Titre: Title:	Comparison of Algorithms for the Optimization of Multi-waveform Networks
Auteur: Author:	Hewan Leul Kedane
Date:	2023
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Kedane, H. L. (2023). Comparison of Algorithms for the Optimization of Multi-waveform Networks [Master's thesis, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/53420/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/53420/
Directeurs de recherche: Advisors:	Alain Hertz, & Andrea Lodi
Programme: Program:	Maitrise recherche en mathématiques appliquées

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Comparison of Algorithms for the Optimization of Multi-waveform Networks

HEWAN LEUL KEDANE
Département de mathématiques et de génie industriel

Mémoire présenté en vue de l'obtention du diplôme de Maîtrise ès sciences appliquées
Mathématiques appliquées

Mai 2023

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Comparison of Algorithms for the Optimization of Multi-waveform Networks

présentée par **Hewan Leul KEDANE**
en vue de l'obtention du diplôme de Maîtrise ès sciences appliquées
a été dûment acceptée par le jury d'examen constitué de :

Louis-Martin ROUSSEAU, président
Alain HERTZ, membre et directeur de recherche
Andrea LODI, membre et codirecteur de recherche
Adulyasak YOSSIRI, membre externe

DEDICATION

to my mom,
you are a gift for me,
thank you for being someone I could always look up to. . .

ACKNOWLEDGEMENTS

This work is a product of long, hard work over several years. It would not have been possible without the help and guidance of many important people.

First and foremost, I would like to express my gratitude to my supervisors, Alain Hertz, and Andrea Lodi, who patiently guided me through the rigours of research. Their in-depth knowledge of optimization and mathematics and their mentorship were valuable assets to the completion of this work. I would also like to thank all my colleagues and friends at the lab for their support. Notably, I am grateful to Mehdi Taobane for his mentorship. Completing my thesis would have been difficult without his advice and guidance throughout the thesis writing process.

I am greatly indebted to my friend Vincent Perreault who very kindly sacrificed a large amount of his time over the years working with me to push this project further until the end. His insight on completing a master's thesis, advice, and emotional support were incredibly monumental for me reaching here. I would also like to thank Josee Lafreniere for her long hours editing my thesis.

Finally, I would like to thank all my family and friends who have been there for me through thick and thin. Without their support, this journey would have been incredibly lonesome and challenging.

RÉSUMÉ

Les facteurs environnementaux, y compris le terrain et la météo, affectent considérablement la vitesse de communication du réseau sans fil. Dans cet article de recherche, le modèle de topologie de réseau maillé sans fil [1] est adapté au modèle de topologie arborescente sans fil pour résoudre le problème général de l'article de thèse [2]. Le modèle considère un graphe de connectivité de secteur pour définir une direction d'antenne pour chaque nœud. De plus, nous avons utilisé une antenne à faisceau unique pour chaque secteur et modélisé les modèles de signal et les interférences par une forme de cône idéale. Nous avons également modélisé un problème de programmation linéaire en nombres entiers mixtes pour trouver le lien le plus encombré du réseau. Un algorithme de recherche locale itérative est utilisé pour résoudre notre problème rapidement et efficacement avec le logiciel Gurobi.

L'algorithme proposé a été évalué en utilisant cinq instances de problème différentes et en analysant un nombre différent de liens par secteur. Le calcul a montré que l'algorithme proposé pouvait trouver le lien le plus encombré du réseau en un temps relativement court. De plus, les performances de l'algorithme sont robustes même lorsque le nombre de liens par secteur augmente. Cependant, la topologie de réseau résultante avait un lien déconnecté dans le réseau. Par conséquent la valeur objectif pour ce réseau devient inférieure au modèle [2]. Pour générer un réseau entièrement connecté, nous avons ajouté une contrainte qui nous aide à réduire le nombre de liens pouvant avoir un débit direct nul. Cela a forcé le modèle adapté à générer une topologie de réseau entièrement connectée. Nous avons également comparé notre algorithme avec [2]. Dans cette comparaison, nous avons observé que la perte de trajet et la marge d'évanouissement affectent de manière significative la qualité de la solution.

ABSTRACT

Environmental factors, including terrain and weather, significantly affect wireless network communication speed. In this research paper, the wireless mesh network topology model [1] was adapted to the wireless tree topology model to solve the thesis paper's general problem [2]. The model considers a sector connectivity graph to define an antenna direction for each node. Also, we used a single-beam antenna for each sector and modeled signal patterns and interference through an ideal cone shape. We also modeled a mixed-integer linear programming problem to find the most congested link in the network. An iterative local search algorithm was used to solve our problem quickly and effectively with Gurobi software.

The proposed algorithm was evaluated using five different problem instances and analyzing a different number of links per sector. The computation showed that the proposed algorithm could find the most congested link in the network in a relatively short time. In addition, the algorithm's performance is robust even when the number of links per sector increases. However, the resulting network topology had a disconnected link in the network. As a result, the objective value for this network was less than the one we found from model [2]. To generate a fully connected network, we added a constraint to help us reduce the number of links that can have zero direct throughputs. This forced the adapted model to generate a fully connected network topology. We also compared our algorithm with [2] and observed that path loss and fade margin significantly affect solution quality.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

PTP	Point-To-Point
PMP	Point-to-MultiPoint
SINR	Signal-to-Noise Ratio
LP	Linear Programming
ILP	Integer Linear Programming
PILP	Pure Integer Linear Programming
MILP	Mixed-Integer Linear Programming
ILS	Iterative Local Search

CHAPTER 1 INTRODUCTION

We use many kinds of technology in our daily lives, many of them making use of wireless communication networks provided by telecommunications. Wireless communication transfers data without a physical connection between two or more nodes. Wireless communication networks use multi-radio waves to connect nodes across the network. Also, wireless communication networks use multi-channels to improve their communication performance in terms of throughput, end-to-end delay, and channel access delay. Such networks have frequent disruptions and failures, which cause the data transmission speed to decrease.

The solution to this problem is to build a high-performance network that considers topology, antenna assignment, channel allocation, and routing constraints. The goal of incorporating all these constraints is to model a network design that will maximize data transmission speeds. This problem has been modeled in various ways by different authors and with different considerations, as we will see in Chapter 4.

1.1 Problem Description

We aim to design a tree network, antenna configuration, and channel assignment to maximize data transmission speed. We aim to adapt the model in paper [1], which is used to solve a wireless mesh topology in order to solve a general wireless tree network problem in [2]. In [1], the authors attempt to model the inner topology of a mesh cluster by optimizing the capacity of the most congested link to maximize the network's data transmission speed. Their network model considers a single-beam sector antenna with fixed alignments and channel assignments with routing. Their signal and interference models depend on a sectored connectivity graph. Conversely, [2] attempts to model a general network problem that includes tree topology, mesh topology, and a combination of both by optimizing the least effective throughput to maximize the network's data transmission speed. In this thesis, the network model considers both single-beam and multi-beam antennas and a channel assignment for each edge. It also includes the physical signal and interference model in the full optimization problem. This paper adapts the method of wireless mesh topology in [1] and uses it to solve the tree topology problem in [2].

The next section discusses the similarities and differences as well as the problem parameters considered in both papers.

Network Topology

Network topology is the systematic arrangement of the nodes in a network and the connections between them, which we refer to as edges. In [2], the network topologies include tree, mesh, and hybrid topologies. In a tree topology, there are many connected nodes arranged like branches, there is at most one edge between any two connected nodes, and those edges do not form any cycles. In a mesh topology, all nodes are connected by more than one edge. When a network topology contains both tree and mesh topologies, we call it a hybrid topology. [1] focuses on the inner part of the mesh cluster. Both papers define the topology as a graph $G(N, E)$ where N denotes the set of vertices and E denotes the set of edges. In [1], the authors wanted to design an inner mesh topology ; however, in [2], the author was interested in designing a global tree topology to maximize the weakest link in terms of data transmission speed. In this work, we adapt the work of [1] in order to solve the problem in [2].

Network Configuration

A network configuration includes a master hub, waveform, and channel assignments to all its edges. A master hub is the central part of the network, which connects all the branches. A master hub also defines a set of successors for every node. Waveforms are a form of communication between nodes in a network. In a network configuration, there are three types of waveforms : point-to-multi-point (PMP), point-to-point (PTP), and mesh. A PMP connection is a one-to-many connection, which has multiple edges from one node to multiple nodes. A PTP connection is a one-to-one connection, with a single edge between two nodes. A mesh connection is a connection between all the nodes in a single mesh cluster, which has multiple edges between nodes. However, in this thesis, we are only interested in the PMP waveform.

A valid waveform assignment must have a valid channel assignment [2]. The process of assigning orthogonal channels to all the nodes within the communication range is called channel assignment, which helps reduce interference. A channel is a range of frequencies allocated for data transmission within the antennas. In the [2] model, every node in a valid channel assignment can use two channels maximum, and each channel has two possible frequencies, for a total of four frequencies that do not interfere with each other. Additionally, two explicit nodes must use different channels. Also, the author models a valid channel assignment as a vertex coloring problem by giving a valid topology and waveform assignment. Similarly, in [1], the authors use four channels maximum and assume no interference between the two different channels. Our model considers that each sector must operate two channels with its own frequencies.

Network Antenna

Two types of antennas are used in the wireless network to exchange data between every connection : single-beam antennas and multi-beam antennas. A single-beam antenna will only transmit and receive signals from one specific direction. There are four types of single-beam antennas : parabolic, panel, omnidirectional, or sector antennas. A multi-beam antenna can transmit and receive signals from multiple directions. In [1], they use a single-beam antenna type, and their sectored connectivity graph predefines the J possible directions of the antennas. [2] models the signal and interference in single-beam antennas using four possible beam widths and any angle $\theta \in [0, 2\pi]$.

Objective

We want to model a tree network topology, channel, and antenna that maximize the weakest link in the network. To model this problem, we first need to measure how the signal-to-noise ratio (SINR) affects the data traffic in the network. To find the SINR, we need to model signal strength and interference. An antenna gain pattern helps determine how strong the transmitted/received signal is from the direction of the antenna alignment to the beam. In [2], the author considers the fade margin and path loss to physically model the signal strength and interference. However, in [1], the authors consider a cone-shaped directional antenna pattern with an equal alignment and an antenna beam width of $360/J$, which is $J = 8$ non-overlapping sectors. There is a perfect signal if the antenna design is in this cone ; otherwise, there is nothing. Then the interference and the signal model become binary variables.

After all this, the author in [2] models the objective function TP_{mn} by considering three different network traffic scenarios that go through edge $(m, n) \in E$: Scenario A is when a single data stream passes between any two nodes. In this case, there is at least one connection between those nodes and the worst direct throughput, so the number of data streams n_{mn} becomes 1. Scenario B is when $|N| - 1$ data streams pass from the master hub to every node or vice versa. Here, all edges do not have the same amount of data, so the problem is finding the most overloaded edge from its direct throughput. Then the number of data streams n_{mn} becomes the number of descendants d_n to every other node, i.e.,

$$d_n = 1 + \sum_{x \ successor \ of \ n} d_x. \tag{1.1}$$

Scenario C is when $|N|(|N|-1)$ data streams pass from every node to every other node. In this case, every node sends and receives signals, so the problem is finding the most overloaded edge from its direct throughput. Then the number of data streams n_{mn} becomes $2d_n(\|N\| - d_n)$.

In [1], the author models the objective function by finding the minimum difference between the aggregate traffic and the effective capacity across all channels and links in the topology. This model is more closely equivalent to Scenario C in [2]. However, in [2], the author measures the maximum amount of data that can be transmitted through each edge. We are interested in maximizing the least throughput by calculating the maximum amount of data transmission through each edge.

Optimization Problem

After all these computations, [2] formulates the following optimization problem for the tree network :

$$\max_{T} \quad \max_{r \in V} \quad \max_{P} \quad \max_{C} \quad \max_{F} \quad \max_{A} \; O, \tag{1.2}$$

where T is a set of valid topologies, P is a set of valid partitions of the master hub node successors, C is a set of valid channel assignments, F is a set of valid frequency assignments, A is a set of valid antenna configurations, and O is given by

$$\alpha(|N| - 1) \left(\min_{[m,n] \in E : n_{mn} \geq 0} \frac{TP_{mn}}{n_{mn}} + \frac{1}{Z} \operatorname*{mean}_{[m,n] \in E : n_{mn} \geq 0} \frac{TP_{mn}}{n_{mn}} \right), \tag{1.3}$$

where (α) is a constant value, and Z is a normalization constant.

In [2], this problem is solved by using a heuristic method. In Chapter 4, we solve this problem approximately by adapting the model and assumptions in [1].

1.2 Contributions

The main contribution of this work is that we adopt a mixed integer linear programming model (MILP) that considers network topology, frequency channel, and antenna assignment for a tree topology. This model includes an algorithm for finding interfering links between nodes. Second, we propose an algorithm that ensures network connectivity and helps to solve the above optimization problem. We include simulation results to validate the effectiveness of our approach.

1.3 Organization

The rest of the thesis is organized as follows. The discussion in Chapter 2 covers the definitions of mixed-integer programming problems and of an iterative local search method. A literature review and related work are provided in Chapter 3. In Chapter 4, we give a mixed-integer programming model and an iterated local search algorithm. The numerical implementation and comparison results are included in Chapter 5. In Chapter 6, we present the conclusions and some directions for future research.

CHAPTER 2 PRELIMINARIES

In this chapter, we briefly introduce the topics of mixed-integer programming and of iterative local searches. The information presented can be used to understand the research topic better.

2.1 Mixed-Integer Programming

An optimization problem is a mathematical problem that aims to find the maximum or minimum value within a given set of parameters. Typically used in engineering, economics, and mathematics, this type of problem involves finding the best solution among all possible solutions by evaluating the objective function value of each one [3]. An optimization problem can be divided into two categories : constrained or unconstrained. In constrained optimization, there are limits or constraints on the maximum or minimum values of the function. In unconstrained optimization, there are no such limits or constraints. One of the possible formulations for constrained programming is the linear programming (LP) problem.

LP problems are used to optimize the value of a function that is subject to a set of linear constraints. These are usually expressed in terms of linear equations and inequalities, which is why the resulting problems are called linear programming problems. LP problems contain a linear objective function, linear constraints, and non-negativity constraints. The general form of LP is

$$\begin{aligned} \min_{x} \text{ or } \max_{x} \quad & cX \\ \text{s.t.} \quad & AX\{\geq, =, \leq\}b \\ & X \geq 0, \end{aligned} \tag{2.1}$$

where $c = (c_1, c_2, \ldots, c_n)^T$, $b = (b_1, b_2, \ldots, b_n)^T$, $X = (x_1, x_2, \ldots, x_n)^T$, and

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}.$$

The most used method to solve LP is the simplex method, an algebraic approach to solving LP problems.

Integer linear programming (ILP) is a mathematical optimization technique. This technique has three different types, namely, pure integer linear programming (PILP), mixed-integer linear programming (MILP), and binary integer linear programming (0-1) [4]. In a PILP, all decision variables only take integer values. In a MILP, only some variables must take integer values. In these problems, the objective is to find an optimal solution that mixes integers and continuous values. In a 0-1 ILP, all the variables take integer values, of either 0 or 1. Such problems occur when there is a need to optimize discrete decisions to satisfy a given set of constraints. The ILP effectively models and solves optimization problems such as scheduling, resource allocation, transportation, and other limited resource decision-making. The ILP is an extension of LP ; however, its integer constraints require more complex algorithms to solve [5]. As a result, all ILP types are classified as NP-hard combinatorial problems, meaning they are significantly more challenging to solve than LP problems.

The branch-and-bound method is the most common approach to solving ILP problems. A branch-and-bound method is an algorithmic approach that divides a problem into subproblems, solves each subproblem by relaxing the integrality requirement and solving an LP, and then combines these subsolutions to determine the optimal solution to the original problem [3]. In this technique, the enumeration of the integer solutions is represented by a tree structure and only the most promising nodes are used to branch at any given time. It keeps track of the best solutions found so far and uses them to determine which nodes should be grown and which should be discarded. This method ensures that the algorithm only explores the most promising solutions, without wasting time on unlikely solutions. In this way, the branch-and-bound method prevents the growth of large trees. The branch-and-cut method uses a branch-and-bound algorithm and a cutting plane method to accelerate the relaxation problem. The Gurobi solver implements this method to solve ILP problems [6]. ILP problems can also be solved using heuristics. In MILP problems, heuristics use problem-specific strategies to find approximate solutions to complex optimization problems. This approach finds all possible solutions that satisfy all constraints and conditions by reducing the problem's time complexity. This approach cannot tell us whether the solution found is the best one or not. However, with branch-and-cut, it is possible to prove optimality within a reasonable time. One of these heuristic methods is described in the following section.

2.2 Iterated Local Search Method

A metaheuristic is a guide or strategy to assist a heuristic algorithm in finding the best solution to a problem [7]. Metaheuristics are designed to provide the best solution to a problem in a reasonable amount of time [8]. They can be used to solve problems with good-quality solutions ; however, they need an efficient way to identify solutions. By combining different heuristics, metaheuristics can achieve good-quality solutions in a shorter period of time, even when the problem has many possible solutions and is challenging to optimize with traditional methods.

Metaheuristics are algorithms that can adapt to changing conditions and environments, meaning that they can solve problems that may not have the same parameters each time. This characteristic sometimes makes them more versatile and efficient than traditional methods. Dividing the algorithm into parts and analyzing each component separately gives better insight into how it works and optimizes more efficiently. Embedded heuristics are often difficult to modify, as they may involve intricate mathematical models and algorithms that, if changed, may lead to unpredictable results. It is therefore essential to ensure that the interface to the object module is designed correctly and intuitively, to apply the heuristic effectively. Generally, metaheuristics can be categorized as either single-solution based (local search) or population based (random search) [7]. Local search metaheuristics are used to find a single optimal solution by making small changes to an existing solution. A local search can meet all these requirements, which can explore a local search space more effectively than other methods, making it more likely to find a good-quality solution. It also allows for the efficient adjustment of parameters, which can be used to fine-tune the search process. On the other hand, random-search metaheuristics are used to find multiple solutions by pseudo-randomly generating new solutions and selecting the best ones.

A local search is a powerful tool for solving complex problems that cannot be solved using traditional methods [9]. However, there may be more effective methods, since a local search can be computationally expensive. Instead, it may be more efficient to use a heuristic search algorithm. For instance, a local search algorithm might attempt to determine the shortest route between two points by beginning with a random solution and then improving it incrementally until the best solution is found. A local search establishes a many-to-one mapping between the set of candidate solutions and the smaller set of solutions to find locally optimal solutions. This type of mapping allows for efficient search space exploration, resulting in a higher likelihood of finding a better solution than the previously established one [8]. A local search aims to find the best solution to a problem by iteratively improving a solution from its current state, by making small, local changes and evaluating the resulting solution. If the resulting solution is better than the current solution, it is accepted, and the process continues until no further improvement can be made. Repeating the search from a different starting point can improve the value found by the local search. For instance, if the first search starts at a local minimum, repeating the search from a different starting point may help the algorithm escape the local minimum and find a better solution. This strategy can be beneficial for finding more effective solutions in a shorter amount of time. Random restarts with independent samplings can sometimes be helpful if the problem is small ; however, such methods are inefficient in large-scale problems.

The iterative local search (ILS) is a technique for finding solutions to MILP problems by starting from an initial solution and using heuristics to improve the solution incrementally [10]. This approach allows for more efficient search space exploration and increases the likelihood of finding a better solution with fewer trials. This iterative local search procedure should produce the best solution when the solution spaces are neither too small nor too large. The ILS algorithm only needs to remember the best-known solution so it has minimal memory requirements [8]. This allows the algorithm to be implemented very efficiently, making it easy to understand and to use. Each new iteration contributes to developing sophisticated optimization techniques, such as iterated descent, chained local optimization, and combinations. This chain of solutions should be periodically updated, allowing for new solutions to be explored as the search progresses, and gradually increasing the algorithm's efficiency. This makes it possible to customize optimization techniques to fit the individual goals and constraints of each problem. However, this process can be time-consuming and may lead to a different solution.

CHAPTER 3 LITERATURE REVIEW

This section briefly covers topics related to our work and highlights their similarities and differences to provide a better understanding of our research. Several recent studies have examined topologies, antenna alignments, channel allocation, and routing algorithms for multi-channel wireless mesh networks. This type of algorithm can help optimize the network's performance by optimizing its usage of resources and energy. They also increase network stability and reliability by improving connectivity and throughput. In the following section, we look at the literature on radio communication networks that is either directly or indirectly connected to our work's objective. Our work is similar to theirs regarding the objective function, channel allocation, signal and interference modeling, and antenna alignments. W also look at how all these articles model their problems and what algorithms they use to solve them.

3.1 Similar Network Design Problems

Here, we view all related works in terms of their objectives and with constraints similar to our work on wireless network design. We compare their objectives and constraints to our own, determining if any solutions can be applied to our wireless network design. We gain insight into the various techniques used to design wireless networks and how they can be applied to our problems. This helps us identify existing solutions gaps and develop new ones that better satisfy our objectives with the given constraints.

A wireless mesh network design, called DMesh, for an internal mesh topology was proposed in [1] ; however, we are interested in designing a wireless tree network design. In their work, the authors proposes a network architecture to improve the performance of wireless mesh networks by incorporating directional antennas with multiple orthogonal channels. This architecture can take advantage of the increased throughput of directional antennas and the decreased interference of having multiple orthogonal channels, thereby improving the overall performance of the mesh network. The paper aims to maximize the capacity of the most congested links to optimize the network topology. By doing this, the network can better accommodate traffic flow and reduce the risk of network overloads and outages. Optimizing the network topology can also increase data transmission speeds and latency efficiency. Our model calculates the maximum throughput achieved per edge, while theirs simulates traffic down to the packet level. This is the same as our objective, except that we are interested in modeling the throughputs for each connection. Their proposed network uses single-beam antennas with fixed possible alignments and a beam width of 45°. This setup ensures that each antenna can extend in eight different directions and that the signal will be strong enough for the network to function correctly. Having a fixed beam width also reduces interference from other devices. The antenna signals are modeled as ideal cones with a specific radius, to compute both the signal and interference models as binary variables. This is because cones are the most efficient shape for antennas to focus the signal and radiate it in a specific direction with minimal loss. We use the same idea to model our problem. Their model assumes that the two channels have different frequencies and that the signal sent through one channel is not affected by the signal transmitted through the other channel. We take this idea into account ; however, in addition to this, we also consider that two nodes may share another node that uses a different channel and that two different channels can interfere with one another. The authors of [1] formulate the whole problem as a mixed integer programming problem and use an iterated local search algorithm to solve it with the MOSEK software solver. The algorithm is designed to generate a feasible solution in a reasonable amount of time while accounting for any constraints. The software solver finds the sub-optimal solution based on the parameter set.

In [11], an inner topology design for the wireless mesh network has been studied. The objective is to maximize network capacity as a function of the demand rate that each traffic stream can meet, carefully balancing the resources used to meet the demand of each traffic stream. This is done by considering the number of users, the demand rate of each user, the characteristics of each user's traffic stream, and the resources available. The network can maximize its capacity by allocating resources to meet the demand rate of each traffic stream. Here, the authors assume that the higher the capacity, the smaller the throughput. Higher capacity means more resources are needed to process any given task, resulting in a decreased data processing rate. As the capacity increases, the resources are spread over a more significant number of functions, which reduces the throughput. Their work is equivalent to our main objective since they define network capacity in terms of network throughput. Their model treats signal and interference as binary variables for single-beam antennas with 6 to 12 fixed angles and a fixed width of 60°. This model allows them to quickly and accurately predict the signal and interference power levels in a given environment, making designing antennas that can provide reliable signal coverage easier and faster. This means that the signal and the interference can be present or absent within a given beam, allowing the model to simulate how the antenna behaves in various realistic scenarios. For example, when a signal is present, the antenna can be modeled to determine how strong the signal is and how much of it is reflected or absorbed. When interference is present, the model can be used to simulate how much of the signal is distorted or blocked by the interference. We also model the signal and interference as binary variables. According to the authors, an orthogonal frequency channel consists of a fixed number of beams, each not interfering with the others but only with itself. This is because the beams are designed to be orthogonal to each other, meaning that they are perpendicular and do not overlap. This allows for multiple beams to be sent or received simultaneously without the risk of interference. Each channel has its distinct frequency, enabling the data to be transmitted accurately and efficiently. The data is sent over multiple channels simultaneously, each with a different frequency. The data can be sent more quickly and accurately using multiple frequencies, as it is less likely to be interfered with or lost in transmission. We take this idea into account ; however, in addition to this, we also consider if two nodes share one another node that must use different channels and that two different channels can interfere with one another. The authors of [11], formulate the problem as an LP problem and use a delayed column generation algorithm with the Cplex software solver.

[12] proposes an inner topology design for a wireless mesh network. This work aims to determine the optimal beam configuration for multi-beam antennas that maximizes throughput between nodes. Since we are interested in using single-beam antennas, this is different from our goal. Their study analyzes the angles and directions of the multiple beams to determine which design will yield the best performance in terms of throughput and throughput stability. Additionally, they evaluate the impact of the beam configuration on other aspects, such as power consumption and interference. These beam configurations enable the highest possible data transfer speeds between nodes, optimizing performance and reliability. By creating multiple beams between the nodes, the data can be sent in various directions at once, which increases the amount of data that can be sent at a given time. Additionally, the increased number of beams increases the reliability of the data transfer, as any disruption to one beam will not affect the other beams. However, this paper aims to maximize a single antenna's activated beam to minimize network throughput. This is because a single antenna's beam is more focused and efficient than multiple antennas and can therefore achieve a higher data rate with fewer resources. Additionally, focusing the beam on the intended receiver reduces interference from unintended receivers and improves network performance. Maximizing the activated beam for a single antenna can increase the network's overall capacity. This helps maximize the network's overall throughput by ensuring each connection can achieve its maximum potential. In this sense, the idea of optimizing the network throughput becomes the same as our work. In [12], the authors assume a cone shape for the antenna pattern and a beam width of $360°/M$, where M is the number of beams. This assumption was made for simplicity, allowing the authors to calculate the antenna gain of each beam without including side lobes. Additionally, a beam width of $360°/M$ ensures that adjacent beams will not interfere with each other. The assumption of a cone shape allows the authors to consider

the antenna's power gain in a specific direction. We also assume this when modeling an antenna pattern to minimize interference and signal. They consider single channels with no interference, implying that there is only one path the data can take to pass. In addition, no other signals can interfere with a signal. This makes the channel simpler and easier to model, eliminating the need to consider interference that could distort the signal. It also makes the system more reliable, as there is no chance of signal disruption. We consider this idea ; however, we also consider that if two nodes share one another, they must use different channels and that two different channels can interfere. In [12], they formulate this problem as a mixed integer programming problem and use an LP rounding algorithm to solve it.

In [13], the author focuses on designing the inner topology of a wireless mesh network. This work optimizes channel utilization with radio interfaces that maximize the network throughput. Maximizing network throughput is our primary objective. Maximizing channel utilization can help accommodate the increased number of users and data traffic the network produces. This helps ensure that users can access the network without experiencing delays or disruptions. In this work, the authors use a single-beam antenna as we do. This offers straightforward interface assignments because the energy is uniformly distributed. Since the strength of the signal is the same in all directions, it is easy to assign an interface to the antenna. Like in our model, the authors consider channel assignment, but only in the sense that two different channels don't interfere. This makes it less likely that two neighboring nodes will use the same channel, resulting in interference that would reduce the network's reliability. The aim is to minimize interference on any given channel by spreading it across as many channels as possible. This improves the overall quality and performance of the network. They use a link conflict weight to model interference physically. This measures the amount of interference experienced by a link due to other links in the network. It considers the power level of the interfering links, the distance between them, and the type of modulation used. This information is used to calculate the signal-to-interference ratio for each connection, which is then used to determine the optimal network configuration. The authors model their problem as an integer linear programming problem and use a heuristic algorithm to solve it.

[14] is also about designing a wireless mesh network by optimizing network throughput to maximize data transmission speed. Optimizing the network throughput can better manage the resources needed for data transmission and minimize network congestion. This increases the data transmission speed and allows for more efficient network communication. This is equivalent to our maximization of the most congested link. The proposed algorithm can identify the most effective routing data by focusing on the most congested link. By considering the likelihood of congestion in different connections, the algorithm can determine the most efficient route for data traffic. It also finds any changes in the network, such as traffic or

congestion, which can further improve the efficiency of the route. This allows for the most efficient data transmission, maximizing the overall network throughput. This transmission type breaks data into smaller packets that can be sent over multiple paths simultaneously and reassembled at destination. This reduces the chance of packet loss and improves the overall transmission speed. However, like paper [1], they simulate traffic for the whole network. This means that the network can be evaluated on a packet level rather than analyzing each node or link. It also allows for simulating large-scale networks with multiple sources and destinations. Like our model, their model assumes a single-beam antenna with a switch beam directional antenna with a fixed transmitting or receiving direction. This allows the signal to be focused in one direction, which increases the signal strength and reduces signal interference from other sources. This can help improve the accuracy of the model. Also, they model the signal pattern of these antennas as cone-shaped within a specific range r. This allows the model to accurately predict the antenna's signal strength at different points in space by considering the distance from the antenna and the radiation angle. If the antenna is inside the cone, there is a perfect signal ; otherwise, there is no signal. The cone acts as a barrier preventing interference. If the antenna is placed inside the cone, the signal is not blocked and can reach the receiver without disruption. The authors model interference as a binary variable by applying the cone shape, which represents the area of interference of a given antenna, allowing binary variables to represent the presence or absence of interference. This model determines which antennas can be used in a given network. We use the same idea for modeling the signal and interference. The work in [14] considers channel assignment, as does our work, but, only in the sense that using the same frequency channel can cause interference. By using the different frequency channels, multiple radios can coordinate to avoid interfering with each other. This reduces the amount of interference that can occur and allows multiple radios to communicate without any issues. In [14], they also consider that interference reduction is proportional to the number of radios assigned to the different frequency channels. This is because multiple radios using different frequency channels have a lower signal-to-noise ratio, reducing the interference between them. In the end, they formulate their problem as a MIP problem and use a CPLEX solver.

3.2 Different Network Design Problems

Here, we review papers focusing on wireless network design problems but with different objectives than ours. These papers provide valuable insights into wireless network design and offer a wide range of approaches that can be applied to different scenarios. We explain how their approach differs from ours.

[15] optimizes a response time and makespan of a spanning tree network. The algorithm works by considering a rooted spanning tree, finding a response time for each node, and then using an iterative approach to reduce the maximum response time while still keeping the network connected. This yields an optimized response time and makespan for the network. The authors can increase the signal strength within a network by using omnidirectional antennas since the antennas can radiate and receive signals from all directions. This, in turn, increases the network's reliability, as data is less likely to be lost. It also reduces power consumption, as the antennas require less power to transmit and receive signals. They introduce a heuristic algorithm to solve this problem.

In [16], the microwave-based wireless backhaul networks installation cost, modeling their signal and interference as binary variables using omnidirectional antennas was optimized. This approach guarantees that the network topology obtained is feasible with respect to the signal strength and interference constraints. Additionally, it reduces the problem's computational complexity, making it easier to solve. They formulate the problem as an integer programming problem, using a heuristic algorithm to solve it.

[17] maximizes traffic capacity for a backhaul topology network. The algorithm uses a Bayesian optimization approach to determine optimal traffic routing for a given topology. By utilizing this approach, it can find the most efficient paths for traffic, maximizing network capacity. The algorithm then uses a search algorithm to determine the optimal routing for each traffic path, thus finding the most efficient path for each connection and maximizing the network's capacity.

[18] proposes a topology control approach for inner wireless mesh networks by using multiple directional antennas to reduce interference while the network is always connected. This approach uses directional antennas to minimize interference by allowing for better signal coverage and more efficient use of the available spectrum. It also increases throughput due to decreased interference and the ability to maintain connection even when nodes are in motion. They use a minimum degree spanning tree to compute interference. This algorithm finds the shortest paths between nodes while minimizing the number of edges. Using this algorithm, each node's interfering links can be reduced since the interference of the directional antennas can be minimized by finding the shortest paths between nodes.

In [19], the cost of designing a wireless backhaul network to increase network performance and reliability was minimized. They use the two-phase network design model from the literature. In the first phase, they design a tree topology network with minimum cost to maximize the network performance. In the second phase, they recover the network from disruption to improve reliability.

[20] also designs a minimum cost backhaul wireless network deployment to satisfy demands. A backhaul wireless network deployment is a system of wireless lines that carry data from one point to another. To minimize cost, it is necessary to find the most efficient and cost-effective solution for network deployment that satisfies all the demands. This involves finding the best route for the network and the most cost-effective hardware and software to use. They use a heuristic algorithm with capacitated facility location algorithm to solve the problem.

In [21], the authors design a network topology in order to minimize cost, reliability, throughput, and traffic pattern simultaneously. They apply an analytical hierarchy process to make a decision for that criterion, which helps to identify which criterion is the most important in the resulting network topology by assigning weight. Then by using their weights, the optimum topology will be found.

[22] proposes a minimum spanning tree topology control algorithm for wireless multihop networks to minimize network interference while maximizing network capacity with an omnidirectional antenna. The algorithm attempts to minimize power consumption by selecting a set of nodes to use as relays while still allowing the network to span the entire area. This is done by finding the path with the minimum power cost while connecting all nodes.

[23] provides and analyses different techniques to design a wireless sensor network. The article examines various design techniques, including topology control, coverage, and data dissemination. It provides an in-depth analysis of the pros and cons of multiple types of hierarchical routing for sensor networks. Moreover, the authors explore the potential application of the techniques in different scenarios.

[24] focuses on maximizing the data transmission speed for the LoRa network by adapting the idea of tree and star topology. Using a tree topology, the energy consumed by each node can be minimized as the data is routed through the nodes in the tree. Furthermore, the star topology allows for more efficient data transmission and provides better coverage.

The authors in [25] focus on maximizing the network lifetime to optimize the connection data rate in the wireless sensor network. They model clustering techniques with data routing as a joint mutual coupling problem. This mutual coupling problem is solved by using the consensus-based iterative algorithm. This ensures the optimization of the network performance while providing a low-cost solution to the problem. The results show a significant improvement in network performance.

[26] proposed a hierarchical distributed hash table called cyclone, to provide optimal routing hops to minimize network scalability problems. Cyclone uses a distributed hash table (DHT) to provide an efficient lookup mechanism for routing. It uses a hierarchical structure, with

each node in the hierarchy responsible for routing requests to the next level of the hierarchy. This reduces the number of hops required to route a request and thus improves network scalability.

The authors in [27] model a mobile wireless sensor network in order to maximize the traffic and end-to-end connection. They use a velocity energy-efficient and link-aware cluster tree approach to model this. Their approach allows for more reliable connections, more efficient use of energy, and improved packet delivery rates while reducing the amount of data that needs to be transmitted.

CHAPTER 4 PROBLEM MODELING

In this chapter, we first look at the old formulation and then our adapted formulation to formulate the topology, channel configuration, and antenna alignment in a joint MILP. We also discuss the algorithm we use to solve this problem.

4.1 Network Model

A fully configured and optimized network is the desired outcome of our problem. The network can achieve the highest level of performance, throughput, and reliability by configuring and optimizing the topology, channels, and antenna configurations.

4.1.1 Network Topology

Topology describes how nodes are connected and how data is transferred between them. It is the physical or logical arrangement of the nodes in a network and how those nodes interact with each other. We consider a connected graph $G(N, E)$, where N represents the set of nodes, and E represents the set of links between nodes. The link between two nodes is established if they are within communication range. The link from m to n is written as (m, n), and the links are symmetrical, i.e., $(m, n) = (n, m)$.

Antenna Configuration

Suppose each node $n \in N$ has J different antenna directions, and each antenna forms $360°/J$ beams. For example, when $J = 4$, each node has four different sector antennas that can form a beam of $90°$. We suppose we have a sectored connectivity graph $G(N, E)$ to model the antennas for our network. To identify which links are in which sectors, we assume we have an $N \times N$ binary location matrix $\theta \in [0, 2\pi]$.

Channel Configuration

We assume that each sector uses two different available orthogonal channels. If links in one sector use one communication channel, the other links in different sectors cannot use that channel, and one link can only use one channel to communicate. However, links in the same sector can use the same channel. Also, if two nodes share another node, they must be on different channels. This causes interference between the two links. Before we present our MILP problem, we introduce the interference model.

Interference Model

We can calculate the signal strength by taking the distance between every pair of nodes sharing the same channel; even if the nodes do not communicate directly, their signal strength can still be determined. We assume an ideal cone-shaped configuration, with a $360°/J$ equal beam, to model the signal strength and interference model using a directional antenna pattern. This model can provide efficient and accurate coverage in the intended area. Suppose the sectored and omnidirectional antennas have the same interference and communication range. An interfering link occurs when two or more transmitters send their signals to the same receiver or when the receiver is within the interference beams of both transmitters. In our case, we model both the signal strength and the interference model as a binary variable, i.e., if the signal is inside the cone shape, we assume we have a perfect signal; otherwise, we have no signal. Here, we must find which links are interfering with each other. To find the interfering links from the connectivity graph, we use the algorithm in [1], where F_{mn} represents the interfering matrix that links (m, n).

4.1.2 Problem Formulation

Now, we formulate the problem as a MILP by providing the sectored connectivity graph, interfering links, channel allocation, and antenna assignment assumptions.

Channel Constraint

Suppose any two nodes, m and $n \in N$, communicate with each other over channel $i = \{1, 2, \ldots, C\}$. Then the unidirectional link channel allocation variable X^i_{mn} is defined as a binary variable, i.e., if there is a link between node m and node n over channel i, the link channel allocation variable becomes 1; otherwise, it becomes 0. We assume that, for two nodes to communicate, they should use the same frequency channel, i.e.,

$$X^i_{mn} = X^i_{nm}, \forall\ m, n \in N, (m, n) \in E, \forall\ i = 1, 2, \ldots, C, \tag{4.1}$$

where C is the number of available channels.

To align the antenna, since we have already defined its direction using the sector, we only need to look at which sector is appropriate for the two communicating nodes. The sector channel allocation variable y^i_{mj} is also defined as a binary variable, i.e., if node m uses sector j over channel i, then the sector channel allocation variable becomes 1 ; otherwise, it becomes 0. According to [1], since both X^i_{mn} and y^i_{mj} are integer variables, to reduce the MILP complexity, the variable y^i_{mj} converts to a continuous variable. The sector channel allocation constraint is as follows :

$$0 \leq y^i_{mj} \leq \sum_{\substack{n \in N: \\ (m,n) \in E, \\ \theta(m,n)=j}} X^i_{mn}, \qquad \forall m \in N, \forall i = 1, 2, \ldots, C \text{ and} \tag{4.2}$$

$$X^i_{mn} \leq y^i_{mj} \leq 1, \qquad \forall n \in N, (m,n) \in E, \theta(m,n) = j \tag{4.3}$$

To minimize interference between nodes, the authors in [1] assumed that each network interface card operates on a separate channel in multi-radio multi-channel wireless network communications. Therefore it is necessary to limit the number of channels per node to the number of NICs available. This constraint is written by

$$\sum_{i=1}^{C} \sum_{j=1}^{J} y^i_{mj} \leq I \quad \forall m \in N \tag{4.4}$$

Additionally, if two links are located in two different sectors, they cannot be assigned to use the same channel. This can be written as

$$\begin{aligned} X^i_{mn} + X^i_{mp} \leq 1, \forall\, m, n, p \in N, (m,n), (m,p) \in E, \\ \forall\, \theta(m,n) \neq \theta(m,p), n \neq p, \forall\, i = 1, 2, \ldots, C. \end{aligned} \tag{4.5}$$

However, they may use the same channel if both links are located within the same sector. In order to maximize the capacity [1] restrict the number of links that may share the same channel in one sector by

$$\sum_{\substack{n \in N, \\ (m,n) \in E}} X^i_{mn} \leq 2, \forall\, m \in N, \forall\, i = 1, 2, \ldots, C. \tag{4.6}$$

Capacity Constraint on a Link

The total capacity is the maximum amount of traffic passing through the network. Assume that node m and n communicate with each other from the direction of m to n over channel

i ; then, the capacity of these two links becomes :

$$c^i_{mn} \leq x^i_{mn} c^0, \forall\ m,n \in N, (m,n) \in E, \forall\ i = 1,2,\ldots,C, \tag{4.7}$$

where c^0 is the nominal data rate of links defined by IEEE 802.11a standards [1]. In other words, if nodes m and n do not communicate with each other from the direction of m to n over channel i, then the total capacity of the link between m and n becomes 0. If nodes p and q are in the interference range of link (m,n) over channel i, then the total capacity between those two links must be less than or equal to the nominal data rate, i.e.,

$$c^i_{mn} + \sum_{\substack{p,q \in N: \\ (pq) \in F_{mn}}} c^i_{pq} \leq c^0, \tag{4.8}$$

Flow Constraint on a Link and for Each Node

Let $a^{sd}_{mn,i}$ be a binary routing variable, i.e., the routing variable is set to 1 if the traffic uses link (m,n) on channel i to get from source node s to destination node d, and is set to 0 if the traffic does not use link (m,n). This variable is not symmetrical. A pair of nodes can be connected through multiple links. Data passing through those links have different latencies for two nodes connected by multiple links using different channels. To avoid packets arriving out of delivery, it is important to use a single link for each flow in order to ensure that the packets arrive in the right place. This constraint is written by

$$\sum_{i}^{C} a^{sd}_{mn,i} \leq 1 \qquad \forall s,d,m,n \in N, (m,n) \in E. \tag{4.9}$$

The aggregate traffic λ^i_{mn} is the sum of all the individual traffic flows, and the traffic between source s and destination d using the link (m,n) from the direction of node m to node n over channel i is given by

$$\lambda^i_{mn} = \sum_{s,d \in N} a^{sd}_{mn,i} \times \gamma^{sd} \qquad \forall m,n \in N, (m,n) \in E, \forall i = 1,2,...,C \tag{4.10}$$

where γ^{sd} is the expected traffic rate between any source and destination nodes. If the aggregate traffic exceeds the effective capacity, the network will become congested and unable to process the traffic. Then, this constraint becomes

$$\lambda^i_{mn} \leq u_1 \times c^i_{mn} \qquad \forall m,n \in N, (m,n) \in E, \forall i = 1,2,\ldots,C \tag{4.11}$$

where u_1 is the maximum link utilization parameter. To ensure network connectivity between any two nodes, we use the difference between the incoming and outgoing flows ; this constraint is given as follows :

$$\sum_{\substack{n\in N\\(m,n)\in E}}\sum_{i=1}^{C} a_{mn,i}^{sd}\times\gamma^{sd} - \sum_{\substack{n\in N\\(nm)\in E}}\sum_{i=1}^{C} a_{nm,i}^{sd}\times\gamma^{sd} = \begin{cases}\gamma^{sd}, & \text{if } s=m\\ -\gamma^{sd}, & \text{if } d=m\\ 0, & \text{Otherwise.}\end{cases} \tag{4.12}$$

Objective Function

All the above constraints determine the feasible region, including the total capacity and the aggregate traffic of nodes m and n. If the total capacity of nodes m and n is greater than the aggregate traffic of nodes m and n, then the resource will be wasted. If the total capacity of nodes m and n is less than the aggregate traffic of nodes m and n, the network's total capacity must be increased to accommodate the traffic. We must assess the balance between resource availability and demand to ensure efficient traffic.

Let δ_{min} be the minimum difference between the total capacity and the aggregate traffic for each link over channel i. If this difference is not compromised, the connection becomes overloaded and congested, leading to slower speeds and increased latency. In this case, δ_{min} provides the most congested link in the network. This can be written as

$$\delta_{min} \leq (u_1 c_{mn}^i - \lambda_{mn}^i) + u_1 c^0 (1 - X_{mn}^i), \qquad \forall m,n\in N,(m,n)\in E, \forall i=1,2,...,C \tag{4.13}$$

which is defined as

$$\delta_{min} = \min_{\substack{m,n\in N,(m,n)\in E\\ I=1,2,...,C}} (u_1 c_{mn}^i - \lambda_{mn}^i) \tag{4.14}$$

Our objective is to maximize the most congested link in the network by balancing the total capacity and the traffic demand for each link. Then the problem becomes a max-min fairness problem, which is different from maximizing the minimum value problem, which is a measure to judge whether allocations are fair or not. The model helps distribute the load across the network more evenly, resulting in better overall network performance. Then, the objective function becomes

$$\max_{X,y,c,a,\lambda,\delta_{min}} \quad \delta_{min} \tag{4.15}$$

Hop Count Constraint

By reducing the loaded links, the model makes the balance between the roughing path ; however, this model may choose the longest path for each source s and destination d, so it is important to find the shortest path between the (s, d) pairs. Let h_G^{sd} be the minimum hop count between each (s, d) pair. Then this constraint is defined by

$$\sum_{\substack{m,n \in N \\ (m,n) \in E}} \sum_{i=1}^{C} a_{mn,i}^{sd} \leq S h_G^{sd},, \qquad \forall s, d \in N \tag{4.16}$$

where $S \geq 1$ is an upper bound on the stretch factor. If $S = 1$, the algorithm becomes the shortest path problem ; otherwise, this constraint becomes relaxed, which also helps get a bigger feasible region [9]

4.1.3 Adapted Constraints

We assume that each node uses a maximum of two different channels. The constraint becomes

$$\sum_{i=1}^{C} \sum_{j=1}^{J} y_{mj}^{i} \leq 2 \quad \forall m \in N. \tag{4.17}$$

In addition to the assumption made in [1], we assume that two nodes must use different channels if they share another node. This assumption allows us to accurately measure the severity of the interference on the common node, so we can best make informed decisions on mitigating it. We model this constraint by

$$\sum_{\substack{n \in N, \\ (m,n) \in E}} X_{mn}^{i} \leq \begin{cases} 1, & \exists n : (m,n) \in E : X_{mn}^{i} (\sum_{\substack{o \in N: \\ (n,o) \in E}} X_{no}^{i}) \geq 1 \\ 10, & \text{Otherwise.} \end{cases} \tag{4.18}$$

The above constraint also ensures that if nodes m and n communicate over channel i and if node n has another connection with node o, then link (n, o) cannot use the same channel i with link (m, n). This helps reduce interference and allows for optimal use of the available channels. It also ensures the maximum possible performance of the network by avoiding interference. However, this constraint is not practical, so we cannot use it in our MILP model. Instead of this model, we prefer to use the following linear equation. We assume that the links on the same channel i connected to node m have a maximum of M links ; we want to try when a number of links $M = 2, 3, \ldots, 10$, and see how good they are according to [2]'s objective function. This constraint becomes

$$\sum_{\substack{n \in N \\ (m,n) \in E}} X^i_{mn} \leq M, \qquad \forall m \in N, \forall i = 1, 2, \ldots, C. \tag{4.19}$$

To calculate the traffic flow between a pair of nodes over channel i, assume that one link can only use one channel. This means that if a node wants to send traffic to another node over channel i, then that link can only use one channel and not any other. This will ensure that the traffic flow between the two nodes over the specified channel is accurately calculated, which is constrained by the following :

$$\sum_{i=1}^{C} X^i_{mn} \leq 1 \qquad \forall m, n \in N, (m,n) \in E. \tag{4.20}$$

The constraints in equation 4.9 and 4.16 limit the number of paths that can be taken between nodes, which means that they can prevent the system from reaching a tree topology. Therefore, these constraints are unnecessary for our work and can be eliminated. By eliminating these constraints, the system can explore more paths and reach the tree topology more efficiently.

The following constraint ensures that the topology found by our adapted MILP prevents the output network from having disconnected edges that do not transmit or receive any data from the rest of the network. We define a binary matrix R_{mn} that is 1 if nodes m and n are connected ; otherwise, the value of R_{mn} becomes 0. Then by considering the possible edges, we can add the following constraint :

$$X^i_{mn} \leq R_{m,n}. \tag{4.21}$$

This ensures that the output network is connected and will remain connected even after the MILP optimization.

Optimization Problem

We formulate the problem as MILP by giving the sectored connectivity graph $G(N, E)$ and all decision variables and parameters. The decision variables and parameters involved in the optimization problem are summarized in Table 4.1 below.

Decision variables and parameters	Description
c^i_{mn}	Total capacity of link m to n on channel i
c^0	Nominal data rate links
$a^{sd}_{mn,i}$	Routing variable
λ^i_{mn}	Aggregate traffic on link m to n on channel i
u_1	Link utilization
γ^{sd}	Expected traffic rate between s and d
h^{sd}_G	Minimum hop count between each (s, d) pair
S	Upper bound on stretch factor

Table 4.1 Summary of decision variables and parameters.

Then the complete MILP problem is written as follows :

$$\begin{aligned}
&\max_{X,y,c,a,\lambda,\delta_{min}} \quad \delta_{min} \\
&\text{s.t.} \quad X^i_{mn} = X^i_{nm} \\
&0 \leq y^i_{mj} \leq \sum_{\substack{n\in N,\\ (m,n)\in E,\\ \theta(mn)=j}} X^i_{mn} \\
&X^i_{mn} \leq y^i_{mj} \leq 1, \quad \forall\theta(mn) = j \\
&\sum_{i=1}^{C}\sum_{j=1}^{I} y^i_{mj} \leq 2 \\
&X^i_{mn} + X^i_{mp} \leq 1, \quad \forall\theta(mn) \neq \theta(mp),\ n \neq p \\
&\sum_{\substack{n\in N,\\ (m,n)\in E}} X^i_{mn} \leq M \\
&\sum_{i=1}^{C} X^i_{mn} \leq 1 \\
&c^i_{mn} \leq X^i_{mn} c^0 \\
&c^i_{mn} + \sum_{\substack{p,q\in N,\\ (pq)\in F_{mn}}} c^i_{pq} \leq c^0 \\
&\quad \lambda^i_{mn} = \sum_{s,d\in N} a^{sd}_{mn,i} \times \gamma^{sd} \\
&\lambda^i_{mn} \leq u_1 \times c^i_{mn} \\
&\quad \sum_{\substack{n\in N,\\ (m,n)\in E}} \sum_{i=1}^{C} a^{sd}_{mn,i} \times \gamma^{sd} - \sum_{\substack{n\in N,\\ (nm)\in E}} \sum_{i=1}^{C} a^{sd}_{(nm),i} \times \gamma^{sd} = \begin{cases} 1, & \text{if s = m} \\ -1, & \text{if d = m} \\ 0, & \text{Otherwise} \end{cases} \\
&\delta_{min} \leq (u_1 \times c^i_{mn} - \lambda^i_{mn}) + u1 \times c^0(1 - X^i_{mn}) \\
&\text{where} \\
&X^i_{mn}, a^{sd}_{mn,i} \in (0, 1), 0 \leq y^i_{mj} \leq 1, 0 \leq c^i_{mn} \leq c^0, \lambda^i_{mn}, \delta_{min} \geq 0,
\end{aligned} \tag{4.22}$$

The above MILP problem can be solved for small instance networks of $|N| \leq 10$. Solving MILP problems with small networks is relatively straightforward and can be done with existing commercial LP/MILP solvers [9]. However, when the size of the problem increases, solving a MILP becomes increasingly difficult due to the complexity of the problem. As the number of nodes in the network increases, the number of variables and constraints in the MILP model also increases, which makes it harder to solve the problem in a reasonable amount of time. Additionally, MILP solvers can become computationally expensive, meaning they require more computing power to solve larger MILP problems. Thus, finding the optimal solutions for large instance networks of $|N| > 10$ is challenging due to the exponential solution space size, meaning the size of the topology, the channel and frequency assignment, and the antenna alignment [2]. A large instance network's topology consists of the number and positions of nodes, and the connections between them, making the solution space vast. As the network size increases, the combinations of these variables increase exponentially, leading to an exponentially large solution space. Channel/frequency assignment is a generalization of the graph coloring problem, which is NP-hard. This means that the graph coloring problem is computationally intractable and cannot be solved in polynomial time. In addition, the problem is so complex, even if the algorithms are exponentially fast, they cannot solve the problem in a feasible amount of time. The solution space for antenna alignment is exponentially large when the problem size is large. This is because the number of variables that need to be considered, such as the antenna's angle, height, and type, its position relative to the ground, and the environment, all contribute to the complexity of the problem. As a result, it is challenging to find a solution that satisfies all the criteria. Additionally, the number of variables may change depending on the type of antenna being used, which further increases the complexity of the problem. Additionally, the number of possible frequencies and antenna alignments that need to be considered for each node and link rises exponentially with the size of the network.

In addition to the solution space problem, interference and signals can also have a global impact across the entire network. Interference affects the network's performance, which can impact the speed and reliability of the connection. It also impacts the amount of data that can be transmitted and received, which can have a global effect on the network as a whole.

Heuristics and approximation algorithms often tackle the above problems [2]. However, these methods may not guarantee the global optimal solution, as they depend on the initial solutions' quality. However, they can help determine the suboptimal solution for a large-scale problem. Here, for our problem, we use a simple and efficient metaheuristic method called the ILS algorithm [9]. The ILS algorithm works by first randomly selecting a solution to the problem and then applying local search methods to refine the solution. Compared to other metaheuristic methods, it explores the search space while finding the best solutions with fewer iterations. This method has proven effective in finding the best approximate solutions to complex optimization problems.

4.2 Algorithm Formulation

We solve this MILP problem by using ILS in [9]. We choose a fully connected topology with random channel i as a starting point. At each iteration, we choose a random pair of nodes $u, v \in N$, and solve the MILP with the two additional constraints. These additional constraints help to ensure that the optimal solution to the MILP is a valid assignment of the two nodes, u and v, in the network. This helps reduce the running time of the algorithm, as well as improve the accuracy of its results. Also, it helps relax all integer constraints on the routing variables. Moreover, all link channel-allocation variables are fixed to their values from the previous iteration, except those with m or n in u, v. This ensures that the channel-allocation variables are not changed during the iteration process unless they are specifically identified by the variables m and n in u and v. The routing variables are converted from their actual real values to Boolean values using a greedy algorithm in steps 8 to 17. The greedy algorithm works by assigning a Boolean value to a routing variable based on the current values of the other variables. The algorithm then assigns a Boolean value to the next variable, and so on, until all variables have been assigned a Boolean value. This process helps reduce the search space and leads to an optimized solution. This simplified problem reduces the problem's complexity and makes it possible to find the suboptimal solution more quickly. By restricting the search space and reducing the number of variables, the problem can be solved more efficiently. This is because the number of non-optimal solutions to evaluate is decreased, so the computation time is faster, and the solution is more accurate. We use GUROBI software to achieve the best possible solution [6].

Algorithm 1 Iterative Local Search

K is the maximum number of iterations.
Put a random channel i for all links.
for $k = 1$ to K **do**
    Randomly choose any two nodes $u, v \in N$ s.t. $(u, v) \in E$
    Solve the MILP problem subject to
        $0 \leq a^{sd}_{mn,i} \leq 1, \forall s, d, m, n \in N, e_{mn} \in E, \forall i \in \{1, 2, \ldots, C\}$
        $X^i_{mn} = X^i_{mn}[k], \forall m, n \in N, e_{mn} \in E, m, n \notin p, q.$
    Set the current value of the link channel variable from the previous iteration.
end for
for each source and destination pair (s, d) **do**
    set $m = s$.
    while $m \neq d$ **do**
        set $a^{sd}_{mn,i} = 1$ where $i, n = \text{argmax}_{i,n}\, a^{sd}_{mn,i}$.
        set $m = n$.
    end while
    for all $m, N \in N$ and all channels $i \in \{1, 2, \ldots, C\}$ **do**
        if $a^{sd}_{mn,i} \neq 1$, then set $a^{sd}_{mn,i} = 0$,
    end for
end for

CHAPTER 5 NUMERICAL EXPERIMENTS

In this section, we design experiments to compare the proposed algorithm with the method in [2] and analyze the results to evaluate the effectiveness of our method.

We used five different network instances given to us by our industrial partner. Each network had 10 connected nodes within an area of 500 m $\times$ 500 m. We assumed that each node has four non-overlapping sectors with a directional antenna beam of 90° and has two frequency channels with four orthogonal channels per sector. Also, the data rate we used for this experiment was 54Mbps with the IEEE 802.11a standard and the upper-bound link was 0.8.

To evaluate the performance of our method and find out which topology is most congested, we changed the maximum number of links per sector for each instance. For each instance, we ran the algorithm nine times. The simulation took five minute per instance. For each, we compared the value of δ_{min} at each iteration and took the maximum value obtained in each instance.

Our algorithm generated a tree topology for each instance with channel edges respecting all the given constraints. At each iteration, we found that the value of δ_{min} tells us which edge experienced the most congestion. Figures 5.1-5.5 represent each instance with its tree topology.

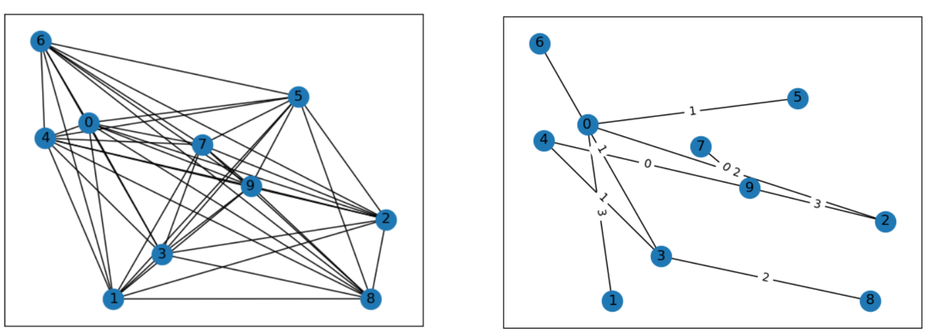

FIGURE 5.1 Instance-1 with its output tree topology

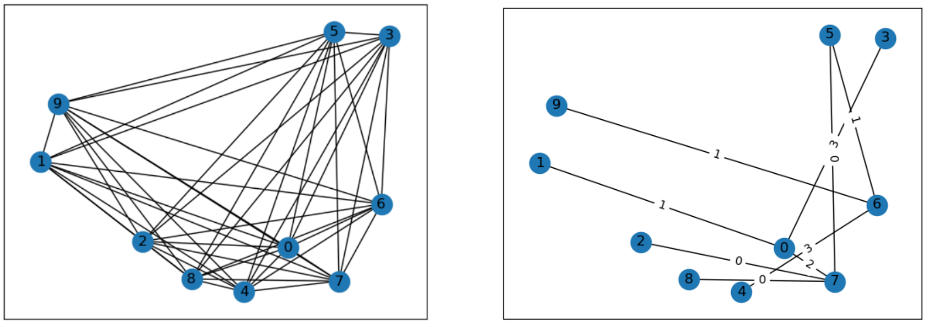

FIGURE 5.2 Instance-2 with its output tree topology

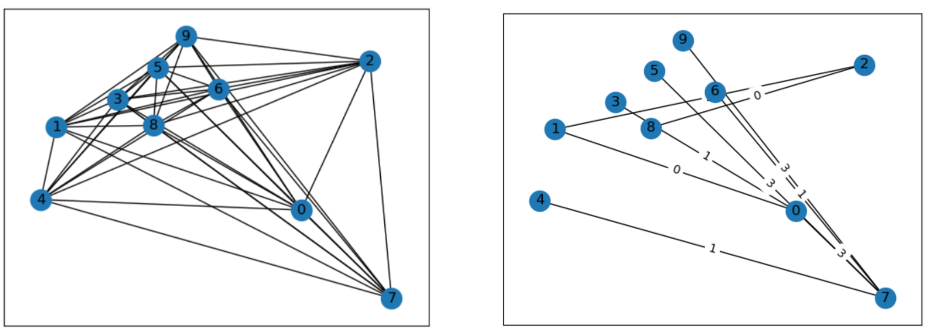

FIGURE 5.3 Instance-3 with its output tree topology

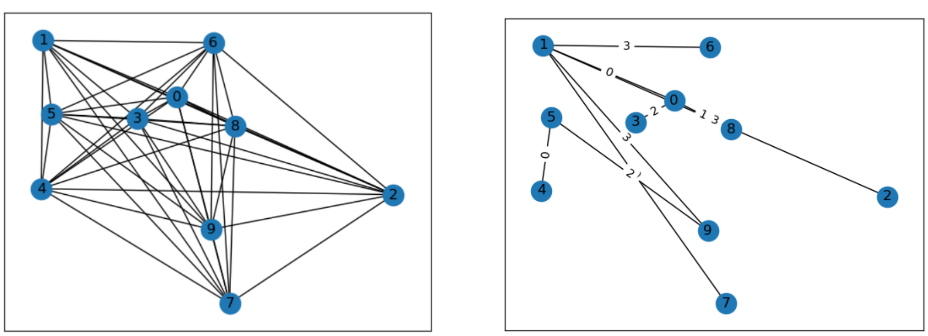

FIGURE 5.4 Instance-4 with its output tree topology

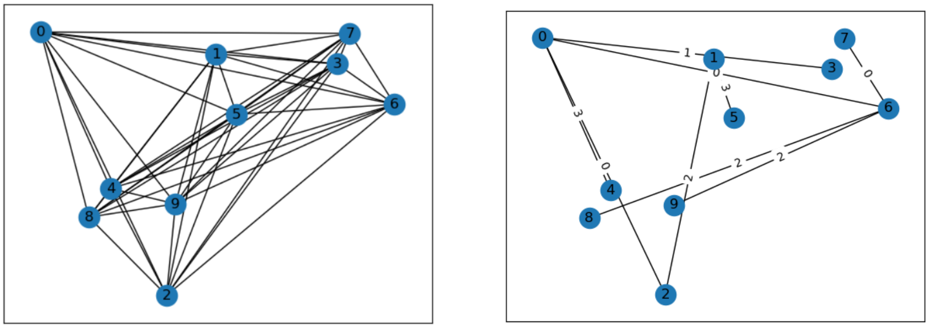

FIGURE 5.5 Instance-5 with its output tree topology

To compare our output network topology with the method in [2], we first solved Equation 1.3 using all the maximum δ_{min} values found from different M at each iteration, for each instance. We used the same relative weight $\alpha = 8/13$ and normalization constant $Z = 78/2$ value as [2]. The best solution for each M value was given the associated minimum effective throughput used in the computation of δ_{min}. We compared the values of δ_{min} with different M values for each instance, which helps determine which network topology at which M value has the most congested links. The result is pictured in Figure 5.6.

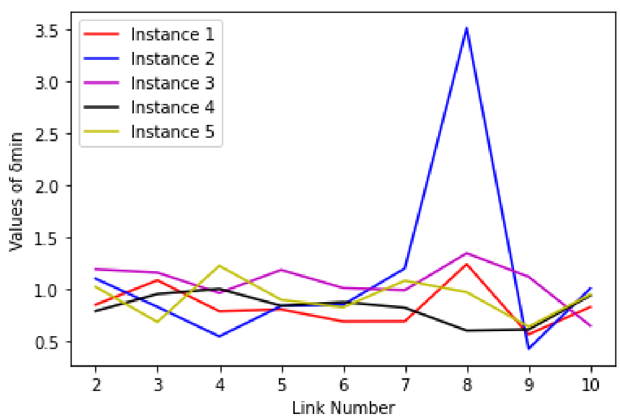

FIGURE 5.6 Comparison of the value of δ_{min} for different M values

The above figure shows that the first, second, and fifth instances experienced the most congestion when the network used a maximum of nine links per sector. The minimum value of δ_{min} was 0.5625, 0.423, and 0.637, respectively. The third instance experienced the most congestion when the network used a maximum of 10 links per sector, with a minimum δ_{min} value of 0.646. The fourth instance experienced the most congestion when the network used a maximum of eight links per sector, with a minimum δ_{min} value of 0.599. This indicates that the more links the sector uses, the more congested the network becomes. The network's capacity decreases due to congestion as the number of links increases. This can lead to a decrease in performance and reliability and an increase in latency. A higher value of δ_{min} indicates that the network has a higher effective capacity or lower aggregate traffic. Since we use directional antennas, which may reduce the size of the interference zone, the number of interfering links may be less. Therefore, the effective capacity becomes higher. To compensate for this difference, it is important to carefully consider the number of links used in the network because the more links the sector has with the same channel, the more interference the topology has. We used the topology with the lowest δ_{min} value for our comparison. A summary of the above figure is given in Table 5.1 below. For each instance, the best solution found is in bold. We

	M = 2	M = 3	M = 4	M = 5	M = 6	M = 7	M = 8	M = 9	M =10
Instance-1	0.848	1.083	0.785	0.803	0.687	0.688	1.238	**0.5625**	0.826
Instance-2	1.099	0.827	0.543	0.836	0.847	1.197	3.509	**0.423**	1.005
Instance-3	1.189	1.157	0.961	1.183	1.009	0.988	1.343	1.117	**0.646**
Instance-4	0.787	0.951	1.000	0.838	0.873	0.819	**0.599**	0.609	0.936
Instance-5	1.020	0.682	1.222	0.896	0.822	1.079	0.969	**0.637**	0.947

Table 5.1 Tree topology performance comparison.

found the best solution for each instance calculated from Equation 1.2. We then compared our topology with the topology found by [2] for the same instances. Figures 5.7-5.11 show the computation between the topology found by our model and by the model in [2]. The red and blue colors represent the channels : light and dark blue are different frequencies in the first channel, light and dark red are different frequencies in the second channel. Broken and solid lines represent the type of edges in the connection : broken lines indicate multiple edges in one connection, and solid lines indicate a single edge in one connection. The green and black dots represent the master hub and all the other nodes, respectively. The arrows represent the edges pointing away from the master hub. The end of this section will provide a comprehensive evaluation of the solution's effectiveness and a discussion of the solution's quality.

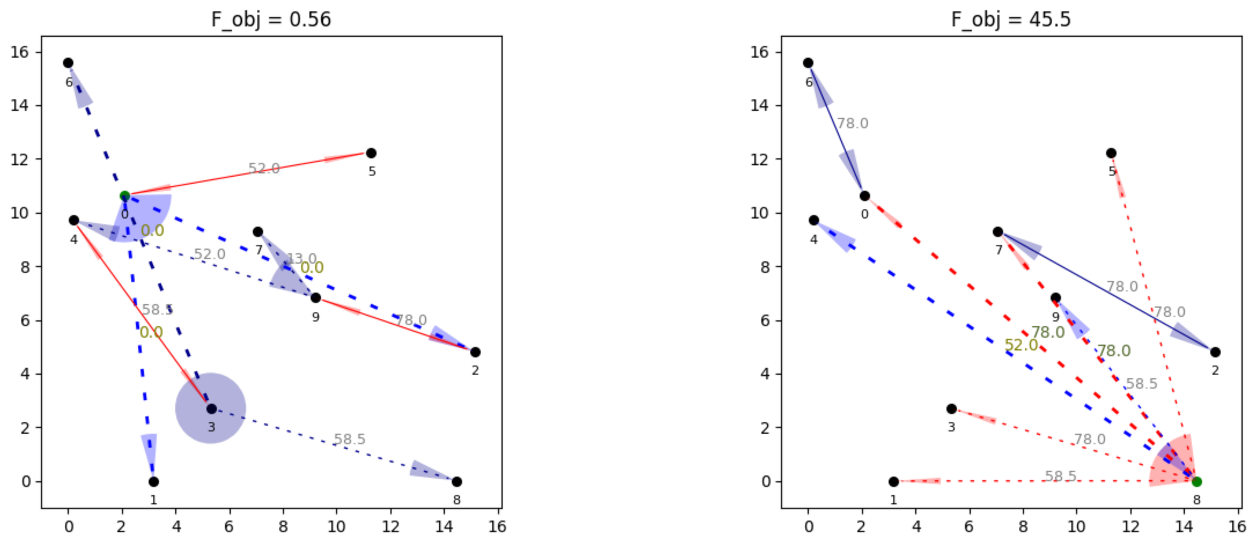

FIGURE 5.7 Instance-1 best topology from our algorithm when we use nine links per sector and [2], respectively.

The left side of Figure 5.7 is the topology generated by our algorithm. Links (0,1), (0,2), and (3,6) are the weakest links in this topology. This means that the minimum value of the most congested link is zero. However, the weakest link has a minimum value of 52.0 in the topology generated by model [2] (right side of Figure 5.7).

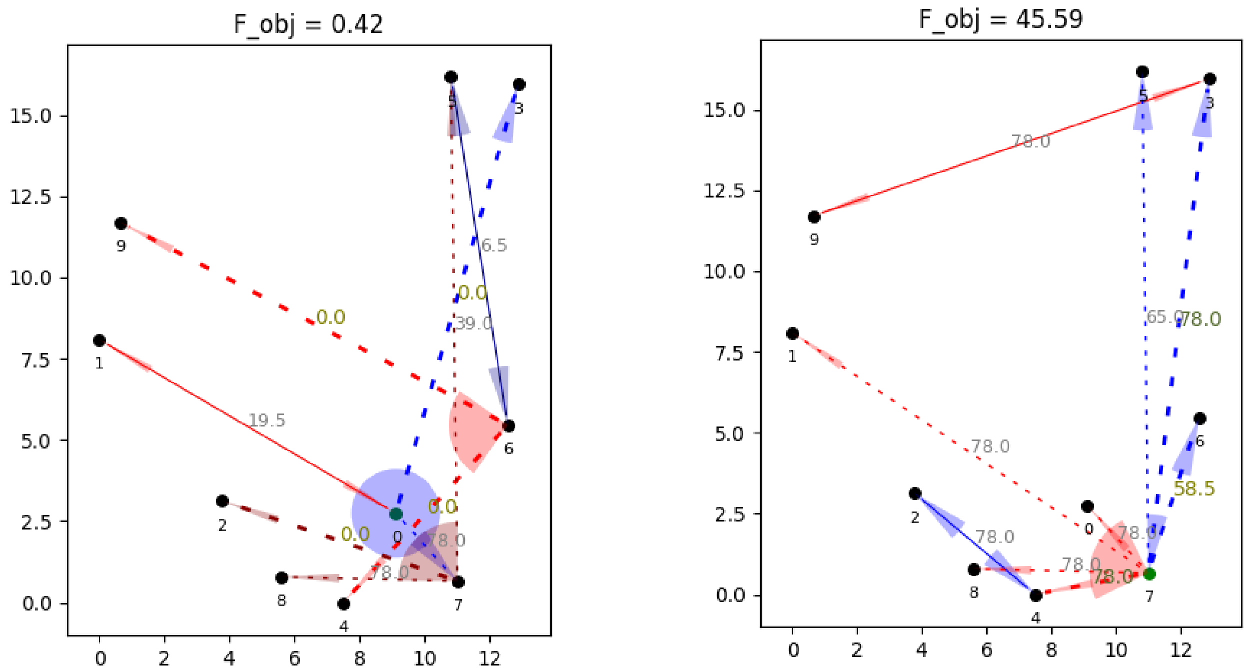

FIGURE 5.8 Instance-2 best topology from our algorithm when we use nine links per sector and [2], respectively.

The left side of Figure 5.8 is the topology generated by our algorithm. Links (0,3), (6,4), (6,9), and (7,2) are the weakest links in this topology. This means that the minimum value of the most congested link is zero. However, the weakest link has a minimum value of 58.5 in the topology generated by model [2] (right side of Figure 5.8).

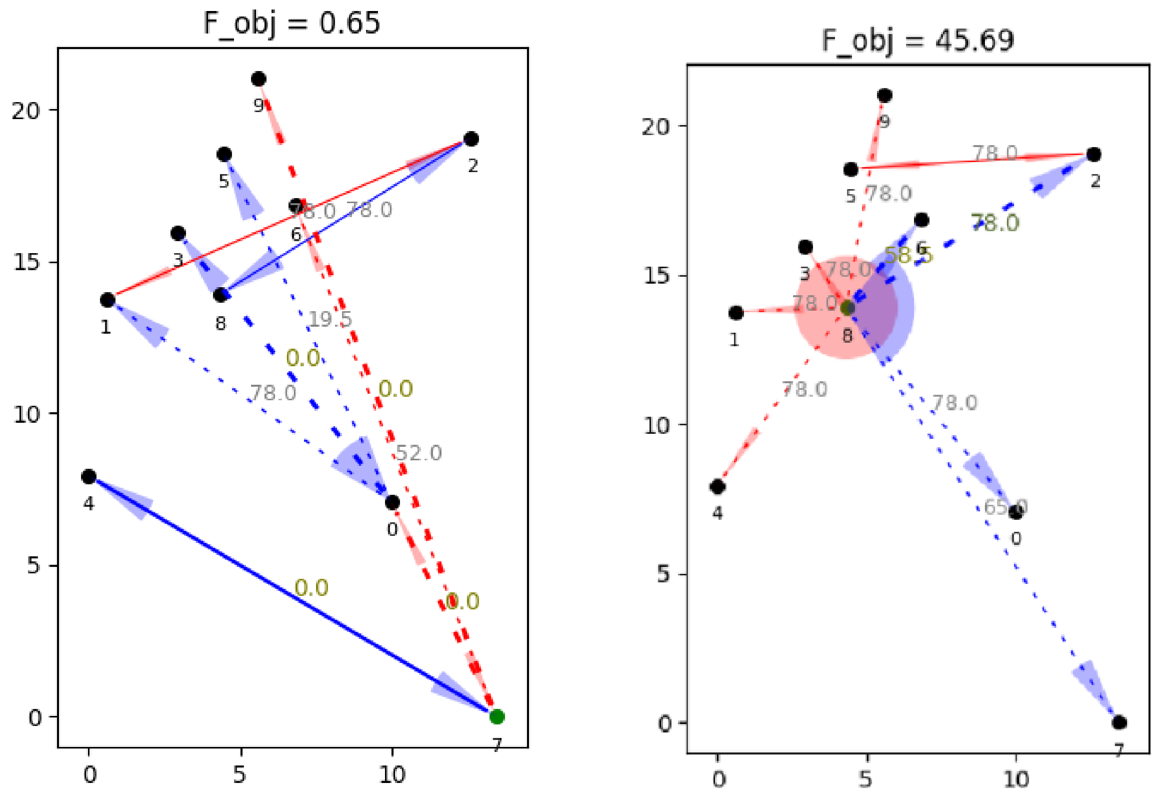

FIGURE 5.9 Instance-3 best topology from our algorithm when we use ten links per sector and [2], respectively.

The left side of Figure 5.9 is the topology generated by our algorithm. Links (0,8), (7,0), (7,4), and (7,9) are the weakest links in this topology. This means that the minimum value of the most congested link is zero. However, the weakest link has a minimum value of 58.5 in the topology generated by model [2] (right side of Figure 5.9).

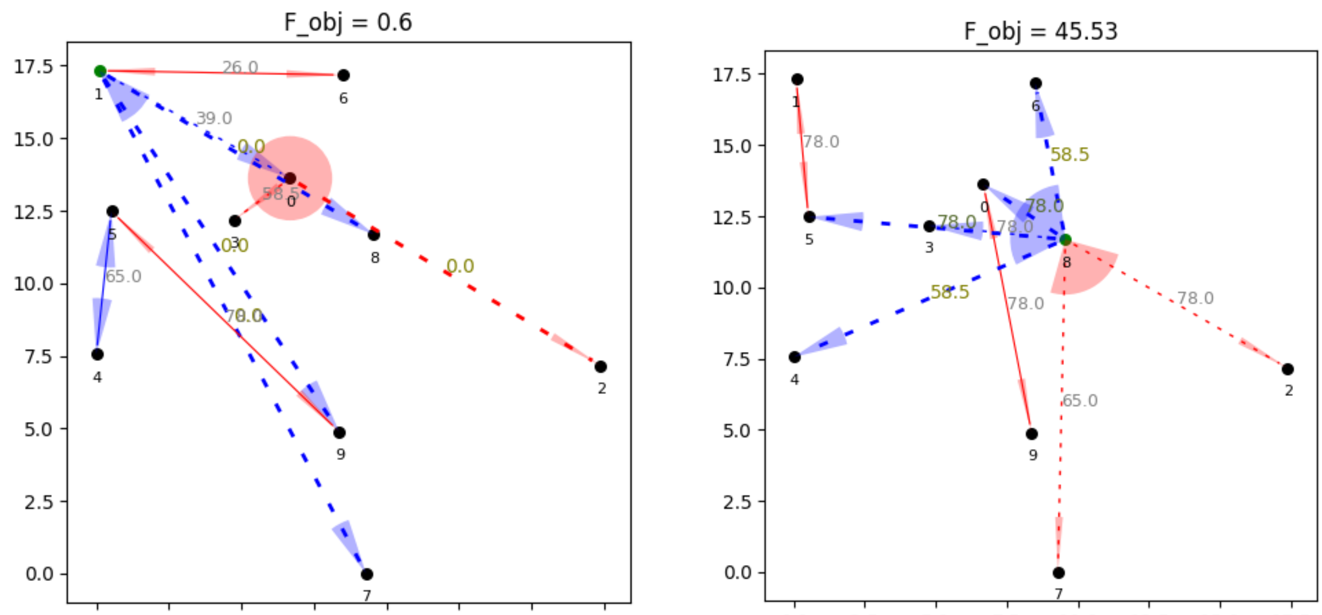

FIGURE 5.10 Instance-4 best topology from our algorithm when we use eight links per sector and [2], respectively.

The left side of Figure 5.10 is the topology generated by our algorithm. Links (0,2), (1,8), and (1,9) are the weakest links in this topology. This means that the minimum value of the most congested link is zero. However, the weakest link has a minimum value of 58.5 in the topology generated by model [2] (right side of Figure 5.10).

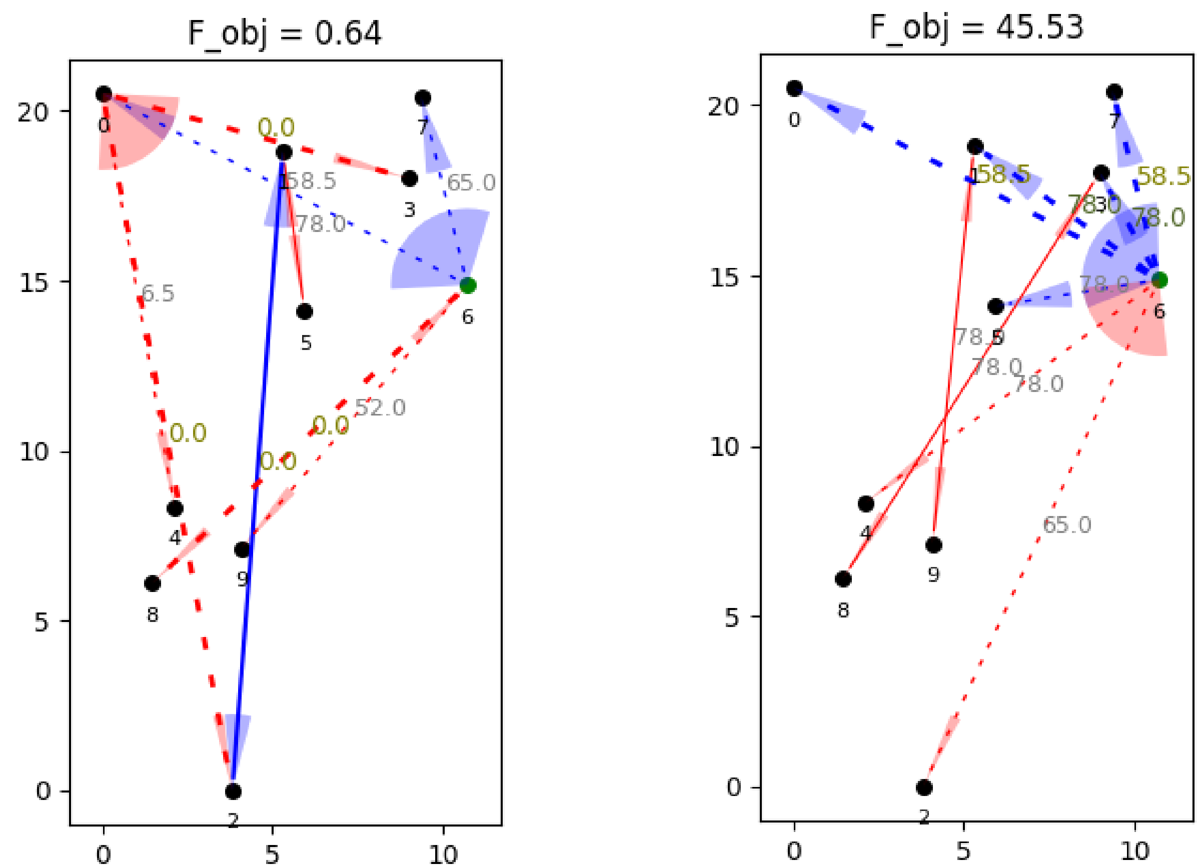

FIGURE 5.11 Instance-5 best topology from our algorithm when we use nine links per sector and [2], respectively.

The left side of Figure 5.11 is the topology generated by our algorithm. Links (0,2), (0,3), (1,2), and (6,8) are the weakest links in this topology. This means that the minimum value of the most congested links is zero. However, the weakest link has a minimum value of 58.5 in the topology generated by model [2] (right side of Figure 5.11).

In order to compare with the approach in [2], we modified as much as possible the closest approach in the literature, i.e., in [2] two components are considered : the path loss and fade margin to model the signal and interference ; whereas, in our case, we only consider the distance between two nodes. The results show that full connectivity is generally hard to achieve because of these two factors, and the objective value of the output topology is affected. In method [2], the generated network topology has a higher objective value than the network topology generated by our model. This difference in the objective value comes from the fact that the weakest link in our model has a zero value, i.e., the edge that disconnects from the network due to our assumption has a zero value, so the first term is zero in Equation 1.2. The remaining term is the mean divided by the normalization constant Z. However, the weakest link in method [2] does not have a value of zero, so method [2] calculates the whole Equation 1.2. From this, we can conclude that path loss and fade margin are important factors in whether two nodes can be connected by an edge, and they also affect the quality of the solution.

The main result so far seems to be that the method adapted from [1] cannot be forced to get a fully connected topology, i.e., the output topologies always have at least one edge that cannot transmit any data and thus disconnects the network. This is a major limitation of this method as it reduces the efficiency of the network. Therefore, further research is needed to find a better solution that can guarantee a fully connected topology. To tackle this limitation, we attempted to force the topology by adding a constraint that only allows edges without interference, given the terrain (path loss and fade margin). However, we cannot simply add the constraint because a constraint is the effect of interference on the signals and how it non-trivially decreases them.

Although we consider interference, it is limited to only binary quantities, like the antenna signals. Furthermore, we consider it in an ideal way, where interfering edges must share a total amount of throughput capacity. By comparison, [2] considers interference as scalar quantities in which the effective signal is the direct signal minus the interfering signal coming from the interfering edges, where all direct signals are supposed to be at maximum strength. Thus in [1], interference reduces the capacity of the edges, depending on how much they need to be used, and in [2], interference reduces the physical signals directly to the point where some might have a zero throughput. Unfortunately, this new constraint only removes about three edges out of 45 possible edges for each instance. We used an ideal situation with a parabolic antenna and no interference to compute this. When an edge is not connected in our results, it is typically for sector antennas (the widest that is not omni). We found that the constraint had limited success in forcing the topology, but that can still be useful to reduce the number of possible edges. However, it is important to note that it is not the best solution and that the remaining edges must be evaluated and tested to ensure they are viable for the given situation. For that, however, we used the same constraint with the possible edges computed using an ideal situation with a sector antenna that is wide enough for a whole sector. These removed about 20 out of 45 possible edges (with instances still connectable). This ensured that the final topology is connected unless the problem came from interference. Here, the problem is that this also removes edges that might have worked with a parabolic antenna, so we restricted the space of topologies further than what might be optimal. However, this new constraint improved the objective value in a network design that maximizes the throughput while ensuring the topology is connected. This is shown in Figures 5.13-5.16.

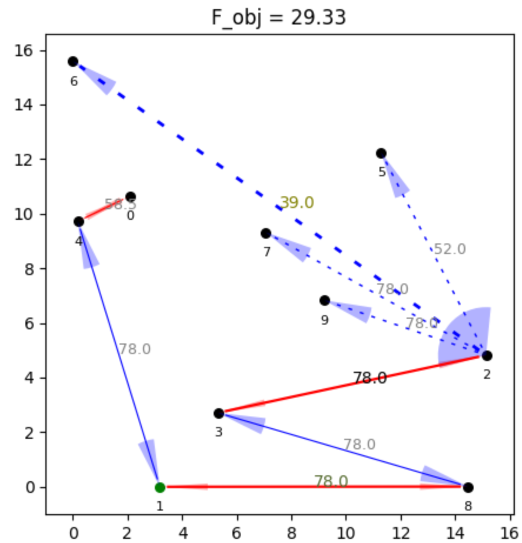

Figure 5.12 Instance-1 connected network topology with its objective value.

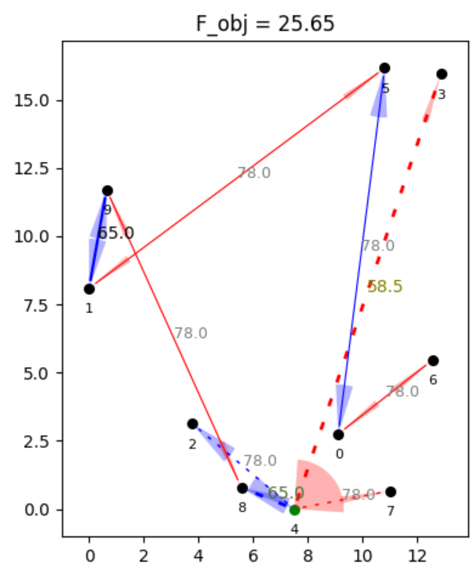

Figure 5.13 Instance-2 connected network topology with its objective value.

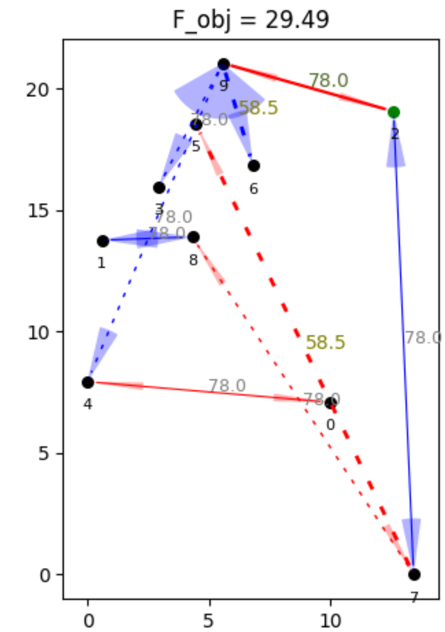

FIGURE 5.14 Instance-3 connected network topology with its objective value.

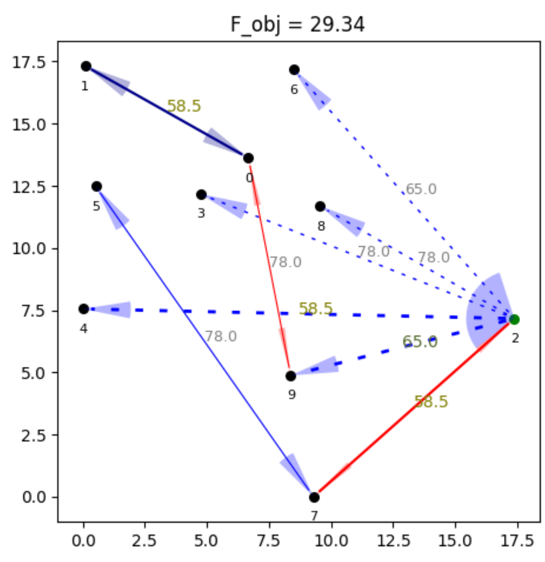

FIGURE 5.15 Instance-4 connected network topology with its objective value.

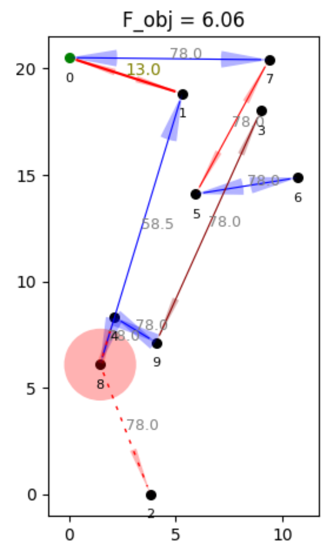

FIGURE 5.16 Instance-5 connected network topology with its objective value.

We have two observations from the above figures. First, we observe that the added constraint worked successfully, which helped us get a fully connected network topology. By limiting the number of connections between nodes, we could ensure that each node had at least one link to the rest of the network. This constraint is important because without it, it is possible that some nodes would remain isolated from the rest of the network and unable to communicate with other nodes. As a result, we achieved a fully connected network topology and got the desired result.

Second, we see that the objective values are still different from those we get from the model in [2]. The objective function is an expression that measures the quality of the topology. When the value of the objective function is high, it indicates that the topology is a good one. On the other hand, when the value of the objective function is low, it is an indication that the topology is not good. For instance, the performance for the first instance from our model was 29.33, while the objective value from the model in [2] was 45.5, meaning that the topology found from our model is not as good as the topology found from model [2]. In general, the topology found by our model is less quality than the topology found from [2]. This difference could be due, first, to different optimization algorithms or parameters used in the optimization process. Also, the model in [2] takes into account more variables than our model and thus arrives at a more accurate estimation of the objective value.

The second reason could be our assumptions used to model the signal and interference. To calculate the signal and interference, [2] model a physical path loss and fade margin. Path loss and fade margin are essential parameters since they affect signal strength, impacting the network's performance. Considering these two factors, it is possible to accurately calculate the signal and interference, thereby maximizing the signal strength and improving the system's overall performance. However, our model assumes signal and interference to be binary variables. This does not help calculate signal strength since the binary variables of signal and interference do not provide enough information to measure signal strength accurately. To accurately measure signal strength, we need to take into consideration factors such as signal frequency, signal power, and noise power. These factors can be calculated from the signal and interference variables, however, our model cannot currently do this. Therefore, this causes the topology found by our model to be lower in quality.

The other reason could be the suboptimality of our ILS algorithm. This is a common problem with ILS algorithms and can affect objective values, leading to different results even when the same optimization algorithms and parameters are used. This means that the local search might get stuck on local minima, leading to suboptimal solutions, and is unable to find a better solution. This can be due to a lack of search space exploration or simply because the initial starting point is too close to the local minimum.

CHAPTER 6 CONCLUSION AND RECOMMENDATIONS

In this paper, we modeled the network design for challenging environments as a tree topology. An ideal cone shape placed the antenna signal in the sector. The cone shape was chosen as it provides the best coverage in terms of signal strength and minimizes interference from other networks. We also modeled the signal and interference as binary variables. By using the cone shape, the antenna is able to direct the signal to a specific area, ensuring that the signal strength is maximized in that area. Modeling the signal and interference as binary variables also made it easier to determine the signal strength in any given area and to identify any interference from other networks. Our model developed each sector to have a maximum of two channel frequencies, and if one sector uses one specific channel, the remaining sectors could not access that channel. We model that, in one sector, there are M different links, and each of them can use a similar channel, but a link can have a maximum of one channel. Ultimately, we formulated our problem as a mixed-integer linear programming problem.

This problem was solved using an iterated local search algorithm, one of the simplest and most powerful metaheuristic algorithms. This allowed us to find a feasible solution in a reasonable time. First, we took a random channel as input. At each iteration, we chose the most congested link from the network and relaxed the binary variables except the chosen ones. We solved this problem using a commercial solver called Gurobi, and the results gave us a possible tree topology with its channel edges. Finally, we evaluated our algorithm and compared it with the existing algorithm.

Based on our experiment, we were able to determine the possible tree topology. The results showed that our algorithm outperformed the existing algorithm in terms of time and accuracy. Our algorithm successfully generated the tree topology with minimal errors, leading to suboptimal solutions for the problem. Our model does not force the original topology to be connected, and due to, that the network topology has disconnected links in the network. In this case, we added additional constraints to the model.

Our study demonstrated that path loss and fade margin are the most important factors in network design. The algorithm identified which factors had the most significant effect on the design of networks by looking at different combinations of parameters and how they affected the quality of the solution. Furthermore, we showed that the model assumptions directly impacted the solution's quality, emphasizing the importance of proper model selection.

6.1 Limitations

The limitations of this work and future work are as follows :

1. We ran our algorithm only for ten nodes. In the future, the algorithm should be tested for the complex problem ($|N| > 10$).
2. We should re-model our MILP by considering the fade margin and the path loss and test the algorithm to see if we get the best solution.
3. Although our algorithm found a suboptimal solution, one of our implementation result topologies is not fully connected due to our assumption. We should consider other factors in addition to distance.

REFERENCES

[1] N. S. Hamami, T. C. Chuah et S. W. Tan, "Joint resource allocation in multi-radio multi-channel wireless mesh networks with practical sectored antennas," dans *2010 International Conference on Computer Applications and Industrial Electronics.* IEEE, 2010, p. 316–321.

[2] V. Perreault, "Tactical wireless network design for challenging environments," Mémoire de maîtrise, Polytechnique Montréal, 2022.

[3] A. U. Chaudhry, "Spectrum requirements for interference-free wireless mesh networks," Thèse de doctorat, Carleton University Ottawa, 2015.

[4] L. A. Wolsey, *Integer programming.* John Wiley & Sons, 2020.

[5] W. John, "Chinneck. : Practical optimization : a gentle introduction," *Systems and Computer Engineering Carleton University Ottawa, Canada, http ://www. sce. carleton. ca/faculty/chinneck/po. html.*

[6] R. Anand, D. Aggarwal et V. Kumar, "A comparative analysis of optimization solvers," *Journal of Statistics and Management Systems*, vol. 20, n°. 4, p. 623–635, 2017.

[7] C. Blum et A. Roli, "Metaheuristics in combinatorial optimization : Overview and conceptual comparison," *ACM computing surveys (CSUR)*, vol. 35, n°. 3, p. 268–308, 2003.

[8] I. Cruz-Aceves *et al.*, "Binary descriptor design for the automatic detection of coronary arteries using metaheuristics," dans *Advanced Machine Vision Paradigms for Medical Image Analysis.* Elsevier, 2021, p. 169–188.

[9] A. H. Mohsenian-Rad et V. W. Wong, "Joint logical topology design, interface assignment, channel allocation, and routing for multi-channel wireless mesh networks," *IEEE Transactions on Wireless Communications*, vol. 6, n°. 12, p. 4432–4440, 2007.

[10] H. R. Lourenço, O. C. Martin et T. Stützle, *Iterated local search.* Springer, 2003.

[11] L. Zhou *et al.*, "On capacity optimization in multi-radio multi-channel wireless networks with directional antennas," dans *2015 IEEE International Conference on Communications (ICC).* IEEE, 2015, p. 3745–3750.

[12] B. Mumey *et al.*, "Topology control in multihop wireless networks with multi-beam smart antennas," dans *2012 International Conference On Computing, Networking and Communications (ICNC).* IEEE, 2012, p. 1020–1024.

[13] M. K. Marina, S. R. Das et A. P. Subramanian, “A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks,” *Computer networks*, vol. 54, n°. 2, p. 241–256, 2010.

[14] W. Zhou, X. Chen et D. Qiao, “Practical routing and channel assignment scheme for mesh networks with directional antennas,” dans *2008 IEEE International Conference on Communications*. IEEE, 2008, p. 3181–3187.

[15] D. England, B. Veeravalli et J. B. Weissman, “A robust spanning tree topology for data collection and dissemination in distributed environments,” *IEEE transactions On parallel and distributed systems*, vol. 18, n°. 5, p. 608–620, 2007.

[16] Y. Li *et al.*, “Multi-objective topology planning for microwave-based wireless backhaul networks,” *IEEE Access*, vol. 4, p. 5742–5754, 2016.

[17] C. Huang et X. Wang, “A bayesian approach to the design of backhauling topology for 5g iab networks,” *IEEE Transactions on Mobile Computing*, 2021.

[18] U. Kumar, H. Gupta et S. R. Das, “A topology control approach to using directional antennas in wireless mesh networks,” dans *2006 IEEE International Conference on Communications*, vol. 9. IEEE, 2006, p. 4083–4088.

[19] P. Leesutthipornchai, N. Wattanapongsakorn et C. Charnsripinyo, “Efficient design techniques for reliable wireless backhaul networks,” dans *2008 International Symposium on Communications and Information Technologies*. IEEE, 2008, p. 22–27.

[20] P. Lin *et al.*, “Minimum cost wireless broadband overlay network planning,” dans *2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM’06)*. IEEE, 2006, p. 7–pp.

[21] N. Kamiyama et D. Satoh, “Network topology design using analytic hierarchy process,” dans *2008 IEEE International Conference on Communications*. IEEE, 2008, p. 2048–2054.

[22] N. Li, J. C. Hou et L. Sha, “Design and analysis of an mst-based topology control algorithm,” *IEEE Transactions on wireless communications*, vol. 4, n°. 3, p. 1195–1206, 2005.

[23] X. Liu, “Atypical hierarchical routing protocols for wireless sensor networks : A review,” *IEEE Sensors Journal*, vol. 15, n°. 10, p. 5372–5383, 2015.

[24] Y. H. Tehrani, A. Amini et S. M. Atarodi, “A tree-structured lora network for energy efficiency,” *IEEE Internet of Things Journal*, vol. 8, n°. 7, p. 6002–6011, 2020.

[25] J. An *et al.*, “Joint design of hierarchical topology control and routing design for heterogeneous wireless sensor networks,” *Computer Standards & Interfaces*, vol. 51, p. 63–70, 2017.

[26] M. S. Artigas *et al.*, “Cyclone : A novel design schema for hierarchical dhts,” dans *Fifth IEEE International Conference on Peer-to-Peer Computing (P2P’05).* IEEE, 2005, p. 49–56.

[27] R. Velmani et B. Kaarthick, “An efficient cluster-tree based data collection scheme for large mobile wireless sensor networks,” *IEEE sensors journal*, vol. 15, n°. 4, p. 2377–2390, 2014.