorithm) to regain 2-connectivity (Fig. 6c). 2-connectivity is restored when the robots reach their goals (Fig. 6d). Please refer to Rabban et al. [36] for more details about the experimental setup. A video of the system in operation can be found here: <http://youtu.be/AYcu3Itwqc4>.

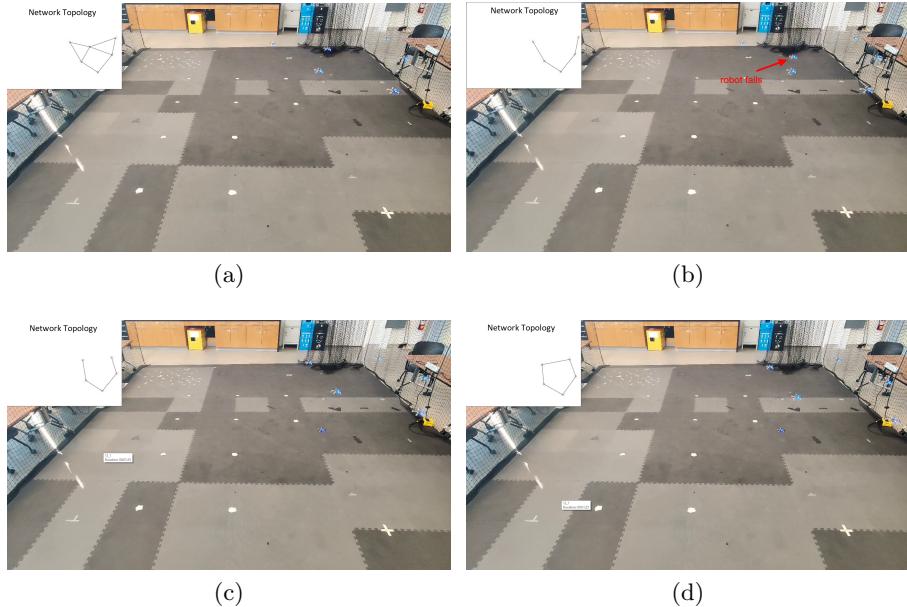


Fig. 6. Six CrazyFlies performing persistent monitoring task. The upper left window in each figure shows the network topology of the team.

7 Conclusion

In this work, we have proposed a scalable algorithm for the FCR problem which has outperformed the existing algorithms. We have also developed a QCP formulation to solve the FCR problem optimally. We have demonstrated the effectiveness of our proposed solution by conducting empirical studies.

In the future, we intend to prove the approximability of the performance of EA-SCR algorithm. We also plan to work on the FCR variant in which the environment contains obstacles. One possible way to deal with obstacles is to discretize the environment and solve a discrete version of FCR. Another direction is to devise the distributed counterpart of the proposed algorithm.

References

1. Y. Rizk, M. Awad, E.W. Tunstel, “Cooperative heterogeneous multi-robot systems: A survey”, *ACM Computing Surveys (CSUR)*, vol. 52(2), pp. 1-31, 2019.
2. Z. Yan, N. Jouandeau, A. Cherif, “A survey and analysis of multi-robot coordination”, *International Journal of Advanced Robotic Systems*, vol. 10(12), pp. 399, 2013.
3. C. Robin, S. Lacroix, “Multi-robot target detection and tracking: taxonomy and survey”, *Autonomous Robots*, vol. 40, pp. 729-760, 2016.
4. L. Huang, M. Zhou, K. Hao, E. Hou, “A survey of multi-robot regular and adversarial patrolling”, *IEEE/CAA Journal of Automatica Sinica*, vol. 4, pp. 894-903, 2019.
5. G. Shi, I.E. Rabban, L. Zhou, P. Tokekar, “Communication-aware multi-robot coordination with submodular maximization”, *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8955-8961). 2021.
6. L. Zhou, P. Tokekar, “Multi-robot coordination and planning in uncertain and adversarial environments”, *Current Robotics Reports*, vol. 2, pp. 147–157, 2021.
7. L. Sabattini, N. Chopra, and C. Secchi, “Decentralized connectivity maintenance for cooperative control of mobile robotic systems,” *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1411–1423, 2013.
8. L. Sabattini, C. Secchi, N. Chopra, A. Gasparri, “Distributed control of multirobot systems with global connectivity maintenance,” *IEEE Transactions on Robotics*, vol. 29(5), 1326-1332, 2013.
9. P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülfhoff, “A passivity-based decentralized strategy for generalized connectivity maintenance,” *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 299–323, 2013.
10. M. M. Zavlanos, M. B. Egerstedt, and G. J. Pappas, “Graph-theoretic connectivity control of mobile robot networks,” *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1525–1540, 2011.
11. M. Minelli, J. Panerati, M. Kaufmann, C. Ghedini, G. Beltrame, L. Sabattini, “Self-optimization of resilient topologies for fallible multi-robots,” *Robotics and Auton. Systems*, vol. 124, p. 103384, 2020.
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, no. 3, pp. 769–787, 2019.
13. S. Kirkland, J. Moliterno, M. Neumann, B. Shader, “On graphs with equal algebraic and vertex connectivity”, *Linear algebra and its applications*, vol. 341(1-3), pp. 45–56, 2002.
14. E. Demaine, M. Hajiaghayi, H. Mahini, A. Sayedi-Roshkhar, S. Oveisgharan, M. Zadimoghaddam, “Minimizing movement.” *ACM Transactions on Algorithms (TALG)*, vol. 5(3), pp. 1–30, 2009.
15. N. Anari, M. Fazli, M. Ghodsi, M. Safari, “Euclidean movement minimization”, *Journal of Combinatorial Optimization*, vol. 32(2), pp. 354–367, 2016.
16. K. Engin, V. Isler, “Minimizing movement to establish the connectivity of randomly deployed robots”, *International Conference on Automated Planning and Scheduling*, vol. 28, pp. 451–458, 2018.
17. P. Basu and J. Redi, “Movement control algorithms for realization of fault-tolerant ad hoc robot networks,” *IEEE network*, vol. 18, no. 4, pp. 36–44, 2004.
18. A. A. Abbasi, M. Younis, and K. Akkaya, “Movement-assisted connectivity restoration in wireless sensor and actor networks,” *IEEE Transactions on parallel and distributed systems*, vol. 20, no. 9, pp. 1366–1379, 2008.

19. S. Lee, M. Younis, and M. Lee, “Connectivity restoration in a partitioned wireless sensor network with assured fault tolerance,” *Ad Hoc Networks*, vol. 24, pp. 1–19, 2015.
20. R. Ramanathan, R. Rosales-Hain, “Topology control of multihop wireless networks using transmit power adjustment”, *IEEE INFOCOM*, vol. 2, pp. 404–413, 2000.
21. S. Engin, V. Isler, “Establishing fault-tolerant connectivity of mobile robot networks” *IEEE Transactions on Control of Network Systems*, vol. 8(2), pp. 667–677, 2021.
22. N. Atay and B. Bayazit, “Mobile wireless sensor network connectivity repair with k-redundancy,” in *Algorithmic Foundation of Robotics VIII*, Springer, pp. 35–49, 2009.
23. J. Bredin, E. Demaine, M. Hajiaghayi, D. Rus, “Deploying sensor networks with guaranteed fault tolerance”, *IEEE/ACM Transaction on Networking*, vol. 18(1), pp. 216–228, 2010.
24. W. Luo and K. Sycara, “Minimum k-connectivity maintenance for robust multi-robot systems,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7370–7377, 2019.
25. W. Luo, N. Chakraborty, K. Sycara, “Minimally disruptive connectivity enhancement for resilient multi-robot teams”, *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11809–11816, 2020.
26. A. Cornejo and N. Lynch, “Fault-tolerance through k-connectivity,” *Workshop on network science and systems issues in multi-robot autonomy: ICRA*, vol. 2, 2010.
27. V. Akram, O. Dagdeviren, B. Tavli, “Distributed k-Connectivity Restoration for Fault Tolerant Wireless Sensor and Actuator Networks: Algorithm Design and Experimental Evaluations”, *IEEE Transactions on Reliability*, vol. 70(3), pp. 1112–1125, 2020.
28. R. E. N. Moraes, C. C. Ribeiro, and C. Duhamel, “Optimal solutions for fault-tolerant topology control in wireless ad hoc networks,” *IEEE Transactions on Wireless Communications*, vol. 8, no. 12, pp. 5970–5981, 2009.
29. J. Hopcroft, R. Tarjan, “Algorithm 447: efficient algorithms for graph manipulation”, *Communications of the ACM*, vol. 16(6), pp. 372–378, 1973.
30. S. Skiena, “The algorithm design manual”, vol. 2, pp. 506.
31. W. Press, “Numerical recipes 3rd edition: The art of scientific computing”, *Cambridge university press*, sec. 16.1, 2007.
32. I. Kuzminikh, A. Snihurov, and A. Carlsson, “Testing of communication range in zigbee technology”, *2017 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*, pp. 133–136, 2017.
33. V. Iordache, R. A. Gheorghiu, M. Minea, and A. C. Cormos, “Field testing of bluetooth and zigbee technologies for vehicle-to-infrastructure applications”, *13th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS)*, pp. 248–251, 2017.
34. L. Gurobi Optimization, “Gurobi optimizer reference manual,” 2020. [Online]. Available: <http://www.gurobi.com>
35. S. Hari, S. Rathinam, S. Darbha, K. Kalyanam, S. Manyam, D. Casbeer, “The generalized persistent monitoring problem”, *American Control Conference (ACC)*, pp. 2783–2788, 2019.
36. M. Ishat-E-Rabban, P. Tokekhar, “Failure-resilient coverage maximization with multiple robots”, *IEEE Robotics and Automation Letters*, vol. 6(2), pp. 3894–3901, 2021.

37. J. Preiss, W. Honig, G. Sukhatme, N. Ayanian, “Crazyswarm: A large nano-quadcopter swarm”, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3299–3304, 2017.