ARL-TR-10113 ● JUNE 2025

Connectivity Lifting for Multi-Agent Systems

by Moshe Hamaoui

NOTICES

Disclaimers

ARL-TR-10113 ● June 2025

Connectivity Lifting for Multi-Agent Systems

Moshe Hamaoui
DEVCOM Army Research Laboratory

REPORT DOCUMENTATION PAGE

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
June 2025	Technical Report	3 June 2024	30 September 2024

4. TITLE AND SUBTITLE
Connectivity Lifting for Multi-Agent Systems

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**

6. AUTHOR(S)
Moshe Hamaoui

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
DEVCOM Army Research Laboratory ATTN: FCDD-RLA-WE Aberdeen Proving Ground, MD 21005	ARL-TR-10113

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES
ORCID: Moshe Hamaoui, 0000-0002-4851-4823

14. ABSTRACT
Enhancing network connectivity in multi-agent systems, particularly for airborne platforms with limited communication ranges, is a crucial challenge in designing resilient teams of autonomous collaborating agents. This report formulates and addresses the Multi-Agent System Connectivity Augmentation problem that seeks to efficiently increase network connectivity subject to the agents' motion and reachability constraints. A two-phased approach is proposed. First, we identify feasible edge sets that meet a specified vertex-connectivity threshold, employing cardinality bounds and pruning techniques to reduce computational complexity. Second, we optimize the spatial arrangement of agents by embedding these edge sets into 3D space, minimizing displacement using second-order cone programming. Numerical experiments demonstrate that the proposed method significantly improves network resilience with minimal agent movement. Furthermore, the application of practical heuristics, such as maintaining initial communication links, achieves substantial reductions in computational time with little or no loss of solution quality. This framework offers a scalable approach to improving the robustness of multi-agent systems with potential for further extensions in decentralized and real-time applications.

15. SUBJECT TERMS
Multi-Agent Systems, Connectivity Augmentation, graph theory, convex optimization, resilient networks, Weapons Sciences

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	C. THIS PAGE		
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UU	20

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Moshe Hamaoui	(410) 306-0968

STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

1. Introduction

Multi-agent systems (MAS) are uniquely suited for spatially distributed and complex tasks, such as search and rescue, surveillance, and perimeter defense. These systems rely on persistent network communications to exchange information and coordinate actions. However, range-limited communication networks can be vulnerable to single points of failure, where the loss of a single agent-node can fracture the network. This report focuses on enhancing the robustness and survivability of agent networks by developing methods to proactively reconfigure agent geometry and corresponding network topology to establish redundant communication pathways before failures occur.

Connectivity augmentation is a classic problem in graph theory and operational research with practical applications in the design of networks that require redundancy and fault tolerance. In its simplest formulation, we are given a graph and asked to find the smallest edge augmentation set that will lift the vertex (or edge) connectivity to a specified threshold. The connectivity of a graph is the minimum number of vertices (or edges) that must be removed to separate the remaining nodes into two or more isolated subgraphs. Menger's theorem states the size of this minimum cut set is equal to the maximum number of disjoint paths between any two vertices. This duality highlights the utility of graph connectivity as a measure of network resilience and survivability.

One of the earliest solutions to the augmentation problem was found by Eswaran and Tarjan (1976) and Plesenik (1976), who formulated a polynomial-time solution for the 2-edge and 2-vertex connectivity problem. Many variations of the problem have been considered, focusing on both directed and undirected graphs, weighted graphs, constrained edge-augmentation sets (Frank and Jordán 2015), planar graphs and geometric graphs (Rutter and Wolff 2008). A review of the literature can be found in (Frank and Jordán 2015).

The goal of this study is to formulate a method for connectivity augmentation appropriate to airborne agent networks with range-limited communication. The graph is no longer abstract, but a consequence of the agents' relative geometry. In this physical setting, edges (or network links) exist only between agents that are within communication range. To add an edge between two agents requires these agents to simply draw closer. However, it is not obvious how to most efficiently restructure the collective agent geometry to lift network connectivity. Specifically, we must consider two key modifications to the classic augmentation problem:

1. **Objective**: Rather than minimizing the size of the edge augmentation set or the cost with respect to prescribed (constant) edge weights, we seek to minimize the transport cost. This cost metric could take different forms, e.g., sum or maximum of all agent displacements, fuel consumption, and time of flight. Feasible flight paths could be further constrained to account for limited platform control.

2. **Optimization variable**: Rather than finding an edge augmentation set, we seek a configuration geometry mapping the relative positioning of agents. Under this construction, the corresponding required edge set may actually discard some pre-existing edges.

The principal contributions of this work include: (1) formulating the Multi-Agent System Connectivity Augmentation (MASCA) problem for physical agent networks, (2) developing a computationally efficient two-phase solution approach that decouples the combinatorial graph problem from the geometric optimization, (3) identifying theoretical properties that significantly reduce the search space, and (4) formulating effective heuristics that achieve near-optimal solutions with significant computational savings.

Because the transport cost depends on the optimal geometry, we cannot optimize with respect to fixed edge weights (for which solutions exist [Frank and Jordán 2015]). This report introduces a methodical approach to solve this problem, involving two key steps: identifying feasible edge sets that satisfy vertex-connectivity requirements and selecting the optimal graph configuration based on geometric and operational considerations in 3D space.

2. Methods, Assumptions, and Procedures

2.1 Notation

We summarize the main notation used throughout the report below.

- $\mathbf{A}$: Adjacency matrix of the proximity graph $\mathcal{G}_\Delta(\mathbf{X})$.

- $\mathbf{A}_2$: Adjacency matrix encoding the union of all possible resulting edge sets after agents move within their reachability constraints.
- C_i: Feasible edge sets that meet the desired connectivity threshold.
- $\{C_i\}$: Collection of feasible edge sets.
- d_{ij}: Euclidean distance between agents v_i and v_j; $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$.
- E: Edge set of the proximity graph $\mathcal{G}_\Delta(\mathbf{X})$.
- E_2: Edge set corresponding to $\mathbf{A}_2$.
- $e_{ij} = \{v_i, v_j\}$: Edge in a graph.
- $\kappa(\mathcal{G})$, $\kappa(\mathbf{A})$, $\kappa(E)$: Vertex connectivity of graph $\mathcal{G}$, adjacency matrix $\mathbf{A}$, or edge set E.
- κ_{ij}: Pairwise vertex connectivity (equivalently, the number of vertex-disjoint paths) between agents v_i and v_j in $\mathcal{G}_2$.
- κ_L: Desired vertex connectivity level.
- κ_0: Initial vertex connectivity of the graph $\mathcal{G}_0$.
- m_d: Number of decision edges (edges that can be added or not).
- m_H: Minimum number of edges in a κ_L-connected graph with n vertices, given by Harary's theorem $m_H = \lceil \frac{\kappa_L n}{2} \rceil$.
- m_k: Number of edges that must be kept (necessary edges).
- n: Number of agents.
- $V = \{v_1, v_2, \ldots, v_n\}$: Set of agents (nodes) in the network.
- $\mathbf{W}$: Mask matrix indicating necessary, decision, and unreachable edges.
- $\mathbf{X}$: Stacked coordinate matrix of agent positions, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^\top \in \mathbb{R}^{n\times 3}$.
- $\mathbf{X}_0$: Initial positions of the agents.
- $\mathbf{X}^*$: Optimal agent positions after movement.
- $\mathbf{x}_i \in \mathbb{R}^3$: Position vector of agent v_i.
- $\mathbf{x}_i^0$: Initial position vector of agent v_i.
- Δ: Communication or interaction range; agents within distance Δ can communicate.
- $\mathbb{S}^n$: Space of symmetric $n \times n$ matrices.
- $|\cdot|$: Cardinality of a set.
- ρ: Maximum allowed displacement for agents; $\|\mathbf{x}_i - \mathbf{x}_i^0\|_2 \leq \rho$.

2.2 Problem Statement

Consider a system of n numerically indexed agents. Let $\mathbf{X}_0 = [\mathbf{x}_1^0, \mathbf{x}_2^0, \ldots, \mathbf{x}_n^0]^\top \in \mathbb{R}^{n\times 3}$ denote the stacked coordinate matrix of initial agent positions. The pairwise Euclidean distances are $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$. Range-limited interaction is represented by the proximity graph $\mathcal{G}_\Delta(\mathbf{X}) = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ and $E \subseteq \{(v_i, v_j) \mid d_{ij} \leq \Delta\}$ are the vertex and edge sets, respectively.

Given an initial configuration $\mathbf{X}_0$, corresponding proximity graph $\mathcal{G}_0 = \mathcal{G}_\Delta(\mathbf{X}_0)$, and desired vertex connectivity κ_L, we can formulate the Multi-Agent System Connectivity Augmentation (MASCA) problem:

$$\begin{aligned} &\text{minimize} && \max_i \|\mathbf{x}_i - \mathbf{x}_i^0\|_2, \quad i = 1, \ldots, n \\ &\text{subject to} && \kappa(\mathcal{G}_\Delta(\mathbf{X})) \geq \kappa_L. \end{aligned} \tag{1}$$

We have implicitly assumed in Problem (1) full agent maneuverability such that all positions $\mathbf{X}$ are reachable from $\mathbf{X}_0$. Further constraints could be added to limit reachability; e.g., maximum travel distance, acceleration, or deflection from the agent's initial velocity vector. Problem (1) is perhaps the simplest formulation of the MASCA problem. Even so, the combinatorial constraint on connectivity paired with the continuous cost function makes this problem challenging. Modern general-purpose global optimization solvers can provide reasonable—if not certifiably optimal—solutions, but this approach is only practicable for modestly sized applications (approximately $n \leq 7$) that can tolerate extended solution times.

Motivated by the need for faster solution times required for in-flight applications, we divide Problem (1) into two more tractable subproblems whose meta-parameters can be tuned to balance speed and cost. Broadly speaking, we first introduce a reachability bound that further constrains the space of reachable proximity graphs. This space is efficiently sampled, resulting in a set of candidate proximity graphs. In the second phase, for each graph, a convex optimization scheme is called to find an embedding in Euclidean space minimizing transport cost subject to spacing constraints. The lowest cost embedding is taken to be the (near) optimal solution $\mathbf{X}^*$. Meta-parameters limiting the numbers of graphs to sample and the number of embedding solutions to consider will be discussed below. Ultimately, this is a relaxation of the original problem, and we are interested in understanding, as best we can, the optimality gap.

2.3 Identifying Feasible Edge Sets

Let ρ be the agents' maximum allowed displacement, $\|\mathbf{x}_i - \mathbf{x}_i^0\|_2 \leq \rho, \quad i = 1, \ldots, n$. We can then define a new adjacency matrix $\mathbf{A}_2$ that encodes the union of all possible resulting edge sets:

$$(\mathbf{A}_2)_{ij} = \begin{cases} 1 & \text{if } d_{ij} \leq \Delta + 2\rho, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The limit $d_{ij} \leq \Delta + 2\rho$ represents the largest bound on d_{ij} such that $\|\mathbf{x}_i - \mathbf{x}_i^0\|_2 \leq \rho$ is still possible. Specifically, if a pair of unconnected nodes were maximally spaced $\Delta + 2\rho$ apart, they would each have to travel a distance ρ directly toward each other to gain adjacency. This is a statement of *local* reachability. A similar statement could be formulated for nodes (v_i, v_k), also—at the extreme—requiring these nodes to likewise move directly toward each other. These pairwise trajectories are not necessarily consistent. Thus, $\mathbf{A}_2$ does not necessarily represent a physically realizable proximity graph; it simply encodes local pairwise reachability. The converse statement, however, is more certain; if two nodes are not adjacent in $\mathbf{A}_2$, then they are certainly not reachable under the transport constraint. The utility of $\mathbf{A}_2$ is in collecting the set of all candidate edges E_2 that could possibly contribute toward $\mathcal{G}_\Delta(\mathbf{X}) \subseteq \mathcal{G}_2(V, E_2)$ after transport.

Our next task is to enumerate the collection of feasible subsets $\{C_i\}$:

$$C_i \in \{\phi \subseteq E_2 \mid \kappa(\phi) \geq \kappa_L\}, \quad \text{for each } i = 1, 2, \ldots, m, \tag{3}$$

where $\kappa(\phi)$ is the vertex connectivity of the graph associated with edge set ϕ, and m is the number of subsets satisfying the connectivity condition.

Even after eliminating the complement edges of $\mathcal{G}_2$, an exhaustive enumeration of the power set of E_2 and direct evaluation of connectivity of each subset would only be tractable for very small n. To facilitate a more efficient enumeration of $\{C_i\}$, we introduce a mask $\mathbf{W} \in \mathbb{S}^n$:

$$W_{ij} = \begin{cases} 1 & \text{if } (\mathbf{A}_2)_{ij} = 1 \text{ and } \kappa_{ij} = \kappa_L, \quad \text{(necessary)}, \\ 0 & \text{if } (\mathbf{A}_2)_{ij} = 0, \quad \text{(unreachable)}, \\ -1 & \text{otherwise}, \quad \text{(decision)}. \end{cases} \tag{4}$$

Among the reachable edges, some are nonnegotiably necessary. Let $\mathbf{K} \in \mathbb{Z}_+^{n \times n}$ be the matrix of pairwise vertex connectivities of $\mathcal{G}_2$, where the matrix entry κ_{ij} gives the number of vertex-independent paths between v_i and v_j. If any $\kappa_{ij} < \kappa_L$, then there can be no reachable configurations which will satisfy (3) at the specified Δ and ρ. We proceed, therefore, under the assumption that $\kappa_{ij} \geq \kappa_L$.

Suppose that for some vertex pair (v_i, v_j) we have $\kappa_{ij} = \kappa_L$, indicating that there are precisely κ_L vertex-disjoint paths between them. While these paths are not generally unique, we can argue that single-edge paths to adjacent nodes are uniquely specified. In particular, if $\kappa_{ij} = \kappa_L$ *and* $(\mathbf{A}_2)_{ij} = 1$, then the corresponding edge e_{ij} is necessarily one of those paths and must be included in all candidate C_i.

Proof by contradiction: Suppose that in $\mathcal{G}_2$ there were κ_L other paths without e_{ij}. Then we could construct an additional path $P = (v_i, v_j)$ composed entirely of e_{ij}. Since P contains no internal vertices, it is clearly independent of any other paths between v_i and v_j. This would lead to $\kappa_L + 1$ paths, contradicting our premise. Thus we conclude that $P = (v_i, v_j)$ is uniquely specified as one of the κ_L paths.

Let m_k be the number of edges that must be kept, and m_d the number of decision (candidate) edges. In general, each feasible edge set C_i will include all necessary edges and some combination of candidate edges. Without any further reduction, the number of possible subsets would be $\sum_{i=0}^{m_d} \binom{m_d}{i}$, and each would have to be evaluated against the connectivity requirement. However, we can further reduce the space of feasible sets by:

1. **cardinality bound**: The combinatorial explosion implied by the summation above follows in part from the size variability of candidate edge sets, $m_k \leq |C_i| \leq m_k + m_d$. Various cardinality bounds have been formulated for the classical connectivity augmentation problem (Frank and Jordán 2015), but are typically developed under the assumption that all existing edges are kept. For the present application, this assumption is not generally true. Instead, we can stipulate a lower-bound based on Harary's theorem on κ-vertex-connected graphs (Harary 1962): the minimum number of edges in a κ-connected graph with n vertices is

$$m_H = \left\lceil \frac{\kappa n}{2} \right\rceil, \tag{5}$$

which furnishes us with a second lower cardinality bound, in addition to m_k.

The bounds can be combined

$$|\{C_i\}| \leq \sum_{i=\max(m_H - m_k, 0)}^{m_d} \binom{m_d}{i}. \tag{6}$$

This is still just an upper-bound on $|\{C_i\}|$ as we have not yet evaluated these sets to filter out those that fail to meet the connectivity threshold. It turns out, however, there is no need to evaluate all combinations in (6), as we shall now show.

2. **Discarding Supersets of Feasible Sets**: If $C_i \subseteq C_i'$ and $\kappa(C_i) \geq \kappa_L$, then $\kappa(C_i') \geq \kappa_L$ since adding edges cannot decrease connectivity. Furthermore, additional constraints (edges) can never lead to a more optimal solution in terms of minimizing movement cost. Thus, we can discard all supersets of C_i without loss of generality.

   One way to implement this is to start with the smallest edge combinations of size $m_k + \max(m_H - m_k, 0)$ and evaluate each against the connectivity condition. Then form the next set of combinations of size $m_k + \max(m_H - m_k, 0) + 1$, discarding supersets of previously found feasible C_i. The process continues until we either reach set size $m_k + m_d$ or there are no more surviving combinations to evaluate. Aside from reducing the number of connectivity evaluations, the resulting smaller $|\{C_i\}|$ also means that there will be fewer feasible sets to consider in the optimization phase.

Algorithm 1 describes the procedure for constructing the collection of feasible edge sets.

2.4 Node-Position Optimization for Feasible Edge Sets

After identifying feasible edge sets that meet the desired vertex-connectivity threshold, the next phase involves determining optimal node positions that realize these edge sets while minimizing the total displacement from the initial configuration. This phase addresses the physical realization of the augmented graph by repositioning agents, within their movement constraints, to establish the necessary connections.

For each feasible edge set $C_i \subseteq E_2$ identified in the previous phase, we seek new positions $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^\top \in \mathbb{R}^{n \times 3}$ by solving the following convex optimization

Algorithm 1 Identification of Feasible Edge Sets for Connectivity Lifting

Input: Initial positions $\mathbf{X}_0$, communication range Δ, movement limit ρ, desired connectivity κ_L

Output: Feasible edge sets $\{C_i\}$, mask matrix $\mathbf{W}$, initial connectivity κ_0

Compute initial adjacency matrix $\mathbf{A}$ based on $\mathbf{X}_0$ and Δ
Calculate initial vertex connectivity $\kappa_0 = \kappa(\mathbf{A})$
if $\kappa_0 \geq \kappa_L$ **then**
    Return current graph; no augmentation needed
end if
Compute reachability adjacency matrix $\mathbf{A}_2$ using $\Delta + 2\rho$ per (2)
Calculate pairwise connectivities κ_{ij} in $\mathcal{G}_2$
if Any $\kappa_{ij} < \kappa_L$ **then**
    No solution exists under given constraints; exit
end if
Construct mask matrix $\mathbf{W}$ using (4)
$m_k \leftarrow$ number of necessary edges ($W_{ij} = 1$)
$m_d \leftarrow$ number of decision edges ($W_{ij} = -1$)
Compute $m_H = \left\lceil \frac{\kappa_L n}{2} \right\rceil$
Initialize feasible edge sets $\{C_i\} \leftarrow \emptyset$
for $m = \max(m_H - m_k, 0)$ to m_d **do**
    Generate combinations of decision edges of size m
    Exclude supersets of previously found feasible sets
    for Each edge combination S **do**
        $C \leftarrow$ necessary edges $\cup$ edge combination S
        Construct graph $\mathcal{G}(V, C)$
        if $\kappa(\mathcal{G}) \geq \kappa_L$ **then**
            Add C to $\{C_i\}$
        end if
    end for
    if No new feasible sets found **then**
        Break loop
    end if
end for
return Feasible edge sets $\{C_i\}$, mask matrix $\mathbf{W}$, initial connectivity κ_0

problem:

$$\underset{\mathbf{X}\in\mathbb{R}^{n\times 3}}{\text{minimize}} \quad \max_i \|\mathbf{x}_i - \mathbf{x}_i^0\|_2 \tag{7}$$

$$\text{subject to} \quad \|\mathbf{x}_i - \mathbf{x}_i^0\|_2 \leq \rho, \quad \forall i = 1, \ldots, n, \tag{8}$$

$$\|\mathbf{x}_i - \mathbf{x}_j\|_2 \leq \Delta, \quad \forall (v_i, v_j) \in C_i. \tag{9}$$

The objective function (7) minimizes the maximum Euclidean distance that any agent moves. Constraint (8) ensures that agents do not exceed their movement capabilities, while constraint (9) enforces the necessary connections as per the feasible edge set. This is a convex optimization problem, as both the objective and the constraints are convex functions of $\mathbf{X}$. Specifically, the objective function is convex as it is the maximum of convex (ℓ_2 norm) functions. The movement and edge constraints are convex as they are norms upper-bounded by a constant. The optimization problem (7–9) is a second-order cone programming (SOCP) problem.

As mentioned in Section 2.3, not every edge set in $\{C_i\}$ is guaranteed to have a realizable embedding $\mathbf{X}$ since $\mathbf{A}_2$ asserts only *pairwise* reachability. Furthermore, for any initial configuration $\mathbf{X}_0$, it is also therefore possible that all $\{C_i\}$ are unreachable. A negative (infeasible) result for all associated SOCP's would certify global infeasibility of the original problem (1).

Otherwise, after solving the optimization problem for all feasible edge sets, we compare the resulting objective values (maximum individual movements). The edge set and corresponding node positions with the minimal objective value are selected as the optimal configuration.

As described thus far, the feasible set enumeration procedure and subsequent node position optimization is certain to find the globally optimal solution to (1), if one exists. While this is more efficient than an exhaustive search, the solution scheme will at some point become impractical as n grows larger. If we are willing to trade optimality for speed, there are several heuristics that can be incorporated:

- Begin feasible edge set search with smaller allowed displacement radius, expanding gradually toward ρ until a feasible set has been found.
- Require that all initial edges are kept. This can drastically reduce the number of possible edge combinations. While there is no reason the optimal solution must retain these edges, numerical studies show this is very often the case.
- Cap the maximum number of feasible edge combinations to consider.
- Cap the number of feasible optimized geometries to consider.

3. Results

The methods outlined in the previous sections were implemented in MATLAB, with convex optimization components handled via CVX (CVX Research 2020) and MOSEK (ApS 2024). In this section, we illustrate and compare several example graph constructions for certified globally optimal solutions and heuristic-driven techniques.

3.1 Keeping Initial Edges

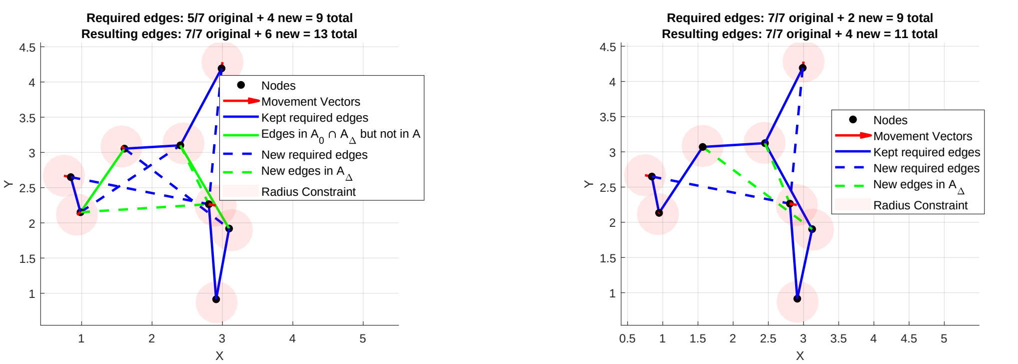
We generated 200 random geometries for $n = 8$ agents in the example set that follows. Given a specified interaction distance of $r = 2$, this corresponds to a set of proximity graphs. Of these, we selected 29 graphs that exhibited an initial vertex connectivity of $\kappa_0 = 1$. These graphs were then augmented with the objective of reaching connectivity threshold $\kappa_L = 2$ subject to maximum displacement radius of $\rho = 0.3$. Both optimal and heuristic solutions are calculated, as prescribed in Sections 2.3 and 2.4.

As mentioned previously, there is no reason to suppose that the initial set of edges E_0 must necessarily show up in the optimal set C_i^*. Indeed, the opposite tends to be true: all these edges are very seldom required to lift connectivity to κ_L. And yet, these edges tend overwhelmingly to reappear—not as *required* edges, but by incidental proximity—in the new proximity graph $\mathcal{G}_\Delta(\mathbf{X}^*)$. Of the 29 selected geometries in this dataset, 23 were feasible. In all 23, the optimal solution did not require all E_0, but the resulting proximity graphs still included them. This can be seen clearly in Figure 1 comparing global optimal and heuristic solutions. Blue lines indicate required edges, while green lines indicate incidental proximity.

For the optimal solution in Figure 1a, one can see by inspection that the blue lines trace out a two-vertex-connected graph using only 9 edges, which is just slightly greater than the Harary lower-bound, $m_H = \lceil \frac{\kappa n}{2} \rceil = 8$. At the same time, all E_0 appear in the resulting proximity graph, as indicated by the blue dashed lines. The heuristic formulation in Figure 1b, which requires all seven initial edges by prescription, also exhibits a total of nine required edges. While the resulting geometries are not identical, these differences are consistent with the objective function that minimizes the maximum displacement and does not penalize smaller displacements, provided that the spacing constraints (required edge constraints) are satisfied.

Practically, both solutions achieve identical objective values. In fact, in all 23 samples, objective values of optimal and heuristic problem formulations matched exactly, even though the latter computed about 35 times faster, on average. Figure 2 shows analogous results for $\kappa_L = 3$ with an average speed-up of approximately $55\times$. These results could have been anticipated simply from the appearance of E_0 in the proximity graphs of all (global) optimal solutions, indicating that the corresponding proximity constraints will not preclude the optimal solution.

This outcome suggests that imposing the requirement to use all pre-existing edges upfront results in minimal loss of efficiency. This approach also significantly reduces the computational search space, accelerating the solution process while maintaining solution quality.



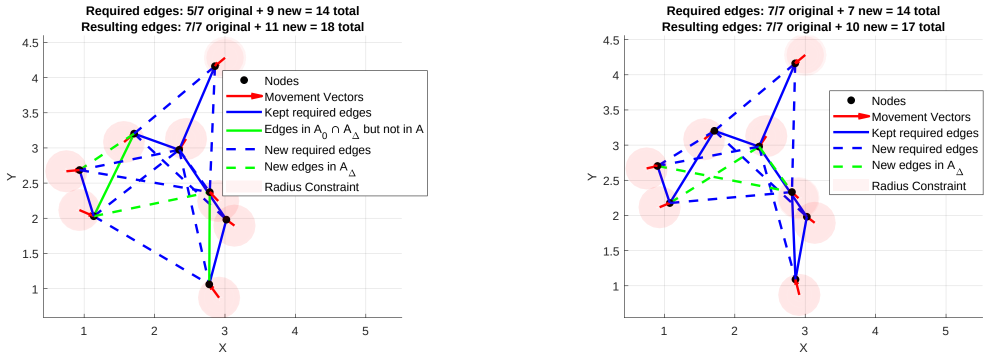
(a) Optimal problem formulation without requiring persistence of initial adjacencies.

(b) Suboptimal problem formulation achieves faster solution by requiring all initially adjacent nodes remain connected.

Figure 1: Lifting to $\kappa = 2$. Comparison of optimal and suboptimal connectivity-lifting problem formulations and solutions. The extra edge constraints in the suboptimal formulation still lead to the same optimal objective value.

4. Conclusion

This report addresses the challenge of connectivity augmentation in MAS with range-limited communication capabilities. We formulated the MASCA problem and proposed a two-phase solution consisting of feasible edge set identification and node-position optimization. The combinatorial nature of the problem presents significant computational challenges. To address this, we introduced strategies to reduce computational complexity, including the use of cardinality bounds, pruning of supersets, and applying heuristics such as preserving initial edges. Additionally,



(a) Optimal problem formulation without requiring persistence of initial adjacencies.

(b) Suboptimal problem formulation achieves faster solution by requiring all initially adjacent nodes remain connected.

Figure 2: Lifting to $\kappa = 3$. Comparison of optimal and suboptimal connectivity-lifting problem formulations and solutions. The extra edge constraints in the suboptimal formulation still lead to the same optimal objective value.

convex optimization was employed to accelerate node-position optimization and further enhance efficiency.

Numerical results demonstrated that our method effectively increases the vertex connectivity of the network while minimizing agent movement. The comparison between optimal and heuristic approaches revealed that introducing a few modest constraints can lead to substantial computational savings without compromising solution quality.

This approach offers a practical framework for enhancing the resilience and robustness of multi-agent communication networks. Future research directions could include 1) developing more efficient approximation algorithms by further exploiting the concept of necessary disjoint pathways (introduced in Section 2.3), 2) incorporating reachability constraints and cost functions that more accurately reflect platform dynamics and control capabilities, and 3) exploring decentralized implementations to improve scalability and adaptability in real-world applications.

References

ApS MOSEK. The MOSEK optimization toolbox for MATLAB manual. Version 10.2.5. 2024. https://docs.mosek.com/latest/toolbox/index.html.

CVX Research Inc. CVX: Matlab Software for Disciplined Convex Programming, version 2.2. https://cvxr.com/cvx. 2020.

Eswaran Kapali P. and Tarjan R. Endre. Augmentation Problems. SIAM Journal on Computing. 1976;5(4):653–665. 10.1137/0205044. https://epubs.siam.org/doi/abs/10.1137/0205044.

Frank András and Jordán Tibor. Graph connectivity augmentation. Handbook of Graph Theory, Combinatorial Optimization, and Algorithms. 2015:313–346.

Harary Frank. The maximum connectivity of a graph. Proceedings of the National Academy of Sciences. 1962;48(7):1142–1146.

Pleseník Ján. Minimum block containing a given graph. Archiv der Mathematik. 1976;27(1):668–672. ISSN: 1420-8938. 10.1007/BF01224737. https://doi.org/10.1007/BF01224737.

Rutter Ignaz and Wolff Alexander. Augmenting the Connectivity of Planar and Geometric Graphs. Electronic Notes in Discrete Mathematics. 2008;31:53–56. ISSN: 1571-0653. https://doi.org/10.1016/j.endm.2008.06.009. https://www.sciencedirect.com/science/article/pii/S1571065308000656.

List of Symbols, Abbreviations, and Acronyms

3D three-dimensional

MAS multi-agent systems

MASCA multi-agent system connectivity augmentation

SOCP second-order cone programming

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
1 (PDF)	DEVCOM ARL FCDD RLD DCI TECH LIB