Resilience by Reconfiguration: Exploiting Heterogeneity in Robot Teams

Ragesh K. Ramachandran, James A. Preiss and Gaurav S. Sukhatme

***Abstract*— We propose a method to maintain high resource availability in a networked heterogeneous multi-robot system subject to resource failures. In our model, resources such as sensing and computation are available on robots. The robots are engaged in a joint task using these pooled resources. When a resource on a particular robot becomes unavailable (e.g., a sensor ceases to function due to a failure), the system automatically reconfigures so that the robot continues to have access to this resource by communicating with other robots. Specifically, we consider the problem of selecting edges to be modified in the system's communication graph after a resource failure has occurred. We define a metric that allows us to characterize the quality of the resource distribution in the network represented by the communication graph. Upon a resource becoming unavailable due to failure, we reconfigure the network so that the resource distribution is brought as close to the ideal resource distribution as possible without a large change in the communication cost. Our approach uses mixed integer semi-definite programming to achieve this goal. We also provide a simulated annealing method to compute a spatial formation that satisfies the inter-robot distances imposed by the topology, along with other constraints. Our method can compute a communication topology, spatial formation, and formation change motion planning in a few seconds. We validate our method in simulation and real-robot experiments with a team of seven quadrotors.**

I. Introduction and Related Work

A heterogeneous multi-robot team, where robots have varied sensing, actuation, communication and computational capabilities, is a promising direction to consider when building a resilient team. Such a team can work together by sharing resources between individual robots to perform complex tasks, thereby being resilient to failures of individual robots. For example, when a particular sensor on a robot fails, it may be able to rely on measurements made by a teammate nearby.

If the team is to perform its task by sharing resources between team members, it is desirable to configure the team with a communication topology such that each robot has access to its teammates' resources (e.g., sensing, computation) within some neighborhood. Under distance-limited communication, specifying such a topology restricts the feasible set of the robots' physical locations.

As a motivating example, consider a multi-target tracking application by a team with two kinds of quadrotors as depicted in Figure 1. The smaller quadrotors are equipped with the sensing capabilities required to sense and identify the targets to be tracked. Each large quadrotor has two kinds of computational resource $P1$ and $P2$ (e.g. GPU and CPU). Smaller quadrotors do not own the computational resources to carry out the computations required for tracking the target of interest. These small quadrotors rely on the larger quadrotors to process their sensor data. As shown Figure 1, if resource $P2$ of a quadrotor fails then the communication network must reconfigure so that $P2$ is available to every member in the team in a few hops.

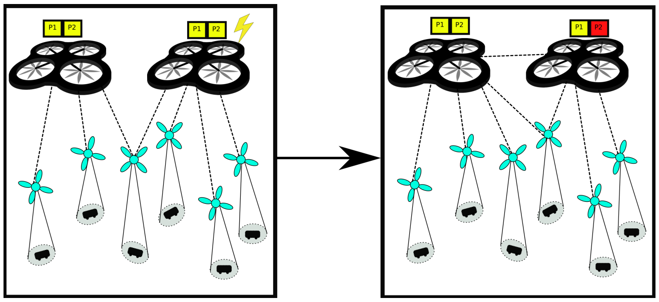


Fig. 1: Reconfiguration of communication network topology after a quadrotor looses a processor, in a multi target tracking scenario (quadrotor image source: Wikipedia).

In this paper, we propose a two-stage method to reduce the impact of a resource failure in a heterogeneous robot team engaged in a task. In response to a resource failure, our method first updates the communication network topology such that the new communication topology is close to the original communication topology and the resource distribution within the network is as close as possible to the ideal resource distribution. An ideal resource distribution is one in which each robot has access to all available resources within a single hop distance. This stage is followed by computing a spatial formation of the robots under which the new topology can be achieved by distance-constrained communications.

In stage one, we use a *mixed integer semi-definite program* (MISDP) formulation to solve the problem of constructing a network topology and generating a pairwise distance between communicating robots. The pairwise distances are constructed such that the communicating robots respect a minimum-distance collision constraint, but are within communication range. Our MISDP objective encourages communicating robots to spread out by maximizing the total pairwise distances between communicating robots.

The pairwise distance output from the MISDP forms the input to the second stage of our approach, where we compute physical locations of the robot team. The communication

*This work was supported in part by the Army Research Laboratory as part of the Distributed and Collaborative Intelligent Systems and Technology (DCIST) Collaborative Research Alliance (CRA).

The authors are with the Department of Computer Science, University of Southern California, Los Angeles, CA 90089, USA rageshku@usc.edu, japreiss@usc.edu, gaurav@usc.edu

topology and desired pairwise distances dictate an optimization objective and constraints on the physical locations of the robots. Optimizing the multi-robot formation is a highly nonconvex, possibly NP-hard problem. We employ a simulated annealing algorithm with penalty functions to escape bad local minima.

In the literature, the problem of resilience in multi-robot system has been studied primarily in the context of constructing communication networks that preclude malicious robots in the network from exerting adverse influence on the network [1], [2]. Researchers have proposed notions of resilience in networks such as *r-robustness* [3] and *p-fraction robustness* [3] to characterize the resilience of the network in achieving consensus in the presence non-cooperative or malicious agents, in terms of network connectivity. In [4], the authors introduce algorithms for building resilient robot networks using this characterization of resilience. Analogous to our work, Hausman *et al.* [5] also develops a notion for reasoning about the quality of networks neighboring to a given network. Since [5] focuses on target tracking applications, [5] compares sensing network topologies in terms of their ability to reduce the expected future uncertainty of the target position.

In this work, we take a different approach to resilience by posing it as a resource sharing problem in a heterogeneous robot team. By doing so, we make the following three contributions:

- We pose the problem of resilience in a heterogeneous robot team in the context of inter-robot resource sharing. As resources become unavailable on robots due to failures, the system reconfigures to bring resources back into communication range as much as possible. We introduce the *inefficacy matrix* – a description of how far is the resource distribution in a configuration is from the ideal resource distribution – and exploit it to drive network reconfiguration.
- We formally pose and solve the configuration generation problem which generates a new communication topology and pairwise distances in response to a failure.
- We give a solution to the formation synthesis problem which generates appropriate values for robot positions satisfying the new configuration.

II. Preliminaries and Notation

For any positive integer $z \in \mathbb{Z}^+$, $[z]$ denotes the set $\{1,2,\cdots,z\}$. We use $\|\cdot\|$ to denote the standard Euclidean 2-norm and the induced 2-norm for vectors and matrices respectively. $\|\mathbf{M}\|_F$ denotes the Frobenius norm of a matrix $\mathbf{M} \in \mathbb{R}^{m_1 \times m_2}$, $\|\mathbf{M}\|_F \triangleq \sqrt{trace(\mathbf{M}^T\mathbf{M})}$. If we arrange the singular values of matrix $\mathbf{M}$ in nondecreasing order, then the i^{th} singular value of $\mathbf{M}$ is denoted by $\sigma_i\{\mathbf{M}\}$. The nuclear norm (trace norm) of $\mathbf{M}$, defined as $\sum_{i=1}^{\min\{m_1,m_2\}} \sigma_i\{\mathbf{M}\}$, is denoted by $\|\mathbf{M}\|_*$. We use $\mathbf{1}^{m_1}$ and $\mathbf{1}^{m_1 \times m_2}$ to represent a vector and matrix of ones of appropriate dimensions, respectively. Similarly, $\mathbf{0}^{m_1}$ and $\mathbf{0}^{m_1 \times m_2}$ denote a vector and matrix of zeros respectively. For any vector $T \in \mathbb{R}^{m_1}$, $Diag(T)$ denotes a matrix with the elements of T along its diagonal. Also, $diag(\mathbf{M})$ outputs a vector which contains the diagonal entries of matrix $\mathbf{M}$ as its elements. $[\mathbf{M}]_{i,j}$ denotes the i,j entry of $\mathbf{M}$. Finally, $\mathscr{S}_+^m$ denotes the space of $m \times m$ symmetric positive semi-definite matrices.

A weighted undirected graph $\mathscr{G}$ is defined by the triplet $(\mathscr{V}, \mathscr{E} \subseteq \mathscr{V} \times \mathscr{V}, \mathbf{A} \in \mathbb{R}_{\geq 0}^{|\mathscr{V}| \times |\mathscr{V}|})$, where $\mathbf{A}$ is the weighted adjacency matrix of the graph. Also, $\overline{\mathscr{E}} = (\mathscr{V} \times \mathscr{V}) \setminus \mathscr{E}$ denotes the edge complement of $\mathscr{G}$. The *graph Laplacian matrix* $\mathbf{L}$ of $\mathscr{E}$ can be computed as

$$\mathbf{L} = Diag(\mathbf{A} \cdot \mathbf{1}^{|\mathscr{V}|}) - \mathbf{A}. \tag{1}$$

We list a few properties of a graph Laplacian matrix used in this article [6]:

$$\mathbf{L} \cdot \mathbf{1}^{|\mathscr{V}|} = \mathbf{0}^{|\mathscr{V}|} \tag{2}$$

$$Trace(\mathbf{L}) = \sum_{1 \leq i < j \leq |\mathscr{V}|} [\mathbf{A}]_{i,j}. \tag{3}$$

III. Problem Statement

We consider a team of n heterogeneous robots labeled sequentially as $[n]$, equipped with different types of capabilities or resources (e.g. sensors, memory, actuators, etc.). The team is assigned with a task of interest. For ease of identification, we denote the robot with label i as i_n. Also, let X_i denote the position vector $[x_i, y_i, z_i]^T \in \mathbb{R}^3$ of i_n. A robot is equipped with at most r distinct types of capabilities/resources. In this paper, we do not distinguish between resources and capabilities of a robot and henceforth, we refer to resources and capabilities interchangeably. The set $[r]$ contains labels of the resources available within in the heterogeneous team.

We assert that all r resources are essential to perform the assigned task. In other words, the team becomes incapable of performing the task of interest if even a single resource among the r resources is unavailable within the team. We assume that each robot can localize itself accurately in the environment. Further, we assume that the robots never lose their localization ability and thus, this capability is not included in the list of r capabilities/resources addressed above. Finally, we assume that the communication graph (defined next) always has edge connectivity of at least 2, i.e., a single failure of a communication link during the task does not partition the communication graph.

We model the communication network using a dynamic undirected graph $\mathscr{G} = (\mathscr{V} = [n], \mathscr{E} \subseteq \mathscr{V} \times \mathscr{V})$, where the edges represent pairs of communicating robots. Based on the communication graph $\mathscr{G}$, we define its closed adjacency matrix $\bar{\mathbf{A}}$[7] as follows:

$$[\bar{\mathbf{A}}]_{i,j} = \begin{cases} 1 & \text{if } (i,j) \in \mathscr{E} \text{ or } (j,i) \in \mathscr{E} \text{ or } i = j \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

In addition, we define the neighborhood distance matrix $\mathbf{D}$, such that,

$$[\mathbf{D}]_{i,j} = \begin{cases} \|X_i - X_j\| & \text{if } (i,j) \in \mathscr{E} \text{ or } (j,i) \in \mathscr{E} \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

For brevity, we also use d_{ij} to denote $[\mathbf{D}]_{i,j}$. Next we define a resilience metric which is a function of the communication network topology and the resource distribution in the robot team. As in [7], we define a binary matrix $\{0,1\}^{n\times r}$, which we refer to as the *resource matrix* and denote by Γ_r. The entries of Γ_r are computed as follows,

$$[\Gamma_r]_{i,j} = \begin{cases} 1 & i_n \text{ has resource } j \in [r] \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

We term the tuple $(\mathscr{G}, \mathbf{D}, \Gamma_r)$ as a *configuration* of the heterogeneous multi-robot team and denote it by $\mathscr{C}$. We define the *inefficacy matrix* $\mathbf{V}$ of a configuration $\mathscr{C}$ as

$$\mathbf{V}(\bar{\mathbf{A}}, \Gamma) = n \cdot \mathbf{1}^{n\times r} - \bar{\mathbf{A}}\Gamma_r. \tag{7}$$

Finally, we define the key quantity used to measure the inability of a heterogeneous multi-robot team with configuration $\mathscr{C}$ to perform the task - the *task inefficacy* of a heterogeneous multi-robot team. The task inefficacy of a heterogeneous multi-robot team with configuration $\mathscr{C}$ is computed as $\|\mathbf{V}\|_*$.

Next, we examine $\mathbf{V}$ to understand why $\|\mathbf{V}\|_*$ is a reasonable metric to quantify the task inefficacy of $\mathscr{C}$. The product matrix $\bar{\mathbf{A}} \cdot \Gamma_r$ encodes information regarding the distribution of resources within the heterogeneous multi-robot team. Particularly, $[\bar{\mathbf{A}} \cdot \Gamma_r]_{i,j}$ describes the exact number of robots in the neighborhood of robot i_n, including itself, with resource j. This matrix becomes $n \cdot \mathbf{1}^{n\times r}$ if each robot is equipped with every resource and the communication graph is fully connected, resulting in the most task-efficient configuration. The inefficacy matrix thus describes how far a configuration is from the ideal configuration. Therefore, we minimize a metric which measures the distance between these matrices with a suitable norm. We found that the *nuclear norm* [8] of the inefficacy matrix works well for our problem and we employ it here. We may explore other matrix norms in future work. When it improves clarity, we will also refer to maximizing the *task efficacy* instead of minimizing the task inefficacy.

We refer to a resource matrix Γ_r as a *feasible resource matrix*, if $\Gamma_r^T \mathbf{1}^n > \mathbf{1}^r$, where $>$ is applied elementwise. We term a resource matrix where $\Gamma_r^T \mathbf{1}^n \ngtr \mathbf{1}^r$ as *infeasible*. Within our framework, any heterogeneous robot team having an infeasible resource matrix lacks the ability to perform the overall task. In this paper, a *resource failure* is a function that consumes a resource matrix (excluding the $n \times r$ matrix of zeros) and generates a resource matrix by setting a random nonzero entry of the consumed resource matrix to zero. A *tolerable resource failure* maps a feasible resource matrix to another feasible resource matrix, while a *catastrophic resource failure* maps a feasible resource matrix to an infeasible resource matrix. Since a catastrophic resource failure renders the team incapable of performing the task, we consider resilience only in the context of tolerable resource failures.

Consider a sequence of countably infinite resource failures $\mathscr{F} = [f_1, f_2, \cdots f_\infty]$ acting on a feasible resource matrix Γ_r sequentially. Now, let $\mathscr{F}_{n_f} = [f_1, f_2, \cdots, f_k, \cdots, f_{n_f}]$ be the first $n_f \in \mathbb{Z}^+$ tolerable resource failures in $\mathscr{F}$, such that $f_{n_f+1} \in \mathscr{F}$ is a catastrophic failure. The k^{th} tolerable resource failure in $\mathscr{F}_{n_f}$ is denoted as f_k. We denote $\Gamma_r[k]$ as the resultant resource matrix after first k tolerable resource failures in $\mathscr{F}_{n_f}$ acted on Γ_r. Mathematically, $\Gamma_r[k] = f_k(f_{k-1}(f_{k-2}(\cdots f_1(\Gamma_r))))$. Also, $\Gamma_r[k]$ can be recursively defined as $\Gamma_r[k] = f_k(\Gamma_r[k-1])$, $\Gamma_r[0] = \Gamma_r$. We term $\mathscr{C}[k-1]$ as the configuration of the heterogeneous multi-robot team before f_k occurred.

We are now in a position to formally define the two main problems addressed in this paper:

Problem 3.1: **Configuration generation:** Given a tolerable resource failure f_k, an associated feasible resource matrix $\Gamma_r[k]$ and a heterogeneous multi-robot team configuration $\mathscr{C}[k-1]$ find a new configuration $\mathscr{C}[k]$ such that,

1) $\mathscr{G}[k]$ is a connected graph,
2) $\|n \cdot \mathbf{1}^{n\times r} - \bar{\mathbf{A}}[k]\Gamma_r[k]\|_* < \|n \cdot \mathbf{1}^{n\times r} - \bar{\mathbf{A}}[k-1]\Gamma_r[k]\|_*$
3) $\|\bar{\mathbf{A}}[k] - \bar{\mathbf{A}}[k-1]\|_F^2 \leq ne$, where $ne \in \mathbb{Z}^+$ is a user-chosen parameter specifying the number of edges to be modified in $\bar{A}[k-1]$ to produce $\bar{A}[k]$ and
4) the sum of distances between communicating robots is maximized.

Problem 3.2: **Formation synthesis:** Given a heterogeneous multi-robot team configuration $\mathscr{C}[k]$, generate coordinates $X_1[k], \ldots, X_n[k] \in \mathbb{R}^3$ for the robots that best realize the given configuration. We describe this problem more precisely in Section IV-B.

The condition of graph connectivity in Problem 3.1 is required to ensure cooperative task performance of the heterogeneous multi-robot team. The second condition in Problem 3.1 states that the new configuration should improve the task efficacy of the multi-robot system. The third condition ensures that graph topologies $\mathscr{G}[k-1]$ and $\mathscr{G}[k]$ differ from each other in at most *ne* edges. The final condition helps robots to spread out in space. The following section describes our solution procedure for these two problems.

IV. Procedure

We explain our methods for the solving Problem 3.1 and Problem 3.2 in this section. When a resource failure occurs, information about the failure is transmitted to a central controller. The central controller checks if the resource failure is tolerable or catastrophic. If catastrophic, it commands the robots to return to their base station as the team no longer retains the necessary resources to perform the assigned task. If the resource failure is tolerable, the central controller uses a two-step procedure to generate a new configuration and an associated set of robot coordinates which realizes the generated configuration in three-dimensional space. Figure 2 gives a schematic illustration of our multi-robot reconfiguration strategy in the event of a tolerable resource failure. We refer to these steps as *configuration generation* and *formation synthesis*. Specifically, configuration generation and formation synthesis steps are solutions to Problem 3.1 and Problem 3.2 respectively. The following subsections describe each step in further detail.

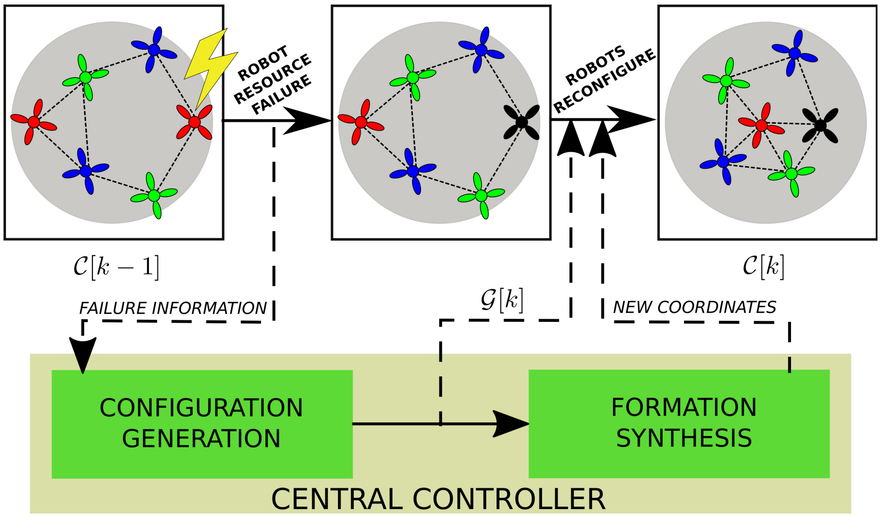


Fig. 2: Basic strategy of our approach. When a resource is lost, *configuration generation* selects edges to modify the communication graph. Then, *formation synthesis* assigns robots to physical locations that support the desired graph topology.

A. *Configuration generation*

We now describe our proposed solution to Problem 3.1. We reformulate the problem as a mixed integer semi-definite program (MISDP) and then solve reformulated MISDP. MISDP formulations are extensively used by researchers for graph topology design problems [9], [10]. We first state our MISDP mathematically; in the following discussion we argue that it captures the facets of Problem 3.1. A MISDP formulation of Problem 3.1 can be written as

$$\underset{\substack{\mathbf{L}\in\mathscr{S}_+^n,\ \mu\in\mathbb{R}_{>0},\\ \Pi\in\{0,1\}^{n\times n}}}{\text{minimize}} \quad \text{Trace}(\mathbf{L}) \tag{8}$$

$$\text{subject to} \quad \mathbf{L}\cdot\mathbf{1}^n = \mathbf{0}^n \tag{9}$$

$$\frac{1}{n}\mathbf{1}^T\mathbf{1}+\mathbf{L}\succeq\mu\mathbf{I} \tag{10}$$

$$diag(\Pi) = \mathbf{1}^n\forall\ i\in[n] \tag{11}$$

$$\Pi = \Pi^T \tag{12}$$

$$[\mathbf{L}]_{i,i}\geq c_{max}\forall\ i\in[n] \tag{13}$$

$$[\mathbf{L}]_{i,j}\geq c_{min}\Pi_{i,j}\forall\ (i,j)\in\mathscr{E} \tag{14}$$

$$[\mathbf{L}]_{i,j}\leq c_{max}\Pi_{i,j}\forall\ (i,j)\in\mathscr{E} \tag{15}$$

$$\|\Pi-\bar{\mathbf{A}}[k-1]\|_F^2\leq ne \tag{16}$$

$$\|\mathbf{V}(\Pi,\Gamma)\|_*<\|\mathbf{V}(\bar{\mathbf{A}}[\mathbf{k}-\mathbf{1}],\Gamma)\|_*. \tag{17}$$

The decision variable $\mathbf{L}$ is a weighted Laplacian of a graph that has the same topology as Π and $\bar{\mathbf{A}}[k]$, and whose weights will be used to compute the desired neighborhood distance matrix $\mathbf{D}[k]$. The range $0 < c_{min} < |\mathbf{L}_{i,j}| < c_{max}$ gives the range of nonzero values of the off-diagonal entries of $\mathbf{L}$. Constraint (9) encodes the property of a graph Laplacian described in Equation (2). It is known [11, Proposition 1] that any undirected graph is connected if and only if $\mathbf{L}+\frac{1}{n}\mathbf{1}^T\mathbf{1}$ is a positive-definite matrix, therefore Constraint (10) ensures that the graph represented by $\mathbf{L}$ must be connected. Constraint (11), Constraint (12) and Constraint **??** models Π as the closed adjacency matrix of a graph. Constraint (15) and Constraint (14) force $\mathbf{L}$ and Π to model the Laplacian matrix and closed adjacency matrix of graphs with same topology, respectively. In other words, the entries of $\mathbf{L}$ are nonzero if and only if the corresponding entries of Π are nonzero. Constraint (16) and Constraint (17) are a direct consequence of the conditions in Problem 3.1

The optimization problem described in Equation (8)-Equation (17) is a MISDP as it involves an integer matrix (Π) and a semi-definite matrix ($\mathbf{L}$) as decision variables. Unlike the integer matrix (binary matrix) Π, whose sole purpose is modeling the closed adjacency matrix of the underlying graph, the Laplacian matrix $\mathbf{L}$ serves two purposes. First, it gives a simple way to incorporate the connectivity constraint of the associated graph in the form of a *linear matrix inequality* [12]. Secondly, we use the off-diagonal entries of $\mathbf{L}$ to compute the neighborhood distance matrix $\mathbf{D}[k]$ associated with the new configuration $\mathscr{C}[k]$. The computation of neighborhood distance matrix $\mathbf{D}[k]$ based on a Laplacian matrix $\mathbf{L}$ is performed as follows:

$$[\mathbf{D}[k]]_{i,j} = \begin{cases} \kappa(|[\mathbf{L}]_{i,j}|-c_{min})+d_{mc} & \text{if}\ \ [\mathbf{L}]_{i,j}<0 \text{ and } i\neq j \\ 0 & \text{otherwise,} \end{cases} \tag{18}$$

where κ (a constant defined as $\kappa := \frac{d_s-d_{mc}}{c_{max}-c_{min}}$, $d_s\in R_{>0}$) is the minimum safe distance between robots and $d_{mc}\in\mathbb{R}_{>0}$ is the minimum distance for non-communicating pairs, such that $d_{mc}>d_s$. Realize from Equation (18) and Figure 3 that the distances between communicating pairs of robots increase as the absolute value of the corresponding Laplacian matrix entries decrease. As a result of Equation (3), maximizing sum of inter-robot distances between communicating robots can be achieved by minimizing the trace of L. Finally, the constraint described in Equation (13) implies that robots with fewer neighboring robots should be close to its neighbors. This enables the robots with weak connectivity to remain

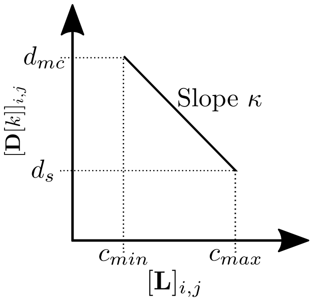


Fig. 3: Map from edge weight $[\mathbf{L}]_{i,j}$ to desired pairwise distance $[\mathbf{D}[k]]_{i,j}$ for nonzero off-diagonal elements in $\mathbf{L}$.

connected with the communication network, even in presence of small external disturbances (e.g. wind). Thus, solving the MISDP described in this section in turn solves Problem 3.1, thereby generating a new configuration $\mathscr{C}[k] = (\mathscr{G}(\bar{\mathbf{A}}[\mathbf{k}] = \Pi)[k], \mathbf{D}[k], \Gamma_r[k])$ for the heterogeneous multi-robot team.

The neighborhood distance matrix $\mathbf{D}[k]$ forms the input to the formation synthesis stage, along with the current formation coordinates to serve as an initial guess in the optimization procedure.

B. Formation synthesis

In this subsection, we introduce a procedure to assign a physical location to each robot based on the desired distances $\mathbf{D}$ specified from the procedure of Section IV-A. We drop the resource failure index $[k]$ for brevity. Let $\{X_i\}_{i=1}^n$, $X_i \in \mathbb{R}^3$ denote the positions of the robots. The primary objective is

$$\underset{X_1,\ldots,X_n}{\text{minimize}} \quad \sum_{(i,j)\in\mathscr{E}} (\|X_i - X_j\|_2 - d_{ij})^2. \tag{19}$$

To demonstrate our method in a smaller indoor experimental space, a minimum-distance constraint is needed, as well as a bounding volume constraint on the whole formation:

$$\begin{aligned}
\text{minimize} \quad & \sum_{(i,j)\in\mathscr{E}} (\|X_i - X_j\|_2 - d_{ij})^2 && && (20)\\
\text{subject to} \quad & d_s \leq \|X_i - X_j\|_2 \leq d_{mc} && \forall\ (i,j) \in \mathscr{E} && (21)\\
& d_{mc} \leq \|X_i - X_j\|_2 && \forall\ (i,j) \in \overline{\mathscr{E}} && (22)\\
& B^{\min} \leq X_i \leq B^{\max} && \forall\ i \in V, && (23)
\end{aligned}$$

where $B^{\min}$, $B^{\max} \in \mathbb{R}^3$ are the minimum and maximum extents of an axis-aligned bounding box, with the operator $\leq$ applied elementwise in (23).

The unconstrained problem (19) is similar to the *graph drawing* [13] and *multidimensional scaling* [14] problems. Graph-drawing variants with constraints similar to (21)–(23) are considered in [15], [16]. In fact, the feasibility problem composed only of the constraints (21) and (23) is a type of *sphere-packing* problem, a class of problems recognized as highly difficult for centuries. The two-dimensional version was shown to be NP-complete by [17], while some other packing variants are not even established as members of NP. Some interesting reviews of packing problems are found in [18], [19].

Algorithm 1 Simulated annealing for formation synthesis

```
procedure SIMULATED-ANNEALING($x_0$ : initial guess)
    $x \leftarrow x_0$
    repeat
        $x' \leftarrow$ PROPOSE($x$)
        if $E(x') < E(x)$ then $x \leftarrow x'$
        else $x \leftarrow x'$ w/ probability $\exp(-T(E(x') - E(x)))$
        $T \leftarrow$ COOLING($T$)
    until stopping criterion met
end procedure
procedure $x' =$ PROPOSE(x)
    sample $j \sim \text{Uniform}([n])$, $d \sim \text{Uniform}([3])$
    sample $\delta \sim \text{Uniform}([-\delta_{\max},\ \delta_{\max}])$
    $x' = x$
    $x'_{j,d} \leftarrow x'_{j,d} + \delta$
end procedure
```

Consequently we cannot hope to optimize (20)-(23) exactly, but in practice, randomized algorithms perform well. In this work, we use a simulated annealing approach with constraints approximated by penalty functions. Other global optimization techniques such as evolutionary algorithms and multi-start gradient methods may be competitive; we leave a more detailed exploration for future work.

1) Simulated annealing: Simulated annealing (SA) [20], [21] is a stochastic global optimization algorithm that can escape poor local minima by occasionally taking random steps that cause an *increase* in the minimization objective. The core algorithmic framework is restated here in Algorithm 1. PROPOSE(x) generates a new value x' in the neighborhood of x, $E(x)$ is the objective or *energy* function, and T is a *temperature* controlling the probability of taking a step that increases the minimization objective, with COOLING(T) implementing a cooling schedule such that T decreases over time. The implementation of PROPOSE(x) in Algorithm 1 is a generic one applicable to many problems with continuous decision variables. We use exponential cooling (COOLING$(T) = \gamma T$ for $0 \ll \gamma < 1$), and our stopping criterion is a fixed number of iterations. These are simple choices but they yielded good solutions in our experiments.

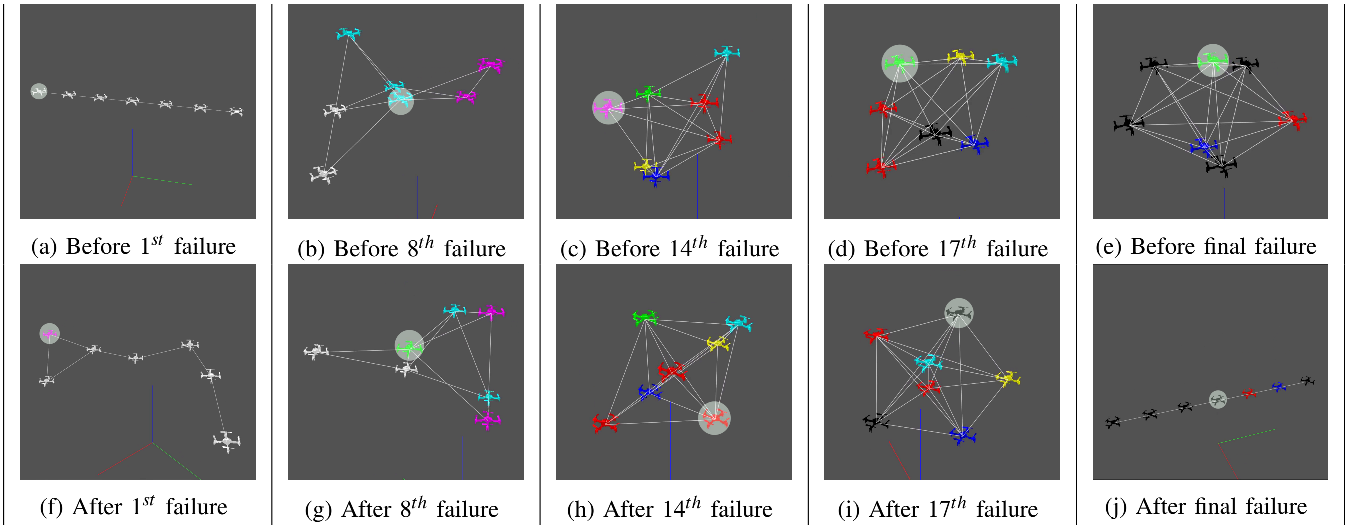
We capture the constraints in the energy function $E(x)$ by adding penalty functions:

$$\begin{aligned}
E(x) = & \sum_{(i,j)\in\mathscr{E}} (\|X_i - X_j\|_2 - d_{ij})^2 + \sum_{(i,j)\in\overline{\mathscr{E}}} p_H(d_{mc} - \|X_i - X_j\|_2)\\
& + \sum_{(i,j)\in\mathscr{E}} \left[p_H(d_s - \|X_i - X_j\|_2) + p_H(\|X_i - X_j\|_2 - d_{mc})\right]\\
& + \sum_{i\in V} \sum_{d\in[3]} p_H(X_{i,d} - B_d^{\max}) + p_H(B_d^{\min} - X_{i,d}),
\end{aligned} \tag{24}$$

where p_H denotes a penalty function satisfying the property

$$\lim_{H\to\infty} p_H(y) = \begin{cases} \infty & : y > 0 \\ 0 & : y < 0. \end{cases} \tag{25}$$

The "hardness" parameter $H > 0$ increases over iterations, analogous to the decay of T. In this work, we use the


(a) Before 1st failure

(b) Before 8th failure

(c) Before 14th failure

(d) Before 17th failure

(e) Before final failure

(f) After 1st failure

(g) After 8th failure

(h) After 14th failure

(i) After 17th failure

(j) After final failure

Fig. 4: Screenshots of a simulation in which seven quadrotors are changing their formation in the event of resource failures. The simulation was visualized using the visualizing tool in our Crazyswarm simulation architecture [22]. The figures in the top row depicts the formation of the quadrotors before the occurrence of a resource failure. The corresponding figures on the bottom portray the formation after 1) the resource failure occurs and is detected, 2) a new communication edge is chosen, 3) the robots move to their new formation. The quadrotor whose resource fails is enveloped using a filled white circle.

exponential penalty $p_H(y) = e^{Hy}$. While the literature on penalty function methods typically includes the property that $p_H(y) = 0$ within the feasible set [23], for example $p_H(y) = \max\{0, Ty^3\}$, we observed solutions with values of (20) closer to optimal using an exponential penalty. We leave further investigation into this observation for the future.

Hyperparameters: In our experiments, we run SA for 20,000 steps. We choose γ such that T decays from 1 to 10^{-8}, and the growth constant for H such that H increases from 1 to 10^3. We let $\delta_{\max} = d_s/10$. Note that tunings of T and H are sensitive to the overall scale of the distances $\mathbf{D}$, d_s, d_{mc} involved in the problem (≈ 1 m in our experiments).

C. Formation change motion planning

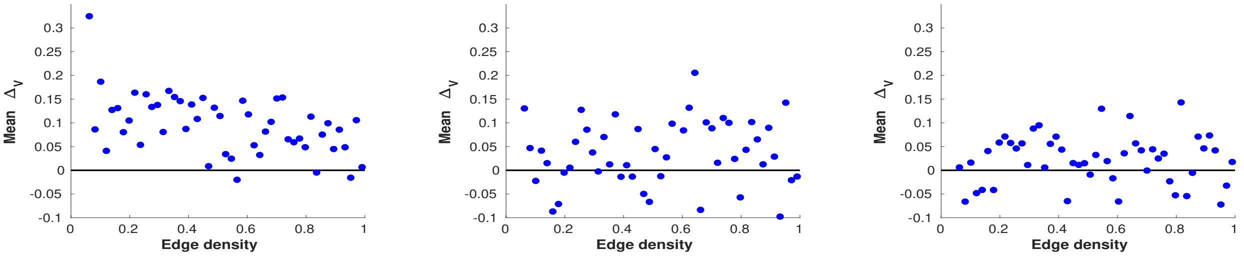
To physically realize the formation change induced by a resource failure, we must ensure that the robots can move between formations without collisions. We employ the method of [22] to plan a set of piecewise polynomial trajectories that transition the robots from configuration $X[k]$ to $X[k+1]$ safely. This method is appropriate for quadrotors in three-dimensional environments with obstacles. We emphasize that other multi-robot motion planning algorithms can be used as needed depending on the type of robot, the task, environment map representation, and so on.

V. Simulation Results

We conducted various simulation experiments to validate our approach. For each simulation experiment, the MISDP problem presented in Section IV-A was solved using MATLAB with YALMIP [24] as the optimization problem modeling interface and SeDuMi [25] as the semi-definite programming solver. We also implemented the simulated annealing technique for formation synthesis (Section IV-B) in MATLAB. Parameters used for the simulations were closely matched with corresponding ones used for the experiments with physical robots described in Section VI. In addition, we choose $c_{min} = d_s$, $c_{max} = d_{mc}$ and $ne = 2$.

For the purpose of illustration, we present results from one simulation experiment. Screenshots of various resource failure instances and the corresponding reconfigurations are presented in Figure 4. This experiment was conducted using seven simulated quadrotors, each having three resources at the start of the experiment. As shown in Figure 4a, the quadrotors' initial positions are predefined such that their communication graph has a line graph topology. A total of 19 resource failures were simulated before the resource matrix of the team became infeasible. A 3-bit RGB color coding system is used to represent resources present in each quadrotor. For example, a white color indicates all three resources, since RGB color for white is $(1,1,1)$. Similarly, if a quadrotor is rendered in yellow, then $(1,1,0)$ color code of yellow represents that the quadrotor possesses all the resources except for the third. Consequently, a quadrotor rendered in black denotes that it has lost all its resources. The multimedia attachment submitted with this manuscript shows the video of this simulation.

In addition, we compared our approach against a strategy in which the robot that suffered from a resource failure connects itself to another random non-communicating robot. We refer to this strategy as *random edge strategy*. For comparison of both strategies, we randomly generated connected graphs with n between 3 and 30. A feasible resource matrix with r between 3 and 20 was constructed based on a predefined *resource percentage* p_r for each randomly generated graph. For a given p_r, we construct a feasible resource matrix by randomly selecting $\lceil \frac{p_r * n * r}{100} \rceil$



(a) $p_r = 20\%$ (b) $p_r = 50\%$ (c) $p_r = 80\%$

Fig. 5: Comparison of our method against random edge selection strategy in reducing the task inefficacy of the heterogeneous multi-robot team (Section V). Our method performs significantly better in low resource availability ($p_r = 20\%$) scenarios.

unique locations from $\mathbf{0}^{n\times r}$ and setting them to 1, such that, the resultant resource matrix is feasible. We define the *edge density* of a graph with n vertices as the ratio of the number of edges in the graph to the number of edges in a full connected graph with n vertices. Finally, we denote $\Delta_{\mathbf{V}}$ as the difference in task efficacy of the new configuration resulting from the random edge addition strategy and our configuration generation strategy. We generated 1000 random connected graphs and 1000 feasible resource matrices, for each $p_r \in \{20, 50, 80\}$. Figure 5 depicts the results from these simulations.

To generate the plots in Figure 5, we divide the interval (0,1] into 50 bins of equal size. The mean $\Delta_{\mathbf{V}}$ of graphs belonging to each bin were computed. Finally, the mean $\Delta_{\mathbf{V}}$ of each bin versus the midpoint of the bin interval were plotted. Note that $\Delta_{\mathbf{V}} > 0$ implies that our approach has increased the task efficacy more than the random edge strategy. Our strategy is designed to improve task efficacy of the team, so it is bound to improve this quantity for any configuration of interest. However, from Equation (7) it is also clear that adding a random edge can not *hurt* task efficacy. Since our MISDP uses constraints to enforce that the task efficacy must be improved but does not explicitly maximize the improvement, it is worthwhile to examine how much is gained in comparison with the random edge strategy. From the scatter plot Figure 5a, since most points lie above zero, we conclude that our strategy increases the task efficacy of the multi-robot team considerably in comparison to random edge strategy, even when the resource percentage is low. Again in Figure 5b, we observe a similar trend in the plot with $p_r = 50\%$. Finally, in Figure 5c, we do not see much difference between the number of points that lie above and below zero. This is expected behavior, since when resource availability in the system is high, the topology of the communication graph has less effect on the task efficacy of the multi-robot team.

VI. Experimental Results

We additionally demonstrate our approach using the Crazyswarm multi-quadrotor experimental platform [26] with a team of seven small quadrotors. We execute the same sequence shown in the simulator screenshots in Figure 4. This experiment illustrates that the formation synthesis and trajectory planning stages of our method produce results that satisfy the collision and kinodynamic constraints of a multi-quadrotor team in the real world.

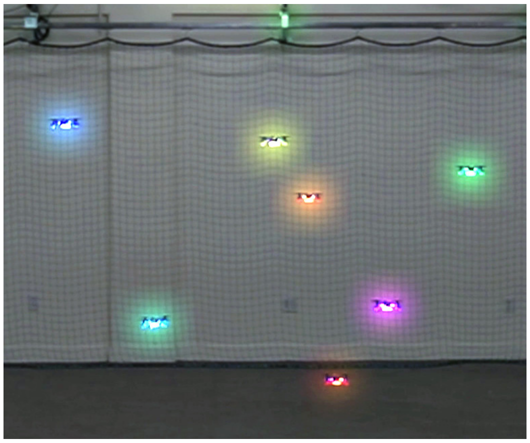


Fig. 6: Real-robot experiment with seven quadrotors. A color glow effect is added to show the resources of each agent. Each of the three resources is mapped to one of the RGB color channels. A video of this experiment is included in the supplementary material.

In our physical experiments, due to programming language interoperability issues we precomputed the sequence of configurations, formations, and multi-robot trajectory plans. However the running time of our method is fast enough for real-time use: in our example problem with seven robots, average computation time was 0.66 s for configuration generation, 2.43 s for formation generation, and 0.15 s for motion planning, resulting in a total of 3.24 s. We also note that our implementation of simulated annealing neglects some opportunities for performance optimization in evaluating $E(x)$ after updating a single coordinate, and we believe a well-tuned implementation could be significantly faster.

VII. Conclusions

In this paper, we described a novel method that enables a heterogeneous team of robots performing a task to reconfigure themselves to a new formation in the event of a robot resource failure. This new formation reduces the impact of resource failure in the multi-robot team by shifting the robots to a new configuration such that the resultant configuration is closer to an 'ideal' configuration without adding much communication cost (number of edges) to the network. Our method has two parts. The first, which is posed as a MISDP, generates the new configuration in the form of a communication graph and the inter-robot distance between the communicating robots. The second part, which is optimized with simulated annealing, computes the robots' spatial coordinates to realize the configuration generated in the previous part in three-dimensional space. We validated our strategy through simulations and demonstrated it using an experiment involving multiple quadrotors. Future work involves incorporating explicit models of resources, namely sensing models, actuation models, and internal computation models, into the strategy.

References

[1] H. Zhang and S. Sundaram, "Robustness of information diffusion algorithms to locally bounded adversaries," in *2012 American Control Conference (ACC)*. IEEE, 2012, pp. 5855–5861.

[2] H. Zhang, E. Fata, and S. Sundaram, "A notion of robustness in complex networks," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 3, pp. 310–320, Sep. 2015.

[3] 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, April 2013.

[4] 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, April 2017.

[5] K. Hausman, J. Müller, A. Hariharan, N. Ayanian, and G. S. Sukhatme, "Cooperative multi-robot control for target tracking with onboard sensing," *The International Journal of Robotics Research*, vol. 34, no. 13, pp. 1660–1677, 2015.

[6] C. Godsil and G. Royle, *Algebraic Graph Theory*, ser. Graduate Texts in Mathematics. New York: Springer-Verlag, 2001, vol. 207.

[7] W. Abbas and M. Egerstedt, "Characterizing heterogeneity in cooperative networks from a resource distribution view-point," *Communications in Information and Systems*, no. 1, pp. 1036–1041, 11 2014.

[8] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM review*, vol. 52, no. 3, pp. 471–501, 2010.

[9] M. Rafiee and A. M. Bayen, "Optimal network topology design in multi-agent systems for efficient average consensus," in *49th IEEE Conference on Decision and Control (CDC)*. IEEE, 2010, pp. 3877–3883.

[10] D. Xue, A. Gusrialdi, and S. Hirche, "A distributed strategy for near-optimal network topology design," in *21st International Symposium on Mathematical Theory of Networked and Systems (MTNS 2014)*, 2014.

[11] M. Sundin, A. Venkitaraman, M. Jansson, and S. Chatterjee, "A connectedness constraint for learning sparse graphs," in *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 151–155.

[12] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. Siam, 1994, vol. 15.

[13] J. X. Zheng, S. Pawar, and D. F. M. Goodman, "Graph drawing by stochastic gradient descent," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2018.

[14] M. Klimenta and U. Brandes, "Graph drawing by classical multidimensional scaling: new perspectives," in *International Symposium on Graph Drawing*. Springer, 2012, pp. 55–66.

[15] R. Davidson and D. Harel, "Drawing graphs nicely using simulated annealing," *ACM Trans. Graph.*, vol. 15, no. 4, pp. 301–331, 1996.

[16] T. Dwyer, "Scalable, versatile and simple constrained graph layout," *Comput. Graph. Forum*, vol. 28, no. 3, pp. 991–998, 2009.

[17] E. D. Demaine, S. P. Fekete, and R. J. Lang, "Circle packing for origami design is hard," *CoRR*, vol. abs/1008.1224, 2010.

[18] H. Alt, "Computational aspects of packing problems," *Bulletin of the EATCS*, vol. 118, 2016.

[19] M. Hifi and R. M'Hallah, "A literature review on circle and sphere packing problems: Models and methodologies," *Adv. Operations Research*, vol. 2009, pp. 150 624:1–150 624:22, 2009.

[20] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[21] D. Henderson, S. H. Jacobson, and A. W. Johnson, "The theory and practice of simulated annealing," in *Handbook of metaheuristics*. Springer, 2003, pp. 287–319.

[22] W. Hönig, J. A. Preiss, T. K. S. Kumar, G. S. Sukhatme, and N. Ayanian, "Trajectory planning for quadrotor swarms," *IEEE Trans. Robotics*, vol. 34, no. 4, pp. 856–869, 2018.

[23] D. G. Luenberger, Y. Ye, *et al.*, *Linear and nonlinear programming*. Springer, 1984, vol. 2.

[24] J. Lofberg, "Yalmip : a toolbox for modeling and optimization in matlab," in *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)*, Sep. 2004, pp. 284–289.

[25] J. F. Sturm, "Using SeDuMi 1.02, a matlab toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11, no. 1-4, pp. 625–653, jan 1999.

[26] J. A. Preiss*, W. Hönig*, G. S. Sukhatme, and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3299–3304.