Network Design for Partial Centralization in Multi-agent Systems

Thesis Proposal

Steven Okamoto
Computer Science Department
Carnegie Mellon University

Committee:
Katia Sycara (Chair)
Anupam Gupta
Srinivasan Seshan
Milind Tambe (University of Southern California)

1 Introduction

Establishing and maintaining effective wireless communication networks is a major challenge for large-scale multi-agent applications in outdoor environments, such as future military operations and disaster response teams. In these applications, agents often do not have access to established communication infrastructure, and so must rely on an ad hoc network to meet their communication needs. Limited communication ranges, communication capacity constraints, signal obstruction (by terrain, foliage, and buildings), agent mobility, and wireless interference restrict the effectiveness of ad hoc networks in such settings. Unmanned autonomous vehicles (UAVs) have been proposed as one way to address these issues by acting as communication relays and supplementing the ad-hoc network of ground-based agents with an aerial backbone network. The key challenge that this proposal addresses is how this aerial network should be designed: where should UAVs be positioned and how should capacity be utilized to meet the communication needs of the system.

Unmanned aerial vehicles have several advantages that favor them for supplementing ground-based ad hoc networks. First, UAVs are often large enough and have sufficient power resources that they can be equipped with sophisticated communication hardware with longer range and higher capacity than those carried by people on the ground or in low-power robots and sensors. Second, UAVs fly in largely obstruction-free airspace, which increases range, capacity, and reliability of communication among UAVs. Third, there are often clearer lines of sight between UAVs and agents on the ground, which similarly improves communication between UAVs and ground-based agents. Fourth, UAVs can often travel more quickly and freely than ground-based agents, which allow them to respond to agents’ movements.

The central problem in deploying a UAV backbone network is determining where the UAVs should be positioned. The UAVs can be used in a variety of ways, for instance by providing connectivity between two disconnected parts of the network, by reducing latency in parts of the network with heavy traffic, or by enhancing robustness against failure by providing alternative routes for information. The desired effect greatly affects the relative merits of different UAV configurations and must be taken into account when designing algorithms for positioning UAVs.
The positioning of UAVs is also heavily influenced by the communication needs of the network. In this proposal, the primary communication pattern that will be considered is partial centralization. Centralization is a fundamental aspect of many multi-agent systems. It arises naturally when agents are asked to collaborate to achieve goals such as data fusion or team plan execution. In data fusion, agents have different data (sensor readings, belief states, etc.) from which other, more useful data is synthesized. The process of aggregating the data in a single location so that it can be fused is an example of centralization. In team plan execution, a group of agents jointly execute tasks from a common, shared team plan structure. Because the individual tasks executed by an agent are frequently dependent on tasks performed by other agents, it is necessary for the agents to coordinate their activities so that the plan can be successfully completed. One way that this can be accomplished is by having the agents communicate plan status information to a single plan monitor, which oversees the execution of the plan. This aggregation of team plan status information is another example of centralization. Centralization is also commonly employed in algorithms for other multi-agent problems, such as resource allocation and network routing, where limited centralization is used to reduce problem complexity.

It is often not feasible, desirable, or necessary for all of the agents to centralize all aspects of operation. Instead, small groups of agents only centralize certain aspects of their collective behavior. For example, only the agents involved with a team plan need to participate in a centralization activity to achieve plan monitoring, and they need only communicate status information relevant to the execution of the team plan. We call this partial centralization, because only a limited subset of the agents are required to centralize limited aspects of their operation.

The central question in partial centralization is determining where information should be centralized for each group. This is important in real-world systems, because the communication and computational abilities of agents are often heterogeneous and limited. Choosing a centralization point poorly can adversely impact system performance. For example, different choices of data fusion points in a sensor network may utilize different amounts of total bandwidth. Choosing a poor data fusion point may entail saturating the entire network, making it unusable for any other use, while choosing a good data fusion point may lead to very little network overhead. Similarly, choosing a bad team plan monitor can greatly reduce the timeliness of information, which can greatly reduce the effectiveness of the team plan monitor in responding to changes. Thus, algorithms for solving the partial centralization problem must account for the network topology, which leads to a mutual interdependence between the UAV placement problem and the partial centralization problem.

A third aspect of network design is the allocation of capacity (or bandwidth) to different centralizing groups. This is necessary to guarantee that the communication needs of the groups are met. Allocating bandwidth along communication links can also serve to simplify routing in the network, which can become a major concern as the networks scale in size. Bandwidth allocation also affects the positioning of UAVs, for example, in areas where there is a high demand, multiple UAVs may be positioned to provide increased capacity.

I seek to address these three interdependent aspects of network design: UAV positioning; partial centralization; and capacity allocation. I will focus on domains in which agents are spatially distributed and have limited communication range, as in commonly in the case in large-scale, outdoor multi-agent systems. This work will begin to address two key challenges:

- Network topology and communication traffic patterns are only partially known. Prior work focuses on problems in which only one of these unknown, for example network topology design for known traffic patterns, or routing and bandwidth allocation given a known topology. In the problems I am solving, only part of the network topology is known (i.e., the ad hoc network between agents), and part of the problem is to adjust the topology through the positioning of UAVs. At the same time, only part of the communication patterns are known (i.e., communication sources), and part of the problem is
to select the communication sinks (i.e., centralization points) and allocate bandwidth to meet those communication demands.

- Partial centralization communication pattern. Most of the existing network design research has focused on multiple independent, single-source-single-destination pairs. Partial centralization is significantly different as a multiple-source-single-destination pattern in which there are dependence relations between members of a centralizing group. At the same time, there are multiple independent groups within the system, and only a part of the system is involved in each group. This differs from many data aggregation problems in sensor networks, where the goal is to ultimately aggregate data streams from the entire network.

2 Related Work

Network design for partial centralization combines three fundamental problems: UAV deployment, the assignment of groups to UAVs, and bandwidth allocation of links. There has been considerable research related to each of these problems individually, but never for all three of them simultaneously.

The UAV placement problem is very similar to facility location problems that have been extensively studied in the operations research community [10, 16, 3, 11]. Given a number of facilities and a set of sites with quantities of goods demanded at each site, the facility location problem is to place the facilities in order to satisfy the demand of the sites for minimum transportation costs of goods from facilities to sites. Variants to the facility location problem include capacitated problems [11, 6] in which there are capacity constraints on facilities or on transportation links, as well as multiple commodity problems [14] where different facilities can provide different types of goods. One major difference between facility location problems and the UAV placement problem in that they assume a direct link between facility and site, while communication from group member to UAV can be through a multihop network.

One area of networking research focuses on how to provide access between two initially disconnected networks through the addition of additional hubs, routers, and bridges. The problems addressed in that literature are similar to the UAV placement problem. One approach is to partition the network into multiple local access networks (LANs) and a backbone network [1, 15]. Each LAN has one node that is designated the access point and is also part of the backbone network. Traffic between nodes in different LANs must first be routed to the access point for the originating LAN, conveyed across the backbone network, and then routed across the destination LAN. Links are considered costly and capacitated. The local access network design problem is to design the local access networks by purchasing LAN edges between nodes so that the total cost is minimized and a known amount of traffic can be routed to the access points, which has similarities to both UAV placement and bandwidth allocation. The access network design problem is NP-hard, but it is known that there exist optimal solutions in which the LANs take the form of trees with the access points as roots [1]. Linear programming formulations have been used to approximately solve the access network design problem in [1] and more recently in [15]. Unlike the network design problem considered here, however, all inter-LAN traffic must be conveyed through the backbone network. In addition, the communication pattern is single-source-single-destination, instead of the centralizing groups we consider here.

The formation of two-tiered communication networks has also been a focus in mobile ad hoc networking [2, 8, 5]. In these networks, there is no pre-existing “backbone network” and the problem is to dynamically create such a network from the underlying ad hoc network. This is accomplished by partitioning the nodes into clusters, and selecting a clusterhead node in each cluster to act as the access node for the backbone network. As a result, all intra-cluster traffic is conducted in multi-hop through the ad hoc network, but inter-cluster traffic must pass through the clusterhead, which then relays it to other clusterheads. The primary focus of research has been in developing techniques of cluster formation and clusterhead selection, such as highest ID [2], highest degree [8], node weight [4], and weighted clustering [5]. These techniques primarily focus on
metrics such as cluster stability and power conservation, and ignore communication costs, capacities, and specific communication requirements. In addition, they assume that the nodes (including the clusterheads) move exogenously, while we actively position the UAVs to meet the demands of the network.

Virtual private network (VPN) provisioning [9] is one of the few areas that consider a group communication pattern. In VPN provisioning, groups of nodes within a network wish to form a subnetwork by reserving bandwidth from the underlying network. Given bounds on the communication demands of the nodes that wish to form the VPN, the VPN provisioning problem is to reserve bandwidth so that any traffic pattern respecting the given bounds can be feasibly routed. Polynomial-time optimal and approximation algorithms were found for some problems, but the capacitated version of the problem is NP-hard [9]. The “group communication” in VPNs differ from that in partial centralization because VPN member communicate with each other, while centralizing group members transmit data to the centralization point. Also, while VPN provisioning allocates bandwidth for a group-oriented communication pattern, it does not address the issue of supplementing the network through additional nodes.

3 Formal Problems and Complexity

In this section we formalize the problems relating to network design in the presence of partial centralization and prove complexity results about some of those problems. We use a graph theoretic abstraction of the communication network in which nodes represent agents and UAVs, and edges represent the potential for communication between the agents or UAVs represented by the end points. The two main problems we are interested in are network augmentation and partial centralization. Given a network topology, the partial centralization problem is to choose centralization points for different groups of nodes and allocate bandwidth along edges to meet the group members’ communication requirements. The network augmentation problem is to supplement an initial network with additional nodes in order to connect specified groups of nodes with sufficient bandwidth. Ultimately the problem of interest is one of performing network augmentation and partial centralization simultaneously.

We are interested in several problem variations that arise from different assumptions. We consider the following assumptions:

- **Edge capacities.** Communication links may have limited capacity (bandwidth) that restrict the amount of data that can be transmitted over the link per unit time. We represent these constraints as capacities on the edges, which are either non-negative real values for limited capacity, or positive infinity for unlimited capacity (also called uncapacitated edges).

- **Edge costs.** Communicating over a link may incur a cost, such as consuming energy or risking detection from enemy forces. This is represented by a cost on the edges, which are non-negative real values. A cost of zero for an edge indicates that communication using that link is cost-free.

- **Edge symmetry.** Communication links may symmetric or asymmetric. With symmetric links, a node that receives data over a link can reply (subject to capacity constraints) to the sender. This is commonly the case when nodes are linked by wired connections or have identical wireless communication ranges. With asymmetric links, a recipient node may not be able to reply to the sender node. This usually arises because the sender has a greater wireless communication range than the recipient. Asymmetric links are represented by directed edges, while symmetric links are represented by undirected edges or directed links in both directions.

- **Objective function.** We consider three objective functions:
  
  - Minimum Deployment: Minimize the number of UAVs required to satisfy all groups.
- Maximum Groups: Maximize the number of satisfied groups given a fixed number of UAVs.
- Maximum Agents: Maximize the number of satisfied agents given a fixed number of UAVs.

In past work, we considered primarily variations of limited and unlimited capacities, zero-costs, symmetric and asymmetric edges, and the minimum deployment and maximum groups objectives. We formalized the problems and showed that even simple variations of the Network Augmentation and Partial Centralization problems are NP-complete [12].

We begin our discussion here by considering the simplest cases of Network Augmentation and Partial Centralization alone.

### 3.1 Network Augmentation

In pure network augmentation, we are given an initial network topology, a finite set of potential locations where UAVs can be deployed, and groups of agents that need to communicate. The simplest case of network augmentation arises when all edges have unlimited (i.e., positive infinity) capacity, are cost-free, and are symmetric. The problem is to deploy UAVs to the potential locations in order to satisfy groups requirements. In this case, group satisfaction reduces to connectivity, so a group of agents is satisfied if all the nodes in the group are connected to each other once the UAVs have been deployed. We have shown that even this simplest network augmentation problem is NP-complete [12]. The formal description follows.

Let the initial network topology be represented as a simple graph $G = (V, E)$, where $V$ is the set of agent nodes, and there exists an edge $(u, v) \in E$ if and only if $u$ and $v$ can directly communicate.

Denote the potential network by the undirected simple graph $G' = (V', E') = (V \cup P, E \cup E_P)$. The set of possible locations where UAVs can be deployed is denoted by $P$, with $V \cap P = \emptyset$, and the set of potential communication links is denoted by $E_P$, with $E_P \cap E = \emptyset$. For each edge $(u, v) \in E_P$, either $u \in P$ or $u \in P$ or both. If $u \in P$ and $v \notin P$, then $(u, v) \in E_P$ represents that a UAV positioned at $u$ can communicate with node $v$; similarly, if $u \notin P$ and $v \in P$, then $(u, v) \in E_P$ means that a UAV positioned at $v$ can communicate with node $u$; and finally, if both $u, v \in P$, then if UAVs were positioned at both $u$ and $v$, they would be able to communicate.

We represent a deployment of UAV by $D \subseteq P$, the set of potential locations where we have deployed UAVs. We denote the actual network instantiated by $D$ by the undirected simple graph $G_D = (V_D, E_D)$ that is the subgraph of $G'$ induced by $V \cup D$.

Let $S = \{S_1, S_2, \ldots, S_K\}$ be the set of agent groups, where $S_i \subseteq V$ for $i = 1, \ldots, K$. We would like all of the nodes within a group $S_i \in S$ to be connected to each other in $G_D$. We refer to this property as group connectivity, and say that the group $S_i$ is connected (or is a connected group). We may also say that $S_i$ is $d$-partially connected, where $d$ is the maximum number of nodes of $S_i$ that are connected in $G_D$. With this terminology, a connected group $|S_i|$ is $|S_i|$-partially connected, while in a 1-connected group none of the group members are connected to each other in $G_D$.

There are three Network Augmentation optimization problem variants, depending on the objective function.

**Problem 1. Minimum Deployment Connected Groups (MinDepCG)** Given an initial network $G = (V, E)$, potential network $G' = (V \cup P, E \cup E_P)$, agent groups $S = \{S_1, \ldots, S_K\}$, find a deployment $D$ with minimum $|D|$ such that all of the groups are connected in $G_D$.

**Problem 2. Maximum Connected Groups (MaxCG)**. Given an initial network $G = (V, E)$, potential network $G' = (V \cup P, E \cup E_P)$, agent groups $S = \{S_1, \ldots, S_K\}$, and integer constant $h \geq 1$, find a deployment $D$ with $|D| \leq h$ such that the maximum number of groups are connected in $G_D$. 

5
Problem 3. Maximum Connected Agents (MaxCA). Given an initial network \( G = (V, E) \), potential network \( G' = (V \cup P, E \cup E_P) \), agent groups \( S = \{S_1, \ldots, S_K\} \), and integer constant \( h \geq 1 \), find a deployment \( D \) with \( |D| \leq h \) such that the sum of partial connectivity over all groups is maximized.

The decision problem related to MinDepCG and MaxCG is the Connected Groups problem:

Problem 4. Connected Groups (CG) Given an initial network \( G = (V, E) \), potential network \( G' = (V \cup P, E \cup E_P) \), agent groups \( S = \{S_1, \ldots, S_K\} \), and integer constants \( h \geq 1 \) and \( k \geq 1 \), is there a deployment \( D \) with \( |D| \leq h \) such that at least \( k \) groups are connected in \( G_D \)?

We have shown that the Connected Groups problem is NP-complete, even for a single group [12].

Theorem 1. CG is NP-complete.

Proof. To show this we must show that CG is in NP and is NP-hard. It is clearly in NP, as we can easily verify in polynomial time whether a set of possible locations \( D \subseteq P \) is a solution or not.

Proof of NP-hardness is done by reduction from the graph Steiner tree (ST) decision problem with unit-weight edges, which is known to be NP-complete [7]. The input to ST is an undirected graph \( G_{ST} = (V_{ST}, E_{ST}) \), a set of terminals \( X_{ST} \subseteq V_{ST} \), and a maximum cost \( k_{ST} \). The problem is to decide whether there exists a tree \( T_{ST} = (W_{ST}, F_{ST}) \) which is a subgraph of \( G_{ST} \) with \( X_{ST} \subseteq W_{ST} \) and at most \( k_{ST} \) edges. Such a tree \( T_{ST} \) is called a Steiner tree.

From an input instance \( G_{ST}, X_{ST}, k_{ST} \) for ST, we construct the following input \( G, G', S, h \):

- \( G = (V, E) \), where \( V = X_{ST} \) and \( E = \{(u, v) \in E_{ST} | u \in X_{ST} \land u \in X_{ST}\} \).
- \( G' = (V \cup P, E \cup E_P) \), where \( P = V_{ST} \setminus X_{ST} \) and \( E_P = \{(u, v) \in E_{ST} | u \notin X_{ST} \lor v \notin X_{ST}\} \).
- \( S = \{S_1\} \), where \( S_1 = X_{ST} \).
- \( h = k_{ST} + 1 - |X_{ST}| \).
- \( k = 1 \).

We now show that \( G_{ST} \) has a Steiner tree with at most \( k_{ST} \) edges if and only if the constructed CG instance has a solution.

Suppose first that \( G_{ST} \) has a Steiner tree \( T_{ST} = (W_{ST}, F_{ST}) \) with at most \( k_{ST} \) edges. We will show that there is a deployment that satisfies the solution conditions for the CG instance. Set \( D = W_{ST} \setminus X_{ST} \). Then clearly

\[
D = W_{ST} \setminus X_{ST} \\
\subseteq V_{ST} \setminus X_{ST} \\
= P \\
= X_{ST} \subseteq V_{ST} \\
\text{(by construction)}.
\]

Note that \( S_1 = X_{ST} \) is connected in the subgraph of \( G' \) induced \( V \cup D \), since \( X_{ST} \) is connected in \( T_{ST} \).

We also get that

\[
|W_{ST}| = |F_{ST}| + 1 \\
\leq k_{ST} + 1 \\
\text{(by assumption)}
\]
so that

\[ |D| = |W_{ST} \setminus X_{ST}| \]
\[ = |W_{ST}| - |X_{ST}| \quad (X_{ST} \subseteq W_{ST}) \]
\[ \leq k_{ST} + 1 - X_{ST} \]
\[ = h \quad \text{(by construction)} \]

Thus \( D \) is a solution to CG.

Assume now instead that there is a solution \( D \) to a CG instance as constructed above. We will show that there must then be a Steiner tree with at most \( k_{ST} \) edges for the Steiner instance. Let \( G_D = (V_D, E_D) \) denote the subgraph of \( G' \) induced by \( D \). Then

\[ |V_D| = |V_D \cap X_{ST}| + |V_D \setminus X_{ST}| \]
\[ = |X_{ST}| + |V_D \setminus X_{ST}| \quad (X_{ST} \subseteq V_D) \]
\[ = |X_{ST}| + |V_D \cap P| \quad (P = V_{ST} \setminus X_{ST}) \]
\[ = |X_{ST}| + |D| \]
\[ \leq |X_{ST}| + h \quad \text{(} D \text{ a solution)} \]
\[ = |X_{ST}| + k_{ST} + 1 - |X_{ST}| \quad \text{(by construction)} \]
\[ = k_{ST} + 1 \]

Because \( D \) is a solution, the nodes in \( S_1 = X_{ST} \) are connected in \( G_D \). Let \( T_D = (V_D, E_{T_D}) \) denote a spanning tree of \( G_D \). Because the nodes in \( X_{ST} \) are connected in \( G_D \), it follows that they are also connected in \( T_D \), and so \( T_D \) is a Steiner tree. Now we get that

\[ |E_{T_D}| = |V_D| - 1 \quad \text{(} T_D \text{ is a tree)} \]
\[ \leq k_{ST} + 1 - 1 \]
\[ = k_{ST} \]

Thus \( T_D \) is a Steiner tree with at most \( k_{ST} \) edges.

Hence \( G_{ST} \) has a Steiner tree with at most \( k_{ST} \) edges if and only if the constructed CG instance has a solution. Therefore CG is NP-complete.

Because other variations of the Network Augmentation problem with the maximum group or minimum deployment objective functions are generalizations of this simple case, it follows that they too are NP-complete.

### 3.2 Partial Centralization

In pure partial centralization, we are given a communication network topology, groups of agents that need to centralize some data, and a set of nodes that are eligible for acting as centralization points. The problem is to select a centralization point for each group that satisfies the group communication requirements. The simplest variations with unlimited capacities and cost-free edges are easily solvable. In this case a group can be satisfied if there exists an eligible node that can be reached from all members of the group. This can be computed by finding connected components in the symmetric edge case or computing reachable sets in the asymmetric edge case, which can be done in polynomial time.

In contrast, the case with limited capacity, cost-free, asymmetric edges is NP-complete. The formal description follows.
We are given a weighted, directed graph $G = (V, E)$ with the weight of each edge $e ∈ E$ denoted by $w_e$. Edge weights represent the bandwidth capacities of each link, and so have either positive, real values (representing finite capacities), or are positive infinity (representing unconstrained bandwidth).

Agent groups $S = \{S_1, \ldots, S_K\}$ seek to aggregate data at a single node in the network, which we term the centralization point of that group, for further processing. We denote by $C ⊆ V$ the set of eligible nodes that can act as centralization points. Each group has a single centralization point to which all members of the group transmit data, and different groups may have different centralization points. Furthermore, a node may be the centralization point for multiple groups. We represent the centralization points using the indicator variables $X_{iv}$ for $s_i ∈ S_i, v ∈ V$, where $X_{iv} = 1$ if $v$ is the centralization point for group $s_i$ and $X_{iv} = 0$ otherwise.

The transmission requirement for an agent group member $s ∈ S_i$ (where $S_i ∈ S$) is denoted by $b^i_s$, which is positive and real-valued. This represents the amount the traffic generated by $s$ that needs to be transmitted to the centralization point for group $S_i$. We seek to reserve bandwidth along edges of $G$ so that there is sufficient bandwidth to transmit $b^i_s$ in a multi-hop fashion from $s$ to the centralization point for $S_i$, for each $s ∈ S_i$. We denote the amount of bandwidth reserved on edge $e ∈ E$ for group $S_i$ by the variable $r^i_e$. The total amount of bandwidth reserved on an edge $e$ for all groups is constrained by $w_e$. If a bandwidth allocation for a group is sufficient for all of the group members to meet their transmission requirements to the group’s aggregation point while respecting bandwidth constraints, we say that the group is satisfied. This is formalized as a network flow problem in which agent group members seek to send flow to the centralization points, with the amount of flow sent on each edge corresponding to the bandwidth reservation for that edge.

The Maximum Satisfied Groups Partial Centralization problem (MaxSG-PC) is to find the centralization point assignments $\{X_{iv}\}$ and bandwidth reservations $\{r^i_e\}$ that maximize the number of satisfied groups. We formally define this problem next.

**Problem 5. Maximum Satisfied Groups Partial Centralization (MaxSG-PC).** Given directed network $G = (V, E)$, weights $\{w_e\}$ for all $e ∈ E$, agent groups $S = \{S_1, \ldots, S_K\}$, eligible nodes $C ⊆ V$, and transmission requirements $b^i_s$ for all $s ∈ V, 1 ≤ i ≤ K$, find centralization assignments $\{X_{iv}\}$ for all $1 ≤ i ≤ K, v ∈ V$, and bandwidth reservations $\{r^i_e\}$ for all $e ∈ E, 1 ≤ i ≤ K$ that maximize

$$\sum_{i=1}^{K} \sum_{v ∈ V} X_{iv}$$

subject to the following constraints:

$$\sum_{v:(u,v)∈E} r^i_{(u,v)} - \sum_{v:(v,u)∈E} r^j_{(u,v)} = \left(b^i_u \sum_{v ∈ V} X_{iv}\right) + \left(X_{iv} \sum_{v ∈ V} b^j_v\right) \quad \text{for } u ∈ V, 1 ≤ i ≤ K \quad (2)$$

$$r^i_e ≥ 0 \quad \text{for } e ∈ E, 1 ≤ i ≤ K \quad (3)$$

$$\sum_{i=1}^{K} r^i_e ≤ w_e \quad \text{for } e ∈ E \quad (4)$$

$$X_{iv} = 0 \quad \text{for } 1 ≤ i ≤ K, v ∈ V \setminus C \quad (5)$$

$$\sum_{v ∈ V} X_{iv} ≤ 1 \quad \text{for } 1 ≤ i ≤ K \quad (6)$$

$$X_{iv} ∈ \{0, 1\} \quad \text{for } 1 ≤ i ≤ K, v ∈ V. \quad (7)$$

The objective function in Equation 1 is to maximize the number of satisfied groups. Constraints for solution feasibility are given in Equations 2 – 7. Equations 2 represents the constraint that the net flow into
a node is conserved. Positive bandwidth reservations are provided by Equation 3. Equation 4 requires that that total bandwidth reservations for all groups on each edge is constrained by the capacity of the edge. Equation 5 ensures that centralization points are only chosen from eligible nodes. Equation 6 represents the constraint that there is at most a single aggregation point for each group. Finally Equation 7 is the integrality constraint on centralization point assignment.

### 3.3 Network Augmentation for Partial Centralization

Network Augmentation for Partial Centralization is the problem of choosing a deployment and selecting the centralization points from among the deployed UAVs. It is a generalization of both the pure network augmentation and partial centralization problems.

As in pure network augmentation, we are given an initial network \( G = (V, E) \) and potential network \( G' = (V', E') = (V \cup P, E \cup E_P) \), but these are now weighted graphs, with the weight of each edge \( e \in E' \) denoted by \( w_e \). Edge weights represent the bandwidth capacities of each link, and so have either positive, real values (representing finite capacities), or are positive infinity (representing unconstrained bandwidth). A deployment \( D \subseteq P \) is the set of potential locations where we will deploy UAVs. In this problem formulation, we will use the indicator variables \( D_p \) for \( p \in P \), where \( D_p = 1 \) if \( p \in P \) and \( D_p = 0 \) otherwise.

As in pure partial centralization, agent groups \( S = \{S_1, \ldots, S_K\} \) seek to centralize data at group centralization points. The set of nodes eligible to serve as centralization points is denoted by \( C \subseteq V \cup P \). If \( v \in C \cap V \), it means that the agent node \( v \) can act as an centralization point, while if \( p \in C \cap P \), it means that a UAV deployed to \( p \) can act as an centralization point. We are primarily interested in the case where the centralization points must be selected from among the UAVs (that is, \( C = P \)), which is not unreasonable given the extremely limited computational power of many typical agent nodes. We use the indicator variables \( X_{ip} \) for \( S_i \in S, p \in D \), where \( X_{ip} = 1 \) if \( p \) is the centralization point for group \( S_i \) and \( X_{ip} = 0 \) otherwise.

The transmission requirements for a group member \( s \in S_i \) (where \( S_i \in S \)) is denoted by \( b^i_s \), which is positive and real-valued. This represents the amount the traffic generated by \( s \) that needs to be transmitted to the aggregation point for group \( S_i \). For a given deployment \( D \), we seek to reserve bandwidth along edges of \( G_D \) so that there is sufficient bandwidth to transmit \( b^i_s \) in a multi-hop fashion from \( s \) to the aggregation point for \( S_i \), for each \( s \in S_i \). We denote the amount of bandwidth reserved on edge \( e \in E_D \) for group \( S_i \) by the variable \( r^i_e \). The total amount of bandwidth reserved on an edge \( e \) for all groups is constrained by \( w_e \).

As before, there are three optimization problems depending on the choice of objective function: Maximum Satisfied Groups (MaxSG), Maximum Satisfied Agents (MaxSA), and Minimum Deployment (MinDep). We will detail the formulation for MaxSG, then briefly sketch how the others are similarly formulated.

The maximum satisfied groups with limited bandwidth problem (MaxSG) is to find the deployment \( \{D_p\} \), centralization assignments \( \{X_{ip}\} \), and bandwidth reservations \( \{r^i_e\} \) that maximize the number of groups that successfully centralize data.

**Problem 6. Maximum Satisfied Groups (MaxSG).** Given initial network \( G = (V, E) \), potential network \( G' = (V', E') = (V \cup P, E \cup E_P) \), weights \( \{w_e\} \) for all \( e \in E' \), agent groups \( S = \{S_1, \ldots, S_K\} \), transmission requirements \( b^i_s \) for all \( s \in V \), \( 1 \leq i \leq K \), and number of UAVs \( h > 0 \), find a deployment \( \{D_p\} \), centralization assignments \( \{X_{ip}\} \) for all \( 1 \leq i \leq K, p \in P \), and bandwidth reservations \( \{r^i_e\} \) for all \( e \in E', 1 \leq i \leq K \) that maximize

\[
\sum_{i=1}^{K} \sum_{p \in P} X_{ip}
\]  

(8)
subject to the following constraints:

\[
\sum_{v : (u,v) \in E} r^i_{(u,v)} - \sum_{v : (v,u) \in E} r^i_{(u,v)} = b^i_u \sum_{p \in P} X_{ip} \quad \text{for } u \in V, \ 1 \leq i \leq K \tag{9}
\]

\[
\sum_{u : (u,p) \in E_P} r^i_{(u,p)} - \sum_{u : (p,u) \in E_P} r^i_{(p,u)} = X_{ip} \sum_{v \in V} b^i_v \quad \text{for } p \in P, \ 1 \leq i \leq K \tag{10}
\]

\[
r^i_e \geq 0 \quad \text{for } e \in E', \ 1 \leq i \leq K \tag{11}
\]

\[
\sum_{i=1}^{K} r^i_e \leq w_e \quad \text{for } e \in E \tag{12}
\]

\[
\sum_{i=1}^{K} r^i_{(u,v)} \leq w_{(u,v)} A_{uv} \quad \text{for } (u, v) \in E_P \tag{13}
\]

\[
A_{pv} = D_p \quad \text{for } (p, v) \in E_P, \ p \in P, \ v \notin P \tag{14}
\]

\[
A_{uv} = D_p \quad \text{for } (u, v) \in E_P, \ p \in P, \ v \notin P \tag{15}
\]

\[
A_{pq} \leq D_p \quad \text{for } (p, q) \in E_P, \ p \in P, \ q \in P \tag{16}
\]

\[
A_{pq} \geq D_p + D_q - 1 \quad \text{for } (p, q) \in E_P, \ p \in P, \ q \in P \tag{17}
\]

\[
X_{ip} \leq D_p \quad \text{for } p \in P, \ 1 \leq i \leq K \tag{18}
\]

\[
\sum_{p \in P} X_{ip} \leq 1 \quad \text{for } 1 \leq i \leq K \tag{19}
\]

\[
\sum_{p \in P} D_p \leq h \tag{20}
\]

\[
A_{uv} \in \{0, 1\} \quad \text{for } (u, v) \in E_P \tag{21}
\]

\[
D_p \in \{0, 1\} \quad \text{for } p \in P \tag{22}
\]

\[
X_{ip} \in \{0, 1\} \quad \text{for } 1 \leq i \leq K, \ p \in P. \tag{23}
\]

The objective function in Equation 8 is to maximize the number of satisfied groups. Constraints for solution feasibility are given in Equations 9 – 23. Equations 9 and 10 represent the constraint that the net flow into agents and UAVs, respectively, is conserved. Positive bandwidth reservations are provided by Equation 11.

Equations 12 requires that that total bandwidth reservations for all groups on each link between agents is constrained by the capacity of the link. The same capacity constraint for edges in $E_P$ is represented by Equation 13, with the additional requirement that bandwidth reservations are only allowed on edges incident to possible locations to which UAVs have been deployed. For edge $(u, v) \in P$, this is represented formally by the indicator variable $A_{uv}$ which is 1 if $(u, v) \in E_D$ and 0 otherwise. Equations 14 and 15 define $A_{uv}$ when only one of the endpoints is in $P$. When both endpoints are in $P$, we would like to define $A_{uv} = D_u D_v$, but this is not a linear constraint as both $D_u$ and $D_v$ are variables. Instead, we use a standard linearization to represent the same definition with two constraints, Equations 16 and 17.

Equation 18 ensures that centralization points are only chosen from deployed UAVs. Equation 19 represents the constraint that there is at most a single centralization point for each group. Equation 20 ensures that no more than $h$ UAVs are deployed. Finally Equations 21 – 23 are the integrality constraints on actual edge indicators, UAV deployment, and centralization point assignment, respectively.

The Minimum Deployment problem (MinDep) is to find a minimum-sized deployment that satisfies all groups.

Problem 7. Minimum Deployment (MinDep). Given initial network $G = (V, E)$, potential network
Input: Instance $I = (G, G', \{w_e\}, S, \{b_i^s\})$: initial network $G = (V, E)$, potential network $G' = (V \cup P, E \cup E_P)$, weights $w_e$ for all $e \in E'$, groups $S = \{S_1, \ldots, S_K\}$, transmission requirements $b_i^s$ for all $s \in V$, $1 \leq i \leq K$.

Output: Solution $O = (\{D_p\}, \{X_{ip}\}, \{r_{ie}\})$: deployment $\{D_p\}$, centralization assignments $\{X_{ip}\}$, bandwidth reservations $\{r_{ie}\}$ for all $e \in E'$, $1 \leq i \leq K$ that minimize

$$\sum_{p \in P} D_p$$

subject to the constraints in Equations 9 – 23, excluding those in Equation 20.

4 Solving MinDep and MaxSG

In past work [12, 13], we solved MaxSG and MinDep both optimally and heuristically using mixed integer linear programming (MILP). Equations 8 – 23 define a MILP formulation of MaxSG, and the formalization of MinDep also defines a MILP formulation. These formulations can be solved optimally using standard MILP techniques, but this is intractable in general (unless $P = NP$). We instead developed heuristics that iterate through the groups, extending a partial solution on each iteration by solving a MILP for the current group.

4.1 Solving MinDep

The basic sequential heuristic algorithm for solving MinDep, MD-Rand-H, iterates in random order and is shown in Algorithm 1. The loop (lines 3 – 6) iterates through the groups, with the problem for the current group solved in line 4 by the subroutine SolveGroup. SolveGroup finds the minimum deployment that satisfies the current group given the partial solution for previous groups. SolveGroup returns the extended partial solution with the extended deployment $D$, the centralization point $Y_j$ for the current group $S_j$, and the bandwidth reservations $\{r_{ie}^s\}$ for the current group and all previously solved groups. It does this by solving a mixed integer program based on the formulation for MinDep, but with the following changes:

1. New constraint

$$D_p = 1 \quad \text{for all } p \in D.$$
This guarantees that the previous deployment found as a partial solution is extended.

2. New constraint

\[ X_{ip} = 1 \quad \text{for all } S_i \in F \text{ where } Y_i = p. \] (26)

This guarantees that the previous aggregation assignments is extended.

3. Constraints in Equations 9, 10, 11, 18, 19, and 23 exist only for \( i \) such that \( S_i \in F \cup \{ S_j \} \), instead of for \( 1 \leq i \leq K \). Together with Equations 25 and 26, this constrains the MILP to only find solutions for the current group \( j \), subject to the partial solutions already found for groups \( S_i \in F \).

It is worth noting that SolveGroup only solves for the integral deployment and assignment variables for the current group, but solves for the continuous bandwidth reservation variables for all previous groups. This does not substantially affect the running time because of efficient, standard LP techniques for solving continuous variables, but permits greater flexibility to the algorithm by allowing bandwidths reservations to change in later iterations. Only the bandwidth reservations for the final iteration are used in the ultimate solution.

We also considered variations in which the next group used to extend the solution is greedily chosen according to the number of additional supplemental nodes required. This is done with respect to a total order relation \( R \) on the deployment size; we consider two such relations, the conventional less-than-or-equal-to (\( \leq \)) and greater-than-or-equal-to (\( \geq \)). The Least First greedy heuristic (MD-LF-H) uses the \( \leq \) operator and chooses the smallest increase in the deployment size at every iteration. The Most First greedy heuristic (MD-MF-H) uses the \( \geq \) operator and is motivated by dealing with the group that imposes the highest cost first.

The general approach is shown in Algorithm 2. The outer while loop from line 3 to 14 iterates over all the groups. The inner for loop from lines 5 to 11 iterates over the groups that have not yet been solved for and finds one that requires the “best” number of additional supplemental nodes according to \( R \). The partial solution is extended for the selected group in line 12, and the group is removed from future consideration in line 13. It is not necessary to explicitly extend the solution for the aggregation assignment, because \( Y_j \) will never be assigned after \( S_j \) is added to \( F \).

While MD-Rand-H calls SolveGroup \( K \) times (and hence must solve \( K \) MILPs), Algorithm 2 calls SolveGroup \( K(K + 1)/2 \) times.

4.2 Solving MaxSG

Like the heuristics for MinDep, the heuristics for MaxSG iterate through the groups and extend the partial solution for at most one group on each iteration. The basic iterative heuristic for MaxSG is shown in Algorithm 3. Unlike MinDep, the limitation on the number of total UAVs available means that all groups may not be able to be satisfied in a MaxSG instance. This is checked in lines 5 – 9, with the solution for the current group used to extend the partial solution if the deployment satisfies the constraint on the maximum deployment size, and discarded otherwise.

5 Results

In this section we present results of empirical evaluation of the heuristic Algorithms 1 – – 3. We randomly generated input instances, then compared the deployment sizes found by the heuristics to the number of groups satisfied by optimal deployments found using the MILP formulation in Equations 24 – 7. In the results reported here, each data point is the average over 50 input instances.
Input: Instance $\mathcal{I} = (G, G', \{w_e\}, S, \{b^*_e\})$: initial network $G = (V, E)$, potential network $G' = (V \cup P, E \cup E_P)$, weights $w_e$, groups $S = \{S_1, \ldots, S_K\}$, transmission requirements $b^*_e$, number of UAVs $h$.

Output: Solution $\mathcal{O} = (\{D_p\}, \{X_{ip}\}, \{r^*_e\})$: deployment $\{D_p\}$, centralization assignments $\{X_{ip}\}$, and bandwidth reservations $\{r^*_e\}$.

1. $D \leftarrow \emptyset$ (set of deployed positions)
2. $F \leftarrow \emptyset$ (set of processed groups)
3. while $F \neq S$ do
   4. $D_{\text{best}} \leftarrow \emptyset$
   5. foreach $S_j \in S \setminus F$ do
      6. $\langle D', Y_j, \{r^*_e\} \rangle \leftarrow \text{SolveGroup} (\mathcal{I}, j, F, D, \{Y_i\})$
      7. if $|D'| \leq h$ then
         8. $D_{\text{best}} \leftarrow D'$
         9. $k \leftarrow j$
   10. end
   11. $D \leftarrow D_{\text{best}}$
   12. $F \leftarrow F \cup \{S_k\}$
13. end
14. Initialize all $\{D_p\}$ and $\{X_{ip}\}$ to 0
15. Set $D_p \leftarrow 1$ for all $p \in D$
16. Set $X_{ip} \leftarrow 1$ for all $Y_i = p$
17. return $\langle \{D_p\}, \{X_{ip}\}, \{r^*_e\} \rangle$

Algorithm 2: General greedy iterative heuristic for MinDep.

---

Input: Instance $\mathcal{I} = (G, G', \{w_e\}, S, \{b^*_e\})$: initial network $G = (V, E)$, potential network $G' = (V \cup P, E \cup E_P)$, weights $w_e$, groups $S = \{S_1, \ldots, S_K\}$, transmission requirements $b^*_e$, number of UAVs $h$.

Output: Solution $\mathcal{O} = (\{D_p\}, \{X_{ip}\}, \{r^*_e\})$: deployment $\{D_p\}$, aggregation assignments $\{X_{ip}\}$, and bandwidth reservations $\{r^*_e\}$.

1. $D \leftarrow \emptyset$ (set of deployed positions)
2. $F \leftarrow \emptyset$ (set of processed groups)
3. foreach $S_j \in S$ do
   4. $\langle D', Y_j, \{r^*_e\} \rangle \leftarrow \text{SolveGroup} (\mathcal{I}, j, F, D, \{Y_i\})$
   5. if $|D'| \leq h$ then
      6. $D \leftarrow D'$
   7. else
      8. $Y_j \leftarrow \text{null}$
   9. end
10. $F \leftarrow F \cup \{S_j\}$
11. end
12. Initialize all $\{D_p\}$ and $\{X_{ip}\}$ to 0
13. Set $D_p \leftarrow 1$ for all $p \in D$
14. Set $X_{ip} \leftarrow 1$ for all $Y_i = p$
15. return $\langle \{D_p\}, \{X_{ip}\}, \{r^*_e\} \rangle$

Algorithm 3: SG-Rand-H: Basic iterative heuristic for MaxSG.
Figure 1: Average deployment sizes found by the optimal algorithm and Algorithms 1 and 2.

Input instances were generated based on a disk graph model. In the results presented here, 100 agent nodes and 64 potential locations were uniformly distributed over a square region of the plane measuring $250 \times 250$. All agent nodes had a range of 25, and all UAVs had a range of 50. The initial network $G = (V, E)$ was formed by adding an edge between two agents if the distance between them was at most the agent range (25). The bandwidth of all edges in $E$ was 10. The potential network $G' = (V \cup P, E \cup E_P)$ was formed by adding an edge between a potential location and another node (agent node or another potential location) whenever the distance between them was at most the supplemental node range (50). This reflects cases where supplemental nodes have higher-gain antennas than sensor nodes, or where environmental factors contribute to longer ranges for supplemental nodes, as when supplemental nodes are airborne and suffer less path loss due to reflection and obstructions. The bandwidth of all edges in $E_P$ was 50.

Group sizes were independently and uniformly distributed between 2 and 7, inclusive, and each input instance could have groups of different sizes. Group membership was chosen independently and uniformly at random from all possible groups of the appropriate size, and sensor nodes could participate in multiple groups. The transmission requirements of each group member was a real number independently and uniformly distributed between 10 and 30. Deployments up to a maximum size of 4 supplemental nodes were considered in these experiments. To solve the MILPs, we used CPLEX, a highly efficient commercial solver.

We solved MinDep problems both optimally using the exact MILP formulation and heuristically using MD-Rand-H, MD-MF-H, and MD-LF-H. All algorithms were run for a maximum of 2 hours on each instance. The average deployment sizes found by the optimal and heuristic algorithms for 1 to 6 groups is plotted in Figure 1. The three heuristics found deployments of similar sizes, although the greedy heuristics significantly outperformed the random heuristic for larger numbers of groups. The optimal algorithm found significantly smaller deployments than the heuristics, with the difference in deployment sizes increasing with the number of groups. However, even for 6 groups, MD-Rand-H finds solutions within 25% of optimal on average, while the MD-MF-H and MD-LF-H find solutions within 14% of optimal on average.

The improvement in MinDep solution quality of the optimal algorithm over the heuristics comes at the cost of increased running time. The average running times are shown in Figure 2, along with the standard errors. While the running times of the optimal algorithm and the heuristics are very close for 1 or 2 groups, they quickly diverge for larger number of groups as the running time of the optimal algorithm increases.
Figure 2: Running time comparison of optimal algorithm and Algorithm 1 and 2

Figure 3: Proportion of instances solved optimally within 2 hours.
dramatically. The error bars show that the variability in running times of the optimal algorithm also increases with the number of groups, while the iterative heuristic exhibits less variability that increases more slowly than the optimal algorithm. This suggests that heuristics may be better suited in cases where the highly variable running times is undesirable.

The running time of the optimal algorithm increases with the number of groups, and this rate increases from 1 to 5 groups, then seems to decrease from 5 to 6 groups. However, this is an artifact resulting from the maximum cut-off time of 2 hours, which introduces an artificial cap on the running time of the optimal algorithm. As the number of groups increases, the proportion of instances that can be solved optimally within 2 hours decreases from 100% for 1 and 2 groups down to about 50% for 6 groups, as shown in Figure 3. The heuristics all terminated within 2 hours and so are not plotted.

We also compared the iterative heuristic for MaxSG to the optimal solution for 1 to 5 groups. Figure 4 plots the number of groups satisfied by deployments of up to 4 UAVs found by Algorithm 3, normalized to the optimal number of groups satisfied. As can be seen, Algorithm 3 initially performs well, but solution quality decreases relative to the optimal as the number of groups increases.

A comparison of the running times of the optimal MILP and Algorithm 3 is given in Figure 5. Both algorithms require similar amounts of time for problems with 1 and 2 groups, but the running time of the optimal algorithm for increases sharply thereafter, while the running time of Algorithm 3 increases much more slowly.

6 Proposed Work

The following is a list of proposed topics I plan to investigate for this thesis.

1. One major direction for future work is develop tractable approximation algorithms for MinDep and MaxSG that would provide provable bounds on solution quality, something that the heuristics developed cannot. It may be possible to do this by exploiting the structure of the MILPs for these problems.
2. The maximum agent objective function is well motivated by applications where even incomplete centralization may yield partial benefits (e.g., in some sensor fusion applications). Whether problems with the maximum agent objective function differ in complexity and approximability from analogous problems with the minimum deployment or maximum group objective functions is an interesting open question.

3. Costly communication is another variation that I intend to address. Communication costs can be included as a secondary optimization criterion, such as finding the lowest cost minimum deployment or lowest cost deployment that maximizes group satisfaction.

4. Wireless interference can severely limit the throughput of an ad hoc wireless network. I will explore the impact of wireless interference on the solution quality of algorithms that do not account for interference. One way in which wireless interference may be mitigated is through topology control, in which devices decrease the signal strength of transmissions, thereby reducing the area of potential interference at the expense of lost range and capacity.

5. I will also investigate dynamic repositioning of UAVs as agents move and communication requirements change. An initial step in this direction is to extend MaxSG to include current positions for UAVs and to represent the allowable transitions to new positions. This may be incorporated with or without additional transition costs incurred for a UAV moving from one position to another.

6. The effects of different routing protocols. The bandwidth allocation scheme provides a form of optimal routing given the constraints of the network, but practical systems will need to operate using more realistic routing protocols. These may greatly affect the network augmentation and partial centralization problems. Initial steps will include examining simpler allowable structures for allocated bandwidth (e.g., trees) that may simplify routing tasks. More in depth analysis will involve analyzing simulated performance using specific ad hoc routing protocols to the performance achievable.

Figure 5: Running time comparison of optimal algorithm and Algorithm 3.
7. Robustness to node and link failure is an important and interesting property. I plan to study how UAV deployment can address robustness issues by providing redundant network paths and adjusting position to respond to failures.

8. Investigate and analyze the network structures and UAV behaviors that result from solving problem instances. I will primarily consider statistical network properties such as degree distribution, flow distribution, and clustering properties. Identifying certain well-known emergent network structures (such as power law degree distributions) may provide insight into how to solve problem instances. This may be especially helpful for solving very large scale instances involving thousands of nodes.

References


