Titre: Title:	Tactical Wireless Network Design for Challenging Environments
Auteur: Author:	Vincent Perreault
Date:	2022
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Perreault, V. (2022). Tactical Wireless Network Design for Challenging Environments [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/10423/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/10423/
Directeurs de recherche: Advisors:	Alain Hertz, & Andrea Lodi
Programme: Program:	Maîtrise recherche en mathématiques appliquées

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Tactical Wireless Network Design for Challenging Environments

VINCENT PERREAULT
Département de mathématiques et de génie industriel

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

Juillet 2022

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Tactical Wireless Network Design for Challenging Environments

présenté par **Vincent PERREAULT**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Antoine LEGRAIN, président
Alain HERTZ, membre et directeur de recherche
Andrea LODI, membre et codirecteur de recherche
Georges KADDOUM, membre

DEDICATION

to Naïma,
the stake to my plant,
who always reminds me that
the only way to truly grow is up…

ACKNOWLEDGEMENTS

I personally would like to thank everyone involved in this project for their invaluable help that they have given me all along this project. Specifically, I would like to thank

- Alain Hertz and Andrea Lodi, whose mathematical and practical insights I have found particularly illuminating for this project, but also for all the next ones;
- the whole initial team on the project, notably Gwenael Poitau and, most importantly, Ghassan Dahman who had the patience to explain to me in detail the ins and outs of the problem, the intuitive understanding of the problem, the expected results and the practical meaning behind every little detail;
- my friends and family who supported me and had to suffer through too many conversations about topologies and multi-beam antennas.

I would also like to acknowledge the financial support for this project without which it would not have been possible. I had the privilege of being granted a Canada Graduate Scholarship for my master's, through the NSERC. I was paid partially by our industrial partner Ultra TCS, through my advisors. I also had access to loans and bursaries for full-time studies from the Government of Québec.

RÉSUMÉ

Dans ce mémoire, nous modélisons le problème de conception de réseau sans fil tactique avec un signal modélisé physiquement et en considérant trois scénarios élémentaires de trafic de données. Nous modélisons ce problèmes pour différents types de topologies possibles, des arbres jusqu'au maillage et tout ce qui se trouve entre les deux. Nous considérons aussi deux types d'antennes : celles à simple faisceau et celles à multiples faisceaux. Nous utilisons des notions de théorie des graphes afin de donner des conditions de validité pour chaque grande partie des réseaux : leur topologie, leur configuration et la configuration de leurs antennes. Nous modélisons finalement un critère d'optimisation basé sur le lien le plus faible du réseau en terme de débit effectif, *i.e.* de vitesse de transmission de données en tenant compte de la congestion du réseau. À ce que nous sachions, nous sommes les premiers à modéliser complètement ce problème combinatoire non-linéaire complexe.

Nous proposons un algorithme à plusieurs niveaux pour résoudre ce nouveau problème. Au plus haut niveau, nous utilisons une méta-heuristique de type recherche local pour résoudre le sous-problème de conception de la topologie. Au niveau moyen, nous utilisons une combinaison d'énumération exhaustive, de méta-heuristiques, d'heuristiques et d'approximations pour résoudre le sous-problème de configuration du réseau. Au plus bas niveau, nous utilisons une heuristique intuitive basée sur la géométrie pour résoudre le sous-problème de la configuration des antennes.

Finalement, nous caractérisons notre approche à travers plusieurs tests. Nous comparons plusieurs façons d'évaluer les maillages. Nous étudions l'effet des paramètres de notre algorithme sur sa performance. Nous comparons quantitativement et qualitativement les meilleurs réseaux trouvés par notre algorithme pour des antennes à simple faisceau avec ceux pour des antennes à multiple faisceaux. Nous comparons également les valeurs d'objectif des meilleurs topologies arbres avec les maillages complets sur les mêmes instances. Nous identifions aussi la plus grande limite de notre approche, qui est sa durée d'exécution, spécifiquement le temps nécessaire pour évaluer la fonction objectif.

Nous concluons en identifiant d'autres limites de notre approche ainsi qu'en esquissant des avenues prometteuses pour poursuivre la recherche suite à ce travail.

ABSTRACT

In this work, we model the tactical wireless network design with a physically-modeled signal in three elementary data traffic scenarios. We model this problem for different restrictions on the type of topologies that are allowed, from trees to full mesh clusters and any combination thereof. We also consider two types of antennas: single-beam and multi-beam. We use the language of graph theory to state validity conditions for each main component of a network: its topology, its configuration and its antenna configurations. We finally model an optimization criterion based on the weakest link in the network in terms of effective throughput, *i.e.* data transmission speeds given the congestion of the network. To our knowledge, we are the firsts to model this complex non-linear combinatorial optimization problem in its entirety.

We propose a multi-level algorithm to solve this novel problem. At the highest level, we use a meta-heuristic local search for the topology design sub-problem. At the mid level, we use a mix of exhaustive enumeration, meta-heuristics, heuristics and approximations to solve the network configuration sub-problem. At the lowest level, we use intuitive geometrically-based heuristics to solve the antenna configurations sub-problem.

We finally evaluate our approach through multiple tests. We compare various ways of evaluating mesh clusters. We study the effects of the algorithm's parameters on its performance. We compare quantitatively and qualitatively the best networks found by our algorithm for single-beam versus multi-beam antennas. We also compare the objective values of the best tree topologies that our algorithm finds with full mesh clusters on the same instances. We also identify the main limitation of our current approach which is its timing, and especially the time it takes to evaluate the objective function.

We conclude by identifying other limitations of our approach and describing promising avenues for further research in the continuation of this work.

TABLE OF CONTENTS

DEDICATION iii

ACKNOWLEDGEMENTS iv

RÉSUMÉ v

ABSTRACT vi

TABLE OF CONTENTS vii

LIST OF TABLES x

LIST OF FIGURES xi

LIST OF SYMBOLS AND ACRONYMS xii

LIST OF APPENDICES xiii

CHAPTER 1 INTRODUCTION 1
- 1.1 Problem Overview 1
  - 1.1.1 Topology 1
  - 1.1.2 Configuration 2
  - 1.1.3 Antenna Types 4
  - 1.1.4 Objective 6
- 1.2 Contributions 6
- 1.3 Outline of the Thesis 6

CHAPTER 2 LITERATURE REVIEW 8
- 2.1 Similar Wireless Network Design Problems 8
- 2.2 Other Wireless Network Design Problems 10

CHAPTER 3 PROBLEM MODELING 13
- 3.1 Problem Input 13
  - 3.1.1 Problem Parameters 13
  - 3.1.2 Problem Instance 15
- 3.2 Network Solution Space 15

3.2.1 Topology 15
3.2.2 Network Configuration 16
3.2.3 Antenna Configuration 18
3.3 Objective Function 19
3.3.1 Signal Modeling 19
3.3.2 Traffic Modeling 23
3.4 Full Optimization Problem 27

CHAPTER 4 ALGORITHMIC STRATEGY 29
4.1 Problem Separation 29
4.2 P_0 Topology Local Search 29
4.2.1 Neighborhoods 29
4.2.2 Taboo Search 32
4.2.3 Initialization 38
4.3 P_1 Network Configuration Exhaustive Enumeration, Approximations, *etc.* . . 40
4.3.1 Master Hub Selection 40
4.3.2 Waveform Assignment 42
4.4 P_2 Antenna Configuration Heuristic 43
4.4.1 Single-Beam Antennas 43
4.4.2 Multi-Beam Antennas 45

CHAPTER 5 NUMERICAL EXPERIMENTS 48
5.1 Mesh Evaluation Performance 49
5.2 Local Search Parameters Impact on Performance 51
5.2.1 Reset 51
5.2.2 Number of Parallel Searches κ 52
5.2.3 Neighborhood Subset Ratio ρ 53
5.3 Single-Beam *vs* Multi-Beam Antennas (Tree Case) 55
5.4 Tree *vs* Mesh Cases 59
5.5 Timing of the Algorithm 59

CHAPTER 6 CONCLUSION AND RECOMMENDATIONS 61
6.1 Summary of Works 61
6.1.1 Problem Modeling 61
6.1.2 Algorithmic Strategy 62
6.1.3 Numerical Experiments 62
6.2 Limitations 63

6.3 Future Research 64
6.3.1 Small Adjustments 64
6.3.2 Broad Research Directions 65

REFERENCES 67

APPENDICES 70

LIST OF TABLES

Table 2.1 Comparison of our Problem with Those Studied in the Literature . . 8

Table 3.1 Direct Throughput Lookup Table 22

Table 5.1 Mesh Evaluation Performance Comparison 50

Table 5.2 Best Tree Networks *vs* Full Mesh Networks by Use Case 55

Table 5.3 Best Tree Networks *vs* Full Mesh Networks by Use Case 59

Table 5.4 Timing in Seconds of the Algorithm by Component 60

Table 6.1 Frequencies as Channels and Intra-Channel Ranking 65

LIST OF FIGURES

Figure 1.1 Example of a Hybrid Network Topology 2
Figure 1.2 Example of a Master Hub 3
Figure 1.3 Example of a Waveform Assignment 3
Figure 1.4 Example of a Channel Assignment 4
Figure 1.5 Example of Antenna Alignments and Selections/Beam Configurations 5
Figure 3.1 Example of an Instance of the Problem 15
Figure 3.2 Topology 16
Figure 3.3 Implicit Tree 16
Figure 3.4 Implicit Arborescence 16
Figure 3.5 Channel Assignment 18
Figure 3.6 Coloring Problem Tree 18
Figure 3.7 Example of a Multi-Beam Antenna with $\phi_a = 0$ and $B_a = \{0, 8, 19\}$. 19
Figure 3.8 Gain Pattern of a Parabolic Antenna with $\phi_a = 0$ on channel 3+ . . . 21
Figure 4.1 Neighbor Topologies (Tree Neighborhood) 30
Figure 4.2 Neighbor Topologies (Mesh Creation Sub-Neighborhood) 30
Figure 4.3 Neighbor Topologies (Mesh Destruction Sub-Neighborhood) 31
Figure 4.4 Neighbor Topologies (Mesh Inclusion Sub-Neighborhood) 31
Figure 4.5 Neighbor Topologies (Mesh Exclusion Sub-Neighborhood) 31
Figure 4.6 Neighbor Topologies (Mesh Fusion Sub-Neighborhood) 32
Figure 4.7 Three Possible Cases of the Single-Beam Alignment Problem 44
Figure 4.8 With $i = 1$ (Requires 4 Activated Beams) 46
Figure 4.9 With $i = 2$ (Requires 5 Activated Beams) 46
Figure 4.10 Antenna Alignment of the Last Example 47
Figure 5.1 Example of Algorithm Run with 1 Hour Limit for $|V| = 30$ 51
Figure 5.2 Performance in Function of the Number of Parallel Searches κ for $|V| = 10$ 52
Figure 5.3 Performance in Function of the Number of Parallel Searches κ for $|V| = 30$ 53
Figure 5.4 Performance in Function of the Neighborhood Subset Ratio ρ for $|V| = 10$ 54
Figure 5.5 Performance in Function of the Neighborhood Subset Ratio ρ for $|V| = 30$ 54
Figure 5.6 The Best Solutions for Each Traffic Scenario for $|V| = 10$ and $D^{-1} = 20$ 56
Figure 5.7 The Best Solutions for Each Traffic Scenario for $|V| = 10$ and $D^{-1} = 50$ 57
Figure 5.8 The Best Solutions for Each Traffic Scenario for $|V| = 20$ and $D^{-1} = 20$ 57
Figure 5.9 The Best Solutions for Each Traffic Scenario for $|V| = 20$ and $D^{-1} = 50$ 58
Figure 5.10 The Best Solutions for Each Traffic Scenario for $|V| = 50$ and $D^{-1} = 20$ 58

LIST OF SYMBOLS AND ACRONYMS

PTP	Point-To-Point
PMP	Point-to-MultiPoint
SINR	Signal-to-Noise Ratio
MIP	Mixed Integer Programming
LP	Linear Programming
MILP	Mixed Integer Linear Programming
ILP	Integer Linear Programming
GRASP	Greedy Randomized Adaptive Search Procedure
MHCP	Matern Hard-Core Process
GNN	Graph Neural Networks

LIST OF APPENDICES

Appendix A Implementation Details and Accelerations 70

CHAPTER 1 INTRODUCTION

Wireless communication is an essential component of most information technology we have today. Standard telecommunications networks usually provide this capability for most cases. However, when these standard networks are unavailable or unusable, as is the case of disaster relief operations and of tactical military operations in foreign territory, temporary networks must be set up to support the necessary information infrastructure.

In such cases, we want to connect multiple key coordinates of a particular region in a single network in which information can be sent to and from any key coordinates, which we call *nodes* in the sequel. The information, which we refer to as *data* in the sequel, is transmitted through antennas located at each node and which operate on specific radio frequencies. In tactical networks, there are typically between 10 and 50 nodes, which is much less than in the standard telecommunications networks which can operate with thousands of nodes.

1.1 Problem Overview

More specifically, the so-called tactical networks in which we are interested have specific constraints arising from the wireless technology that they use. These constraints affect the possible topologies, configurations and antenna types that the networks can have. Given these constraints, our objective is to optimize the network design in order to maximize the data transmission speeds in the network.

1.1.1 Topology

The *topology* of a network describes how the nodes are connected to one another in terms of direct links, which we refer to as *edges* in the topology. A communication between any pair of nodes in the network will have to go through a path of such edges.

We consider either tree topologies, mesh topologies, or any combination thereof.

Tree Topology Any connected topology in which there are no cycles of edges. In a tree topology, there is a single path that exists between every pair of nodes.

Mesh Cluster A set of nodes which can be modeled as a clique in the topology, *i.e.* a such that there is an edge between every pair of these nodes. However, not all of these edges are used depending on their strength and the specific data traffic in the cluster. In a mesh cluster, there can be more than one path that exists between two nodes. A topology which is a single mesh cluster is called a full mesh cluster.

Hybrid Topology Any topology that has both a global tree structure as well as the possibility of containing one or multiple mesh clusters.

For instance, in the example hybrid topology below, there are three distinct mesh clusters which each contain only three nodes. They are the colored triangles (*i.e.* cliques of size 3) and two of them share a node (the blue and the red clusters).

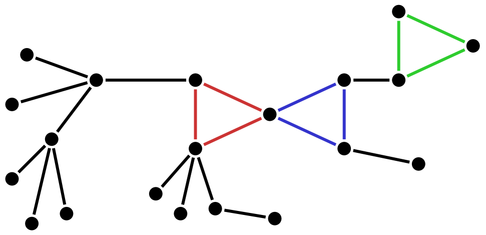

Figure 1.1 Example of a Hybrid Network Topology

This topological constraint comes from the three considered types of waveforms, which will be described in the next subsection.

1.1.2 Configuration

A fully configured network, in addition to having a defined topology, has a master hub (or gateway) as well as waveform, channel and frequency assignments to all of its edges.

Master Hub

A *master hub*, or gateway, is a node of the network which has the additional role of being a central node to which all the other nodes can send or from which they can receive data if necessary. The master hub is necessary in cases where an information must be collected from all the nodes of the network or where an information must be distributed to all the nodes of the network.

For instance, in Figure 1.2, the master hub is pictured in green.

Waveforms

A *waveform* is associated with a communication protocol that is used in a connection between nodes. We consider three types: Point-To-Point (PTP), Point-to-MultiPoint (PMP), Mesh.

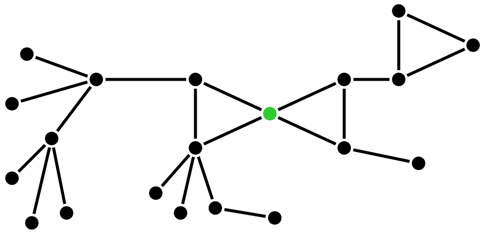

Figure 1.2 Example of a Master Hub

PTP A Point-To-Point connection is between two nodes. Such a connection is comprised of a single edge.

PMP A Point-to-Multipoint connection is between a node and multiple nodes. Such a connection is thus comprised of multiple edges, *i.e.* the edges that go from the one node to the other nodes.

Mesh A Mesh connection is between all the nodes of a single mesh cluster. Again, such a connection is comprised of multiple edges, *i.e.* all the edges between the nodes that are inside the mesh cluster.

A network with only PTP and PMP waveforms will have a tree topology while a network with only the mesh waveform will be a single mesh cluster. By having the possibility of all three, the networks we consider can have any combination of tree and mesh topology.

For instance, in our example topology, the only possible waveform assignment is given below where PTP connections are solid, PMP connections are dashed and Mesh connections are dotted.

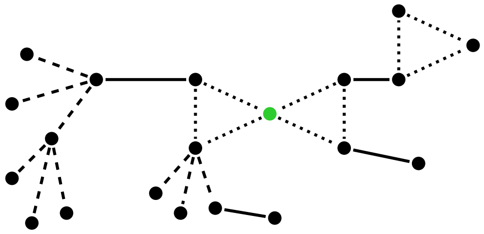

Figure 1.3 Example of a Waveform Assignment

Channels and Frequencies

A *channel* is a frequency band and every connection is assigned a channel as well as a specific carrier *frequency* within that channel at which the data is transmitted through the antennas.

We consider two channels: NATO bands[1] 3+ (2000 - 2400 MHz) and 4 (4500 - 5000 MHz). For each channel, we consider two possible frequencies: 2000 MHz and 2400 MHz for channel 3+ and 4500 MHz and 5000 MHz for channel 4. These four frequencies are chosen so that they are all orthogonal to one another, *i.e.* they do not interfere with one another. The only interference within the network comes from connections that use the same frequency.

In a real-world application, each node would have two antennas and a single radio connecting these antennas. Furthermore, the two antennas would be operating on different channels since a radio can use each channel only for one antenna, hence the name "channel". As such, for our problem, a node can only be connected to at most two connections and each of them should be on a separate channel. The only constraint for the assignment of the carrier frequencies is that they belong to the assigned channels.

For instance, a channel assignment is pictured in the example below where channel 3+ is in blue and channel 4 is in red.

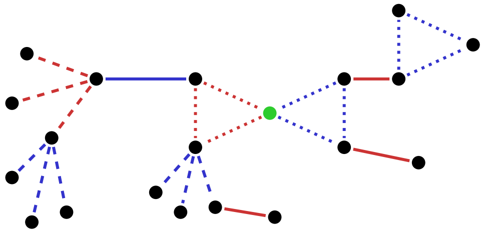

Figure 1.4 Example of a Channel Assignment

1.1.3 Antenna Types

We are interested in two different categories of antennas: single-beam antennas and multi-beam antennas. We are mainly interested in the latter and characterizing the advantage they offer compared to the former, if any.

Omni-Directional Antennas Such antennas transmit and receive signal in all directions (360°) simultaneously. They create the most interference since their signal goes in every direction, irrespective of the position of the intended transmitter/receiver(s). They are required for the Mesh waveform.

[1] Also known respectively as NATO bands S and C, these were since renamed NATO bands E and G respectively [1]. We keep the old terminology here as it was given to us by our industrial partner who submitted the problem.

Single-Beam Antennas As the name suggests, such antennas transmit and receive signal along a particular direction. There are three types with different beamwidths: parabolic (15°), panel (30°) and sector (100°).

Multi-Beam Antennas The multi-beam antennas that we consider have 24 separate beams distributed evenly around. At one time, up to 7 of these beams can be activated with each new activated beam reducing the power of all activated beams. These antennas also have an omni-directional mode in which none of the beams are activated and signal is transmitted/received in all directions. This omni-directional mode is required for the Mesh waveform.

In our comparison of the single and multi-beam antennas, we group the omni-directional antenna with the single-beam antennas. The latter is a newer type of antenna and it has the potential to reduce the amount of interference in the network compared to the single-beam antennas. This is because, for multi-beam antennas, the intended receivers don't have to be angularly close to each other and the signal is only transmitted in their direction whereas single-beam antennas with large beamwidths can waste a lot of signal in the regions between the receivers. For both categories, we consider the scenario in which every node has access to two antennas, one for each of the two channels. This makes it so that every node can be in up to two connections, thus allowing any hybrid topology for the network, an example of which is pictured below.

In the figure, our example is completed with the antenna alignments and the antenna type selection (in the case of single-beam antennas) or the alignments and beam configurations (in the case of multi-beam antennas, here pictured with only 8 possible beams for readability). The antennas are represented by the transparent circles (for omni-directional) and by the transparent circular sectors (for beams) with the associated colors of their connections. For this example, since the receivers are all close to one another for all PMP connections, both cases look the same. However, in the general case, a multi-beam antenna can have multiple activated beams that are all in very different directions.

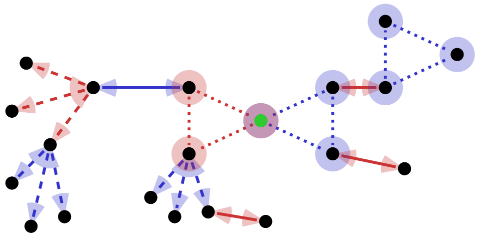

Figure 1.5 Example of Antenna Alignments and Selections/Beam Configurations

1.1.4 Objective

Given a fully-configured network with respect to its topology, configuration and antennas, we can model the maximum speed of data transmission in megabits per second for each of the edges in the network, which we refer to as their *throughput* in the sequel. Given these direct throughputs and a data traffic scenario, we can also compute the effective throughput for each of the edges, which also takes into account their congestion. The objective of our optimization problem is to maximize the network's weakest link, *i.e.* its slowest edge in terms of effective throughput.

1.2 Contributions

There are two main contributions in this work. The first is the mathematical modeling of the full optimization problem with all the practical constraints associated with the network's topology, configurations and antennas as well as their effects on the objective function, including the physical modeling of the transmitted signal with interference. We also take into account three different data traffic scenarios in the network and their impact on the effective throughputs. To the best of our knowledge, this is the first time that this full optimization problem has been formulated.

The second main contribution is our algorithm which tackles directly this complex nonlinear combinatorial optimization problem. We do this with a sophisticated problem separation in three levels and a variety of sub-strategies such as meta-heuristics, exact methods, greedy methods, graph theoretic methods and simple heuristics. Again, to the best of our knowledge, this is the first time that this full optimization problem has been tackled directly. Moreover, the results that our algorithm outputs achieve to both surprise our industrial partners as well as confirm some of their expectations in terms of the shapes of the preferred solutions and how they vary according to the different possible scenarios. Previously, they had to resort to exhaustive enumeration (brute force) for small instances of the problem (≤ 10 nodes) or had to do it manually for larger instances.

1.3 Outline of the Thesis

In the following, a brief literature review (Chapter 2) will present similar problems found in the literature as well as how they are solved algorithmically. Then, our mathematical modeling of the full optimization problem will be given (Chapter 3). Afterwards, our algorithmic strategy to solve this full problem will be detailed (Chapter 4). The thesis will conclude with a chapter on numerical experiments to characterize our algorithm and a conclusion (Chapter 5). In Appendix A, will be presented several implementation details that provide small accelerations in the algorithm used for the numerical experiments.

CHAPTER 2 LITERATURE REVIEW

Many problems similar to this one have been researched in the literature. In this section, we give a brief overview of some of these and how they compare to our own, both in terms of the problem itself and how they solve it.

The comparison table below presents an overview of the similarity and differences of the problem formulation of the reviewed literature.

Table 2.1 Comparison of our Problem with Those Studied in the Literature

Problem	Ours	Hamami2010	Zhou2015	Mumey2012	Li2016	England2007	Huang2021
Optimization Criterion	**Least Throughput**	**Most Congested Link**	Network Throughput	Total Capacity	Cost, Delay	Data Loss, Power, Delay	Capacity for Higher Traffic Variations
Designed Topology	**Tree and/or Mesh**	Mesh Inner Topology	Mesh Inner Topology	Mesh Inner Topology	**Tree**	**Tree**	Rooted Directed Acyclic Graph
Type of Antenna	**Single-Beam or Multi-Beam**	**Single-Beam**	**Single-Beam**	**Multi-Beam**	Omni-Directional	Omni-Directional	Omni-Directional
Antenna Alignment	**Angle** $\in [0, 2\pi[$	Fixed Possible Sectors	Fixed Possible Sectors	Fixed	None	None	None
Beam Pattern Modeling	**Physical**	Ideal Cone Shape	Ideal Cone Shape	Ideal Cone Shape	–	–	–
Channel/ Frequency Modeling	**Channel and Frequency**	**Channel**	**Channel**	None	None	None	None
Interference Modeling	**Physical**	0-1	0-1	None	0-1	None	None
Signal Modeling	**Physical**	0-1	0-1	0-1	0-1	0-1	0-1

2.1 Similar Wireless Network Design Problems

The most similar network design problems also optimize the throughput in network design for radio communication through antennas. However, they do not consider challenging terrains and, so, we did not find any in the literature that models its signal physically as we have. As such, none of the reviewed literature takes into account the physical properties of the specific terrain where the network is situated, which determine the amount of signal that is lost between every pair of points. This is problematic since in a tactical setting, such as in disaster relief operations, these physical properties are capital in determining the optimal topology of the network.

Hamami2010 [2]

This conference paper focuses on the design of the inner topology of a full mesh cluster, *i.e.* the topology inside of a single mesh cluster that contains all the nodes. For us, the inner topology of the mesh clusters is not the main focus and we solve it with a heuristic.

They optimize this inner mesh topology to maximize the capacity of the most congested link. This is equivalent to our maximization of the least effective throughput. However, they simulate the traffic down to the data packet level, *i.e.* they consider every interaction between all the nodes that communicate directly to simulate how the smallest unit of a bigger data (a packet) moves from one node to another in the network. In opposition to this very low-level simulation, we stay at a high level by only modeling the maximal effective throughputs that can go through each edge. This allows us to consider more levels of complexity in the network design (waveform/frequency assignments, antenna configuration) and a more accurate modeling of the signal.

The kind of network that they consider uses single-beam antennas with 8 fixed possible alignments and a fixed beam width of 45° (*i.e.* 1/8 of 360°). They model the signal pattern of these antennas as an ideal cone shape with a specific radius within which there is perfect signal and outside of which there is none. These ideal cone shapes are then used to compute the interference also as an all-or-nothing quantity, which results in an all-or-nothing signal modeling. This is in contrast to our problem which, in the case of single-beam antennas, considers 4 possible beam widths and models its signal and interference physically.

Finally, their problem takes into account the channel assignment as does ours, but, only in the sense that two different channels cannot interfere with one another. Since we model the signal physically, the channels and their specific signal frequencies also affect the strength of the signals, the widths of the beams and the interference.

By modeling the signal as an all-or-nothing variable, they can formulate their problem as a Mixed Integer Programming (MIP) problem, which they solve with the iterated local search solver MOSEK.

Zhou2015 [3]

This conference paper also focuses on the design of the inner topology of a full mesh cluster.

They maximize the network capacity as the ratio of the satisfied demand of traffic for every traffic stream (indirect stream of data between two nodes in the network). They view this network capacity as a measure of the network throughput. This is somewhat related to our problem although we measure the network throughput directly for three different typical traffic scenarios whereas they solve their problem given specific traffic streams and their demands.

As in Hamami2010, they consider single-beam antennas of fixed widths (60°) with fixed possible angles (between 6 and 12) and they model both signal and interference as an all-or-nothing value.

Also as in Hamami2010, they consider a fixed number of channels (between 1 and 6) only as orthogonal signal frequencies that do not interfere with one another, but only with themselves.

Their all-or-nothing modeling of signal allows them to formulate their problem as a Linear Programming (LP) problem which they solve via delayed column generation with Cplex and a custom MatLAB code to generate the full problem formulation.

Mumey2012 [4]

This conference paper is also about the design of the inner topology of a full mesh cluster, but this one focuses on the beam configurations of the multi-beam antennas, given pre-computed maximal throughputs between the pairs of nodes.

They maximize the total capacity of the network given that the more beams are activated for a single antenna, the smaller the throughputs this antenna can have for all of its connections.

The antenna alignments are fixed, the beam pattern modeling is again an ideal cone shape of width $360/M°$where M is the number of beams, which they study between 4 and 12. This is again in contrast to our modeling where the antenna alignment is still very important in our multi-beam case to maximize the signal and minimize the interference.

They use a single channel which means that their networks do not have interference.

Their formulation of the problem results in a Mixed Integer Linear Programming (MILP) problem which they solve heuristically by solving an LP relaxation of the problem multiple times in a specific fashion.

2.2 Other Wireless Network Design Problems

Other similar network design problems also address network design for radio communication, although they are interested in optimizing different criteria than throughput. Their use-case is very different than our setting, focusing more on backhaul networks that are between the low-level network at the user level and the high-level core network (typically the internet).

Although their approaches to solving their problems are interesting, the problems themselves are not similar enough to our own to deserve meaningful comparison.

Li2016 [5]

This article optimizes the cost of installation and the communication delay in a tree network connecting a “base station controller” (connected to higher-level core networks) and multiple “base transceiver stations” (connected to end users). The former can be seen as the root of the network while the latter can be seen as the end users of this network.

The antennas are omni-directional with all-or-nothing interference and signal modeling.

Their formulation results in an Integer Linear Programming (ILP) problem which they solve using a shortest paths heuristic and a local search.

England2007 [6]

This article presents a simple heuristic based on minimum spanning trees to optimize data loss, power and delay in tree topologies with omni-directional antennas.

Huang2021 [7]

This article maximizes the capacity for higher traffic variations in a network connecting a “donor base station” to multiple “user base stations”. The topology can be any directed acyclic graph, which are more general than directed trees (arborescences).

The antennas are again omni-directional with no interference and all-or-nothing signal modeling.

They use a Bayesian approach to iteratively learn the structure of the topology.

Other Telecommunications Network Design Problems

Other less similar network design problems include the design of permanent networks which use other communication mediums such as free space optics [8]-[9], fiber [10], or LoRa [11]. Such network design problems usually have other objectives such as cost, power consumption and reliability. Other network design problems consider radio signals but for different kinds of topologies such as sets of trees [12]-[13], star-star topologies [14], or other more general types of topologies created via clustering and routing [15]. These typically prioritize installation costs and power consumption over the performance of the network. Finally, other network design problems arise in many other telecommunication application such as fleet clustering between ships and “ship-masters” [16], formation control of robot swarms [17], and information flow between communicating agents [18]. Each of these problems has its specific set of objectives.

CHAPTER 3 PROBLEM MODELING

Tactical wireless network design is a complex non-linear combinatorial optimization problem. As presented in the literature review, its complexity can be pushed as far as simulating data packet delivery within specified traffic streams to evaluate the practical throughputs of the network. However, we limit ourselves to a higher level characterization of the throughput quality of the networks that describes the maximal effective throughputs that can go through each edge. This allows us to model the physical signal as well as the network solution space to a much more realistic degree.

3.1 Problem Input

The input of our problem is a set of problem parameters as well as a particular instance for which a network must be designed.

3.1.1 Problem Parameters

Multiple parameters are necessary to specify the use case of the problem. They are briefly presented below, but will be explained more as the problem modeling is presented further.

Tree, Mesh or Hybrid Topology This parameter describes whether the topology of the designed network should be a tree, a single full mesh cluster or a hybrid topology with a global tree structure as well as the possibility of one or multiple localized mesh structure(s).

Single-Beam or Multi-Beam This parameter describes whether the antennas at each node are single-beam antennas (parabolic, panel, sector or omni-directional) or multi-beam antennas with 24 possible beams.

Traffic Scenario Relative Weights $(\alpha_A, \alpha_B, \alpha_C)$ These relative weights represent for which of the three possible traffic scenarios (A, B and/or C) the network should be optimized. Depending on the use case, a network might be expected to support specific traffic scenarios more frequently than others, if at all. The three specific traffic scenarios that they represent will be presented in Section 3.3.2.

Number of Frequencies per Channel Although we generally assume 2 possible frequencies per channel (2000 MHz and 2400 MHz for channel 3+ and 4500 MHz and 5000

MHz for channel 4), another use case would have a single frequency per channel (2200 MHz for channel 3+ and 4500 MHz for channel 4). In the sequel we assume the former use case since it is the more general problem due to its non-trivial frequency assignment, with an exponential number of possible frequency assignments when everything else is fixed.

Maximum Number of Nodes per PMP Connection In a PMP connection between a node and multiple other nodes, there can be a physical limit on the number of the latter. In the sequel, we assume a limit of 10 nodes per PMP connection. This means that a node can be connected to at most 10 other nodes on a single PMP connection.

PTP Throughput Coefficient $\lambda_{\mathbf{PTP}} \geq 1$ This coefficient describes how much more throughput can go through a PTP connection than a PMP connection for the same signal. A PMP connection uses the same local channel for multiple edges so the channel must be shared in time between all the edges, *i.e.* only one of these edges actually communicates at once. Since a PTP connection is comprised of a single edge, its waveform protocol does not have to account for channel sharing and it can thus transmit more throughput at once. In the sequel, we assume a PTP throughput coefficient of $\lambda_{\text{PTP}} = 2$.

Mesh Throughput Evaluation Parameters The Mesh waveform protocol has multiple subtleties related to its special structure. As in PMP connections, it has to do local channel sharing in time because all the edges inside of a mesh cluster use the same channel on the same connection. Unlike PMP connections, this sharing in time is much more complex and dynamic. A way to model this without having to simulate the traffic inside the mesh is to consider collision domains inside mesh clusters, *i.e.* the set of edges that have to be silent for each edge to be able to transmit. This is a costly evaluation however, so, in the general case, we usually abstract the impact of the collision domains as a Mesh throughput coefficient $\lambda_{\text{Mesh}} < 1$. There is another issue in mesh clusters that was hinted at previously in the literature review. What we called the inner topology of a mesh, then, relates to the mesh routing problem in our framework, *i.e.* how traffic inside of a mesh is distributed in the many possible paths that exist between every pair of mesh nodes. In our modeling, a greedy prioritization parameter $\xi \in [0, 1]$ will appear to modulate with how much care we solve this routing problem, with $\xi = 0$ being a very naive approach and $\xi > 0$ being a less naive greedy approach. This is explained in more details in section 3.3.2.

With these parameters out of the way, we can begin to describe our modeling of the problem.

3.1.2 Problem Instance

An instance of our problem is comprised of a set of nodes V, their (x_v, y_v) relative coordinates in kilometers, $\forall v \in V$, as well as the two physical properties that characterize the possible edges between them. Given two nodes $u, v \in V$, the two physical properties are the path loss p^f_{uv} and the fade margins m^f_{uv}, both expressed in decibels. Together, they describe the signal that is lost in a transmission between nodes $u \in V$ and $v \in V$ on signal frequency $f \in F$ in MHz where $F := \{2000, 2400, 4500, 5000\}$. As such, these properties are symmetric ($p^f_{uv} = p^f_{vu}$ and $m^f_{uv} = m^f_{vu}$) and the matrices p^f and m^f are also symmetric, $\forall f \in F$.

For instance, an example of a problem instance is pictured.

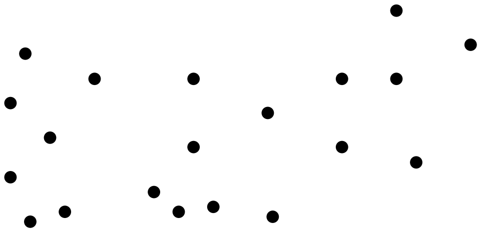

Figure 3.1 Example of an Instance of the Problem

3.2 Network Solution Space

The desired output of our problem is an optimized and fully configured network. Such a network is defined in order by its topology, its configuration and its antenna configurations.

3.2.1 Topology

The topology of a network describes its global tree structure as well as the memberships of its mesh clusters. It can be represented by an undirected graph (V, E) where E is the network's edges $[u, v]$ with $u, v \in V$. In this graph, we represent every mesh cluster as a clique, *i.e.* a set of nodes such that E contains every possible edge between two of these nodes. This is just an abstraction since most these edges would have a zero throughput, meaning that the corresponding nodes cannot communicate directly. This clique modeling is possible because a mesh cluster of size 2, the clique of which would be an edge, would be more advantageous as a PTP connection than a Mesh connection in terms of throughput in the hybrid case.

The case in which the topology must be a tree is trivial, *i.e.* the topology (V, E) must be a tree. The case in which the topology must be a single full mesh is also trivial, *i.e.* the topology (V, E) is a clique. We now present the more interesting hybrid case.

To make sure that the topology has a global tree structure, we construct a second undirected graph which we call the implicit topology $(V \cup K, E^*)$ or implicit tree (Figure 3.3). To construct it, we add a new implicit node $k \in K$ for each clique of size greater than 2 (*i.e.* mesh cluster), we remove the edges of the previous cliques and we add an edge between the implicit nodes $k \in K$ and the explicit nodes $v \in V$ that made up these cliques in (V, E). A valid topology in the hybrid case is such that its implicit topology is a tree, hence its other name of implicit tree.

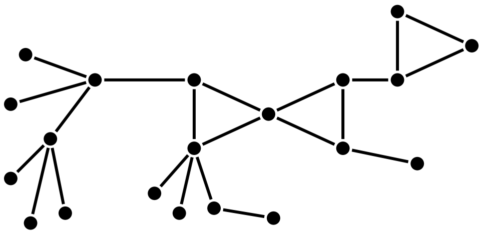

Figure 3.2 Topology

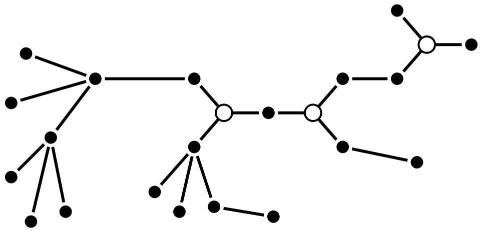

Figure 3.3 Implicit Tree

Another constraint is that a valid topology must admit at least one master hub such that there exists at least one valid waveform and channel assignments on it. For instance, this means that a node cannot be shared by three or more mesh clusters or it would require more than two channels. For the same reason, a node cannot have tree edges if it is also a member of two mesh clusters.

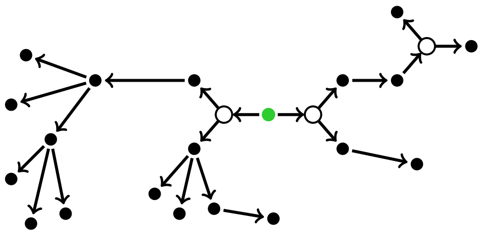

Figure 3.4 Implicit Arborescence

A final constraint, which also applies to the strictly tree case, is that a valid topology must admit at least on master hub such that there exists at least one valid waveform assignment on it such that every PMP has less edges than or equal to the maximum of 10.

3.2.2 Network Configuration

A network configuration is comprised of a selected master hub, a waveform assignment as well as channel and frequency assignments.

Master Hub Selection

A valid master hub is any explicit node $r \in V$. In conjunction with the implicit tree, it defines an implicit arborescence $(V \cup K, \vec{E}^*)$ (Figure 3.4), of which it is the root. This implicit arborescence defines, for every implicit node $w \in (V \cup K)$, a single predecessor except for r which does not have a predecessor. It also defines a set of successors for every implicit node $w \in (V \cup K)$. For instance, the leaves of the implicit arborescence have an empty set of successors.

Given these definitions, we can attribute a number of explicit descendants d_w to every implicit node $w \in (V \cup K)$. It is given by

$$d_w = \mathbb{1}(w \in V) + \sum_{x \text{ successor of } w} d_x, \tag{3.1}$$

where $\mathbb{1}(P) = 1$ if the proposition P is true and $\mathbb{1}(P) = 0$ otherwise. This number of explicit descendants will be necessary to describe the congestion of each edge for the computation of the effective throughput for each traffic scenario.

Waveform Assignment

A waveform assignment can be seen as a partition of the edges into connections. Since we consider only two possible channels, there are relatively few valid waveform assignments on a given topology with a given master hub.

A valid waveform assignment is such that

1. for every mesh cluster, there is a single Mesh connection that contains all of its edges;
2. the tree edges between the master hub $r \in V$ and all of its successors are partitioned in one or two connections;
3. the tree edges between any other explicit node $v \in (V \setminus \{r\})$ and all of its successors form a single connection.

In the last two cases, the connections are PTP if they contain a single edge or PMP if they contain multiple edges. By tree edges, we mean the edges $e \in (E \cap E^*)$, *i.e.* the explicit edges that are not part of a mesh clique. As mentioned previously, a valid waveform must admit a valid channel assignment in which every node is in up to two connections, so, if the master hub $r \in V$ is part of a mesh cluster, its other tree edges if any must form a single connection and, if the master hub is in two different mesh clusters, it cannot have other tree edges.

With such a definition of a valid waveform assignment, we can see that, given a topology and a master hub, the number of valid assignments is upper bounded by the number of partitions of the master hub's successors in 2 sets, unless it is part of a mesh cluster in which case there is a single valid assignment if any.

Channel and Frequency Assignments

A valid channel assignment assigns a channel to every connection such that two connections that share an explicit node must be on different channels. Given a topology, a master hub and a waveform assignment, the channel assignment problem is equivalent to a $|C|$-coloring problem on a graph where each connection is a node and every set of connections that share an explicit node form a clique. Since we consider only $|C| = 2$ and the topology is constructed around an implicit tree, we have that the resulting coloring graph is a tree and there is only 2 possible 2-colorings on it.

This is pictured in our running example below. The only other 2-coloring (not pictured) inverts the assigned channels.

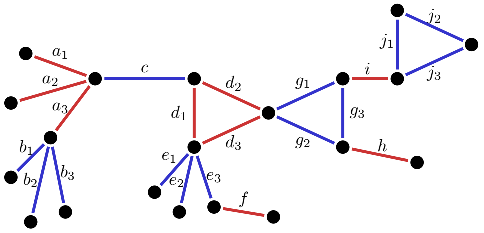

Figure 3.5 Channel Assignment

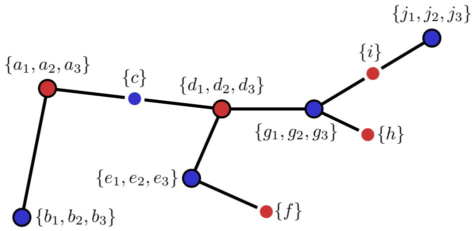

Figure 3.6 Coloring Problem Tree

A valid frequency assignment assigns to every connection one of the two possible frequencies belonging to the assigned channel. There are 2 to the power of the number of connections such valid frequency assignments when everything else is fixed. This becomes a big difficulty later in the algorithmic strategy because of this exponential size. In the case where there is a single frequency per channel, the frequency assignment becomes trivial.

3.2.3 Antenna Configuration

For every node in every connection, an antenna is required to transmit and receive data. All of these antennas need to be configured individually.

Single-Beam Antennas

In the case of single-beam antennas, an antenna configuration is comprised of the choice of a type of antenna $t_a \in \{\text{parabolic}, \text{panel}, \text{sector}, \text{omni-directional}\}$ as well as an alignment $\phi_a \in [0, 2\pi[$ relative to the azimuth if $t_a \neq$ omni-directional.

Multi-Beam Antennas

In the case of multi-beam antennas, an antenna configuration is comprised of an alignment $\phi_a \in [0, 2\pi[$ relative to the azimuth as well as a set of activated beams $B_a \subset [\![0, 23]\!]$ with $0 \leq |B_a| \leq 7$ where the beam 0 is centered on the alignment ϕ_a. When $B_a = \emptyset$, we mean that the antenna is set to its omni-directional mode. For instance, in the figure below, a multi-beam antenna with alignment $\phi_a = 0$ is pictured with 3 activated beams given by $B_a = \{0, 8, 19\}$.

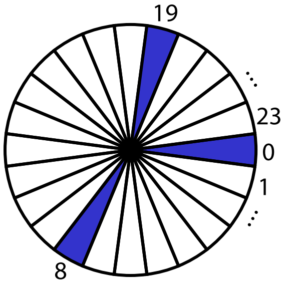

Figure 3.7 Example of a Multi-Beam Antenna with $\phi_a = 0$ and $B_a = \{0, 8, 19\}$

3.3 Objective Function

The optimization criterion of our problem is the least effective throughput of the network. We wish to maximize its weakest link so as to guarantee the highest possible worst-case data transmission speed throughout the network. In order to model this objective, we must first model the physical signal itself and then model how it affects the data traffic in the network.

3.3.1 Signal Modeling

The physical signal modeling begins with the antennas and the physical properties of the terrain. Part of the transmitted signal is the desired useful signal, but the rest contributes to the interference between all the connections that use the same frequency. This interference, combined with the native noise of the receiving antennas, creates a background of noise against which the useful signal competes. This natural competition gives rise to the final Signal-to-Noise Ratio (SINR) that determines the direct throughput that can go through each edge of the network.

Antenna Gain Pattern

The antenna gain pattern describes the strength of the signal that is transmitted/received by an antenna in a particular direction as it relates to the alignment of its beam, for single-beam antenna, or beams, for multi-beam antennas. In the simpler case of single-beam antennas, this signal strength from an antenna of type t_a at node u in the direction of node v on frequency f is given in decibels by

$$g_{uv}^{t_a f} = g_{\max}(t_a, f) - 3\,\frac{\log_{10}|\cos(\Delta_\phi(\phi_a, x_u, y_u, x_v, y_v))|}{\log_{10}|\cos(\Delta_{3dB}(t_a, f)/2)|}, \tag{3.2}$$

where $g_{\max}(t_a, f)$ is the gain at the maximal direction of ϕ_a, $\Delta_\phi(\phi_a, x_u, y_u, x_v, y_v)$ is the angle deviation between this maximal direction and the direction towards v, and $\Delta_{3dB}(t_a, f)$ is the 3 dB beam width, *i.e.* the width of the beam at which there is a 3 dB loss in signal. For an omni-directional antenna, this is reduced to

$$g_{uv}^{\text{omni-directional}f} = g_{\max}(\text{omni-directional}, f). \tag{3.3}$$

We use

$$g_{\max}(t_a, f) = \begin{cases} 21\text{ dB}, & \text{if } t_a = \text{parabolic and channel}(f) = 3+ \\ 28\text{ dB}, & \text{if } t_a = \text{parabolic and channel}(f) = 4 \\ 16\text{ dB}, & \text{if } t_a = \text{panel and channel}(f) = 3+ \\ 22\text{ dB}, & \text{if } t_a = \text{panel and channel}(f) = 4 \\ 14\text{ dB}, & \text{if } t_a = \text{sector and channel}(f) = 3+ \\ 15\text{ dB}, & \text{if } t_a = \text{sector and channel}(f) = 4 \\ 6\text{ dB}, & \text{if } t_a = \text{omni-directional and channel}(f) = 3+ \\ 8\text{ dB}, & \text{if } t_a = \text{omni-directional and channel}(f) = 4 \end{cases}$$

and

$$\Delta_{3dB}(t_a, f) = \begin{cases} 20^\circ, & \text{if } t_a = \text{parabolic and channel}(f) = 3+ \\ 8^\circ, & \text{if } t_a = \text{parabolic and channel}(f) = 4 \\ 40^\circ, & \text{if } t_a = \text{panel and channel}(f) = 3+ \\ 16^\circ, & \text{if } t_a = \text{panel and channel}(f) = 4 \\ 110^\circ, & \text{if } t_a = \text{sector and channel}(f) = 3+ \\ 90^\circ, & \text{if } t_a = \text{sector and channel}(f) = 4. \end{cases}$$

Plotting the gain pattern in front of a parabolic antenna on channel 3+ is pictured in Figure 3.8.

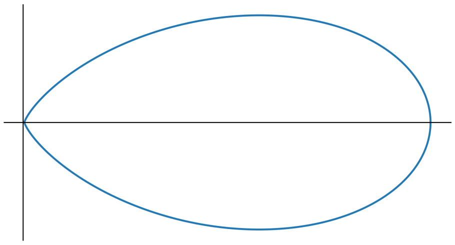

Figure 3.8 Gain Pattern of a Parabolic Antenna with $\phi_a = 0$ on channel 3+

For a multi-beam antenna at node u on frequency f, the signal strength in the direction of node v is given by

$$g^f_{uv} = 10\,\log_{10} \sum_{b\in B_a} \exp_{10}\left(\left(g_{\max}(|B_a|, f) - 3\,\frac{\log_{10}\left|\cos\left(\Delta_\phi(\phi_a + b\frac{2\pi}{24}, x_u, y_u, x_v, y_v)\right)\right|}{\log_{10}|\cos\left(\Delta_{3dB}(f)/2\right)|}\right)\Big/\,10\right), \tag{3.4}$$

where the gain at the maximal direction $g_{\max}(|B_a|, f)$ now depends on the number of activated beams $|B_a|$. Again, in the case where $B_a = \emptyset$, this is reduced to

$$g^f_{uv} = g_{\max}(0, f). \tag{3.5}$$

We use

$$g_{\max}(|B_a|, f) = \begin{cases} 13 - 10\log_{10}(|B_a|)\text{ dB}, & \text{if } |B_a| > 0 \text{ and } \text{channel}(f) = 3+ \\ 15 - 10\log_{10}(|B_a|)\text{ dB}, & \text{if } |B_a| > 0 \text{ and } \text{channel}(f) = 4 \\ 4\text{ dB}, & \text{if } |B_a| = 0 \text{ and } \text{channel}(f) = 3+ \\ 6\text{ dB}, & \text{if } |B_a| = 0 \text{ and } \text{channel}(f) = 4 \end{cases}$$

and

$$\Delta_{3dB}(f) = \begin{cases} 60^\circ, & \text{if channel}(f) = 3+ \\ 50^\circ, & \text{if channel}(f) = 4. \end{cases}$$

Signal Strength

Given the antenna signal strengths from node u to node v and from node v to node u, the base signal strength in decibel-milliwatts is then given by

$$s^f_{uv} = 30 + g^{t_a f}_{uv} + g^{t_b f}_{vu} - p^f_{uv}, \tag{3.6}$$

where p^f_{uv} is the path loss and where we assume a transmitted power of 30 dBm.

Interference

These base signal strengths can be computed between every pair of nodes that use the same frequency, irrespective of whether they should be directly communicating or not. This gives rise to the interference on the useful signal between directly communicating nodes u and v. It is given linearly (*i.e.* not in decibels) by

$$i_{uv}^{f} = \sum_{\substack{w \text{ s.t. it has an} \\ \text{antenna on } f \text{ on a} \\ \text{different connection}}} \left(\exp_{10}\left(\frac{s_{uw}^{f}}{10}\right) + \exp_{10}\left(\frac{s_{wv}^{f}}{10}\right) \right). \tag{3.7}$$

Signal-to-Noise Ratio

With the useful signal s_{uv}^{f} and the interference i_{uv}^{f} that it competes with, we can finally compute the SINR between directly communicating nodes u and v. It is given by

$$S_{uv} = s_{uv}^{f} - m_{uv}^{f} - 10\,\log_{10}\left(i_{uv}^{f} + \exp_{10}\left(\frac{NP}{10}\right)\right), \tag{3.8}$$

where m_{uv}^{f} is the fade margin and NP is the receiver antenna's noise power in dBm. Assuming a 20 Mhz bandwidth and a 10 dB noise figure, we use

$$NP = -174 + 10\,\log_{10}\left(20 \cdot 10^{6}\right) + 10.$$

Direct Throughput

Given the SINR S_{uv} between nodes u and v, their direct throughput TP_{uv} in megabits per second can be taken from Table 3.1 [19].

Table 3.1 Direct Throughput Lookup Table

SINR (dBm)	Throughput (Mbps)
$S < 2$	0
$2 \leq S < 5$	6.5
$5 \leq S < 9$	13
$9 \leq S < 11$	19.5
$11 \leq S < 15$	26
$15 \leq S < 18$	39
$18 \leq S < 20$	52
$20 \leq S < 25$	58.5
$25 \leq S < 29$	65
$29 \leq S$	78

This is the direct throughput of a PMP connection. If the waveform of the associated connection is PTP, then this direct throughput is multiplied by λ_{PTP}. If the waveform is Mesh and we do not model collisions, the direct throughput is multiplied by λ_{Mesh}.

3.3.2 Traffic Modeling

Once we have computed the maximal direct throughputs that every edge in the topology can transmit in either directions, we can model how they affect the traffic within the network.

Traffic Scenarios

We want to maximize the network's worst-case scenario in terms of least effective throughput, *i.e.* we are looking for an objective function under the form

$$\min_{\substack{[u,v]\in E:\\ n_{uv}>0}} \frac{TP_{uv}}{n_{uv}},$$

which is then maximized where n_{uv} is the number of simultaneous data streams that go through edge $[u,v] \in E$. However, to determine n_{uv}, which is a measure of the congestion of each edge, we need to know in what kind of traffic scenario the network is operating. Since this varies significantly on the actual use case of the network, we consider three different scenarios that span the whole range of possibilities. They are the following.

Scenario A : a single data stream from any explicit node to any other explicit node, for a total of 1 single data stream at any given time.

Scenario B : data streams from every explicit node to the master hub or, equivalently, from the master hub to every other explicit node, for a total of $|V|-1$ data streams at once.

Scenario C : data streams from every explicit node to every other explicit node, for a total of $|V|(|V|-1)$ data streams at once.

The problem parameters include relative weights $(\alpha_A, \alpha_B, \alpha_C)$ for these three scenarios such that the final objective function evaluates a weighted combination of these scenarios with

$$\alpha_A \min_{\substack{[u,v]\in E:\\ n^A_{uv}>0}} \frac{TP_{uv}}{n^A_{uv}} + \alpha_B \min_{\substack{[u,v]\in E:\\ n^B_{uv}>0}} \frac{TP_{uv}}{n^B_{uv}} + \alpha_C \min_{\substack{[u,v]\in E:\\ n^C_{uv}>0}} \frac{TP_{uv}}{n^C_{uv}}.$$

Each weight can take any nonnegative value describing the relative frequency of the associated scenario with 0 meaning that the network is expected to never be in such a scenario.

In scenario A, the bottleneck of the network in the worst-case scenario is the edge $[u, v] \in E$ that is necessary to at least one direct or indirect communication with the worst direct throughput. In the case of a network with a strictly tree topology, this is simply

$$n_{uv}^{A} = 1, \quad \forall [u, v] \in E.$$

In the strictly mesh or the hybrid case, we need to introduce the notion of mesh routes, *i.e.* the optimal paths $\dddot{R}_{uv}^{X}$ for the traffic scenario X between every pair of nodes u, v within a same mesh cluster $k \in K$. The so-called mesh routing problem of defining such optimal paths will be detailed in the next subsection. Assuming these paths have already been defined, the general hybrid case, $\forall [u, v] \in E$, is given by

$$n_{uv}^{A} = \begin{cases} 1, & \text{if } [u, v] \in E^{*} \\ \mathbb{1}\Big(\exists w, x \in V[k],\ [u, v] \in \dddot{R}_{wx}^{A}\Big), & \text{if } \exists k \in K,\ [u, v] \in E[k], \end{cases} \tag{3.9}$$

where $V[k] \subseteq V$ is the set of nodes that belong to a mesh cluster $k \in K$ and $E[k] \subseteq E$ is the set of its edges that form a clique in the topology (V, E). As such, the objective function of scenario A can also be rewritten

$$\min_{\substack{[u,v] \in E: \\ n_{uv}^{A} > 0}} \frac{TP_{uv}}{n_{uv}^{A}} = \min \left\{ \min_{[u,v] \in (E \cap E^{*})} TP_{uv}, \quad \min_{k \in K} \min_{\substack{u,v \in V[k]: \\ u \neq v}} \dddot{T}\dddot{P}_{uv}^{A} \right\},$$

where $\dddot{T}\dddot{P}_{uv}^{A}$ is the worst direct throughput on the mesh route $\dddot{R}_{uv}^{A}$.

In traffic scenario B, every explicit node is sending a data stream to the master hub except for the master hub r itself. The edge connecting a leaf of the implicit arborescence to the rest of the network carries a single data stream, but as we get nearer to the master hub, the edges accumulate more data streams from their descendants as the data streams converge towards the master hub. The bottleneck in this scenario is the most overloaded edge with respect to its direct throughput. Under the form

$$\min_{\substack{[u,v] \in E: \\ n_{uv}^{B} > 0}} \frac{TP_{uv}}{n_{uv}^{B}},$$

we have that, $\forall [u, v] \in E$,

$$n_{uv}^{B} = \begin{cases} d_v, & \text{if } (u, v) \in \vec{E}^{*} \\ \displaystyle\sum_{w \in V[k]} d_w \, \mathbb{1}\Big([u, v] \in \dddot{R}_{wx}^{B}\Big), & \text{if } \exists k \in K,\ [u, v] \in E[k] \text{ with } (x, k) \in \vec{E}^{*}. \end{cases} \tag{3.10}$$

For a tree edge $(u, v) \in \vec{E}^*$, the number of data streams that it carries is simply the number of explicit descendants d_v of explicit node v as defined by the implicit arborescence. For mesh edges, each edge carries the weighted sum of all the paths that go through it from every node in the mesh $w \in V[k]$ to the predecessor x of the mesh $k \in K$, *i.e.* $x \in V[k]$ such that $(x, k) \in \vec{E}^*$.

In traffic scenario C, all explicit nodes are sending data to all other nodes. Each edge carries all the signals of its descendants to the rest as well as all the signals coming from the opposite direction. The bottleneck is again the most overloaded edge with respect to its direct throughput. In this case, the congestion, $\forall [u, v] \in E$, is given by

$$\mathrm{n}^C_{uv} = \begin{cases} 2\, d_v(|V| - d_v), & \text{if } (u, v) \in \vec{E}^* \\ 2\, (|V| - d_k) \sum\limits_{w \in V[k]} d_w\, \mathbb{1}\Big([u, v] \in \dddot{R}^C_{wx}\Big) + 2 \sum\limits_{\substack{w_1, w_2 \in V[k]: \\ w_1 \neq x, \\ w_2 \neq x, \\ w_1 < w_2}} d_{w_1} d_{w_2}\, \mathbb{1}\Big([u, v] \in \dddot{R}^C_{w_1 w_2}\Big), & \text{if } \exists k \in K,\ [u, v] \in E[k] \text{ with } (x, k) \in \vec{E}^*. \end{cases} \tag{3.11}$$

In practice, because an objective function with only minima is mostly flat and makes the optimization more difficult, we prefer the objective function

$$\begin{aligned} O \;=\; & \alpha_A \left(\min_{\substack{[u,v] \in E: \\ n^A_{uv} > 0}} \frac{TP_{uv}}{n^A_{uv}} + \frac{1}{Z} \operatorname*{mean}_{\substack{[u,v] \in E: \\ n^A_{uv} > 0}} \frac{TP_{uv}}{n^A_{uv}} \right) \\ & + \alpha_B \left(\min_{\substack{[u,v] \in E: \\ n^B_{uv} > 0}} \frac{TP_{uv}}{n^B_{uv}} + \frac{1}{Z} \operatorname*{mean}_{\substack{[u,v] \in E: \\ n^B_{uv} > 0}} \frac{TP_{uv}}{n^B_{uv}} \right) \\ & + \alpha_C \; (|V| - 1) \left(\min_{\substack{[u,v] \in E: \\ n^C_{uv} > 0}} \frac{TP_{uv}}{n^C_{uv}} + \frac{1}{Z} \operatorname*{mean}_{\substack{[u,v] \in E: \\ n^C_{uv} > 0}} \frac{TP_{uv}}{n^C_{uv}} \right), \end{aligned} \tag{3.12}$$

where Z is a large renormalization constant and mean is the usual arithmetic mean. Given our direct throughput table that has a maximum of 78 Mbps, we use

$$Z = \frac{78}{2}.$$

We renormalize the objective of scenario C so that the magnitude of its values are comparable with those of scenarios A and B. This ensures that the relative weights directly adjust how much each scenario is optimized rather than being additionally modulated by different magnitudes of values. We use a constant of $|V| - 1$ since it is the minimum value of a non-zero n^C_{uv}, corresponding to an edge $[u, v]$ connecting an explicit leaf to the rest of the topology.

Mesh Routing

In order to evaluate the traffic inside of a mesh cluster, we must first define the optimal paths $\dddot{R}^X_{w_1w_2}$ for all pairs of nodes $w_1, w_2 \in V[k]$ and for all scenarios $X \in \{A, B, C\}$. Since this is by itself a very complex combinatorial optimization problem, we only wish to find a suboptimal solution of good quality. Our most naive solution is to take

$$\dddot{R}^A_{w_1w_2} = \dddot{R}^B_{w_1w_2} = \dddot{R}^C_{w_1w_2}$$

as the best path between w_1 and w_2 assuming that no other traffic is being streamed in the mesh. We do this using the classic Floyd-Warshall shortest path algorithm [20], where we maximize the quality $\dddot{T}\dddot{P}^X_{w_1w_2}$ of each path, which is taken as the smallest direct throughput that it goes through. This is the optimal solution for scenario A (if mesh collisions are not considered), but it is not necessarily the case for the other scenarios as such an algorithm will prioritize the same best partial routes for multiple pairs of nodes. This, in turn, will overload these preferred partial routes and their effective throughput will be greatly diminished. For the other two scenarios, we wish to distribute the data streams in as many different paths as possible to avoid this problem.

For this, we propose another slightly less naive approach, which is to take

$$\dddot{R}^B_{w_1w_2} = \dddot{R}^C_{w_1w_2},$$

and greedily construct these optimal paths by batch of pairs of nodes using the Floyd-Warshall algorithm. We introduce the greedy prioritization parameter $\xi \in [0, 1]$ and take the number of batches as

$$n_{\text{batch}} = \left\lceil (V[k](V[k]-1))^{\xi} \right\rceil,$$

and we distribute all the paths equally in these batches. With $\xi = 0$, we have a single batch of $V[k](V[k]-1)$ paths, *i.e.* the naive solution. With $\xi = 1/2$, we have approximately $\sqrt{V[k](V[k]-1)}$ batches of $\sqrt{V[k](V[k]-1)}$ paths. With $\xi = 1$, we have $V[k](V[k]-1)$ batches of a single path. The difficulty then becomes how to order the paths in the batches to get the best paths. Our idea is to prioritize the paths with the biggest weights in equations (3.10) and (3.11) which define the congestions for scenarios B and C, respectively.

We prioritize the most the paths that go from the predecessor of the mesh x to all the other nodes w and we order these paths by descending weight d_w. For the rest of the paths from w_1 to w_2, we order them by descending weight $d_{w_1}d_{w_2}$. For both the first and the second sets of paths, we order at random the paths that have the same weight. This randomness allows us to run this greedy routing approach multiple times and use the one with the best final objective value O.

Mesh Collisions

Mesh collision domains model the local edges within a mesh cluster that must be silent in order for a specific edge to be able to communicate. When we do not take them into account, we multiply the mesh direct throughputs by the coefficient $\lambda_{\text{Mesh}} < 1$ to model this behavior. To take them into account, we modify the denominators $n^A_{w_1w_2}$, $n^B_{w_1w_2}$ and $n^C_{w_1w_2}$ for every mesh edge $[w_1, w_2]$ such that

$$n^{*X}_{w_1w_2} = n^X_{w_1w_2} + \sum_{\substack{\text{all used edges } [w_3,w_4] \\ \text{that must be silent} \\ \text{for } [w_1,w_2] \text{ to} \\ \text{communicate}}} n^X_{w_3w_4}. \tag{3.13}$$

For an edge $[w_1, w_2]$, the used edges $[w_3, w_4]$ that must be silent, *i.e.* its collision domain, are all the edges such that either w_1 is a neighbor in a certain sense of w_3 or the same for w_1 and w_4, w_2 and w_3 and w_2 and w_4. The neighborhood relationship that we mean here between, for instance, nodes w_1 and w_3 is that the SINR $S_{w_1w_3}$ is above a certain threshold, which is 6 dB below the minimal SINR for direct throughput in Table 3.1. In our case, since the minimal SINR to have a non-zero throughput is 2 dB, this threshold is -4 dB.

3.4 Full Optimization Problem

Given all of these definitions, our modeling of the full tactical wireless network design optimization problem can now be informally stated as

$$\max_{\text{valid topology}} \max_{r \in V} \max_{\substack{\text{valid partition} \\ \text{of } r\text{'s successors} \\ \text{in 2 sets}}} \max_{\substack{\text{valid channel} \\ \text{assignment}}} \max_{\substack{\text{valid frequency} \\ \text{assignment}}} \max_{\substack{\text{valid antenna} \\ \text{configurations}}} O. \tag{3.14}$$

This combinatorial optimization problem is both complex because of the non-linearity of its objective function O and because of its massive solution space. The first maximum on valid topologies has a solution space which has a size that is exponential with $|V|$. The number of valid waveform assignments is also bounded exponentially with the number of successors to the master hub, which depends on the topology and the master hub. Finally, the number of valid frequency assignments, given the rest, is exponential in the number of connections. The exploration of such a massive solution space is difficult, but it is made even more difficult by the non-linearity of O and, especially, the global effect of interference in networks.

The previous problem statement is for the strictly tree and general hybrid cases. For the strictly mesh case, *i.e.* the topology being a single full mesh cluster, the problem is simplified substantially to

$$\max_{r \in V} \; \max_{f \in F} \; O,$$

because there is a single valid topology, a single valid waveform assignment with a single frequency and a single valid antenna configuration for every antenna (omni-directional antennas or multi-beam antennas in their omni-directional mode). We solve this special case purely by exhaustive enumeration on r and f. In the next chapter on algorithmic strategy, we will focus on the strictly tree and general hybrid cases.

CHAPTER 4 ALGORITHMIC STRATEGY

We solve this complex non-linear combinatorial optimization problem with a sophisticated problem separation that allows us to tackle each sub-problem with a variety of methods ranging from exact methods and approximations to meta-heuristics and heuristics.

4.1 Problem Separation

We separate the full optimization in three nested sub-problems.

$$\underbrace{\max_{\text{valid topology}}}_{\substack{P_0 \\ \text{Local Search}}} \underbrace{\max_{r\in V} \max_{\substack{\text{valid partition} \\ \text{of } r\text{'s successors} \\ \text{in 2 sets}}} \max_{\substack{\text{valid channel} \\ \text{assignment}}} \max_{\substack{\text{valid frequency} \\ \text{assignment}}}}_{\substack{P_1 \\ \text{Exhaustive Enumeration, Approximations, \textit{etc.}}}} \underbrace{\max_{\substack{\text{valid antenna} \\ \text{configurations}}}}_{\substack{P_2 \\ \text{Heuristic}}} O \tag{4.1}$$

We define

P_0 as finding the optimal topology, given solvers for P_1 and P_2;

P_1 as finding the optimal network configuration, given a topology and a solver for P_2;

P_2 as finding the optimal antenna configurations, given a topology and a network configuration.

4.2 P_0 Topology Local Search

Given solvers for P_1 and P_2, we solve P_0 with meta-heuristic local search in the space of topologies. Because of the nested structure of the three sub-problems, this local search is the outermost loop of our full algorithm.

4.2.1 Neighborhoods

The most defining characteristic of any local search is the neighborhood structure used to move in the space of solutions. We have two different neighborhoods: one to change the global tree structure (tree neighborhood) and one to change the mesh cluster memberships (mesh neighborhood). The former is used in both the strictly tree and general hybrid cases while the latter is used only in the hybrid case. As such, in the hybrid case, our local search is a variable neighborhood search.

Tree Neighborhood

Our tree neighborhood N_{tree} is the edge-swap neighborhood. To construct the neighbor topologies, for each tree edge $e \in (E \cap E^*)$ in the current topology, we remove it, which disconnects the topology into two connected components, and we consider every possible way of reconnecting these two connected components with another edge. Every such neighbor is a valid topology (provided that it admits a valid network configuration) and this newly added edge automatically becomes a tree edge in the resulting topology.

En example of such neighbors is presented in the figure below where the edges in blue were "swapped".

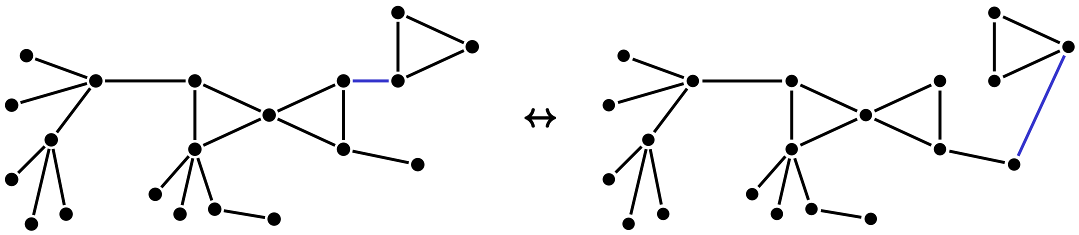

Figure 4.1 Neighbor Topologies (Tree Neighborhood)

Implementation details for this neighborhood, such as the edge ordering for the first improvement neighborhood search, are presented in appendix A.

Mesh Neighborhood

Our mesh neighborhood N_{mesh} consists of all the topologies that can be obtained by either adding/removing an adjacent node to a mesh cluster or by creating/destroying a mesh cluster.

More specifically, a mesh cluster can be created from a node and its tree neighbors if the originating node is not already in a cluster and it is not a leaf (*i.e.* it has more than one tree neighbor).

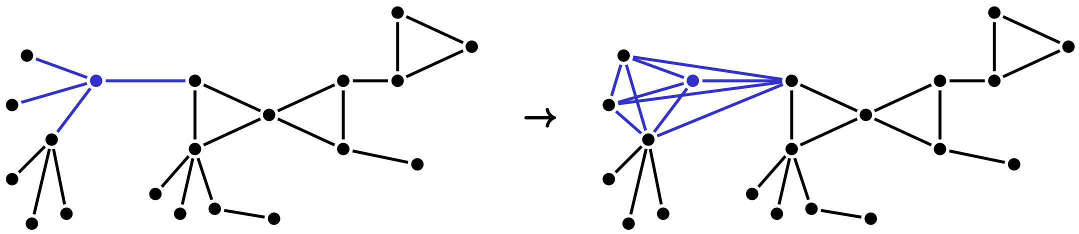

Figure 4.2 Neighbor Topologies (Mesh Creation Sub-Neighborhood)

Conversely, a mesh cluster of size 3 can be destroyed into 3 nodes linked by 2 tree edges.

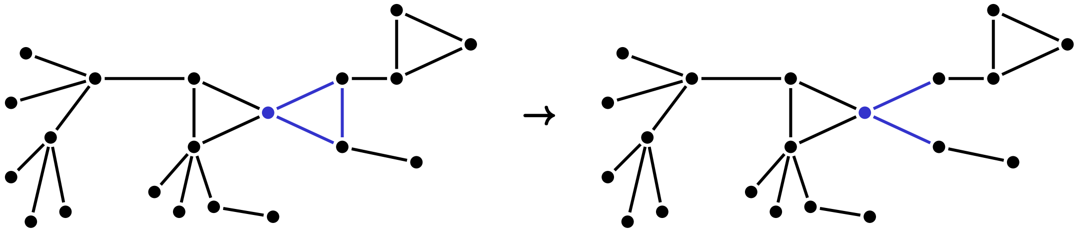

Figure 4.3 Neighbor Topologies (Mesh Destruction Sub-Neighborhood)

A node adjacent to a mesh cluster can be included into it.

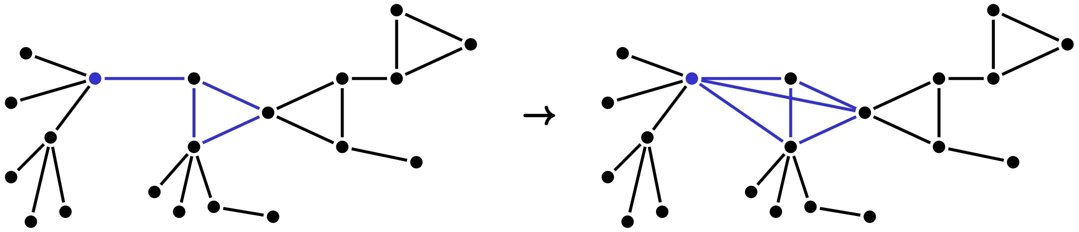

Figure 4.4 Neighbor Topologies (Mesh Inclusion Sub-Neighborhood)

Conversely, a node inside of a mesh cluster of size 4 or more can be excluded from it. In that case, the two resulting connected components can again be reconnected by any possible tree edge.

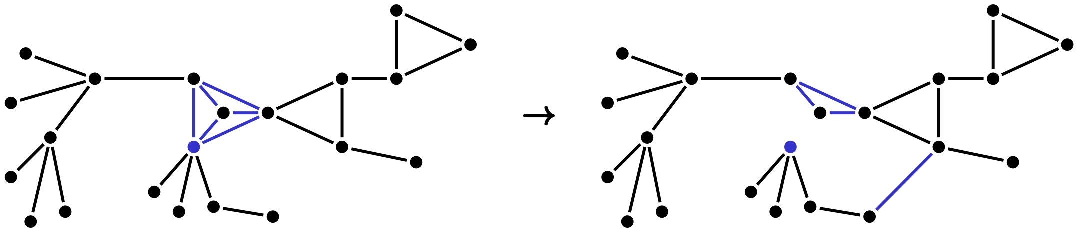

Figure 4.5 Neighbor Topologies (Mesh Exclusion Sub-Neighborhood)

Finally, two mesh clusters which are adjacent can be fused into a single mesh cluster.

Implementation details for this neighborhood, such as the sub-neighborhood inter-ordering and intra-ordering for the first improvement neighborhood search, are presented in appendix A.

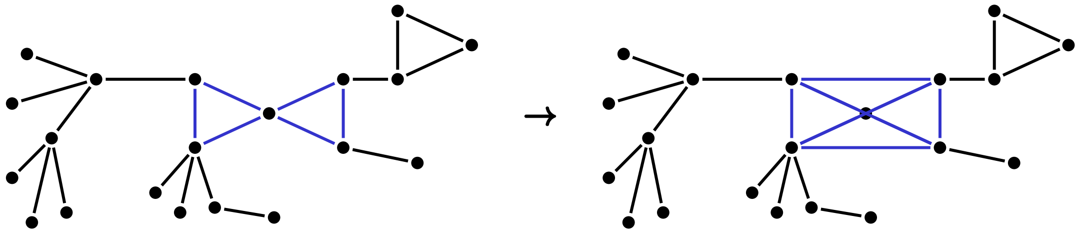

Figure 4.6 Neighbor Topologies (Mesh Fusion Sub-Neighborhood)

4.2.2 Taboo Search

For P_0, our meta-heuristic local search is a Taboo search. The specificity of a Taboo search, compared to a regular local search, is that it explores more of the solution space than just the closest local minima because it is never allowed to move in the direction from which it came. There is a caveat to this rule if moving in such a direction explores a never before seen region. This is possible because of a Taboo list that records the last few moves or, rather, the opposites of the last few moves which then become Taboo for the next few moves, *i.e.* we are not allowed to do them unless they allow us to reach a new best, which means a new region to explore.

The pseudo-code for a generic Taboo search is presented below.

Algorithm 4.1 Taboo Search Pseudo-Code

```
Initialize the solution $x \leftarrow x_0$, the best solution $x^* \leftarrow x_0$ and the Taboo list $L \leftarrow [\,]$.
While stopping conditions are not met,
    Find $x' \in \arg\max_{x' \in N(x)} \; H(x')$
                  s.t. $x' \notin L$ or $H(x') > H(x^*)$.
    Make move $x \leftarrow x'$ and add the opposite move to $L$.
    If $x$ is the new best, update $x^* \leftarrow x$.
Return the best solution $x^*$.
```

Taboo Beam Search

An extension of Taboo search called sequential fan candidate list search or beam search [21] runs κ parallel Taboo searches and, at every iteration, it only keeps the κ best solutions among the neighborhoods of the κ previous solutions. They might all come from the same previous solutions or all from different solutions. When multiple new solutions come from the same previous solution, that previous solution and its Taboo list is duplicated. The previous solutions that do not generate any of the new solutions are removed along with their Taboo list. This allows the search to explore more promising solutions than by always only taking the best solution locally. Sometimes the best solution can be just two moves away, but if the first move to get there is not as good as some other move in a completely different direction, a normal Taboo search will miss it. By using a Taboo beam search, we allow ourselves more room to find such solutions.

The pseudo-code for a generic Taboo beam search is presented below.

Algorithm 4.2 Taboo Beam Search Pseudo-Code

```
Initialize the solutions $\mathcal{X} \leftarrow (x_0)$, the best solutions $\mathcal{X}^* \leftarrow (x_0)$, the best solution
  $x^* \leftarrow x_0$ and the Taboo lists $\mathcal{L} \leftarrow ([\,])$.
While stopping conditions are not met,
    For $x \in \mathcal{X}$ and $L \in \mathcal{L}$,
        Find the $\kappa$ best moves with $\max_{x' \in N(x)} \; H(x')$
                                          s.t. $x' \notin L$ or $H(x') > H(x^*)$.
    Keep only the best $\kappa$ moves $\mathcal{M}$ that result in different solutions.
    For each $M \in \mathcal{M}$, corresponding $x \in \mathcal{X}$ and $L \in \mathcal{L}$,
        Make move $M$ to $x \in \mathcal{X}$ and add its opposite to $L \in \mathcal{L}$.
        If resulting solution $x$ is one of the $\kappa$ bests, add it to $\tilde{\mathcal{X}}^*$ keeping only the $\kappa$
          bests and, if the best, update $x^* \leftarrow x$.
Return the best solution $x^*$.
```

Taboo Lists

For each neighborhood, we keep two taboo lists: one for the last few dropped components and one for the last few added components. For each move of the tree edge-swap neighborhood, the dropped component is the tree edge that is removed and the added component is the tree edge that reconnects the topology. However, since it is the opposite of the move that becomes Taboo, the dropped tree edge goes in the Taboo add list (we cannot add it again for the next few moves) and the added tree edge goes in the Taboo drop list (we cannot drop it again for the next few moves).

For each move of the mesh neighborhood, depending on the sub-neighborhood, the dropped component that goes in the Taboo add list and the added component that goes in the Taboo drop list can be a cluster (creation, destruction) or a node (inclusion/exclusion). The fusion sub-neighborhood does not need to be recorded in the Taboo lists since it does not have a converse sub-neighborhood. This is because separating a cluster in two would have an exponential number of neighbors and the same move can be achieved by a series of exclusions, creation and inclusions.

The other defining characteristic of the Taboo lists which has a big impact on the performance of the algorithm is their lengths, *i.e.* for how many moves does a component have Taboo drop/add status. A good rule of thumb is to set the Taboo drop/add list length proportional to the square root of the approximate number of possible components to be dropped/added. For the taboo drop list of the tree neighborhood, this would be the number of tree edges in the current topology. As such, we use a taboo drop list of size

$$\left\lfloor \frac{1}{2}\sqrt{|V| - \sum_{k\ \ K} (|V[k]| - 1)\ \ - 1} \right\rceil , \tag{4.2}$$

where $\lfloor x \rceil$ means x rounded to the nearest integer. For the taboo add list of the tree neighborhood, this would be the number of possible tree edges that can be added, *i.e.* those that are not already inside of a mesh cluster. As such, we use a taboo add list of size

$$\left\lfloor \sqrt{\frac{|V|\,(|V| - 1)}{2} - \sum_{k\in K} \frac{|V[k]|\,(|V[k]| - 1)}{2}} \right\rceil . \tag{4.3}$$

For the taboo drop list of the mesh neighborhood, this would be the sum of the number of clusters of size 3 and the number of mesh nodes. As such, we use a taboo drop list of size

$$\left\lfloor \sqrt{|\{k \in K \mid |V[k]| = 3\}| + \sum_{k\in K} |V[k]|} \right\rceil . \tag{4.4}$$

For the taboo add list of the tree neighborhood, this would be the sum of the number of non-leaf nodes and the number of adjacent nodes to mesh clusters. Because we do not want to have to analyse the topology every time, which would also make the list length less stable in time, we use the maximum number of non-leaf nodes and the average number of adjacent nodes to mesh clusters. For this, we assume an average number of adjacent node per mesh node of 1. As such, we use a taboo add list of size

$$\left\lfloor \frac{1}{2}\sqrt{|V| - 2 + \sum_{k\in K} |V[k]|} \right\rceil . \tag{4.5}$$

Although these taboo list lengths have not been fully fine-tuned and optimized, these parameters yield good results in practice.

P_0 Variable Neighborhood Taboo Beam Search Algorithm

Given the previous definitions, our P_0 topology local search solver is a variable neighborhood Taboo beam search with the previously defined neighborhoods and Taboo lists. However, we need a few final definitions before we can present its algorithm.

We define

$$F = \max_{r \in V} \max_{\substack{\text{valid partition} \\ \text{of } r\text{'s successors} \\ \text{in 2 sets}}} \max_{\substack{\text{valid channel} \\ \text{assignment}}} \max_{\substack{\text{valid frequency} \\ \text{assignment}}} P_2, \tag{4.6}$$

where the antenna configurations are given by our P_2 heuristic solver. We can evaluate this objective function with a full exhaustive enumeration on the network configuration. However, the number of such network configurations is massive because of the potentially exponential size of valid partitions of the master hub's successors and of the definitely exponential size of the valid frequency assignments in the number of connections. In practice, F can only be evaluated for very small instances $|V| \leq 10$ and even then, it is too slow to be used in a local search framework. To overcome this problem, we have formulated three different pseudo-objectives that approximate F in different ways.

Our first pseudo-objective $\overline{F}$ is the one that we use the most often since we use it for the neighborhood search of each parallel Taboo search at every iteration. It is given by

$$\overline{F} = \max_{r \in V} \max_{\substack{\text{valid partition} \\ \text{of } r\text{'s successors} \\ \text{in 2 sets}}} \max_{\substack{\text{valid channel} \\ \text{assignment}}} \overline{P}_2, \tag{4.7}$$

where we assume a single frequency per channel. Since using a single frequency per channel removes the networks ability to mitigate interference, we do not consider interference for this pseudo-objective. As such, $\overline{P}_2$ uses the SINR $\overline{S}_{uv}$ given by

$$\overline{S}_{uv} = s^f_{uv} - m^f_{uv} - NP \tag{4.8}$$

to compute the direct throughouts $\overline{TP}_{uv}$.

Our second pseudo-objective $\hat{F}$ is used as an upper bound to tell us if it is worth evaluating our third and significantly more costly pseudo-objective $\tilde{F}$. $\hat{F}$ is given by

$$\hat{F} = \max_{r \in V} \max_{\substack{\text{valid partition} \\ \text{of } r\text{'s successors} \\ \text{in 2 sets}}} \max_{\substack{\text{valid channel} \\ \text{assignment}}} \hat{P}_2, \tag{4.9}$$

where, again, the antenna configurations are provided by our P_2 heuristic solver. Like $\overline{P}_2$, $\hat{P}_2$ also does not consider interference, but, additionally, it does not consider a single frequency

f to compute the signal strengths. Rather, for every component of the signal strength computation, it uses the specific frequency of the channel that outputs the strongest signal. As such, for their respective computations, it uses the frequency $\hat{f}_p$ for the minimal path loss, the frequency $\hat{f}_m$ for the minimal fade margin, the frequency $\hat{f}_g$ for the maximal max gain and the frequency $\hat{f}_{3dB}$ for the maximal 3 dB beam width. Since those maximixal frequencies are different for the different components, the resulting direct throughputs are not possible for a practical antenna. For a strictly tree topology, $\hat{F}$ is an actual upper bound on F. However, because of the routing and collision problems, this is not the case for topologies containing mesh clusters.

Our final pseudo-objective $\tilde{F}$ provides a feasible network for

$$\max_{r \in V} \max_{\substack{\text{valid partition}\\\text{of } r\text{'s successors}\\\text{in 2 sets}}} \max_{\substack{\text{valid channel}\\\text{assignment}}} \max_{\substack{\text{valid frequency}\\\text{assignment}}} P_2$$

by greedily assigning frequencies breadth-first to each connection, starting from the connections of the master hub and ending with the connections of the leaves. We have that $\tilde{F} \leq F$ and, in practice, it is usually close to F or equal to it.

Given these definitions, our algorithm for the P_0 topology local search in the hybrid case is given on the next page.

For the tree case, instead of a neighborhood change, a reset is done where the Taboo lists are emptied and the current topologies are reset to the κ best topologies $\tilde{\mathcal{T}}^*$ that have been visited so far in terms of $\tilde{F}$.

Stopping conditions and other parameters

Our algorithm has multiple parameters, which each can have a significant impact on its performance.

Number of Parallel Searches $\kappa \in \mathbb{N}^*$ This parameter determines how much room we allow the algorithm to find good topologies that are close in the space as opposed to exploring the space greedily by moving in the single best direction at every iteration with $\kappa = 1$. There is a tradeoff between the diversity of the topologies at each iteration and the time it takes to search the neighborhoods of these topologies.

Neighborhood Subset Ratio $\rho \in \;]0, 1]$ This parameter determines how much of the neighborhood we explore in each neighborhood search. With $\rho = 1$, we explore the

Algorithm 4.3 P_0 Variable Neighborhood Taboo Beam Search Algorithm Pseudo-Code

```
Initialize the neighborhood $N \leftarrow N_{\text{tree}}$, the topologies $\mathcal{T} \leftarrow (T_0)$, the best topologies
  $\tilde{\mathcal{T}}^* \leftarrow (T_0)$, the best neighborhood topology $\overline{T}^* \leftarrow T_0$, the best topology $\tilde{T}^* \leftarrow T_0$
  and the Taboo lists $\mathcal{L} \leftarrow ([\,])$.
While stopping conditions are not met,
    If neighborhood change conditions are met,
        Change neighborhood $N$, empty $\mathcal{L}$ and reinitialize $\mathcal{T}$ to the best $\kappa$ from $\tilde{\mathcal{T}}^*$.
    For $T \in \mathcal{T}$ and $L \in \mathcal{L}$,
        Find the $\kappa$ best moves of $\max_{T' \in N(T)} \overline{F}(T')$
                                        s.t. $T' \notin L$ or $\overline{F}(T') > \overline{F}(\overline{T}^*)$.
        If one of the $\kappa$ best moves is such that $\overline{F}(T') > \overline{F}(\overline{T}^*)$, update $\overline{T}^* \leftarrow T'$.
    Keep only the best $\kappa$ moves $\mathcal{M}$ that result in different topologies.
    For each $M \in \mathcal{M}$, corresponding $T \in \mathcal{T}$ and $L \in \mathcal{L}$,
        Make move $M$ to $T \in \mathcal{T}$ and add its opposite to $L \in \mathcal{L}$.
        Evaluate pseudo-objective $\hat{F}$ on resulting topology $T$.
        If $\hat{F}(T)$ is greater or equal to the worst $\tilde{F}$ value of the $\tilde{\mathcal{T}}^*$,
            Evaluate pseudo-objective $\tilde{F}$ on $T$ and, if one of the $\kappa$ bests, add it to $\tilde{\mathcal{T}}^*$
              keeping only the $\kappa$ bests and, if the best, update $\tilde{T}^* \leftarrow T$ and empty its
              $L$ except for this last move.
If $|V| > 10$,
    Return the best topology $\tilde{T}^*$.
Else,
    Evaluate the best topologies $\tilde{\mathcal{T}}^*$ with the real objective $F$ and return the best
      topology $T^*$.
```

entire neighborhood[1]. With $\rho < 1$, we explore a random subset of the neighborhood with an expected ratio of ρ (implementation details for this mechanism are presented in appendix A). This allows the Taboo search to do more iterations for the same amount of time and creates more variety in the parallel searches. Since each iteration requires many costly resolutions of combinatorial problems, the size of which depends on $|V|$, having more iterations for the same amount of time is very helpful, especially for larger instances. For a meta-heuristic such as Taboo search in general, both more iterations and more diversity entail a greater exploration of the solution space. However, there is a tradeoff between the exploration of the solution space and the quality of the visited

[1] This is not exactly true. Since we use a first improvement neighborhood search instead of a best improvement search, we stop the neighborhood exploration whenever we both have encountered a strictly improving topology and we have already seen $\kappa - 1$ other non-Taboo topologies.

solutions.

Stopping Conditions We have implemented three different stopping conditions: a time limit, a maximum number of iterations and a maximum number of iterations since the last new κ-best topology in $\tilde{\mathcal{T}}^*$. For the numerical experiments, we have only used the time limit since the available computation time was our biggest practical constraint.

Neighborhood Change Conditions In the hybrid case, we have implemented two different neighborhood change conditions and a neighborhood change is triggered if one of these conditions is true. The first is a maximum number of consecutive iterations in the same neighborhood, which we have set to 25, and the second is a maximum number of consecutive iterations in the same neighborhood since the last new κ-best topology in $\tilde{\mathcal{T}}^*$, which we have set to 12.

Reset Conditions In the strictly tree case, we have implemented a single reset condition : a maximum number of iterations since the last new κ-best topology in $\tilde{\mathcal{T}}^*$, which we have set to 25. As with the neighborhood change conditions, this parameter was not fine-tuned in a systematic way. Rather they are values which yield good results in practice.

4.2.3 Initialization

When initializing our algorithm, we compute many quantities that provide accelerations in the implementation of our first improvement neighborhood searches. In doing so, we can also find if the instance admits possible mesh clusters in the hybrid case. Finally, the initialization provides the initial topologies for our local search algorithm.

It begins by ordering all the possible edges $[u, v]$ $(\forall u, v \in V$ with $u < v)$ by increasing distance

$$d_{uv} = \underset{f \in F}{\text{mean}}\ p^f_{uv} + m^f_{uv}.$$

With these distances, we can compute an approximate direct throughput $TP_{d_{uv}}$ that assumes a perfect alignment of the antennas, a parabolic antenna in the single-beam case, a single activated beam in the multi-beam case, a PMP wavefom, and no interference. For any edge $[u, v]$ such that $TP_{d_{uv}} = 0$, we do not consider this edge a possible tree edge for the rest of the algorithm. The ordering itself is also used directly in the neighborhood search implementations (see appendix A and A for details).

In the hybrid case, the distances also allow us to compute an approximate mesh direct throughput $\dddot{T}\dddot{P}_{d_{w_1 w_2}}$ that assumes omni-directional antennas in the single-beam case, the omni-directional mode in the multi-beam case, a Mesh waveform, and no interference. These approximate mesh direct throughputs allow us to pre-compute the maximal possible mesh clusters for the instance (maximal in the sense of $\subseteq$). More specifically, for each possible Mesh direct throughput value z, which we can obtain by multiplying the possible direct throughputs in the Table 3.1 by λ_{Mesh}, we compute every maximal set of nodes that could form a mesh cluster. The conditions for a set of nodes to be able to form a mesh cluster are the following:

1. every node has a minimum degree of 2 inside the mesh cluster;
2. no single node would separate the cluster into two connected components if it were removed (if that is the case, then two mesh different clusters that share this node would be acceptable);
3. the minimal degree of a cut in the cluster is 2. This last condition subsumes condition 1, but is much more computationally intensive (see appendix A for details on its implementation).

Defining possible mesh clusters at a minimal direct throughput level z in this way implies that, for a same level z, two such possible mesh clusters share at most one node, which is an important property to design the initial topologies. These sets of possible mesh clusters for each value z are also used throughout the algorithm in the mesh neighborhood search implementation (see details in appendix A). Moreover, if no such possible mesh clusters are found for the minimum non-zero z value, then the algorithm switches to the strictly tree case.

In the general hybrid case, for each value z that admits distinct sets of possible mesh clusters, plus the added value $z = +\infty$ for which no mesh clusters are possible, we construct a distinct minimum spanning tree on top of the partial implicit tree defined by the possible mesh clusters. In the strictly tree case, we only construct the minimum spanning tree corresponding to $z = +\infty$. We construct the minimal spanning trees with the classic Kruskal algorithm [20] with the tree edges order provided by d_{uv}.

In the hybrid case, the first time that the neighborhood changes to the mesh neighborhood, instead of starting from the κ best topologies $\tilde{\mathcal{T}}^*$, we start from these initial minimum spanning trees, except for the strictly tree topology in order to encourage mesh topology exploration. Moreover, if one of these initial minimum spanning trees is a single full mesh cluster, it is not used for the initial tree neighborhood since it has no neighbors in that neighborhood.

4.3 P_1 Network Configuration Exhaustive Enumeration, Approximations, *etc.*

Given a topology and a solver for P_2, we solve P_1 with

- an enumeration heuristic for the master hub selection,
- exhaustive enumeration or a meta-heuristic for the waveform assignment,
- exhaustive enumeration of the two possible channel assignments (as detailed in section 3.2.2), and
- exhaustive enumeration or approximations of the frequency assignments (as detailed in section 4.2.2).

4.3.1 Master Hub Selection

Previous equations (4.7) and (4.9) which respectively defined the pseudo-objectives $\overline{F}$ and $\hat{F}$ were an abuse of notation with respect to

$$\max_{r \in V} \ldots$$

since our approach for the master hub selection is not always an exact method. Rather we use a heuristic which gives us a subset of the the nodes $R(T) \subset V$, depending on the topology T, for which we actually do exhaustive enumeration

$$\max_{r \in R(T)} \ldots$$

The master hub has two main impacts on the objective function. The first impact is that it is the only node which can have two outgoing PMP connections. Such a node can have up to 20 tree edges connected to it while other nodes can only have up to 11 (1 predecessor tree edge and up to 10 in a outgoing PMP connections). This advantage not only allows it to have more tree edges, but also it has more freedom to partition its tree edges in sets that are more favorable for the signal strength of its antennas and for the interference that it causes in the rest of the network, by dividing the tree edges among two different frequencies.

The second impact is in the computation of the traffic of scenario B in which all nodes send a data stream to the master hub (or in the other direction by symmetry). Where it is located in the topology determines towards where the congestion increases in the edges, with the congestion being maximal at the tree edge connected to the master hub with the greatest number of explicit descendants.

For both impacts, the explicit leaves of the topology (*i.e.* leaves in the explicit topology) are bad candidates for the topology as the node to which they are connected would be better in both counts. Therefore, to construct $R(T)$, we begin by disregarding all the explicit leaves of the topology. The implicit leaves which are not explicit leaves correspond to mesh nodes which only have mesh edges. These can still be favorable master hubs if they are central to many routes in their mesh cluster(s).

We are left with many potential candidates. To filter them further entails the possibility of missing the actual best master hub. However, many of these candidates usually seem unsuitable for the optimal master hub for a variety of reasons and each of their evaluation requires a big computational cost. For that reason, we proceed with further filtering.

We consider three properties for the remaining candidates to evaluate how likely they are of being the optimal master hub.

1. their degree in the implicit arborescence, *i.e.* their tree edge degree summed with the number of mesh clusters of which they are a member. This gives us an information both on the potential of the first impact as well as how it can potentially distribute the congestion for the second impact. We wish to maximize this property.

2. the number of mesh clusters of which they are a member. This information is important by itself, again, relating to how it can potentially distribute the congestion for the second impact. We wish to maximize this property also.

3. their eccentricity, *i.e.* the greatest distance between the candidate and every other node, where we consider that each tree edge has a distance of 1 and two nodes that are in the same mesh cluster have a distance of 1. In graph theory, the center of a graph is defined as the node with the least eccentricity. As such, the eccentricity gives a valuable information since the center would be the optimal choice of master hub based on the second impact alone for a tree with the same direct throughput in each edge. We wish to minimize this property.

We then order their values in increasing order for the maximized properties and in decreasing order for the minimized property. For every candidate node, we sum its ranking in all three properties with the 2^{nd} being multiplied by the average number of nodes per mesh. We then have a value for each candidate node such that a node which would be the best in all three properties would have the highest value. Given the distribution of these values, we compute the inclusive quantile of each value such that the highest value has a quantile of 1 and the lowest value has the lowest quantile, greater than 0. We finally define our set $R(T)$

as

$$R(T) = \{v \in (V \setminus \texttt{explicit\_leaves}) \mid \texttt{quantile}_v \geq 1/2\}\,. \tag{4.10}$$

We note that this set could be equal to the set of nodes that are not explicit leaves. Typically, however, it saves between 0 and 40 % of the computing time and it keeps the optimal master hub, although, in some cases, it misses it.

In the specific case of a full mesh cluster, a custom implementation provides a significant acceleration (for implementation details, see appendix A).

4.3.2 Waveform Assignment

As for the master hub selection, our approach for the waveform assignment is not always an exact method either. In the typical case, we do exhaustive enumeration, but, for the costly greedy pseudo-objective $\tilde{F}$, when more than 64 partitions are possible (*i.e.* more than 7 tree neighbors and no mesh clusters), we switch to a meta-heuristic known as Greedy Randomized Adaptive Search Procedure (GRASP). This is because the valid number of partitions can be as big as 2^{19} in the worst case, so we want to avoid computing all the assignments for this costly pseudo-objective.

For the other two pseudo-objectives $\overline{F}$ and $\hat{F}$, waveform assignment can be accelerated in a way that mitigates this issue (for details on the acceleration, see appendix A). Moreover, the evaluation of $\overline{F}$ is necessary for our GRASP waveform assignment for $\tilde{F}$.

GRASP Waveform Assignment for $\tilde{F}$

GRASP is a very straightforward meta-heuristic. Its generic pseudo-code is presented below.

Algorithm 4.4 GRASP Pseudo-Code

```
Initialize the best solution $x^* \leftarrow \emptyset$.
While stopping conditions are not met,
    Construct incrementally an initial solution $x_0$ in a randomized greedy way by
      selecting each new element of the solution at random from a restricted
      candidate list containing the elements with the least greedy incremental costs.
    Do a local search from solution $x_0$ and obtain the best visited solution $x$.
    If $x$ is the new best, update $x^* \leftarrow x$.
Return the best solution $x^*$.
```

As a meta-heuristic, it finds its strength in exploring the solution space in many iterations.

However, in our case, each evaluation is a costly greedy frequency assignment so we cannot afford many iterations. For GRASP to be interesting, it should cost fewer evaluations than exhaustive enumeration, which limits its performance. Nonetheless, it still is the best approach we currently have for large waveform assignment problems for $\tilde{F}$.

We construct our initial solutions by taking the best partitions found by evaluating the cheaper pseudo-objective $\overline{F}$. For our local search, we use the classic steepest descent on the neighborhood which considers every way of switching a successor from one set to the other. To limit the evaluations, we do a single GRASP iteration. For a partition of n_{succ} successors, we do $n_{\text{succ}} - 4$ iterations of the local search, for a total of

$$n_{\text{succ}}^2 - 5\, n_{\text{succ}} + 5$$

$\tilde{F}$ evaluations when taking into account the fact that we do not need to evaluate the previous neighbor starting at the second iteration.

The performance of this GRASP waveform assignment is poor, but, without it, our algorithm could stall for extremely long periods of time whenever it would need to evaluate $\tilde{F}$ on a topology with a large n_{succ}.

4.4 P_2 Antenna Configuration Heuristic

Given a topology and a network configuration, we solve P_2 with a geometrically-based heuristic. For both single-beam and multi-beam antennas, the case in which an antenna a is in a Mesh connection is trivial, *i.e.* an omni-directional antenna for single-beam and the omni-directional mode with $B_a = \emptyset$ for multi-beam. The PTP case is also trivial, *i.e.* a parabolic antenna aiming directly at the other node for single-beam and a single-beam $B_a = \{0\}$ aimed directly at the other node for multi-beam. In the sequel we consider the PMP case.

4.4.1 Single-Beam Antennas

For an antenna a at node $u \in V$ connected to antennas at nodes $V_a \subset V$ on frequency $f \in F$, we wish to find an alignment $\phi_a \in [0, 2\pi[$ and a type of antenna $t_a \in \{\text{parabolic}, \text{panel}, \text{sector}\}$ that maximize the strength of the signals.

We first define the set Θ_a of the angles between node u and the nodes from V_a as

$$\Theta_a = \{\Delta_\phi(0, x_u, y_u, x_v, y_v) \mid v \in V_a\}\,.$$

We order these angles from 0 to 2π in the ordered set (θ_i) where $\theta_i \in \Theta_a$, $\forall i \in [\![0, |V_a| - 1]\!]$. Three examples are pictured in the Figure 4.7.

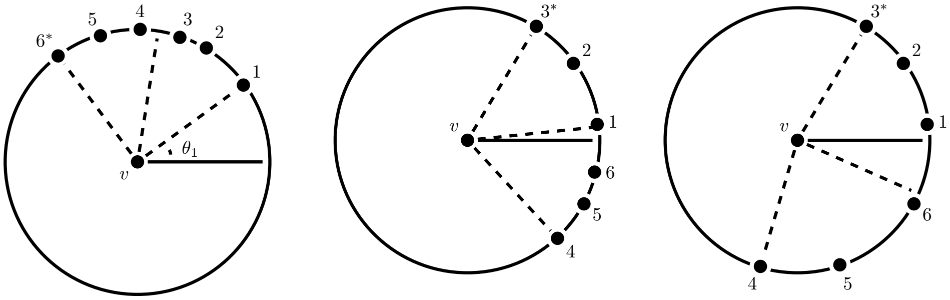

Figure 4.7 Three Possible Cases of the Single-Beam Alignment Problem

The antenna can only connect to all the necessary nodes if its 3 dB beam width $\Delta_{3dB}(t_a, f)$ is large enough to cover their angular spread. In order to formalize this, we define the following counterclockwise angular distance

$$\mathrm{d}^{\frown}_{\theta}(\theta_1, \theta_2) = \begin{cases} \theta_2 - \theta_1, & \text{if } \theta_2 \geq \theta_1 \\ 2\pi + \theta_2 - \theta_1, & \text{if } \theta_2 < \theta_1. \end{cases}$$

The condition for an antenna of type t_a to be wide enough is given by

$$\frac{\pi}{180} \Delta_{3dB}(t_a, f) \geq 2\pi - \max_{i \in [\![0, |\Theta_a| - 1]\!]} \mathrm{d}^{\frown}_{\theta}(\theta_i, \theta_{i+1 \bmod |\Theta_a|}).$$

We determine the antenna type t_a by taking the type with the smallest 3dB beam width that is wide enough. If $t_a =$ sector is not wide enough, we use the omni-directional antenna.

Otherwise, let $i^* \in [\![0, |\Theta_a| - 1]\!]$ represent the last node of the spread (in counterclockwise) with

$$i^* \in \arg \max_{i \in [\![0, |\Theta_a| - 1]\!]} \mathrm{d}^{\frown}_{\theta}(\theta_i, \theta_{i+1 \bmod |\Theta_a|}).$$

For instance, in the figure above, such nodes have an asterisk.

We approximate the optimal alignment $\phi_a \in [0, 2\pi[$ with the bisector of the spread, pictured as the middle dashed line in the figure above. Formally, it is given by

$$\overline{\phi}_a = \begin{cases} \frac{1}{2}\left(\theta_{i^*+1 \bmod |\Theta_a|} + \theta_{i^*}\right), & \text{if } \theta_{i^*+1 \bmod |\Theta_a|} \leq \theta_{i^*} \\ \frac{1}{2}\left(\theta_{i^*+1 \bmod |\Theta_a|} + \theta_{i^*}\right) - \pi, & \text{if } \theta_{i^*+1 \bmod |\Theta_a|} > \theta_{i^*} \text{ and } \theta_{i^*} \geq 2\pi - \theta_{i^*+1 \bmod |\Theta_a|} \\ \frac{1}{2}\left(\theta_{i^*+1 \bmod |\Theta_a|} + \theta_{i^*}\right) + \pi, & \text{if } \theta_{i^*+1 \bmod |\Theta_a|} > \theta_{i^*} \text{ and } \theta_{i^*} < 2\pi - \theta_{i^*+1 \bmod |\Theta_a|}. \end{cases}$$

For instance, in the first sub-figure above, $\overline{\phi}_a$ would be equal to $1/2\,(\theta_1+\theta_6)$. In the second sub-figure, $\overline{\phi}_a$ would be equal to $1/2\,(\theta_3+\theta_4)-\pi$. In the third sub-figure, $\overline{\phi}_a$ would be equal to $1/2\,(\theta_3+\theta_4)+\pi$.

4.4.2 Multi-Beam Antennas

For an antenna a at node $u \in V$ connected to antennas at nodes $V_a \subset V$, we wish to find an alignment $\phi_a \in [0, 2\pi[$ and a set of activated beams $B_a \subseteq [\![0, 23]\!]$ that maximize the strength of the signals.

We define, for a beam $b \in [\![0, 23]\!]$, with beam 0 starting at angle θ, the angular range $\Phi_b(\theta) \subseteq [0, 2\pi[$ covered by beam b as

$$\Phi_b(\theta) = \begin{cases} [l, u[\,, & \text{if } u > l \\ [l, 2\pi[\cup [0, u[\,, & \text{otherwise} \end{cases}$$

with

$$l = \left(\theta + \frac{2\pi}{24} b\right) \bmod 2\pi \quad \text{and} \quad u = \left(\theta + \frac{2\pi}{24}(b+1)\right) \bmod 2\pi$$

where we assume an effective beam width of $\frac{2\pi}{24}$. For instance, the range of the beam 0 is given by

$$\Phi_0(\theta) = \begin{cases} \left[\theta, \theta + \frac{2\pi}{24}\right[\,, & \text{if } \theta + \frac{2\pi}{24} < 2\pi \\ [\theta, 2\pi[\cup \left[0, \theta + \frac{2\pi}{24} - 2\pi\right[\,, & \text{otherwise.} \end{cases}$$

Antenna a being connected to multiple nodes, it may require more than a single beam, but more activated beams decrease the strength of the signal of each beam. We wish to find an alignment which partitions the connected nodes in as few beams as possible. As with the single-beam antennas, We define the set Θ_a of the angles between node u and the nodes from V_a given by

$$\Theta_a = \{\Delta_\phi(0, x_u, y_u, x_v, y_v) \mid v \in V_a\}$$

and we order these angles from 0 to 2π in the ordered set (θ_i) where $\theta_i \in \Theta_a$, $\forall i \in [\![1, |V_a|]\!]$.

To find the best partition of the nodes into activated beams, we try, for each of ordered node $i \in [\![1, |V_a|]\!]$, a configuration of the beams in which the node i is at the minimum angle of the range covered by the central beam 0. This is pictured in figures 4.8 and 4.9 for the first 2 nodes of an example with 5 connected nodes (and only 8 beams for clarity).

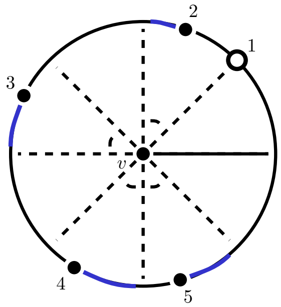

Figure 4.8 With $i = 1$ (Requires 4 Activated Beams)

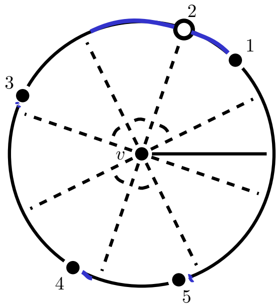

Figure 4.9 With $i = 2$ (Requires 5 Activated Beams)

For each such configuration, we compute the number of necessary beams

$$\sum_{b=0}^{23} \max_{j \in [\![1,|V_a|]\!]} \mathbb{1}\left(\theta_j \in \Phi_b(\theta_i)\right).$$

The minimum number of necessary activated beams is then

$$z = \min_{i \in [\![1,|V_a|]\!]} \sum_{b=0}^{23} \max_{j \in [\![1,|V_a|]\!]} \mathbb{1}\left(\theta_j \in \Phi_b(\theta_i)\right).$$

If more than a 7 activated beams are necessary, it is preferable to switch the antenna to its omni-directional mode with $\phi_a = 0$ and $B_a = \emptyset$.

Otherwise, we note i^* the index of the node with the minimum number of beams z. From this, we define, for every beam $b \in [\![0, 23]\!]$, the set $J_b \subseteq [\![1, |V_a|]\!]$ of connected nodes that are partitioned into that beam with

$$J_b = \left\{j \in [\![1, |V_a|]\!] \mid \theta_j \in \Phi_b(\theta_{i^*})\right\}.$$

The antenna's set of activated beams $B_a \subset [\![0, 23]\!]$ is then given by

$$B_a = \{b \in [\![0, 23]\!] \mid |J_b| > 0\}. \tag{4.11}$$

For instance, in Figure 4.8 above, the set of activated beams would be $B_a = \{0, 2, 4, 5\} \subset [\![0, 7]\!]$.

If ϕ_a were to be equal to $\theta_{i^*} + \frac{\pi}{48}$, then the node corresponding to i^* would be exactly on the border of the central beam and therefore would not be served well. On the other end of the activated beams' angular ranges, we have some extra angular space, pictured in blue in the figures above. This extra angular space is equal to

$$\xi = \min_{b \in B_v^{f*}} \min_{j \in J_b} \left(\left(\left(\theta_{i^*} + \frac{2\pi}{24}(b+1) \right) \bmod 2\pi - \theta_j \right) \bmod 2\pi \right).$$

For instance, in Figure 4.8 above, this extra angular space is limited by the angle between the node 2 and the end of beam 0 which serves it.

The antenna's alignment $\phi_a \in [0, 2\pi[$ is then centered on the connected nodes with

$$\phi_a = \left(\theta_{i^*} + \frac{\pi}{48} - \frac{1}{2}\xi \right) \bmod 2\pi. \tag{4.12}$$

For the last example, this is pictured in the figure below.

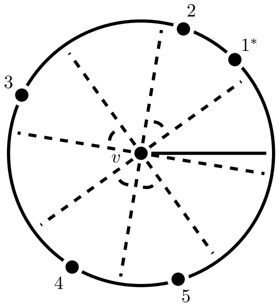

Figure 4.10 Antenna Alignment of the Last Example

CHAPTER 5 NUMERICAL EXPERIMENTS

We evaluate the performance of our algorithmic framework with a series of tests. We then compare the best networks found by our algorithm and how they differ depending on the use case.

For all following numerical experiments, we have used traffic scenario relative weights of

$$\alpha_A = \frac{1}{13}, \quad \alpha_B = \frac{4}{13}, \quad \alpha_C = \frac{8}{13}.$$

We have chosen these weights in order to give more weight to the more challenging traffic scenarios in terms of congestion in the network.

For all the following tests, we were constrained by our available computing time. More instances and runs of the algorithm per instance would be necessary to obtain statistically robust characterizations of, both, the performance of our algorithm and the kinds of solutions that it can find.

Problem Instance Generation

The problem instances used in the numerical experiments were generated by our industrial partner. They were simulated by taking into account different practical aspects such as the relative positions of the nodes as well as the effect of terrain diffraction.

The terrain details were obtained from the SRTM3 Shuttle Radar Topography Mission data [22]. Simulations were run using random areas within the USA and Canada.

The coordinates of the nodes were simulated following a type II Matern Hard-Core Process (MHCP) with a minimum distance of 0.5 km. The MHCP is a type of Poisson point process that maintains a specified minimum distance between the nodes [23]. For each simulated run, the dispersion of the communication nodes was characterized by the inverse of the average density D^{-1}, given in squared kilometers by node. Inverse densities D^{-1} of 20 km^2/node and 50 km^2/node were used, with $D^{-1} = 20$ km^2/node corresponding to the dense case and $D^{-1} = 50$ km^2/node corresponding to the case where the nodes are far away from each other.

At each node, a random antenna height between 5 and 15 m above the terrain was assumed.

For each inter-node link, the multiple knife-edge diffraction is calculated using the Deygout model [24] where the terrain details are approximated using multiple screens.

The path losses p_{uv}^{f} are then computed such that they reflect the inter-node distances and the operating frequency f.

The fade margins m_{uv}^f are finally computed using the Olsen-Segal model assuming 99% link reliability [25].

5.1 Mesh Evaluation Performance

We evaluated the performance of our different mesh evaluation methods on full mesh clusters with multi-beam antennas for instances of 10, 20 and 50 nodes and for inverse densities D^{-1} of 20 and 50 km^2/node. For each number of nodes $|V|$ and inverse density D^{-1}, 10 different instances for which the full mesh cluster was connectable were used for the tests. In the case of 50 nodes and $D^{-1} = 50$ km^2/node, out of the 200 generated instances, no such instance was found.

For each instance, we computed the pseudo-objective $\tilde{F}$ corresponding to the greedy frequency assignment as well as the time taken for the evaluation. We compared

1. the naive routing approach $\xi = 0$ with no collisions and a $\lambda_{\text{Mesh}} = 1$,
2. the greedy routing approach $\xi = 1$ (with no collisions and $\lambda_{\text{Mesh}} = 1$), run 5 times where the best pseudo-objective was taken, and
3. the same 5 runs of the greedy routing approach $\xi = 1$, but with collisions where, again, the best pseudo-objective was taken.

By comparing 1 and 2, we get a sense of the importance of the routing problem for the pseudo-objective as well as the performance of our approaches. One of the way we do this is by recording how often routing corresponding to $\xi = 1$ run 5 times is actually better or equal to $\xi = 0$ (in the column informally named $0 \leq 1$). By comparing 2 and 3, we see the importance of the mesh collisions for the pseudo-objective. By comparing 1 and 3, we can start to estimate appropriate values of the parameter λ_{Mesh} which models the mesh collisions as a multiplicative coefficient. For all values except how often $\xi = 1$ is better than $\xi = 0$, we only report the geometric means.

The results are given in the Table 5.1 on the next page.

Although $\xi = 1$ run 5 times is always better or equal to $\xi = 0$ run once for $|V| = 10$ with dense nodes, its performance seems to decrease both with the number of nodes and how sparse the nodes are located on the terrain. For $|V| = 50$ with dense nodes, the geometric mean of its performance is even slightly worse than $\xi = 0$ run once with only being higher or equal 60 % of the time. However, we do see that the routing problem has a significant impact

Table 5.1 Mesh Evaluation Performance Comparison

$\lvert V \rvert$	D^{-1}	Pseudo-Objective $\tilde{F}$			$0 \leq 1$	$\hat{\lambda}_{\text{Mesh}}$	Time (s)		
		$\xi = 0$	$\xi = 1$	Coll.			$\xi = 0$	$\xi = 1$	Coll.
10	20	8.79	15.07	1.08	100 %	0.12	0.02	1.35	2.76
10	50	4.67	5.69	0.74	70 %	0.16	0.02	1.37	2.19
20	20	4.92	8.99	0.49	90 %	0.10	0.10	28.49	38.67
20	50	2.82	3.04	0.35	50 %	0.12	0.16	28.18	33.69
50	20	5.14	5.08	0.29	60 %	0.06	1.92	2244.76	2396.21

on the objective function. For $|V| = 10$ with dense nodes, the performance is doubled by the greedy routing algorithm run 5 times. For larger instances, the routing problem becomes even harder and our greedy heuristic does not seem to hold much value. This is especially the case when we take into consideration the running time required for $\xi = 1$ run 5 times compared with the single run of $\xi = 0$. As $|V|$ increases, the running times quickly become unmanageable and unusable for a local search framework where $\tilde{F}$ is evaluated up to κ times per iteration.

Looking at the difference between $\xi = 1$ without and with mesh collisions, the collision phenomenon has a very significant negative impact on the effective throughputs of the mesh clusters. Its additional computing time is also impractical for use in the local search framework.

The estimations $\hat{\lambda}_{\text{Mesh}}$ of the parameter λ_{Mesh} which take into account both the routing problem and the collision problem vary dramatically with the number of nodes $|V|$ and the density of the nodes. Moreover, for the density of nodes, they vary in the opposite direction of the relative performance of $\xi = 1$ compared to $\xi = 0$. This suggests that using a single multiplicative coefficient to model the complex phenomenon of mesh collisions is not enough to properly evaluate networks with mesh clusters.

Since we have no guarantee on the accuracy of our evaluation of the mesh clusters within networks (with $\xi = 0$), we therefore consider that we cannot adequately evaluate hybrid networks, which means that their optimization according to such an imperfect measure is of little interest. In the sequel, we will evaluate the performance of our local search algorithm only in the strictly tree case, the modeling of which we are confident is representative of the associated networks' quality in terms of effective throughput.

5.2 Local Search Parameters Impact on Performance

We evaluated the dependency of our algorithm's performance on its local search framework, including its reset component, as well as its parameters κ and ρ (for multi-beam antennas).

5.2.1 Reset

Below is a typical run of our algorithm with an early time limit. Specifically, it pictures the progression of the $\kappa = 2$ topologies of the algorithm in terms of their pseudo-objective values ($\tilde{F}$, labeled greedy, above, and $\overline{F}$, labeled chan_avg, below) through time for an instance with $|V| = 30$ for one hour. In the sub-figure of pseudo-objective $\overline{F}$ below, we see the two current solutions and how they branch out into the next two current solutions through time. In the sub-figure above of pseudo-objective $\tilde{F}$, we only see the solutions for which the pseudo-objective $\tilde{F}$ was evaluated. We also see in red the progression of the value of the best topology $\tilde{T}^*$ visited so far.

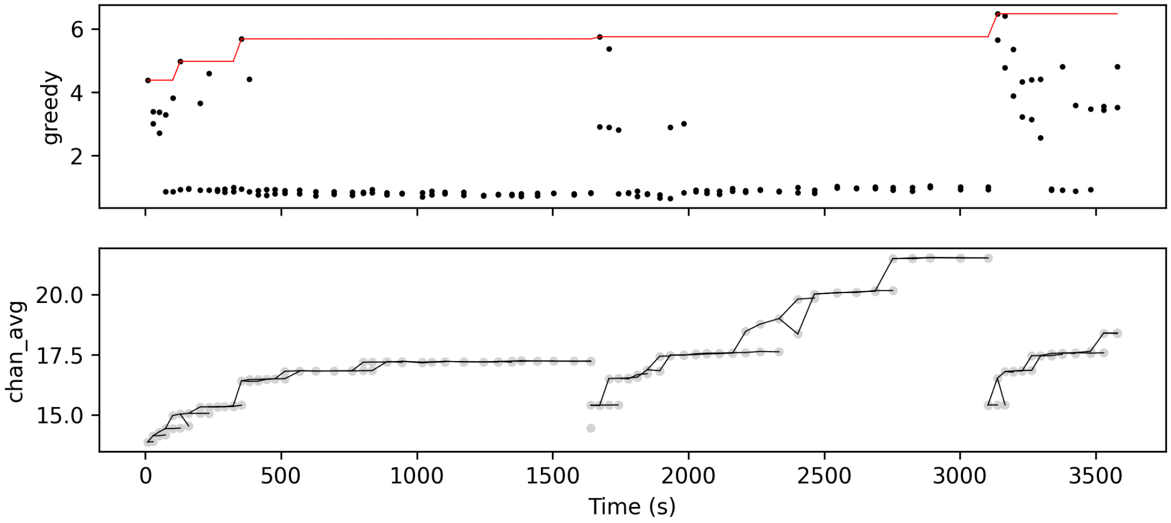

Figure 5.1 Example of Algorithm Run with 1 Hour Limit for $|V| = 30$

There are many interesting details in this figure. First, we see that the greedy frequency assignment $\tilde{F}$ has a value well below the pseudo-objective $\overline{F}$ which does not take interference into account. In fact, we see that, for most good topologies found by the neighborhood search with $\overline{F}$, their actual performance is destroyed by interference in $\tilde{F}$. Second, we see that the pseudo-objective $\overline{F}$ below, which is the one that is directly optimized by the local search framework, does increase in time (until each reset). The only mechanisms which direct the algorithm in good directions for $\tilde{F}$ are the taboo lists being emptied for every new best topology $\tilde{T}^*$ and the reset mechanism. In the beginning of the run, either through the first mechanism and/or simply through the solutions that the neighborhood searches suggested, the value of $\tilde{T}^*$ also increases with time. Third, we see that each reset (where the algorithm restarted from the two best topologies $\tilde{\mathcal{T}}^*$ visited thus far) allowed the searches to reach new higher values of $\tilde{T}^*$.

5.2.2 Number of Parallel Searches κ

To evaluate how the performance of our algorithm changes with the number of parallel searches κ, we ran the algorithm with different numbers $\kappa \in \{1, 2, 4, 8\}$ on the same instances and for the same time limits with a neighborhood subset ratio of $\rho = 1/2$. Two instances with $|V| = 10$ nodes and two instances with $|V| = 30$ nodes with, in each case, one instance with $D^{-1} = 20$ km^2/node and the other with $D^{-1} = 50$ km^2/node. For $|V| = 10$, the algorithm was run three times with a time limit of 10 minutes for each instance and, for $|V| = 30$, it was run only once with a time limit of 1 hour for each instance. In both cases, the densities did not impact these performances and the results were aggregated by number of nodes $|V|$ by normalizing the pseudo-objective $\tilde{F}$ (such that 1 is the initial worst value and 0 is the best value found by all runs). We compared the aggregate progression of these normalized objectives in time as well as the aggregate progression of the so-called primal integrals which integrates these normalized objectives in normalized time. The primal integral helps to give a sense how quickly the algorithm makes progress. The results are pictured in figures 5.2 and 5.3.

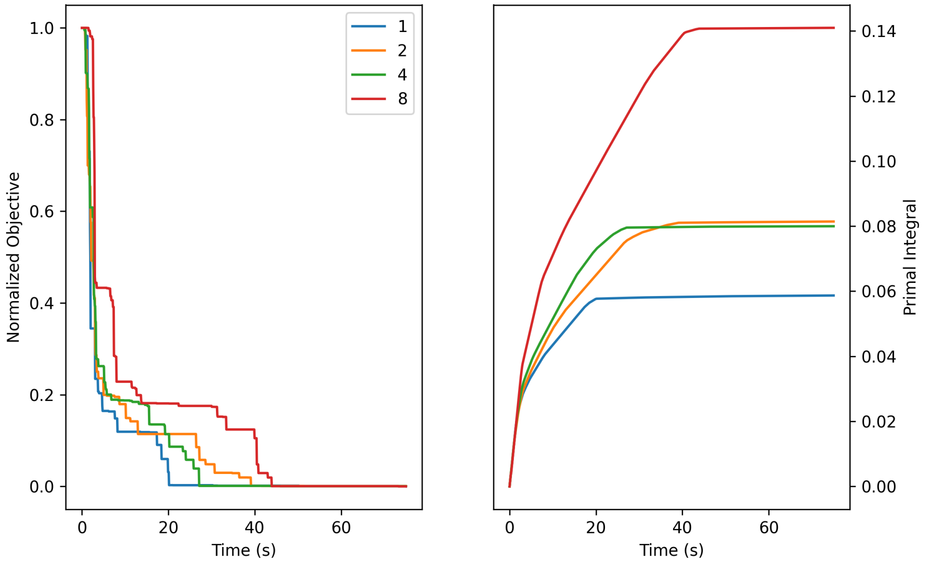

Figure 5.2 Performance in Function of the Number of Parallel Searches κ for $|V| = 10$

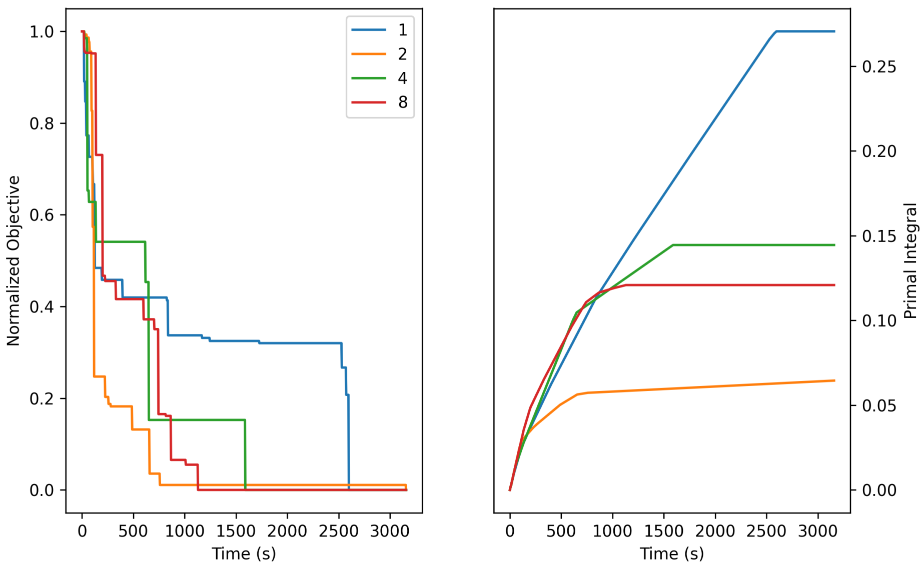

Figure 5.3 Performance in Function of the Number of Parallel Searches κ for $|V| = 30$

These figures show that, while a single parallel search seems to be the quickest for small instances ($|V| = 10$), it performs poorly as the size of instance and the associated solution space increase. This is because a single parallel search can miss other close promising solutions and can waste a lot of time by going off on a tangent and staying lost until some combination of luck in the neighborhood search and the reset mechanism allows it to find its way back. For larger instances, here represented by $|V| = 30$, $\kappa = 2$ parallel searches seem to have the best tradeoff between diversity and number of iterations. For smaller instances ($|V| = 10$), $\kappa = 2$ also had the second best aggregate trajectory according to the primal integrals, especially early in the search.

For the following tests, we will fix the number of parallel searches at $\kappa = 2$.

5.2.3 Neighborhood Subset Ratio ρ

To evaluate how the performance of our algorithm changes with the neighborhood subset ratio ρ, we ran the algorithm with different ratios $\rho \in \{1, 1/2, 1/4\}$ on the same instances as for κ and for the same time limits and numbers of runs. Again, both for $|V| = 10$ and $|V| = 30$, the densities did not impact the performances and the results were aggregated in the same way as before by number of nodes $|V|$. The results are pictured in figures 5.4 and 5.5.

For both instance sizes and especially for the larger instances ($|V| = 30$), the neighborhood subset ratio $\rho = 1/2$ seems to have the best tradeoff between number of iterations and quality

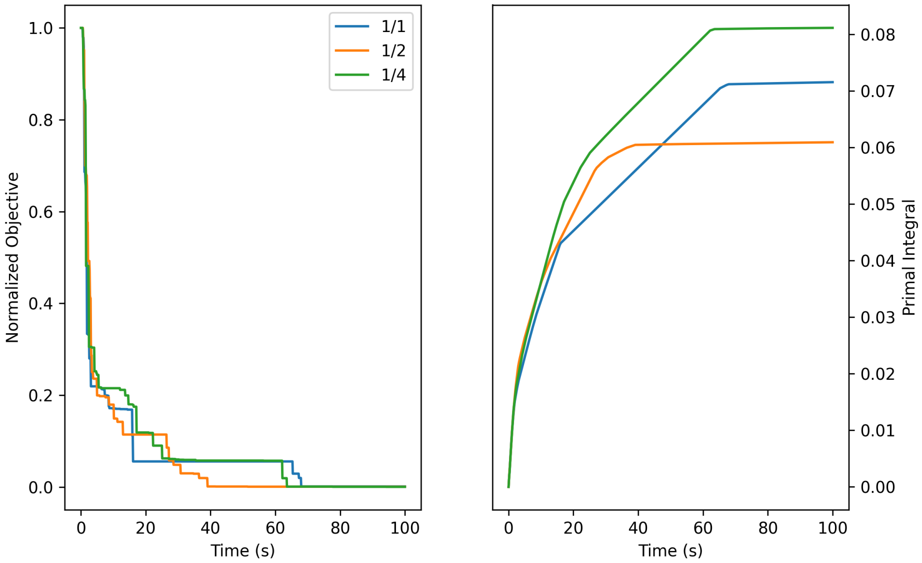

Figure 5.4 Performance in Function of the Neighborhood Subset Ratio ρ for $|V| = 10$

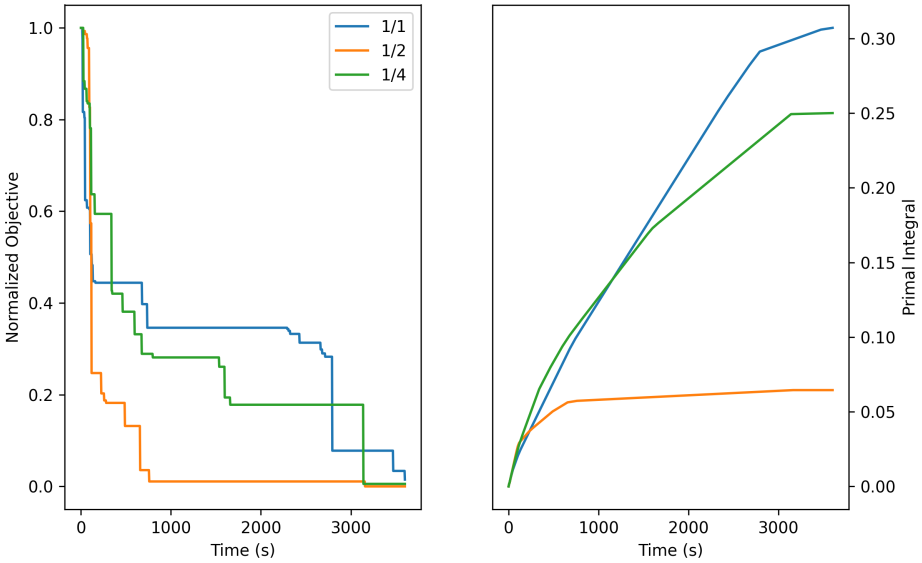

Figure 5.5 Performance in Function of the Neighborhood Subset Ratio ρ for $|V| = 30$

of the solutions. For the following tests, we will fix the neighborhood subset ratio to $\rho = 1/2$.

Ideally, a finer grid of ratios ρ should be compared for a more fine-tuned parameterization, but, again, the limited computing time available restricted us to these coarse candidates.

5.3 Single-Beam *vs* Multi-Beam Antennas (Tree Case)

Now that we have roughly parameterized our algorithm, we can use it to test whether multi-beam antennas are really advantageous when compared to single-beam antennas and in which cases. Also, we can see what are the shapes of topologies that are preferred depending on the use case. To do this comparison, we considered two instances with $|V| = 10$ nodes (one with $D^{-1} = 20$ km^2/node and the other with $D^{-1} = 50$ km^2/node), two instances with $|V| = 20$ nodes (again with $D^{-1} \in \{20, 50\}$ km^2/node) and one instance with $|V| = 50$ nodes and with $D^{-1} = 20$ km^2/node. We ran our algorithm with $\kappa = 2$ and $\rho = 1/2$ on these instances both with single-beam antennas and with multi-beam antennas. For the instances with $|V| = 10$, we ran the algorithm three times with a time limit of 10 minutes. For the instances with $|V| = 20$, we ran the algorithm two times with a time limit of 30 minutes and for the intance with $|V| = 50$, we ran the algorithm only once with a time limit of 1 hour. The relative traffic scenario weights used by the algorithm was the same as before, but the algorithm also recorded the best topology for each of the three traffic scenarios $X \in \{A, B, C\}$. This allows to compare the use of single-beam *vs* multi-beam antennas for different numbers of nodes $|V|$, for different densities of nodes D^{-1} and for different traffic scenarios X.

The results are given in the table below where, the best overall objective with our specified weights $(\alpha_A, \alpha_B, \alpha_C)$ is given directly by O and, for each individual traffic scenario (A: a single indirect communication; B: all nodes send to the master hub; C: all nodes send to all other nodes), the best objective is given by the associated minimum effective throughput that is used in the computation of O. The italicized results indicate that the associated topology could not connect (hence the minimum of 0 for all traffic scenarios). For each matchup, the type of antennas with the best found solution in bold.

Table 5.2 Best Tree Networks *vs* Full Mesh Networks by Use Case

$\lvert V\rvert$	D^{-1}	Overall		Scenario A		Scenario B		Scenario C	
		Sing.	Multi	Sing.	Multi	Sing.	Multi	Sing.	Multi
10	20	**58.46**	53.83	**156.00**	117.00	39.00	**52.00**	**56.16**	52.00
10	50	**58.46**	46.58	**156.00**	78.00	31.2	**39.00**	**56.16**	50.14
20	20	**32.88**	19.01	**78.00**	65.00	**22.29**	13.00	**29.06**	12.47
20	50	**36.59**	13.19	**156.00**	39.00	**26.00**	7.31	**29.64**	12.47
50	20	*1.59*	**2.31**	*0.00*	**6.50**	*0.00*	**0.81**	*0.00*	**1.17**

Contrary to what was believed, in most cases, single-beam antennas had the net advantage. The multi-beam antennas were only better for scenario B in small instances ($|V| = 10$) and for all scenarios in a very large instance ($|V| = 50$) since no connectable topology was found for single-beam antennas. The caveat is that these results were obtained with very tight time limits (especially for the larger instances). This is apparent in the number of iterations that the algorithm had time to make in those time limits: over 400 iterations for $|V| = 10$, about 80 iterations for $|V| = 20$ and just under 20 iterations for $|V| = 50$. It is possible that, given more time, single-beam networks could also find connected topologies for $|V| = 50$ and even better networks than those found with multi-beam antennas given the same computing time.

The best topologies found for each case are pictured in figures 5.6 - 5.10.

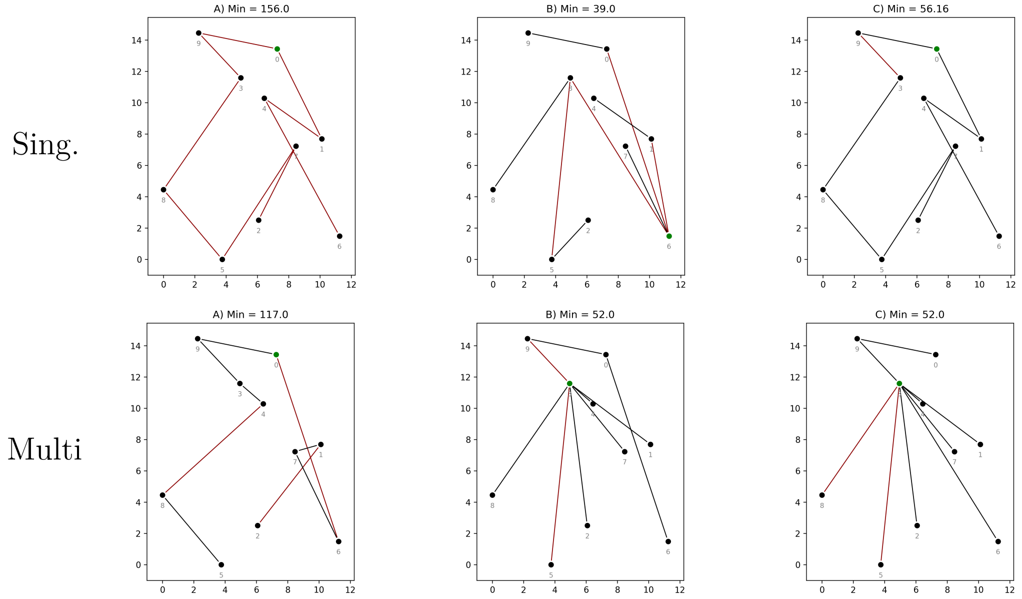

Figure 5.6 The Best Solutions for Each Traffic Scenario for $|V| = 10$ and $D^{-1} = 20$

For each topology, the selected master hub is in green and the limiting edges in terms of effective throughput are pictured in red.

Although, each traffic scenario was not optimized directly and we are probably missing better topologies, there are still important features that we can qualitatively say about the shapes of the topologies found for each traffic scenario. Scenario A (a single indirect communication) prefers long PTP chains which optimize the best direct throughputs in every edge. However, if communication delay is an important factor (not modeled in our formulation of the problem), chain topologies maximize the delay between the two ends of the chain. Scenario B (all nodes sending data to the master hub or vice versa) tends to favor deep trees which are centered on the master hub. As we expected, their limiting edges tend to be close to the master hub and have a large number of descendants. Scenario C (all nodes sending data to all other nodes) seems to be vary between the chains of scenario A and the trees of scenario B. Again as we expected, its limiting edges seem to always be connecting two big

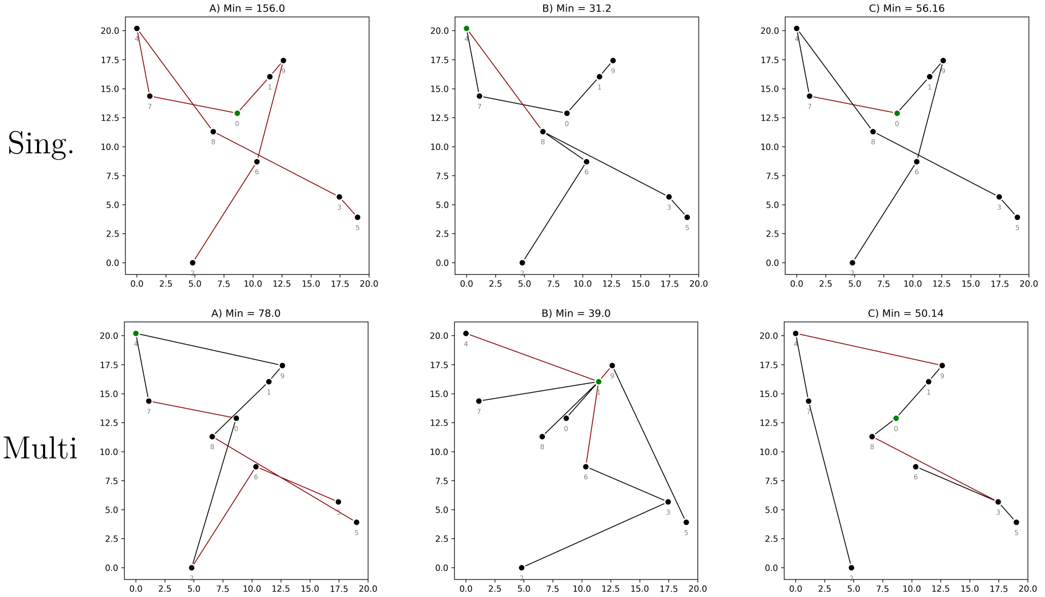

Figure 5.7 The Best Solutions for Each Traffic Scenario for $|V| = 10$ and $D^{-1} = 50$

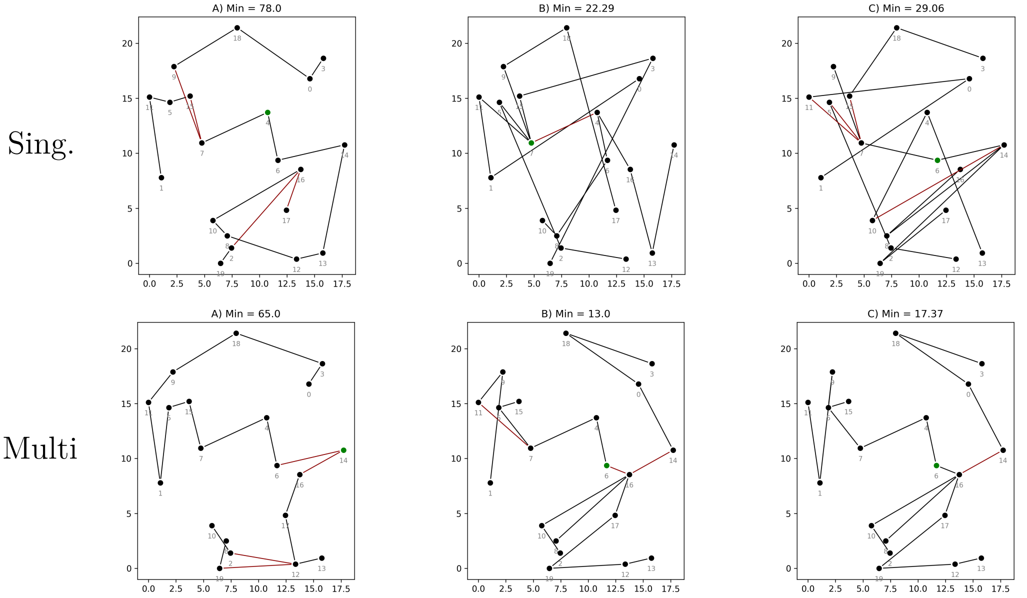

Figure 5.8 The Best Solutions for Each Traffic Scenario for $|V| = 20$ and $D^{-1} = 20$

parts of the topology.

When the nodes are less densely distributed on the terrain (higher D^{-1}), longer PTP chains are prioritized inside the topology because they make more solid connections in terms of direct throughput, which counters the bigger difficulty in connection resulting from the additional lost signal from the longer distances.

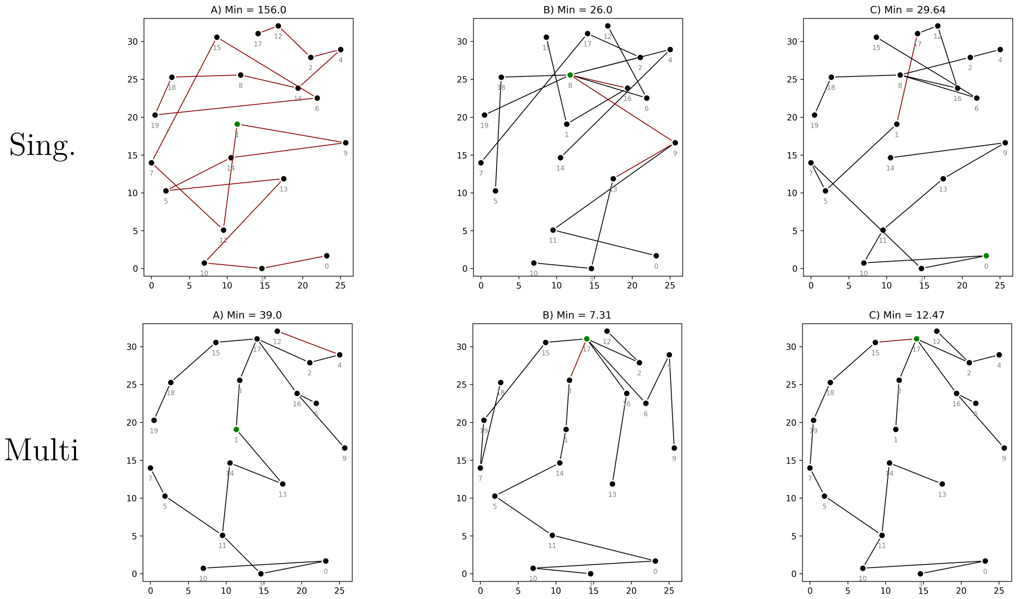

Figure 5.9 The Best Solutions for Each Traffic Scenario for $|V| = 20$ and $D^{-1} = 50$

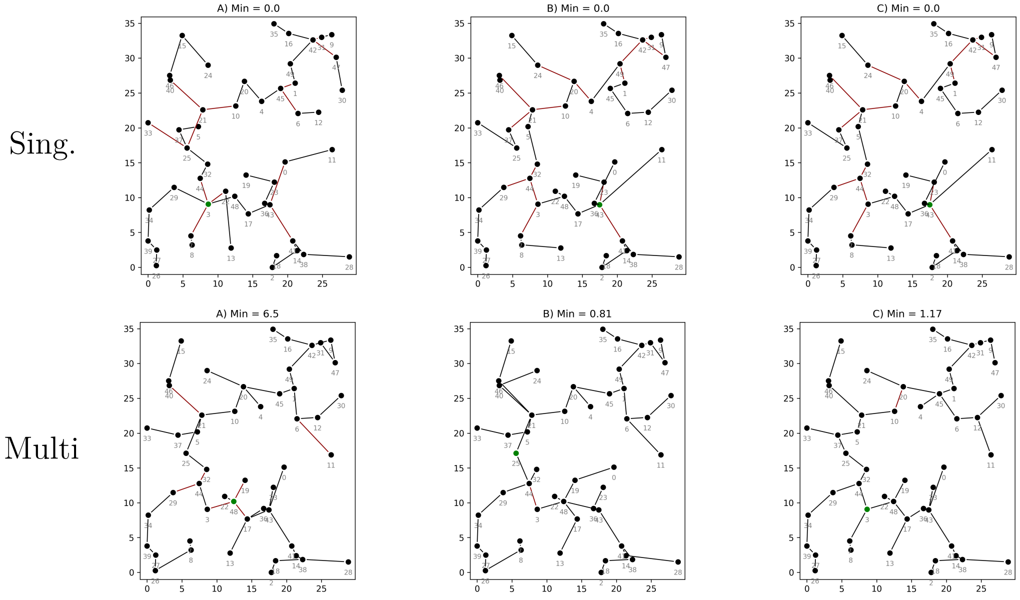

Figure 5.10 The Best Solutions for Each Traffic Scenario for $|V| = 50$ and $D^{-1} = 20$

As the size of the instance $|V|$ increases, it becomes a much harder problem to find connected topologies given the additional interference and exponential size of the combinatorial optimization problem of the topology alone.

5.4 Tree *vs* Mesh Cases

Since we cannot meaningfully evaluate hybrid topologies, one question that we can still test is in which use case is strictly tree topologies better than full mesh topologies in terms of effective throughput and in which use case is it the opposite. To do this comparison, we considered two instances with $|V| = 10$ nodes (one with $D^{-1} = 20$ km^2/node and the other with $D^{-1} = 50$ km^2/node), two instances with $|V| = 20$ nodes (again with $D^{-1} \in \{20, 50\}$ km^2/node) and one instance with $|V| = 50$ nodes and with $D^{-1} = 20$ km^2/node. For these instances, we have already computed the evaluation of the full mesh cluster in the mesh evaluation test from section 5.1. We have also already ran our algorithm in the strictly tree case for these instances in section 5.3.

The results are given in the table below.

Table 5.3 Best Tree Networks *vs* Full Mesh Networks by Use Case

$\|V\|$	D^{-1}	Overall				Traffic Scenario *A*				Traffic Scenario *B*				Traffic Scenario *C*			
		Single-Beam		Multi-Beam		Single-Beam		Multi-Beam		Single-Beam		Multi-Beam		Single-Beam		Multi-Beam	
		Tree	Mesh	Tree	Mesh	Tree	Mesh	Tree	Mesh	Tree	Mesh	Tree	Mesh	Tree	Mesh	Tree	Mesh
10	20	**58.46**	1.76	**53.83**	0.87	**156.00**	6.50	**117.00**	3.25	**39.00**	1.30	**52.00**	0.54	**56.16**	1.24	**52.00**	0.64
10	50	**58.46**	1.21	**46.58**	0.74	**156.00**	4.33	**78.00**	2.17	**31.20**	1.18	**39.00**	0.5	**56.16**	0.70	**50.14**	0.59
20	20	**32.88**	0.61	**19.01**	0.54	**78.00**	4.33	**65.00**	4.33	**22.29**	0.34	**13.00**	0.17	**29.06**	0.20	**12.47**	0.19
20	50	**36.59**	0.56	**13.19**	0.34	**156.00**	4.33	**39.00**	2.17	**26.00**	0.22	**7.31**	0.14	**29.64**	0.21	**12.47**	0.18
50	20	*1.59*	**0.59**	**2.31**	0.40	*0.00*	**6.5**	**6.50**	4.33	*0.00*	**0.05**	**0.81**	0.04	*0.00*	**0.05**	**1.17**	0.05

For all use cases except for one, strictly tree topologies give networks better effective throughputs than full mesh topologies. The only matchup that the full mesh cluster won is against the single-beam antennas for a very large instance of $|V| = 50$ nodes, but, again, this is probably more an issue of computing time than it is a win for mesh topologies. Full mesh topologies are plagued with the mesh collision phenomenon which destroys any chance they would have of beating tree networks. To be fair, our method for evaluating mesh clusters is far from representing their theoretical optimal performance, but their practical deployment protocols are also ad hoc in their approach and cannot hope to achieve actual optimal performance either.

5.5 Timing of the Algorithm

The main limiting aspect of our algorithm is its timing. In the next table, we have presented averaged times for a single neighborhood search (with $\rho = 1/2$), a single evaluation of the $\tilde{F}$ pseudo-objective and for a single iteration of one parallel search (one neighborhood search and one $\tilde{F}$ evaluation), for both tree and mesh cases. For the mesh case, we have presented the timings considering an $\tilde{F}$ evaluation with $\xi = 0$, with $\xi = 1$ and with $\xi = 1$ and mesh collisions where, again, $\xi = 1$ is run five times.

Table 5.4 Timing in Seconds of the Algorithm by Component

$\lvert V \rvert$	Tree			Mesh						
	N_{Tree}	$\tilde{F}$	Iter.	N_{Mesh}	$\xi = 0$		$\xi = 1$		Coll.	
					$\tilde{F}$	Iter.	$\tilde{F}$	Iter.	$\tilde{F}$	Iter.
10	0.02	0.30	0.32	0.01	0.02	0.03	1.36	1.37	2.47	2.48
20	0.04	3.12	3.16	0.02	0.13	0.15	28.34	28.36	36.19	36.21
50	0.18	92.97	93.15	0.20	1.92	2.12	2244.76	2244.96	2396.21	2396.41

As the table shows, for the tree case, the pseudo-objective $\tilde{F}$ alone with its greedy frequency assignment takes between 93.8 % and 99.8 % of the time taken by the entire algorithm. The computation of this pseudo-objective restricts how many iterations the algorithm can do in a limited amount of time, which holds back its performance since it decreases significantly its ability to explore the solution space. This problem is exacerbated by larger instance sizes and their exponential effect on the size of the solution space.

For the mesh case, the main problem is mostly the routing problem, but also, to a lesser extent, the computation of the mesh collisions. As long as these cannot be adequately done in a reasonable time, our algorithm will not be able to use its hybrid formulation.

CHAPTER 6 CONCLUSION AND RECOMMENDATIONS

We now conclude this thesis with a brief summary, followed by an account of the limitations of the presented work as well as avenues for future research.

6.1 Summary of Works

This work focused on the problem of tactical wireless network design for challenging environments. We modeled this problem, then, we proposed an algorithm to solve it and, finally, we empirically evaluated this algorithm through numerical experiments.

6.1.1 Problem Modeling

We modeled the tactical wireless network design with a physically-based signal with practical parameters and a variety of elementary data traffic scenarios (A: a single indirect communication; B: all nodes sending data to the master hub or vice versa; and C: all nodes sending data to all other nodes).

We modeled this problem for different use cases in two separate axes. We modeled the problem for strictly tree topologies (PTP and PMP waveforms), for strictly full mesh topologies (Mesh waveform), and for hybrid topologies (all three waveforms). We also modeled the impact of either single-beam antennas or multi-beam antennas on the network performance.

We modeled validity conditions for the network design problem for each main network component :

1. a topology;
2. a network configuration, *i.e.* a selected master hub, a waveform assignment on the edges and channel and frequency assignments on the connections;
3. antenna configurations, *i.e.* alignments and antenna types for single-beam antennas and alignments and activated beam configurations for multi-beam antennas.

This network design was optimized according to its weakest edges in terms of effective throughput for all appropriate traffic scenarios, weighted by the relevance for the use case.

To the best of our knowledge, we are the first to model this complex non-linear combinatorial optimization problem in its entirety.

6.1.2 Algorithmic Strategy

We proposed an algorithm to solve this problem for all cases by separating it in three nested sub-problems P_0, P_1 and P_2. The P_0 topology design sub-problem is solved with parallel Taboo searches that use two neighborhoods (tree and mesh) and a variety of pseudo-objectives $\overline{F}$, $\hat{F}$ and $\tilde{F}$ that approximate the real objective F in different ways and with different levels of accuracy (with $\tilde{F}$ being the closest).

The P_1 network configuration sub-problem is solved with a mix of a enumeration heuristic for the master hub selection, exhaustive enumeration and GRASP for the waveform assignment, exhaustive enumeration for the channel assignment and approximations for the frequency assignment which give rise to the aforementioned pseudo-objectives.

The P_2 antenna configurations sub-problem is solved with intuitive geometrically-based heuristics.

6.1.3 Numerical Experiments

We evaluated our algorithm through multiple tests. We first looked at the evaluation of mesh clusters, which suggested that the computation time of our best full evaluation method was too long for a proper evaluation of mesh clusters in the setting of our local search algorithm. Therefore, we strictly focused on the tree case for the rest of the testing.

We then studied the impact of our local search parameters on the performance of the algorithm. We looked at the reset mechanism, the number of parallel searches κ and the neighborhood subset ratio ρ. We found that $\kappa = 2$ and $\rho = 1/2$ worked best in our testing, which was done for multi-beam antennas.

With our parametrized algorithm, we compared the best solutions found by our algorithm with single-beam *versus* multi-beam antennas. We found, to our surprise, that the networks with single-beam antennas had higher effective throughputs for all traffic scenarios and for all sizes, except for very small instances ($|V| = 10$) for scenario B. For very large instances ($|V| = 50$), our testing proved inconclusive because of the limited computing time, although multi-beam antennas performed better with the early one hour time limit. We also found that traffic scenario A (a single indirect communication) favored chain topologies, traffic scenario B (all nodes sending data to the master hub or vice versa) favored tree topologies centered on the master hub and was limited by the edges at the master hub with the most descendants, and traffic scenario C (all nodes sending data to all other nodes) favored something in between scenarios A and B and was limited by the edges that connect two big parts of the network.

We also compared strictly trees topologies, found by our algorithm, *versus* full mesh clusters, evaluated with our best method and considering mesh collisions. We found that the tree topologies always had higher least effective throughputs in all traffic scenarios and for all use cases. Again, for single-beam antennas for a very large instance of $|V| = 50$ nodes, our algorithm could not find a connected network in one hour, but we believe that given more time, such a network could be found by our algorithm and it would certainly beat the full mesh topology (as the best unconnected tree network already has a better objective than the full mesh).

We also looked at the timing of our algorithm and noted that it was easily the most limiting aspect in its current form. More specifically, the evaluation time of the pseudo-objective $\tilde{F}$, which is κ times per iteration, takes between 93.8 % and 99.8 % of the time taken by the entire algorithm.

6.2 Limitations

The main limitations of the current work are three-fold: the mesh routing problem, the timing of the algorithm and the testing methodology.

The mesh routing problem is a complex combinatorial problem worthy of study in itself. The naive approach we have taken to solve it is mostly heuristic and it yields a poor performance, especially in terms of computing time.

Our algorithm, in its current form, is too slow, especially in the way that it computes its most accurate pseudo-objective $\tilde{F}$ which can take up to 99.8 % of the computing time. If this can be reduced significantly, then the local search will have time to do many more iterations, which should result in better networks for all cases in instances with $|V| > 10$ nodes.

The way we tested our algorithm was entirely constructed around our computing time constraints for this thesis. For our results to be more statistically robust, we would need to do each of the tests

1. on many more instances,
2. for many more runs of the algorithm per instance, and
3. for much longer time limits per run (for $|V| > 10$).

In doing so, we could also begin to characterize the variance of our algorithm, especially as ρ decreases towards 0.

6.3 Future Research

There are two different types of avenues for future research: small adjustments of the current method and broader research directions.

6.3.1 Small Adjustments

Here are some of the small adjustments that can quickly be made to our current method.

Mesh Routing

Our current mesh routing approach with $\xi \in [0, 1]$ could be improved with an iterative method that starts with $\xi = 0$ and progressively increases ξ until we are satisfied in a way such that the limiting links of the current iteration are reordered in the beginning (prioritized) for the next iteration.

Hybrid Topology Initialization

The way we initialize the hybrid algorithm could be improved by adding a validity check for each new edge in the Kruskal minimum spanning tree algorithm, such that an edge is invalidated if it can not lead to a valid waveform assignment that admits a 2-channel assignment. This would make sure that all initial topologies are valid for the hybrid case.

Multi-Beam Antenna Configuration Heuristic

The geometrically-based heuristic for the multi-beam antenna configuration problem assumes a beam width of $2\pi/24$ for the activated beam configuration. This could be improved by using the actual 3 dB beam width of each beam. Fewer beams would be activated to reach all the nodes and, so, each beam would have a stronger signal, leading to higher least effective throughputs in the resulting networks.

Parameters Characterization and Fine-Tuning

We have roughly optimized the parameters κ and ρ in our numerical experiments, but these could be optimized much further with more testing, as well as all the other parameters such as stopping/reset/neighborhood change conditions and Taboo list lengths.

6.3.2 Broad Research Directions

Here are some of the broader research directions that could be explored to continue this work.

Mesh Routing

If the study of networks with mesh clusters or networks which are full mesh clusters is important, then the mesh routing problem is of the most immediate concern. The way it is solved dictates whether hybrid topologies can be adequately evaluated in practical time, *i.e.* whether our algorithm can be used in the hybrid case. These two aspects of accuracy and timing are both necessary for the hybrid case to be explored as well as for evaluation of strictly full mesh topologies.

$\tilde{F}$ Pseudo-Objective Acceleration

The timing of our most accurate pseudo-objective $\tilde{F}$ currently takes up to 99.8 % of the computing time of our algorithm and is the big reason that good networks cannot be found for large instances ($|V| > 10$) in practical time. Either accelerating significantly this pseudo-objective or finding different accurate pseudo-objectives that can be computed in less time would be necessary to take the current (strictly tree) version of the algorithm to its next step.

One way could be to view the waveform, channel and frequency assignments as one big combinatorial optimization problem $P_{1.5}$ where we wish to assign frequencies to the edges such that they correspond to valid waveform, channel and frequency assignments, given a topology and a master hub.

$$P_{1.5} = \max_{\substack{\text{valid partition}\\ \text{of } r\text{'s successors}\\ \text{in 2 sets}}} \max_{\substack{\text{valid channel}\\ \text{assignment}}} \max_{\substack{\text{valid frequency}\\ \text{assignment}}} P_2$$

Each frequency f can be represented as (c, r) where c is its channel and r is its intra-channel ranking, as in the table below.

Table 6.1 Frequencies as Channels and Intra-Channel Ranking

f	c	r
2000	0	0
2400	0	1
4500	1	0
5000	1	1

This new problem $P_{1.5}$ could be solved by any form of variable neighborhood search with the following three neighborhoods:

1. a waveform assignment neighborhood that switches a successor of r from one set (*i.e.* channel) to the other;
2. a channel assignment neighborhood that switches all channels;
3. a frequency assignment neighborhood that switches the ranking of a connection;

such that the ranking stays the same when the channel is switched and vice versa.

Machine Learning Accelerations

Machine Learning could potentially provide many accelerations in our current framework, both for the P_0 neighborhood searches, but also for the P_1 network configurations. For the neighborhood searches, some structures that could be learned are

- which promising edges to be dropped for the edge-swap N_{tree}; and
- which promising edges to be added, given a dropped edge, for the edge-swap N_{tree}.

Both are currently being studied with Graph Neural Network (GNN) edge classification in ongoing work by Defeng Liu for his doctoral studies.

For the network configurations, the only structure that we see potential to be learnt is which promising master hubs to select (perhaps with GNN node classification). If it works, it would be very important since it could help accelerate significantly the evaluation of $\tilde{F}$ for the strictly tree case.

Finally, another difficult problem where Machine Learning could help is the routing problem of finding optimal paths in the mesh inner topology. This could perhaps be done with a methodology similar to that of the new Generative Flow Networks [26].

Optimal Network Characterization by Use Case

The algorithm in its current form would be interesting enough to fully characterize the best types and shapes of tree networks by use case, if it were given enough computing time.

REFERENCES

[1] L. A. B. Belov, S. M. Smolskiy, and V. N. Kochemasov, *Handbook of RF, Microwave, and Millimeter-Wave Components.* Norwood, MA: Artech House, 2012.

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

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

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

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

[6] D. England, B. Veeravalli, and 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, no. 5, pp. 608–620, 2007.

[7] C. Huang and X. Wang, "A bayesian approach to the design of backhauling topology for 5g iab networks," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2021.

[8] I. K. Son and S. Mao, "Design and optimization of a tiered wireless access network," in *2010 Proceedings IEEE INFOCOM*, 2010, pp. 1–9.

[9] T. Ning and H. Jiexu, "Topology control for free-space optical networks," in *2015 IEEE International Conference on Communication Software and Networks (ICCSN)*, 2015, pp. 245–249.

[10] R. Zhao, H. Liu, and R. Lehnert, "Topology design of hierarchical hybrid fiber-vdsl access networks with aco," in *2008 Fourth Advanced International Conference on Telecommunications*, 2008, pp. 232–237.

[11] Y. H. Tehrani, A. Amini, and S. M. Atarodi, “A tree-structured lora network for energy efficiency,” *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 6002–6011, 2021.

[12] I. Gódor and G. Magyar, “Cost-optimal topology planning of hierarchical access networks,” *Computers Operations Research*, vol. 32, no. 1, pp. 59–86, 2005.

[13] P. Lin *et al.*, “Minimum cost wireless broadband overlay network planning,” in *2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks(WoWMoM’06)*, 2006, pp. 7 pp.–236.

[14] J. Petrek and V. Sledt, “A large hierarchical network star—star topology design algorithm,” *European Transactions on Telecommunications*, vol. 12, no. 6, pp. 511–522, 2001.

[15] J. An *et al.*, “Joint design of hierarchical topology control and routing design for heterogeneous wireless sensor networks,” *Comput. Stand. Interfaces*, vol. 51, no. C, p. 63–70, mar 2017.

[16] L. Mroueh *et al.*, “Topology design of fully connected hierarchical mobile ad-hoc networks,” in *2016 International Symposium on Wireless Communication Systems (ISWCS)*, 2016, pp. 104–108.

[17] S. Chen, X. Ding, and X. Chen, “Formation control of robot swarm based on community division and multilevel topology design via pining,” in *The 26th Chinese Control and Decision Conference (2014 CCDC)*, 2014, pp. 1631–1636.

[18] X. Li and Y. Xi, “Double-layer topology design based on physical communication network,” in *2014 13th International Conference on Control Automation Robotics Vision (ICARCV)*, 2014, pp. 925–930.

[19] W. P. Keith Parsons. (2019) Mcs table. [Online]. Available: https://d2cpnw0u24fjm4.cloudfront.net/wp-content/uploads/802.11ac-VHT-MCS-SNR-and-RSSI.pdf

[20] T. H. Cormen *et al.*, Eds., *Introduction to Algorithms*, 3rd ed. Cambridge, MA: The MIT Press, 2009.

[21] F. Glover and M. Laguna, *Tabu Search*. Assinippi Park Norwell, MA: Kluwer Academic Publishers, 1997.

[22] U.S. Geological Survey (USGS). (2016) Srtm3 shuttle radar topography mission north american data version 2.1. [Online]. Available: dds.cr.usgs.gove/srtm/version2_1/SRTM3/North_America/

[23] M. Kiderlen and M. HÖrig, “Matérn’s hard core models of types i and ii with arbitrary compact grains,” Centre for Scotchastic Geometry and Advanced Bioimaging (CSGB), Tech. Rep., 2013. [Online]. Available: https://data.math.au.dk/publications/csgb/2013/math-csgb-2013-05.pdf

[24] H. Sizun, *Radio Wave Propagation for Telecommunication Applications.* Paris, France: Springer-Verlag, 2003.

[25] S. Loyka, A. Kouki, and F. Gagnon, “Fading prediction on microwave links for airborne communications,” in *IEEE 54th Vehicular Technology Conference. VTC Fall 2001. Proceedings (Cat. No.01CH37211)*, vol. 4, 2001, pp. 1960–1964 vol.4.

[26] E. Bengio *et al.*, “Flow network based generative models for non-iterative diverse candidate generation,” in *Advances in Neural Information Processing Systems*, M. Ranzato *et al.*, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 27 381–27 394. [Online]. Available: https://proceedings.neurips.cc/paper/2021/file/e614f646836aaed9f89ce58e837e2310-Paper.pdf

APPENDIX A IMPLEMENTATION DETAILS AND ACCELERATIONS

In this appendix, we present important implementation details that would have otherwise encumbered the body of the thesis.

A.1 P_0 Potential Mesh Minimum Cut Implementation

To verify that the minimum cut of a potential mesh cluster is 2, we try every possible way to expose a 1-cut. If we cannot find a 1-cut, then the potential mesh cluster is verified as a possible mesh cluster. If we do find a 1-cut, it exposes two new potential mesh clusters.

Algorithm A.1 Potential Mesh Cluster Minimum 2-Cut Verification Implementation

```
Let the potential set of cluster nodes be $V[k']$ and its cluster edges $E[k']$ such that
  $\forall [w_1, w_2] \in E[k']$, $\overset{\cdots\cdots}{TP}_{d_{w_1 w_2}} \geq z$.
For $[w_1, w_2] \in E[k']$ ,
    Remove $[w_1, w_2]$ from $E[k']$.
    Iteratively construct the connected components $W_1$ and $W_2$ respectively starting
      from $w_1$ and $w_2$ and stop construction if their intersection becomes non-empty.
    Add $[w_1, w_2]$ back to $E[k']$.
    If $W_1 \cap W_2 = \emptyset$ ,
        $[w_1, w_2]$ exposed a 1-cut in the potential mesh cluster defined by the set of
          nodes $V[k']$.
        In doing so, it revealed both $W_1$ and $W_2$ as other potential mesh clusters if,
          respectively, $|W_1| > 2$ and $|W_2| > 2$.
        Break.
Return the verified possible mesh cluster $V[k']$ or the two new potential mesh
  clusters $W_1$ and $W_2$, depending on the case.
```

A.2 P_0 Neighborhood Implementations

In this section, we present important implementations details for the neighborhood searches used at every iteration of the P_0 main Taboo beam search loop.

κ Best Neighbors

For both neighborhoods (tree and mesh), less than κ best neighbors may be returned by the neighborhood search if it evaluated less than κ non-Taboo neighbors. This becomes more common as the neighborhood subset ratio ρ gets closer to 0.

Random Neighborhood Subset Search

For both neighborhoods, when using a neighborhood subset ratio $\rho < 1$, our algorithm randomly samples (without replacement) a subset of the possible neighbors to evaluate. For instance, in the tree neighborhood, for every tree edge in the current topology that can be dropped, there are n_{add} possible tree edges that can be added to reconnect the topology, of which our algorithm randomly samples a subset such that the expected size of this subset is $\rho\, n_{\text{add}}$. In the mesh neighborhood, for each sub-neighborhood, a subset of the neighbors is randomly sampled in the same way.

In general, if a subset of $n_{\text{component}}$ components must be randomly sampled, our algorithm samples either

$$\lfloor \rho\, n_{\text{component}} \rfloor \quad \text{or} \quad \lceil \rho\, n_{\text{component}} \rceil$$

of those components such that the latter has a probability of

$$\rho\, n_{\text{component}} - \lfloor \rho\, n_{\text{component}} \rfloor .$$

If the components are not ordered, they are sampled with equal probability. If the components are ordered $(\texttt{comp}_i)$ with $i \in [\![0, n_{\text{component}} - 1]\!]$, then they are sampled such that each component $\texttt{comp}_i$ has a relative probability

$$\frac{n_{\text{component}} - i}{n_{\text{component}}^2 + n_{\text{component}}(n_{\text{component}} - 1)/2},$$

i.e. such that the components that are ordered first have a linearly higher probability than those ordered last.

A.2.1 Tree Neighborhood First Improvement Search

In the following, we present the implementation details of the first improvement neighborhood search for the tree neighborhood.

Revised Taboo Add List Length

Since in the initialization we compute which tree edges are possible, our initial Taboo add list length, which depended on the number of possible tree edges that can be added, needs to be revised. We correct it with a coefficient describing the ratio of tree edges which are possible out of the $|V|\,(|V| - 1)/2$ theoretically possible tree edges. Our revised Taboo add list length becomes

$$\left\lfloor \sqrt{\frac{|\{(u,v) \in V^2 \mid u < v,\ TP_{d_{uv}} > 0\}|}{|V|\left(|V|-1\right)/2} \left(\frac{|V|\left(|V|-1\right)}{2} - \sum_{k \in K} \frac{|V[k]|\left(|V[k]|-1\right)}{2}\right)} \right\rceil . \tag{A.1}$$

Neighbor Ordering

For each tree edge in the current topology that can be dropped, there is a set of possible tree edges which can reconnect the topology. We order these addable tree edges $[u, v]$ by increasing distance d_{uv}, which was defined in the initialization, and we sample a proportion ρ of these edges, as described previously.

The algorithm evaluates these neighbors in the following fashion.

Algorithm A.2 Tree Neighborhood First Improvement Search

```
Initialize the set of current tree edges to visit $E' = E \cap E^*$.
Initialize the ordered set of best moves $\mathcal{M} = ()$
While first improvement search conditions are not met and $|E'| > 0$,
    Select at random (uniformly) an edge $e \in E'$.
    If $e$ has never been selected, compute its set of addable edges $E'_e$, order it and
      sample it in proportion $\rho$.
    Remove the first addable edge $e'$ from $E'_e$ and if $|E'_e| = 0$, remove $e$ from $E'$.
    Evaluate the neighbor with tree edge $e$ swapped with $e'$, and if non-Taboo and
      one of the $\kappa$ bests (or Taboo and best ever evaluated), add the move $(e, e')$ to
      $\mathcal{M}$ and keep only the $\kappa$ bests.
Return the best moves $\mathcal{M}$.
```

A.2.2 Mesh Neighborhood First Improvement Search

In the following, we present the implementation details of the first improvement neighborhood search for the mesh neighborhood.

Neighbor Ordering

For the creation sub-neighborhood, for each candidate node (non-leaf non-mesh node), the potential mesh cluster that can be created is comprised of this node and all of its tree neighbors. We check this potential cluster against the different levels of maximal possible clusters computed in the initialization to find the maximal subset of this potential cluster that is possible. To be considered possible, the subset must be of size greater or equal to 3 and form a single connected component in the pre-computed inner topology of the associated maximal possible cluster, restricted to this subset. If no such subset can be found for the candidate node, than this candidate node is discarded. If such a subset can be found, we compute the minimal degree of its nodes in the restricted pre-computed inner topology. We order the clusters that can be created in this sub-neighborhood, first, by decreasing maximum associated throughput level z and, then, for clusters with the same z value, by decreasing minimal degree. Once they are ordered, we sample them in the proportion ρ as described previously.

For the inclusion sub-neighborhood, for each candidate node (node adjacent to a current cluster), as for the creation neighborhood, we check the resulting cluster against the different levels of maximal possible clusters. If it can be included in a single connected component in one of the associated restricted inner topologies, then it is considered. Again, we order these candidates, first, by decreasing maximum associated throughput level z and, then, for candidates with the same z value, by decreasing minimal degree. Once they are ordered, we sample them in the proportion ρ as described previously.

For the exclusion sub-neighborhood, for each current cluster, we can check against the maximal possible clusters to see which one with the highest value z contains this current cluster in single connected component in the restricted pre-computed inner topology. We can then compute the degree of each of its candidate nodes (nodes in the current cluster) in the associated restricted pre-computed inner topology. This allows us to order the candidate nodes by increasing degree. Then, for each candidate node, there is a set of possible tree edges which can reconnect the topology. We order these addable tree edges $[u, v]$ by increasing distance d_{uv} and we sample a proportion ρ of these edges, as described previously.

For the destruction sub-neighborhood, for each candidate cluster (current cluster of size 3), we sample (uniformly) in the proportion ρ a subset of the three possible ways of destroying this cluster into 3 nodes connected by 2 tree edges. In our implementation, this sub-neighborhood is grouped with the exclusion sub-neighborhood, as presented in Algorithm A.3.

For the fusion sub-neighborhood, we sample (uniformly) in the proportion ρ a subset of the pairs of clusters which can be merged together (pairs of current clusters that share a node).

We evaluate these neighbors as described in Algorithm A.3.

A.3 P_1 Accelerated Master Hub Selection for Full Mesh Clusters

Because of the single valid waveform assignment possible for every master hub as well as the omni-directional requirement, the direct throughputs are the same for all master hubs for a given frequency $f \in F$. They only need to be computed once. Only the routing problem

Algorithm A.3 Mesh Neighborhood First Improvement Search

```
Initialize the set of appliable sub-neighborhoods $SN$ for the current topology.
Initialize the ordered set of best moves $\mathcal{M} = ()$
While first improvement search conditions are not met and $|SN| > 0$ ,
    Select at random (uniformly) a sub-neighborhood $sn \in SN$.
    If $sn =$ creation ,
        If the first time creation is selected, compute the set of possible clusters
          $K'_{\text{creation}}$, order it and sample it in proportion $\rho$.
        Remove the first candidate cluster $k'$ from $K'_{\text{creation}}$ and if $|K'_{\text{creation}}| = 0$,
          remove creation from $SN$.
        Evaluate the neighbor with cluster $k'$ created, and if non-Taboo and one of
          the $\kappa$ bests (or Taboo and best ever evaluated), add the move $(\emptyset, k')$ to $\mathcal{M}$
          and keep only the $\kappa$ bests.
    Else if $sn =$ inclusion ,
        If the first time inclusion is selected, initialize the set of current clusters with
          remaining candidate nodes $K'_{\text{inclusion}} = K$.
        Select at random (uniformly) a current cluster $k \in K'_{\text{inclusion}}$.
        If the first time $k$ is selected, compute the set of possible nodes to be included
          $V^k_{\text{creation}}{}'$, order it and sample it in proportion $\rho$.
        Remove the first candidate node $v'$ from $V^k_{\text{creation}}{}'$ and if $|V^k_{\text{creation}}{}'| = 0$, remove
          $k$ from $K'_{\text{inclusion}}$ and if $|K'_{\text{inclusion}}| = 0$, remove inclusion from $SN$.
        Evaluate the neighbor with node $v'$ included in $k$, and if non-Taboo and one
          of the $\kappa$ bests (or Taboo and best ever evaluated), add the move $(\emptyset, (k, v'))$
          to $\mathcal{M}$ and keep only the $\kappa$ bests.
    Else if $sn =$ exclusion ,
        If the first time exclusion is selected, compute the set of current clusters with
          remaining candidates $K'_{\text{exclusion}} = K$.
        Select at random (uniformly) a current cluster $k \in K'_{\text{exclusion}}$.
        If $V[k] > 3$ ,
            If the first time $k$ is selected, initialize the set of current nodes with
              remaining candidates $W^k_{\text{exclusion}}{}' = V[k]$ and order it.
            Select the first node $w'$ from $W^k_{\text{creation}}{}'$ to exclude and, if the first time $w'$ is
              selected, compute the set of possible edges to be added $E^{kw'}_{\text{exclusion}}{}'$, order it
              and sample it in proportion $\rho$.
            Remove the first candidate edge $e'$ from $E^{kw'}_{\text{exclusion}}{}'$ and if $|E^{kw'}_{\text{exclusion}}{}'| = 0$,
              remove $w'$ from $W^k_{\text{exclusion}}{}'$ and if $|W^k_{\text{exclusion}}{}'| = 0$, remove $k$ from $K'_{\text{exclusion}}$
              and if $|K'_{\text{exclusion}}| = 0$, remove exclusion from $SN$.
            Evaluate the neighbor with node $w'$ excluded from $k$ and edge $e'$ added to
              the topology, and if non-Taboo and one of the $\kappa$ bests (or Taboo and
              best ever evaluated), add the move $((k, w'), e')$ to $\mathcal{M}$ and keep only the $\kappa$
              bests.
        Else,
            If the first time $k$ is selected, compute the set of possible nodes $W^k_{\text{destruction}}{}'$
              to which the other two should be connected and sample it (uniformly) in
              proportion $\rho$.
            Remove the first candidate node $w'$ from $W^k_{\text{destruction}}{}'$ and if
              $|W^k_{\text{destruction}}{}'| = 0$, remove $k$ from $K'_{\text{exclusion}}$ and if $|K'_{\text{exclusion}}| = 0$, remove
              exclusion from $SN$.
            Evaluate the neighbor by destroying cluster $k$ with node $w'$ connected to
              the other two, and if non-Taboo and one of the $\kappa$ bests (or Taboo and
              best ever evaluated), add the move $(k, w')$ to $\mathcal{M}$ and keep only the $\kappa$
              bests.
    Else if $sn =$ fusion ,
        If the first time fusion is selected, compute the set of pairs of clusters $K'_{\text{fusion}}$
          and sample it in proportion $\rho$.
        Remove the first pair of clusters $(k'_1, k'_2)$ from $K'_{\text{fusion}}$ and if $|K'_{\text{fusion}}| = 0$,
          remove fusion from $SN$.
        Evaluate the neighbor with clusters $k'_1$ and $k'_2$ merged, and if non-Taboo and
          one of the $\kappa$ bests (or Taboo and best ever evaluated), add the move
          $(\emptyset, (k'_1, k'_2))$ to $\mathcal{M}$ and keep only the $\kappa$ bests.
Return the best moves $\mathcal{M}$.
```

(for $\xi > 0$) and the effective throughputs differ between possible master hubs and those have to be computed for all $r \in V$.

A.4 P_1 Accelerated Waveform Assignment for $\overline{F}$ and $\hat{F}$

If there is only one or two valid partitions, we evaluate them normally. If there is more than two partitions (*i.e.* the set of the master hub's successors V_{succ} is such that $|V_{\text{succ}}| > 2$ and the master hub is not in a mesh), we do the following.

We evaluate the topology with partition $\{V_{\text{succ}}, \emptyset\}$ once with the first channel and once with the second channel. With these two evaluations, we get all the effective throughputs we need to evaluate all of the valid partitions, except for the effective throughputs of the tree edges connecting the master hub to its successors. For those, we only need to consider the restricted topology $(V[\{r\} \cup V_{\text{succ}}], E[\{r\} \cup V_{\text{succ}}])$ containing only the master hub and its successors. With this restricted topology, we get the necessary direct throughputs which we can then divide by the constant d_v for each successor $v \in V_{\text{succ}}$.